大規模言語モデル群への routing タスクにおける埋め込みモデルと多数決併用の分析

田村拓也　榎本昌文　秋元康佑　小山田昌史
NEC データサイエンスラボラトリー
{tamura-takuya,masafumi-enomoto,kosuke_a,oyamada}@nec.com

概要

専門性の異なる多様な大規模言語モデル (LLM) が利用可能な状況において，省コストで質の良い生成物を得るために，タスクに応じて適切なモデルを選択する routing 手法が提案されている．既存の routing 手法は LLM がタスクを正しくこなす度合いを推定するために，タスクの埋め込みを用いて類似過去タスクを参照するが，埋め込みモデルの違いが性能に与える影響を十分に分析できていない．本研究では埋め込みモデルによる routing 性能の差異を分析した．その結果，埋め込みによる性能の差異はほとんど無い一方，全タスクで平均的に良い回答を行う LLM に偏ってタスクを割り当てるため，依然として routing 手法に改善の余地があることが分かった．また，複数の候補モデルを routing 手法で選択した後に，多数決で誤答を省くことで，更に性能が改善することも新たに発見した．

1　はじめに

2023 年には，LLM の顕著な発展に伴い Llama-2 [1] や Mistral [2]，Qwen [3] などの小規模かつ高性能な LLM が数多く公開された．これらの小規模 LLM は追加訓練のコストが低いため，ドメイン特化型の追加学習済みモデルが派生的に構築されている．ただし，追加学習時の訓練データに含まれないドメインにおいて事前学習済みモデルよりもモデル性能が低下する「破壊的忘却」が起こり，この影響は特に小規模なモデルほど大きい [4, 5, 6]．このような専門性の異なる LLM が多数利用可能な状況では，解きたいタスクに適した LLM を予め選択することで，良質な生成物を得ることが期待できる．

先行研究では，与えられた各タスクを解くにあたって，最も高品質な回答を生成できると推定される LLM へタスクを割り当てる routing 手法が提案されている [7, 8]．Shnitzer ら [7] は，過去のベンチマークから検索された解きたいタスクの類似事例の正誤を参照することで当該タスクの正答率を推定し，routing を行う手法を提案した．ここにおける類似性は，LLM へ入力されるプロンプト文たちの埋め込み間の距離によって定義される．この埋め込み空間の距離に基づく割り当てによって，無作為な割り当てよりも高い性能を達成することが示された．しかし，タスクの類似性の捉え方が割り当てに直接寄与するにも関わらず，埋め込みモデルの違いによる routing 性能への影響が分析されていない．加えて，誤った割り当てを行った際の詳細な傾向も不明瞭であるため，どのような方向に手法改善の余地があるか分からない．そのため本稿では，この Shnitzer ら [7] に倣った正答率推定手法に基づく routign 手法について，以下のリサーチクエスチョン（RQ）に関する実験および分析を行った．

RQ-1: 類似インスタンスの検索に利用する埋め込み空間を変えると，routing 性能がどれほど変化するか？　Shnitzer ら [7] は，インスタンスの埋め込み化を Sentence-BERT[9] のみで行なっている．一方で現在文埋め込みモデルには，プロンプトに基づきコンテキストを考慮する手法 [10, 11, 12, 13] や，インストラクションを考慮する手法 [14] など多様なモデルが提案されている．そこで本稿では，埋め込みモデルの違いによる routing 性能への影響を調査した．結果，単語埋め込みの平均ベクトルを文埋め込みとするモデルから LLM に基づくモデルまで，いずれも routing 性能はおおむね同じであった．

RQ-2: 現状の routing と理想的な routing にはどのような差があるか？　本稿では，routing の基準となる推定正答率に基づく LLM のランキングと実際の正誤に基づく理想的なランキングとの差を調査した．結果，スペアマンの順位相関係数は最大でも 0.228 と高くないものの，NDCG@5[15] では 0.625

と高い値が得られた．また，routing の誤り分析を行なったところ，理想的な rouitng に比べて実際の routing は全タスクで平均的に良い回答を行う LLM へ偏って割り当ててしまう傾向があり，依然として埋め込み手法を含む routing 手法の改善に余地があることが示唆された．

