対話型検索のためのクエリ書き換えにおける大規模言語モデルの効果分析

阿部健也 [1]　竹岡邦紘 [2]　小山田昌史 [2]
[1] 筑波大学大学院　[2]NEC データサイエンスラボラトリー
s2321672@u.tsukuba.ac.jp
{k_takeoka,oyamada}@nec.com

概要

対話型検索は，対話の最後に与えられるユーザの質問に適合する文書を検索するタスクであり，対話の文脈に依存した質問を適切に書き換えるクエリ書き換えが主要なアプローチの一つである．大規模言語モデル (LLM) によって作成したクエリ書き換えで学習する手法は提案されているが，どのような場面でどの程度有効かは十分に調べられていない．本研究では，LLM による書き換えやそのデータを学習したモデルがどのような状況において有効であり，人手のデータを使う場合に比べてどのような性質があるかを調査した．実験の結果，LLM を利用すると元の質問との重複を避ける傾向にあり，元の質問から大きく書き換える必要のない事例を苦手とすることがわかった．

1　はじめに

対話型検索 (conversational search) は，対話において最後に与えられたユーザの質問に適合する文書を検索するタスクである．近年は TREC で対話型検索に取り組むコンペティション [1] が開催され，大規模なデータセット [2] が作成されるなど活発に研究されている．対話型検索が一般的な検索と最も異なる点は，省略や共参照が頻繁に発生する対話の文脈に依存した自然文の質問であり，対話履歴に基づいて質問の意図を理解する必要があるところだ．

この課題に対するアプローチには，対話型クエリ書き換え (Query Rewriting) と対話型密検索 (Conversational Dense Retrieval) の 2 つがある [3]．対話型密検索は対話をそのまま入力して適切な検索結果が得られるように学習するアプローチ [4, 5, 6] であり，一方で対話型クエリ書き換えは一般的な検索エンジンに入力するクエリへと変換するアプローチである．しかし，対話型密検索には検索モデルが固定されてしまうという欠点がある．一方で対話型クエリ書き換えは検索モデルに依存しないという点で実用性が高く，T5 を利用して質問を書き換える T5QR[7] や BERT を利用して対話履歴から必要な単語を選択して拡張する QuReTeC[8] などが提案されている．最近提案された LLM4CS[9] では LLM を用いてクエリの書き換えを行うことで従来の手法を上回る精度を達成し，いくつかの工夫を加えることで人手書き換えの精度も上回った．この他にもクエリ書き換えの手法は多数提案されている [10, 11, 12, 13, 14] が，これらの手法の多くは人手やルールベースの書き換えデータを学習している．しかし，人手の書き換えは作成コストが高く，ルールベースの書き換えは精度が高くない．

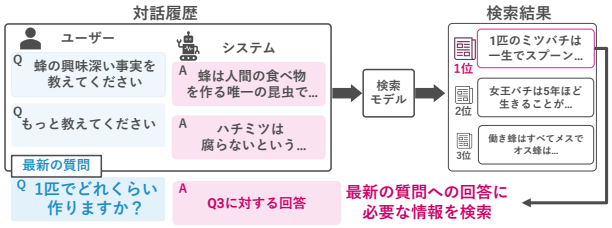

図 1　対話型検索の概要 (対話型検索では図中の検索結果を評価する．)

この課題を解決するために大規模言語モデル (LLM) で正解となる書き換えデータを生成しモデルを学習するという手法が提案されている [15]．この論文は LLM の書き換えを学習データとした書き換えモデルで検索精度を一部改善することに成功した．これらは人手の書き換えの代わりに LLM による書き換えを用いる方が有効な可能性があることを示唆している．しかし，[15] では一部のケースにおいて LLM による書き換えをベースにした手法が有効に動作しないケースが報告されていたものの，その明確な理由は特定することができていなかった．

LLM による書き換えは人手による書き換えと比較するとコストが低く，大量のデータを容易に生成できるため，どのような状況で LLM が効果的に機能するか，またはそうでないかが明らかになれば生成したデータに対してフィルタリング等 [16] を行うことで低コストで高品質な学習データを用意できると考えられる．

