Proportionate Diversification of Top-k LLM Results using Database Queries

Thinh On[1,*], Subhodeep Ghosh[1,2], Mengnan Du[1] and Senjuti Basu Roy[1]

[1]*New Jersey Institute of Technology*

[2]*HIT India*

Abstract

Result diversification aims to return relevant results that cover a variety of perspectives. Attribute-based diversification groups results by shared attributes (e.g., genre for movies) and selects a proportional number of items from each group based on their distribution in the underlying data. However, large language models (LLMs) are not designed to produce proportionally diverse results. In this work, we propose leveraging external data sources to determine the distribution of groups related to a query and prompt LLMs to produce proportionally diverse results. This can improve result diversity by representing groups in proportion to their prevalence. Specifically, we first argue the benefits of making top-k results from LLMs proportionally diverse. We then show how to use external benchmark databases to enable proportional diversity. Finally, we outline a framework that prompts LLMs with proportionality information from external data and discuss challenges in automating this process. Our approach provides a path to overcoming LLMs' limitations in producing proportionally diverse responses.

Keywords

top-k Diversification, large language models (LLMs), prompting LLMs, querying database

1. Introduction

The goal of result diversification is to return results that are relevant as well as cover different intents. In the data management community, returning top-k diverse results of a query has been extensively studied, and there exists many seminal works that propose objective functions and efficient algorithms to return results that are diverse and representative [1, 2, 3]. In this work, we study proportional diversity in top-k, which amounts to grouping items in the result set based on the query condition that share common attribute values (e.g., different genres for movies) and selecting only a limited number of items from each group that represent their proportional distribution. Large language models (LLMs), such as ChatGPT [4], Claude [5], Alpaca [6], Vicuna [7], etc, on the other hand, do not return results that are proportionally diverse. In this work, we first argue the benefits of making top-k results from LLMs proportionally diverse. We then present how to leverage external benchmark databases to enable proportional diversity. Finally, we outline a computational framework that prompts LLMs with proportionality information from external data sources and discuss research challenges to automate such an outcome.

As a concrete example, a leading open source LLM

*Corresponding author.
✉ to58@njit.edu (T. On); sg2646@njit.edu (S. Ghosh); mengnan.du@njit.edu (M. Du); senjutib@njit.edu (S. B. Roy)

ChatGPT provides the following response of the query below.

Example 1. Query : top-10 movies from different genres.

ChatGPT Response. *Certainly! Here's a list of top 10 movies from different genres to cater to a range of preferences:*

- *Action: "The Dark Knight" (2008) - A thrilling superhero film that explores the morally complex world of Batman as he faces off against the Joker.*
- *Comedy: "Anchorman: The Legend of Ron Burgundy" (2004) - A hilarious satire about a clueless news anchor and his antics in the male-dominated world of broadcasting.*
- *Drama: "The Shawshank Redemption" (1994) - A powerful drama set in a prison, following the friendship and resilience of two inmates in the face of adversity.*
- *Romance: "Eternal Sunshine of the Spotless Mind" (2004) - A unique love story that blends science fiction elements with an exploration of memories and relationships.*
- *Science Fiction: "Blade Runner" (1982) - A visually stunning dystopian film that delves into themes of artificial intelligence, identity, and what it means to be human.*
- *Fantasy: "The Lord of the Rings: The Fellowship of the Ring" (2001) - The first installment of the epic fantasy trilogy that follows a group of heroes on a quest to save Middle-earth from darkness.*

SQL response to find proportion of highly rated (average rating greater than 6.5) movies per genre

User Input (find 10 diverse movies based on genre) → Text To SQL Transform → Query an external SQL database

Step 1
```
SELECT genre, COUNT(genre)
From Movie group by genre;
+----------+
| Count    |
+----------+
| drama 20,000 |
+----------+
| action 15,000 |
+----------+
| comedy 25,000 |
+----------+
```

Step 2
```
SELECT Count (Movie_name)
From movie INNER JOIN Ratings
On
movie.M_id=Rating.Movie_id
where genre ='Drama'
group by Rating.(Movie_id)
Having Avg(rating) > 6.5
+----------+
| Count    |
+----------+
| 10,000   |
+----------+
```

Step 3
```
SELECT Count (Movie_name)
From movie INNER JOIN Ratings
On
movie.M_id=Rating.Movie_id
where genre ='Action'
group by Rating.(Movie_id)
Having Avg(rating) > 6.5
+----------+
| Count    |
+----------+
| 5,000    |
+----------+
```

Step 4
```
SELECT Count (Movie_name)
From movie INNER JOIN Ratings
On
movie.M_id=Rating.Movie_id
where genre ='Comedy'
group by Rating.(Movie_id)
Having Avg(rating) > 6.5
+----------+
| Count    |
+----------+
| 5,000    |
+----------+
```