RQ-3: routing と多数決を併用することにより，性能を向上できるか？ Shnitzer ら [7] は，推定正答率の最も高い LLM 単体にタスクを割り当てている．一方で，LLM の推論において複数の回答候補による多数決が有効に働くことが知られている [16, 17, 18]．そこで本稿では，routing 時に複数の LLM を選んで回答候補を生成したのち多数決によって集約する手法についても実験し，ランダムにサンプリングされた LLM による多数決に比べて高い性能を得ることを新たに発見した．

2 手法

本研究では，解きたいインスタンスに対して最も適した LLM 群を選択しそれらによる回答を集約する問題を解く．具体的な問題設定は次の通りである．新たなインスタンス x が与えられた際に，利用可能な指示学習済みの大規模言語モデルの集合 $\{m_1, \cdots, m_M\}$ のなかから正答しやすいと推定される上位 w モデル $\{m_1^*, \cdots, m_w^*\}$ を選出し，実際に推論を行った後，回答の集約を行う．

2.1 未知インスタンスに対する正答率推定

本研究では，Shnitzer ら [7] に倣い，評価済みベンチマークデータ内にある類似インスタンスの正誤情報を用いる手法を採用した．ここに，ベンチマーク D を，インスタンス x と候補となる各 LLM m_i による生成物，そしてそれに対する評価スコア $s_i(x)$ の３つ組の集合と定義する．

$$D = \{(x, m_i(x), s_i(x))\} \tag{1}$$

まず，ベンチマークの各インスタンスのプロンプト文のうち，指示文や few-shot 事例を除いたインスタンス固有のテキストに対する埋め込みベクトルを予め用意する．ベンチマークに存在しない新たなのインスタンス x が与えられた際には，同様に文埋め込みベクトルへ変換したもの $\phi(x)$ をクエリとして kNN 探索を行い，過去のベンチマークから類似事例 $\{x'\}$ を k 件検索する[1)]．これらの類似事例におけるモデル m_i による平均正答率を x に対する**推定正答率** s_{m_i} とする．

$$s_{m_i}(x) = \frac{1}{k} \sum_{x' \in \mathrm{kNN}(\phi(x), D)} s_i(x') \tag{2}$$

2.2 多数決による回答集約

Shnitzer ら [7] は，単一の最良モデルへの割り当てのみに焦点を当てていた．本研究では，多数決による回答集約が効果的であることを示した先行研究 [16, 17, 18] を踏まえ，routing に加えて多数決による回答集約を行った．具体的には，候補となる各モデル $\{m_i\}$ の推定正答率 $\{s_{m_i}\}$ を得た後，$\{s_{m_i}\}$ の値が大きい上位 w モデルを取り出し，それらのモデルによる出力の多数決をとる．なお，簡単のため，本稿における実験では選択式もしくは数値による回答が可能なタスクのみを利用した．

3 実験

本稿における実験では，埋め込みモデルによる routing 性能への影響の評価と，routing 時に選択された複数の LLM の回答を多数決で集約することによる効果を評価する．

3.1 実験設定

3.1.1 データセット

本稿における実験では，多様なドメインのタスクを含む場合に適切なモデルへ割り当てが可能かを評価するため，57 のタスクを含む MMLU [19]，数学ドメインに特化した GSM8K（1 タスク）[20]，17 のタスクを含む BBH Multi-Choice [21] のテストデータからなるデータセット (計 75 タスク/11,037 インスタンス) を作成した．なお，ここではインスタンスの集合をタスクと定義する．評価にあたっては，作成したデータセットのうち各タスクごとに 20%のインスタンスを無作為にサンプリングした計 2,216 インスタンスを routing のテストデータとし，残りの 8,821 事例を routing における過去事例として利用した．

3.1.2 候補となる LLM

2023 年 12 月現在，Mistral-7B は小規模 LLM の中で最も高性能な LLM の一つであり，多様なデータセットによる追加学習済みモデルも多数公開されている．本実験では，Huggingface の Open LLM Leaderboard[2)] の上位モデルのうち Mistral-7B をベー