そこで，本研究では既存研究 [15] の実験で用いられたデータセットに加えて，2 つのデータセットを用いて，人手と LLM によるクエリの書き換えが検索結果に及ぼす影響，および学習データとしてのそれぞれの書き換えの効果についての分析を行った．実験では 3 つのデータセットの結果の評価だけでなく，成功例や失敗例の分析を行い，以下のような知見が得られた．**(1)** LLM による書き換えは元の質問の単語との重複を避ける傾向にある．**(2)** LLM による書き換えを学習したモデルも同様に単語との重複を避ける傾向がある．**(3)** LLM とその書き換えで学習したモデルは元の質問から大きく書き換える必要のない事例を苦手とする．

2 実験

2.1 問題設定

対話型検索は，対話を通じたユーザからの質問 q_i とその対話履歴 $C_i = \{q_1, r_1, q_2, r_2, \ldots, q_{i-1}, r_{i-1}\}$ が与えられた時に，質問 q_i に対する返答 r_i に必要な文書 d_i^+ を文書集合 D から検索するタスクである．対話型検索においてユーザの質問 q は代名詞による置き換えや単語の省略が存在するため，それまでの対話履歴 C を利用してそれらを解決して，質問 q を $\hat{q}$ に変換することをクエリ書き換えと呼ぶ．この書き換えた $\hat{q}$ は文脈に依存せずに情報要求を表す簡潔な質問であることが望ましい．このような $\hat{q}$ に変換することで検索モデルの種類に関わらず検索ができることがクエリ書き換えの強みである．よって，本研究ではクエリ書き換えに注目する．本研究ではクエリ書き換え手法のベースとして T5QR[7] を用いる．T5QR は T5 の入力として質問とそれまでの対話履歴を与え，対話履歴に基づいて書き換えた質問を出力する．この時，T5 は人間の書き換えを正解として学習を行う．クエリ書き換え手法は他にも存在するが，LLM が基本的に自然文を生成するため，今回は自然文の書き換えを生成する T5QR を選択した．

2.2 LLM による書き換えデータ生成

人間と LLM の書き換えモデル（T5QR）の学習データとしての効果を検証するために LLM で書き換えを生成する際の方法について述べる．入力として用いるのはユーザの質問 q_i とその対話履歴 C_i である．具体的に LLM に適切なクエリ書き換えを生成させるためのプロンプトは付録 B 表 B に示した．また，同様に付録 B 表 5 に，[15] において使用されたプロンプトも記載した．この 2 つのプロンプトはどちらも対話履歴に基づいてユーザの質問を書き換えるように指示している[1)]．[15] では学習データの生成時に理想的な書き換えを含んだ対話の例をいくつか提示する Few-Shot な設定であったが，本実験の生成時には Zero-Shot な設定で生成を行う．

2.3 データセット

以前の研究 [11, 12, 13] に従って，in-domain 設定のためのデータセットとして QReCC[2] を利用した．また，本実験ではデータセットとして TREC CAsT 2019，2020 [1, 17] も利用する．QReCC は 8 万件のターン（質問と返答のペア）が含まれる英語での対話 1.4 万件のデータセットである．コーパスには Web ページ 100 万件を分割したパッセージ 5400 万件が含まれる．追加のデータセットとして利用する CAsT-19，20 には対話がそれぞれ 50，25 件で対話ターンが 479，208 ずつ用意され，検索対象はパッセージ 3900 万件である．3 つのデータセットの質問には適合文書が与えられているため，この適合文書をどの程度検索できているかで手法を評価する．

2.4 実験設定

クエリ書き換えの性能を比較するために 2 つの検索モデルで実験を行った．検索モデルは従来の研究で用いられていることの多い BM25[18] と GTR[19] を使用した．BM25 のハイパーパラメータは $k1 = 0.82$ と $b = 0.68$ と設定した．GTR は T5 のエンコーダ部分を用いたベクトル検索モデルであり，クエリの最大入力トークン数を 64，文書の最大入力トークン数を 384 とした．上位 100 件を検索し，評価指標は MAP@100，MRR@100，Recall@10(R@10) を用いた．全ての指標が高い値ほど良い評価で，MRR は検索結果に最初に登場する適合文書が上

1) 2 つのプロンプトで結果に大きな影響はなかった．

表 1 異なる書き換え手法を用いた場合の対話型検索の評価結果.