SQL to Text Transform → **Calculate proportionate representation of movies per genre in 10 results** → Prompt Summarize (find 5 drama, 3 action, 2 comedy movies) → LLMs → Output

Movie

M_id	Movie name	genre	Actor_id	Production company_id	Award id
1	Citizen Kane	Drama	853453	863453	123
2	The Dark Knight	Action	853454	863453	124
3	Psycho	Thriller	853455	863453	179

Rating

Movie_id	User_id	rating	date
1	ABC	4	01/01/2018
2	ABC	1	12/10/2021
3	DEF	2	1/10/2002

User

user_id	User_name	contact_name	email
1	ABC	131655	of@gmail.com
2	DEF	131656	tf@gmail.com
3	GHI	161657	hf@gmail.com

Actor

actor_id	name	Gender	Birth country
853453	Orson Welles	M	US
853454	Christian Bale	M	UK
853455	Janet Leigh	F	US

Production Company

Prod company_id	Name	Owner	Headquarter
863453	RKO pictures	David Sarnoff	NYC
863454	Warner Bros	Warner Bros. Pictures Group	CA
863455	Paramount pictures	National Amusements	CA

Award

Award id	Award name	Award category id
123	Academy Award	3
124	Golden Globe	4
179	Edgar Award	1

Award Details

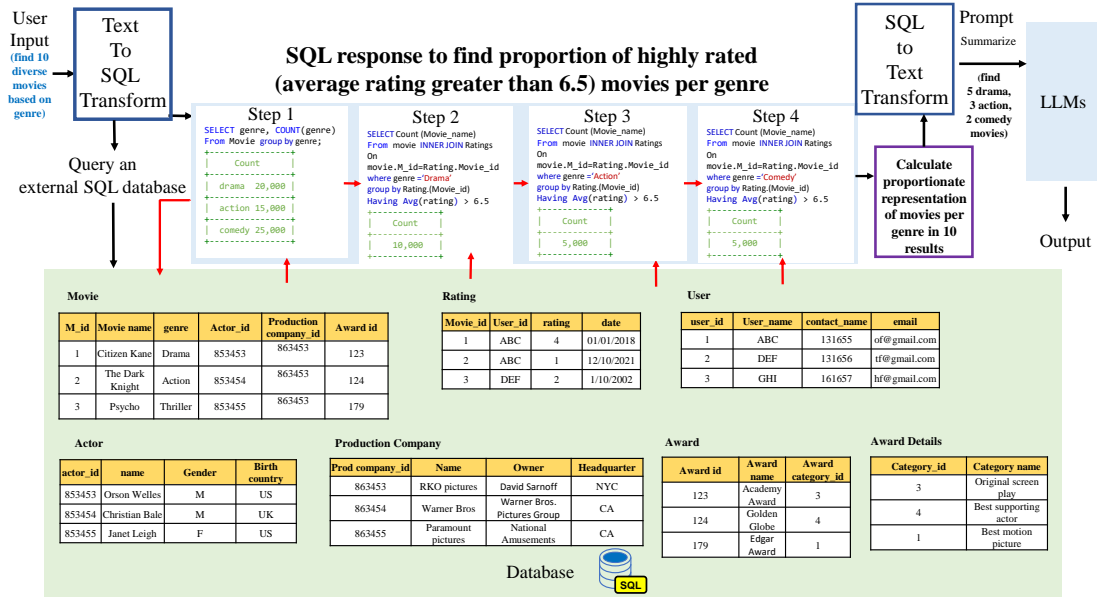| Category_id	Category name
3	Original screen play
4	Best supporting actor
1	Best motion picture

Database

Figure 1: Proposed Framework to Make LLMs Proportionally Diverse.

- *Thriller: "Psycho" (1960) - A classic suspenseful film by Alfred Hitchcock that explores the chilling story of a motel owner with a dark secret.*
- *Animated: "Toy Story" (1995) - The groundbreaking animated film that introduced audiences to a world where toys come to life, combining humor, heart, and adventure.*
- *Mystery: "Gone Girl" (2014) - A gripping thriller that keeps audiences on the edge of their seats as a man becomes the prime suspect in his wife's disappearance.*
- *Historical: "Schindler's List" (1993) - A poignant and harrowing portrayal of the Holocaust, based on the true story of Oskar Schindler, a German businessman who saved over a thousand Jewish lives.*

These represent a diverse range of genres and have achieved critical acclaim for their storytelling, performances, and impact on popular culture. It is easy to infer that the LLM as is returns one movie per genre (i.e., the proportional distribution of movies per genre is uniform). However, if an external data source is looked at (such as, IMDB database), it could be seen that the top-rated movies (average rating 6.5 for example) exhibit different proportional distributions per genre.