1) 本稿における実験では，パラメータチューニングの結果 $k = 100$ を採用した．

2) https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard

スとした 7 モデルを routing の候補 LLM として採用した．具体的には，Mistral-7B-Instruct-v0.2[2]，数学推論に特化したデータセット MetaMathQA[22] による MetaMath-Mistral-7B，OpenOrca データセット [23] またはそのサブセットによる Mistral-7B-OpenOrca[23]，Mistral-7B-SlimOrca[24]，GPT-3.5/4 の出力を利用し Conditional-RLFT を行った OpenChat-3.5-1210[25]，Open Heremes による OpenHermes-2.5-Mistral-7B，OpenHeremes や Magicoder データセットによる Dolphin-2.6-Mistral-7b を利用した．

3.1.3 埋め込みモデル

Shnitzer ら [7] は，各インスタンスの埋め込み化にあたって Sentence-BERT[9] 方式のモデル all-mpnet-base-v2 のみを採用している．ただし，現在では LLM をベースとしたモデルやインストラクションを考慮するモデルなど多様で高性能な埋め込みモデルが提案されている．そこで本稿における実験では，MTEB リーダーボード[3]で上位に位置する文埋め込みモデルと，現在では性能の高いモデルはないものの以前は広く利用されていた単語埋め込みの平均に基づくモデルの計 8 モデルについて比較を行った．具体的には，Glove による単語埋め込みモデルの平均[4]，テキストの類似度判定タスクのパフォーマンス向上を図った Sentence-BERT[9] 方式のモデル roberta-large-nli-mean-tokens，all-mpnet-base-v2，より幅広いタスクのパフォーマンス向上を図った T5 ベースのモデル sentence-transformers/sentence-t5-large[10]，sentence-transformers/gtr-t5-large[11]，プロンプトに基づくコンテキストを考慮するモデル embaas/sentence-transformers-e5-large-v2[12]，それら同等の性能をプロンプトなしでも達成可能なモデル thenlper/gte-large[13]，インストラクションをに基づいて埋め込み位置を配置するモデル hkunlp/instructor-large[14] [5]を利用した．

3.2 各タスクにおける候補 LLM の正答率

まず，タスクごと最も性能が高い LLM が異なることを確認するため，各タスクにおける各候補 LLM の性能を評価した．表 1 に各 LLM の平均正答率を，図 1 には各タスクごとに最も高い正答率の LLM を示す．図 1 によれば，OpenChat-3.5-1210 が平均的に最も強いモデルである一方，SlimOrca-Mistral-7B は MMLU に強く，OpenHeremes-2.5-Mistral-7B は BBH-MC に強いなど LLM によって異なる傾向がみられる．また，MetaMath-Mistral-7B はどのタスクでも最も正答率の高い LLM に含まれていないものの，表 1 によれば GSM8K タスクにおいて 3 番目に正答率の高いモデルとなっており，各モデルで適不適があることが確認できる．

Model	Avg	BBH-MC	GSM8K	MMLU
OpenChat-3.5-1210	**0.498**	**0.533**	0.455	0.488
SlimOrca-Mistral-7B	0.495	0.460	0.587	**0.504**
OpenHermes-2.5-Mistral-7B	0.468	**0.533**	**0.705**	0.371
OpenOrca-Mistral-7B	0.446	0.498	0.447	0.415
Mistral-7B-Instruct-v0.2	0.418	0.399	0.375	0.439
Dolphin-2.6-Mistral-7b	0.384	0.414	0.436	0.353
MetaMath-Mistral-7B	0.251	0.187	0.568	0.225
Avg	0.423	0.432	0.510	0.399