		QReCC			CAsT-19			CAsT-20		
	書き換え手法	MAP	MRR	R@10	MAP	MRR	R@10	MAP	MRR	R@10
BM25	質問のみ	8.85	9.28	15.49	8.40	31.90	3.89	2.87	9.92	2.54
	人手	38.30	39.66	62.55	21.84	61.32	10.95	12.98	37.73	14.26
	LLM	34.97	36.28	56.22	16.93	57.09	9.37	10.48	33.98	11.37
	T5QR（人手）	32.83	34.04	54.21	**17.88**	**54.44**	**9.07**	8.23	26.11	8.49
	T5QR（LLM）	**33.43**	**34.72**	**55.59**	16.27	53.45	8.48	**8.62**	**27.74**	**9.14**
GTR	質問のみ	11.49	12.11	18.73	11.53	43.56	6.38	7.44	23.14	6.38
	人手	41.22	43.1	66.05	24.2	75.87	15.02	24.9	63.65	22.5
	LLM	38.92	40.74	62.66	20.4	66.04	12.61	21.66	56.71	19.89
	T5QR（人手）	36.42	38.16	59.58	**21.16**	**69.47**	**12.97**	**17.93**	**47.94**	**15.78**
	T5QR（LLM）	**37.04**	**38.75**	**60.93**	19.82	63.01	12.07	17.47	46.58	15.67

位であるほど高い評価となる．MAP と Recall はどちらも適合文書をできるだけ多く検索できることを重視しており，特に MAP は順位も考慮する．書き換えモデルの T5QR では入力の最大を 384 として出力の最大を 64 とした．また，書き換えモデルとしての LLM には OpenAI が提供する API の gpt-3.5-turbo-instruct[2)3)] を使用した．

2.5 比較手法

実験において書き換え手法として比較を行うものについて述べる．**質問のみ**: ユーザの質問をそのまま使う．**人手**: 各データセットで提供されている人手による書き換えを使う．**LLM**: LLM で書き換えたクエリを使う．**T5QR(人手)**: 人手で作った書き換えデータを学習した T5 モデルで生成したクエリを使う．**T5QR(LLM)**: LLM で作った書き換えデータを学習した T5 モデルで生成したクエリを使う．

2.6 実験結果

表 1 は 3 つのデータセットにおける評価結果である．T5QR（人手）と T5QR（LLM）の間で評価が高かったものを太字で，全体で最も高い評価のものを下線で表す．実験の結果からクエリ書き換え手法に関して以下のことがわかった．**(1)** 全てのデータセット，検索手法において人手での書き換えの精度が最も高い．これは [9] の結果と同様で LLM を最大限に活用するような方法では人間を超える精度が発揮できる一方で，今回の実験のように Zero-Shot な設定で書き換えのみを行う場合，人手の結果を超えるのは難しい．**(2)** QReCC においては T5QR の学習に人手よりも LLM の書き換えを使った場合の方が精度が高い．[15] のような Few-shot 設定ではなく，今回のような Zero-Shot な設定においても，LLM の書き換えを学習に使うことは有効であると考えられる．また (1) と (2) の結果から，直接の書き換えとしては人手の方が良いにも関わらず，学習の効果としては LLM の方が有効に作用するケースがあると分かる．**(3)** CAsT-19，20 では CAsT-20 の BM25 を除いて，T5QR（人手）が T5QR（LLM）を上回る．**(4)** CAsT-19 でのみ直接 LLM を利用するよりも人間の書き換えで学習した T5QR で書き換えた方が検索結果が良い．今回の設定では他のデータで学習済みの検索モデルをそのまま利用しているため，クエリ書き換えの質が検索精度に大きく影響する．そのため (3)，(4) についての結果ではデータセット間でのクエリの性質の違いが関わっていると考えられ，表 2.5 の実例分析において詳しく述べる．

2.7 実例分析

本節では実例分析により人間や LLM の書き換えの傾向にどのような違いがあるのかを明らかにする．まず，書き換え前後における 1 クエリあたりの単語数と重複数を図 2.5 に示す．LLM による書き換えは一貫して人手による書き換えよりも単語数が多いにも関わらず，元の質問との重複数は人手よりも少ない．この傾向はそれぞれの書き換えを学習データとして利用した際にも同様である．また，人手と LLM の書き換えについて CAsT における実例を表 2.5 に示す．CAsT-19，20 に共通して人間の書

2) https://platform.openai.com/docs/models/gpt-3-5
3) 同時期に実験に取り組んでおり，[15] と実験設定が異なるため結果が異なるものの，API で使用するモデルバージョンの違いが主要因であることを確認している．