As a toy example, as shown in Figure 1, 10,000 out of 20,000 *Drama* movies have average ratings higher than 6.5, whereas, 5,000 out of 15,000 *Action*, and 5,000 out of 25,000 *Comedy* movies have similar high ratings, respectively. In proportion, therefore, it is desirable to see 5 *Drama, 3 Action,* and *2 Comedy* movies in the top-10 results, which represents the preference of the IMDB users more appropriately in the returned LLM answers. Such idea of proportionate representation is explored in multiple recent works [8, 9] and bears close connection to make the results diverse and fair.

Problem Definition 1. Proportionally diversified top-k Results. *Given a user query q, integer k, user specified attribute A with ℓ domain size (on which the results are to be diversified), a proportionality constraint defined over a single attribute containing ℓ different groups G_1, $G_2,.,G_\ell$, requires that the representation of each group G_i is a_i top-k results, where $\sum_i^\ell a_i = k$. Generalizing this, if proportionality is defined over a set R of different attributes with a required representation on each group of each attribute, a proportional top-k result must simultaneously satisfy proportionate representation for all attributes in R.*

2. Proposed Framework

The proposed framework is presented in Figure 1, which is motivated by ChatDB [10]. The user writes a query to the LLM, the LLM connects with external databases to retrieve count information, the produced query results are to converted in natural language texts to prompt LLM, LLM produces final results and summarizes it. The

development of the framework thus requires solving the following four fundamental tasks.

A. Convert user query to LLMs to a series of SQL queries. Given the user query (e.g., find top-10 movies based on genre), an external data source (e.g., IMDB) is looked at and a series of SQL queries are submitted. Using the running example, the first query (step 1) retrieves different movie genres and their respective counts that are present in the database. The second, third, and fourth queries (corresponding to steps 2,3, and 4, respectively) query for each of the retrieved genre (in this example, these are, *drama, action, and comedy*) and finds the count of each kind which has average rating higher than some threshold (in this case > 6.5). In general, the goal is to make a sequence of SQL queries that are required to produce count information of the groups of interests based on the query. The Text-to-SQL solutions [11, 12, 13] could be leveraged for that. Even though in this preliminary work, we manually generate those queries.

B. Compute proportion based on arbitrary query condition. The next task is to produce proportion of each group in top-k based on the count information retrieved from the query engine. For that, we leverage our recent research results [8, 14] that studies computational challenges of computing proportional representation. When the query is defined on a single attribute with ℓ different domains (e.g., using the running example, $\ell = 3$). Using the running example, as shown in Figure 1, 50% (10,000 out of 20,000) Drama movies have average ratings higher than 6.5, whereas, 33% (5,000 out of 15,000) Action, and 20% (5,000 out of 25,000) Comedy movies have similar high ratings, respectively. Therefore, $a_{drama} = \frac{.5}{.5+.33+.2} = 5$, whereas, $a_{action} = 3$, $a_{comedy} = 2$. Indeed, it could be seen that the external data-source has a high proportion of highly rated drama movies compared to comedy movies, even though there are more comedy movies that drama movies in the results. These numbers indicate that in the returned results drama should have higher representation than comedies. However, computing proportion for arbitrarily complex condition is non-trivial. In [14], we prove that it is NP-hard just to decide whether there exists a feasible solution that satisfies the proportionality requirement defined over 3 or more attributes.

C. SQL to Text Transform. The next step of the process is to consume the proportion generated by the SQL results and convert them in natural language like texts that LLMs understand. Using the running example, this is equivalent to prompting the LLMS to return *"5 drama movies, 3 action movies, and 2 comedy movie"*. As before, there exists the challenge of automatically translating computed proportions to natural language texts. For which, existing SQL-to-Text solutions could be used [15, 16].

D. Finetune LLMs to Summarize Results. The final challenge is to finetune LLMs so that it could summarize back to the user the reasoning behind the returned results. In the context of the running example, it may say *"5 out of 10 movies are drama, because 50% drama movies are very highly rated,..."*

3. Preliminary Results

In this section, we evaluate the effectiveness of the proposed framework as a proof of concept, demonstrating its ability to prompt LLMs to generate answers with proportional diversity by linking to an external database. We first translate user intents into SQL statements to query information and calculate proportional statistics from the database. We then prompt the models with the proportional information to produce diversified answers. For instance, a request for "10 diverse movies by genre" requires counting movies of each genre in the database, computing the proportion for each genre, and normalizing the proportions to the number of movies requested (10 in this example). This process allows for proportional representation of genres in the generated movie list.

3.1. IMDb Dataset

We use the IMDb dataset by Kaggle [1] for a case study of our framework. This dataset is a subset of a larger IMDb dataset which spans a comprehensive collection of movies over several decades. The dataset is formatted as a relational database with 3 tables, namely movieList, ratings, and regions which encompass movies in 25 different genres across 70 languages. We first convert source data files to comma-separated value (CSV) format which is readable by SQL. Next, the CSV files are imported to MySQL by removing delimiters between values/records (e.g., comma, \n) while maintaining the original table schema (see Table 1 for the database schema). The structure of the dataset allows users to query and filter the data based on specific research questions, which fits the goals of our case study.