表 1 各候補 LLM の正答率（各行がモデル，各列がタスクを示す．各データセットで最も正答率の高いモデルを太字で示す．）

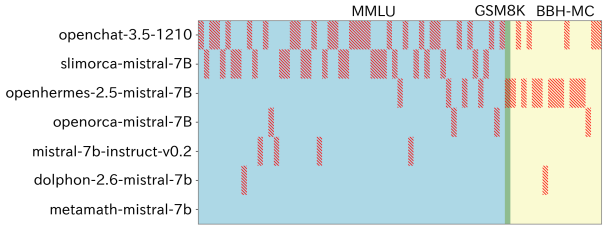

図 1 各タスクにおいて最も正答率の高い LLM を可視化した図（各行がモデル，各列がタスクを示す．各タスクで最も正答率の高いモデルを斜線で示す．）

3.3 各埋め込みモデルを用いた routing の結果

本節では，埋め込みモデルにより routing 性能がどのように変化するかについて述べる．routing 手法による推定正答率の最も高い top-1 LLM の回答を用いるだけでなく，ランキング上位の複数 LLM の回答を用いる使用事例も考えられる．例えば，使用者は top-n の回答を確認して，その中から最終的な回答を選択することができる．そのため，top-1 LLM へ routing した際の正答率のほか，routing 手法を LLM のランキング手法とみなしたときの性能を併せて評価した．具体的には，まず LLM 群が生成した回答に基づいて，正答が上位に位置する理想的なランキングを用意する．そして，推定正答率に基づくランキングと理想的なランキングの spearman の順位相関係数と NDCG@5 を算出した．表 2 に，各埋め込みモデルに基づいて top-1 LLM へ routing した場合の正答率と，相関係数，NDCG@5 の結果を

3） https://huggingface.co/spaces/mteb/leaderboard
4） https://huggingface.co/sentence-transformers/average_word_embeddings_glove.6B.300d
5） InstructOR におけるインストラクションは，’Represent the task domain for estimating how the LLM answers well: ’ とした．

示す．これによると，7 つの候補 LLM の平均正答率 (0.394) やベストモデルである OpenChat-3.5 の正答率 (0.498) に比べ，いずれの埋め込みモデルによる routing も高い正答率 (0.528 ~ 0.545) が得られたことが確認できる．ただし，埋め込みモデル間で正答率，相関係数，NDCG@5 のいずれもほとんど差がみられない．さらに，タスク/インスタンス単位で理想的な routing ができた際に達成されるオラクルの正答率について算出したところ，依然として routing に改善の余地があることが確認できる．

Embedding Model	Spearman	NDCG@5	Routing Top1 Acc
glove_average_word_embeddings	0.226	0.623	0.528
roberta-large-nli-mean-tokens	0.227	0.622	0.533
all-mpnet-base-v2	0.233	0.625	0.534
sentence-t5-large	0.228	0.623	0.535
gtr-t5-large	0.227	0.625	0.540
e5-large-v2	0.224	0.623	0.545
gte-large	0.220	0.621	0.530
instructor-large	0.226	0.623	0.530
Average of 7 LLMs	-	-	0.394
Best LLM (OpenChat-3.5-1210)	-	-	0.498
Oracle by Task[7]	-	-	0.584
Oracle by Instance (proposal)	-	-	0.865

表 2　各埋め込みモデルに基づいて Top1 LLM へ routing した場合の相関係数，NDCG@5，正答率（各行は埋め込みモデルを示す．また，正答率の比較のため，候補 LLM の平均とベストとタスク/インスタンス単位で理想的な routing ができた際に達成されるオラクルの正答率を示す．）

次に，僅差であるが最も routing 性能の高い埋め込みモデル e5-large-v2 を用いて routing の誤り事例を分析した．ここで，図 2 に routing された LLM と routing すべき LLM の関係を示す．これによれば，本来多様なモデルの回答に正答が含まれているにも関わらず平均的に最も強いモデルである OpenChat-3.5 へ誤って割り当ててしまうことが多い傾向がある．逆に，弱いモデルへ誤って割り当ててしまう事例はほとんど見られない．

3.4　上位 LLM 群への routing と多数決による集約の結果

top-1 モデルのみを利用する場合には依然として routing 誤りや生成失敗による誤答の懸念が残る．そこで，少数の誤答を省くことが可能な多数決を併用することによりランダムにサンプリングされた LLM による多数決よりもさらに性能が向上することを検証する．図 3 に，e5-latge-v2 に基づいて routing された上位 w モデルを用いて多数決とった場合，および，ランダムにサンプリングされた w モデルによって多数決をとった場合の正答率を示す．ここで，図中の点 A,B の大小関係により routing で