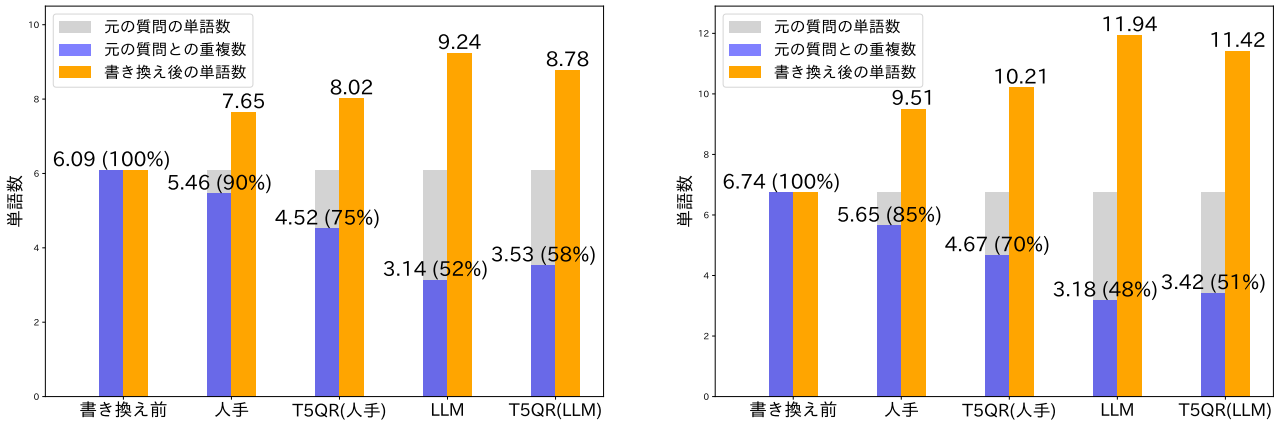

図 2　書き換え前後における 1 クエリあたりの単語数と重複数．() 内は書き換え前後の単語の重複数を書き換え前の単語数で割った値である．（左：CAsT-19　右：CAsT-20）

	書き換え前	人間による書き換え	LLM による書き換え
CAsT-19	(a) How was it received?	How was the Neverending Story film received?	✓ What were the reviews for the Neverending Story film?
	(b) How has it changed the way TV is watched?	How has Netflix changed the way TV is watched?	✗What impact has Netflix had on television viewing habits?
	(c) What is anemia?	What is anemia?	✗What are the symptoms and causes of anemia?
CAsT-20	(d) When should you consider surgery?	When should you consider shoulder surgery?	✓ At what point should surgery be considered for a shoulder injury?
	(e) Why are so many dying?	Why are so many bees dying?	✗What is causing the high death rate among bees?

表 2　質問の書き換えの成功例と失敗例: LLM の書き換えを人間の書き換えと比較して検索結果の評価結果が全ての指標で上がっているものを成功（✓），下がっているものを失敗✗としている．

き換えは文章の形が大きく変わることは少なく，省略や共参照の解決のみにとどまることが多い．特に CAsT-19 では，(c) の例のように，書き換えがなくても検索できているケースが確認された．これらのことから，LLM による書き換えが必ずしも質問の意図を損なわない一方で質問を冗長にしてしまい，検索精度のばらつきが生じる原因となっていると考えられる．他にも例 (e) の “Why are so many bees dying?” では dying が検索において重要な単語であり，その単語が LLM による書き換えでは失われてしまうため検索に失敗している．これは LLM に「書き換えること」を指示していることや LLM 自体の性質が原因であると考えられる．この性質から 2.6 章の (3)，(4) の結果について以下のような考察ができる．CAsT-19 では大きな書き換えを必要としない質問を特に多く含むため，LLM が有効に作用せず，大きく書き換えを行わない人間の書き換えを学習した T5QR に劣ったと考えられる．同様に，CAsT-19 のみで T5QR(LLM) が BM25，GTR の両方で T5QR（人手）を下回ったことも単語の重複を避ける性質を学習しているためであると考えられる．一方で，CAsT-19 と CAsT-20 の成功例があるように冗長であることは悪いケースのみではなく，成功例の (a) や (d) にあるように映画の人気についての話題では “review”，肩の怪我についての話題で “injury” という単語が追加されるような書き換えにより検索結果が改善するケースも存在する．