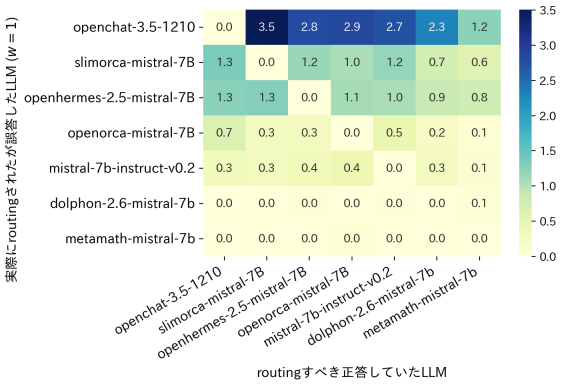

図 2　誤答インスタンスにおける routing すべき LLM と実際に routing された LLM の関係（行列の (i, j) 成分は i 番目の LLM に routing されたものの誤答しており，実際には j 番目の LLM が正答であったインスタンスがテストデータに占める割合を示す．単位は%）

得られた上位 2 モデルによる多数決は，routing による top1 モデルよりも性能を改善できる．また，A,C の大小関係により，ランダムに選ばれた 2 モデルによる多数決に比べて，routing による top1 モデルの方が高い性能を得られる上，推論に必要なコストを半減できることが確認できた．

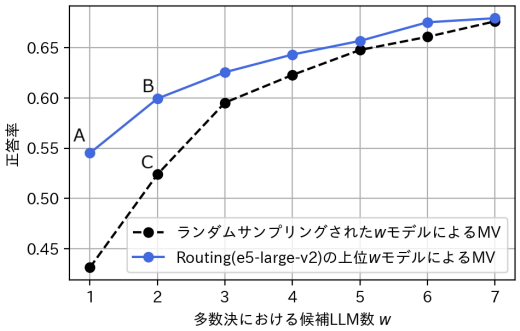

図 3　候補 LLM 数とそれらの回答を多数決させたときの正答率の関係（x 軸は多数決における候補 LLM 数 w を，y 軸は多数決により集約された回答の正答率を示す．）

4　おわりに

本研究では，専門性の異なる多様な LLM が利用可能な状況において，インスタンスに応じて適切な LLM を選択する routing 手法で利用される埋め込みモデルと，routing と多数決併用の効果について分析を行った．その結果，埋め込みによる性能の差異はほとんど無い一方，全タスクで平均的に良い回答を行う LLM に偏ってタスクを割り当てるため，依然として routing 手法に改善の余地があることが分かった．また，複数の候補モデルを routing 手法で選択した後に，多数決によって誤答を省くことで，更に性能が改善することも確認できた．

参考文献

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. Canton-Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. **CoRR**, Vol. abs/2307.09288, , 2023.

[2] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de Las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. Le Scao, T. Lavril, T. Wang, T. Lacroix, and W. El Sayed. Mistral 7b. **ArXiv**, Vol. abs/2310.06825, , 2023.

[3] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, B. Hui, L. Ji, M. Li, J. Lin, R. Lin, D. Liu, G. Liu, C. Lu, K. Lu, J. Ma, R. Men, X. Ren, X. Ren, C. Tan, S. Tan, J. Tu, P. Wang, S. Wang, W. Wang, S. Wu, B. Xu, J. Xu, A. Yang, H. Yang, J. Yang, S. Yang, Y. Yao, B. Yu, H. Yuan, Z. Yuan, J. Zhang, X. Zhang, Y. Zhang, Z. Zhang, C. Zhou, J. Zhou, X. Zhou, and T. Zhu. Qwen technical report. **CoRR**, Vol. abs/2309.16609, , 2023.

[4] Y. Luo, Z. Yang, F. Meng, Y. Li, J. Zhou, and Y. Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. **ArXiv**, Vol. abs/2308.08747, , 2023.

[5] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma. Investigating the catastrophic forgetting in multimodal large language models. **ArXiv**, Vol. abs/2309.10313, , 2023.

[6] Y. Lin, L. Tan, H. Lin, Z. Zheng, R. Pi, J. Zhang, S. Diao, H. Wang, H. Zhao, Y. Yao, and T. Zhang. Speciality vs generality: An empirical study on catastrophic forgetting in fine-tuning foundation models. **ArXiv**, Vol. abs/2309.06256, , 2023.