3　結論

本研究では対話型検索におけるクエリ書き換えモデルの学習において，正解データを人間と LLM の書き換えとした場合の効果の違いについて分析を行った．その結果，LLM は質問の意味を保った書き換えが可能であるものの，検索において重要な単語を書き換えてしまうケースがあることが明らかになった．これは，LLM が単語の重複を避けようとするという性質によるもので，特に元の質問から大きく書き換える必要のない質問を苦手とすることがわかった．また，LLM による書き換えで学習した T5 も同様の性質を示した．今後の課題としては，LLM が人手の書き換えと比較して有効である事例にどのような傾向を明らかにすること，本実験を踏まえて LLM を利用したロバストなクエリ書き換えデータ生成方法を検討することが挙げられる．

参考文献

[1] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. Trec cast 2019: The conversational assistance track overview. arXiv preprint arXiv:2003.13624, 2020.

[2] Raviteja Anantha, Svitlana Vakulenko, Zhucheng Tu, Shayne Longpre, Stephen Pulman, and Srinivas Chappidi. Open-domain question answering goes conversational via question rewriting. arXiv preprint arXiv:2010.04898, 2020.

[3] Jianfeng Gao, Chenyan Xiong, Paul Bennett, and Nick Craswell. Neural approaches to conversational information retrieval. arXiv preprint arXiv:2201.05176, 2022.

[4] Shi Yu, Zhenghao Liu, Chenyan Xiong, Tao Feng, and Zhiyuan Liu. Few-shot conversational dense retrieval. In Proceedings of the 44th International ACM SIGIR Conference on research and development in information retrieval, pp. 829–838, 2021.

[5] Kelong Mao, Zhicheng Dou, Hongjin Qian, Fengran Mo, Xiaohua Cheng, and Zhao Cao. Convtrans: Transforming web search sessions for conversational dense retrieval. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 2935–2946, 2022.

[6] Hongjin Qian and Zhicheng Dou. Explicit query rewriting for conversational dense retrieval. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pp. 4725–4737, 2022.

[7] Sheng-Chieh Lin, Jheng-Hong Yang, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. Conversational question reformulation via sequence-to-sequence architectures and pretrained language models. arXiv preprint arXiv:2004.01909, 2020.

[8] Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. Query resolution for conversational search with limited supervision. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 921–930, 2020.

[9] Kelong Mao, Zhicheng Dou, Haonan Chen, Fengran Mo, and Hongjin Qian. Large language models know your contextual search intent: A prompting framework for conversational search. arXiv preprint arXiv:2303.06573, 2023.

[10] Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. Contextualized query embeddings for conversational search. arXiv preprint arXiv:2104.08707, 2021.

[11] Fengran Mo, Jian-Yun Nie, Kaiyu Huang, Kelong Mao, Yutao Zhu, Peng Li, and Yang Liu. Learning to relate to previous turns in conversational search. arXiv preprint arXiv:2306.02553, 2023.

[12] Kelong Mao, Zhicheng Dou, Bang Liu, Hongjin Qian, Fengran Mo, Xiangli Wu, Xiaohua Cheng, and Zhao Cao. Search-oriented conversational query editing. In Findings of the Association for Computational Linguistics: ACL 2023, pp. 4160–4172, 2023.

[13] Fengran Mo, Kelong Mao, Yutao Zhu, Yihong Wu, Kaiyu Huang, and Jian-Yun Nie. Convgqr: Generative query reformulation for conversational search. arXiv preprint arXiv:2305.15645, 2023.

[14] Zeqiu Wu, Yi Luan, Hannah Rashkin, David Reitter, Hannaneh Hajishirzi, Mari Ostendorf, and Gaurav Singh Tomar. Conqrr: Conversational query rewriting for retrieval with reinforcement learning. arXiv preprint arXiv:2112.08558, 2021.

[15] Fanghua Ye, Meng Fang, Shenghui Li, and Emine Yilmaz. Enhancing conversational search: Large language model-aided informative query rewriting. arXiv preprint arXiv:2310.09716, 2023.

[16] Zhuyun Dai, Vincent Y Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B Hall, and Ming-Wei Chang. Promptagator: Few-shot dense retrieval from 8 examples. arXiv preprint arXiv:2209.11755, 2022.