[7] T. Shnitzer, A. Ou, M. Silva, K. Soule, Y. Sun, J. Solomon, N. Thompson, and M. Yurochkin. Large language model routing with benchmark datasets, 2023.

[8] K. Lu, H. Yuan, R. Lin, J. Lin, Z. Yuan, C. Zhou, and J. Zhou. Routing to the expert: Efficient reward-guided ensemble of large language models, 2023.

[9] N. Reimers and I. Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, **Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)**, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics.

[10] J. Ni, G. Hernandez Abrego, N. Constant, J. Ma, K. Hall, D. Cer, and Y. Yang. Sentence-t5: Scalable sentence encoders from pretrained text-to-text models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Findings of the Association for Computational Linguistics: ACL 2022**, pp. 1864–1874, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[11] J. Ni, C. Qu, J. Lu, Z. Dai, G. Hernandez Abrego, J. Ma, V. Zhao, Y. Luan, K. Hall, M.-W. Chang, and Y. Yang. Large dual encoders are generalizable retrievers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 9844–9855, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[12] L. Wang, N. Yang, X. Huang, B. Jiao, L. Yang, D. Jiang, R. Majumder, and F. Wei. Text embeddings by weakly-supervised contrastive pre-training. **ArXiv**, Vol. abs/2212.03533, , 2022.

[13] Z. Li, X. Zhang, Y. Zhang, D. Long, P. Xie, and M. Zhang. Towards general text embeddings with multi-stage contrastive learning. **ArXiv**, Vol. abs/2308.03281, , 2023.

[14] H. Su, W. Shi, J. Kasai, Y. Wang, Y. Hu, M. Ostendorf, W. t. Yih, N. A. Smith, L. Zettlemoyer, and T. Yu. One embedder, any task: Instruction-finetuned text embeddings. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 1102–1121, Toronto, Canada, July 2023. Association for Computational Linguistics.

[15] C. J. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. N. Hullender. Learning to rank using gradient descent. **Proceedings of the 22nd international conference on Machine learning**, 2005.

[16] A. Lewkowycz, A. J. Andreassen, D. Dohan, E. Dyer, H. Michalewski, V. V. Ramasesh, A. Slone, C. Anil, I. Schlag, T. Gutman-Solo, Y. Wu, B. Neyshabur, G. Gur-Ari, and V. Misra. Solving quantitative reasoning problems with language models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, **Advances in Neural Information Processing Systems**, 2022.

[17] X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. In **The Eleventh International Conference on Learning Representations**, 2023.

[18] D. Oniani, J. Hilsman, H. Dong, F. Gao, S. Verma, and Y. Wang. Large language models vote: Prompting for rare disease identification, 2023.

[19] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. **Proceedings of the International Conference on Learning Representations (ICLR)**, 2021.

[20] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. **ArXiv**, Vol. abs/2110.14168, , 2021.

[21] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H. W. Chung, A. Chowdhery, Q. Le, E. Chi, D. Zhou, and J. Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Findings of the Association for Computational Linguistics: ACL 2023**, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics.

[22] L. L. Yu, W. Jiang, H. Shi, J. Yu, Z. Liu, Y. Zhang, J. T. Kwok, Z. Li, A. Weller, and W. Liu. Metamath: Bootstrap your own mathematical questions for large language models. **ArXiv**, Vol. abs/2309.12284, , 2023.

[23] W. Lian, B. Goodson, E. Pentland, A. Cook, C. Vong, and Teknium. Openorca: An open dataset of gpt augmented flan reasoning traces. https://https://huggingface.co/Open-Orca/OpenOrca, 2023.

[24] W. Lian, B. Goodson, E. Pentland, A. Cook, C. Vong, and Teknium. Slimorca: An open dataset of gpt-4 augmented flan reasoning traces, with verification, 2023.

[25] G. Wang, S. Cheng, X. Zhan, X. Li, S. Song, and Y. Liu. Openchat: Advancing open-source language models with mixed-quality data. **ArXiv**, Vol. abs/2309.11235, , 2023.