[17] Jeffrey Dalton, Chenyan Xiong, and Jamie Callan. Cast 2020: The conversational assistance track overview. In In Proceedings of TREC, 2021.

[18] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. Foundations and Trends® in Information Retrieval, Vol. 3, No. 4, pp. 333–389, 2009.

[19] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernández Ábrego, Ji Ma, Vincent Y Zhao, Yi Luan, Keith B Hall, Ming-Wei Chang, et al. Large dual encoders are generalizable retrievers. arXiv preprint arXiv:2112.07899, 2021.

[20] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations, pp. 38–45, 2020.

A　実験設定の詳細

表 3　データセットのサイズ

train データ	dev データ	test データ
57150	6351	8209（16451）

QReCC のデータセットの詳細は 3 の通りで，test データは 16451 件用意されているが，[13, 14, 15] などと同様に注釈づけをやり直して作られた 8029 件をテストデータとして扱う．また，学習時には T5 のベースとして transformers ライブラリ [20] の t5-base[4] を利用した．学習率は 5×10^{-5}，バッチサイズは 48 で 1 epoch 毎に dev データでの損失を計算し，2 回連続で損失が下がらなければ学習を止める early stopping を行った．学習はシードを変えて 3 回行い，それぞれのモデルの書き換えで検索を行い，3 回の平均をとった．

B　入力したプロンプトの例

本実験と [15] のプロンプトを表 B と表 5 に示す．本実験で用いた gpt-3.5-turbo-instruct では 2 つのプロンプトで大きな差は見られなかった．[15] でも同じような意味の他のプロンプトであれば適用できると述べられている．

Please infer the intent of the input question from the context and reformulate into a question for Google search to de-cotextualize it.
Question: How does water freeze?
Response: Freezing happens when the molecules...
Question: What happens to its molecules?
Response: When frozen, water molecules take...
Question: Why isn't the bottom of the ocean frozen?
Rewrite:

表 4　本実験で使用したプロンプト

表 5　[15] による Zero-Shot 設定のプロンプト

Given a question and its context, decontextualize the question by addressing coreference and omission issues. The resulting question should retain its original meaning and be as informative as possible, and should not duplicate any previously asked questions in the context.
Context: [Q: How does water freeze?
A: Freezing happens when the molecules...
Q: What happens to its molecules?
A: When frozen, water molecules take...]
Question: [Why isn't the bottom of the ocean frozen?]
Rewrite:

表 6　[15] による Zero-Shot 設定のプロンプト

4）https://huggingface.co/t5-base

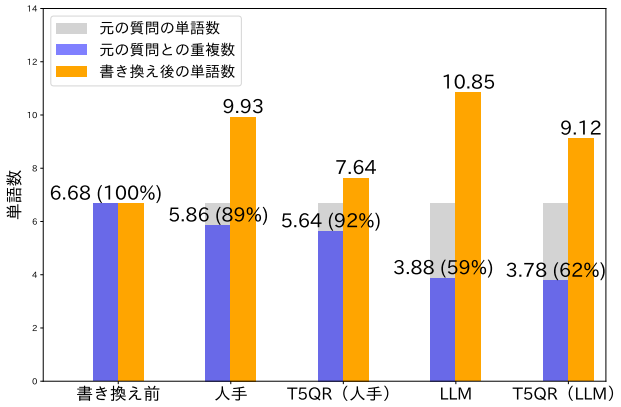

図 3　QReCC における書き換え前後における 1 クエリあたりの単語数と重複数

	QReCC		CAsT-19		CAsT-20	
	MRR	R@10	MRR	R@10	MRR	R@10
我々の設定	36.28	56.22	57.09	9.37	33.98	11.37
[15] の設定	43.85	62.32	48.13	7.90	27.59	9.08

表 7　LLM のプロンプト，バージョンの違いによる評価結果への影響

C　QReCC における単語数と重複数

QReCC における書き換え前後での単語の重複数とその割合を図 D に記載した．CAsT のデータと同様に LLM の書き換えでは単語数が長く，元の質問からの重複を避ける傾向にあることがわかる．

D　LLM のバージョンによる評価結果の違い

また，[15] の設定と我々の設定での結果の比較を表 D に示した．[15] の設定は QReCC での結果は良いものの出力が安定しないため CAsT データセットでは本実験の設定が上回っている．