3.2. Implementation Details

To simplify the task that requires various tools, we unify MySQL and GPT-3.5-Turbo Model API developed by OpenAI [2] into a single Python interface. First, we link the IMDb database to MySQL and establish connection between MySQL and Python using `pymysql` library. As a result, SQL statements could be written directly in Python interface and the proportion could be computed from query outputs. Next, we summarize the proportion information into a prompt and feed into the GPT Model API

Table 1
Schema of the IMDb database.

Table	Major attributes	Description
movieList	tconst	alphanumeric unique identifier of the title
	titleType	type/format of the title (e.g., movie, tvseries, video, etc.)
	primaryTitle	the more popular title / the title used by the filmmakers on promotional materials at the point of release
	originalTitle	includes up to three genres associated with the title
	genres	La Haine (1995, French)
ratings	tconst	alphanumeric unique identifier of the title
	averageRating	weighted average of all the individual user ratings
	numVotes	number of votes the title has received
regions	titleId	a tconst, an alphanumeric unique identifier of the title
	title	the localized title
	region	the region for this version of the title
	language	the language of the title

to produce final outputs. We use four queries in Table 2 to demonstrate that the prompts summarized from SQL results are able to steer GPT to produce outputs that are proportionally diverse.

3.3. Result Analysis

Table 2 compares the answers of the GPT model with and without using prompts summarized from SQL results. The queries contain specific movie attributes (e.g., language, genre) to facilitate our investigation of proportion in the answers. The results show that prompting GPT with information summarizing relevant SQL results enables it to generate proportionally diverse answers that better match the distribution in the external IMDb database. Without these prompts, GPT tends to return answers skewed toward some attributes. For the query "find top-10 tv series from genres {comedy, documentary}", GPT without the prompt returns 9 comedies and 1 documentary, indicating a skew toward the comedy genre. This is not proportionate to the actual distribution of these genres in IMDb. However, after prompting GPT with the proportion of these genres derived from IMDb (e.g. "return 4 comedies and 6 documentaries"), GPT is able to generate an answer that matches this distribution. For query 1 in Table 2, the proportion of English is counter-intuitively high because English is the most popular language. However, the proportion English : French : Italian is reasonable as we only filter movies with rating greater than 8.0. This demonstrates that providing proportionality information from external databases as a prompt can guide large language models to generate answers that are proportionally diverse with respect to specified attributes.

4. Open Problems

In this section, we discuss research challenges regarding achieving proportionally diverse top-k results from LLMs, including automatically transform user queries into SQL queries, automatically transform SQL results into LLM prompts, and finetuning LLMs.

4.1. Text-to-SQL

Automatically converting natural language user queries into SQL queries is a challenging task that requires addressing several complex issues. First, the system must understand the meaning and intent behind the user's query. This requires handling ambiguities in language, modeling complex query structures, and leveraging domain knowledge about the database. Second, the system needs to translate the user's intent into the appropriate SQL constructs, such as SELECT, FROM, WHERE, GROUP BY, ORDER BY, etc. While simple queries may have a straightforward translation, translating more complex queries requires breaking them down into a series of steps while maintaining the overall semantics. Third, evaluating the quality of the generated SQL queries, both in terms of relevance and precision, is critical to building an effective text-to-SQL system. By addressing these issues, a robust system can be built to automatically convert human instructions into SQL queries.

4.2. SQL to LLM Prompt

There are several key challenges in automatically transforming SQL queries and results into natural language prompts for LLMs. First, it is challenging to map numerical values, proportions, and counts retrieved from a database into appropriate quantifiers and aggregates in

Table 2
Comparison of GPT's answers to original queries and to summarized prompts.

	Query	Answers to original queries	Answers to summarized prompts
1	find top-10 movies in languages {English, French, Italian} with average rating > 8.0 (ground truth proportion = 2 : 3 : 5)	The Godfather (1972, English) La Haine (1995, French) Cinema Paradiso (1988, Italian) Pulp Fiction (1994, English) Amelie (2001, French) Life is Beautiful (1997, Italian) The Shawshank Redemption (1994, English) Breathless (1960, French) 8½ (1963, Italian) The Grand Budapest Hotel (2014, English)	The Dark Knight (2008, English) Pulp Fiction (1994, English) Amelie (2001, French) La Haine (1995, French) Blue is the Warmest Color (2013, French) Life is Beautiful (1997, Italian) Cinema Paradiso (1988, Italian) La Dolce Vita (1960, Italian) The Great Beauty (2013, Italian) The Conformist (1970, Italian)
	Total	**4 English : 3 French : 3 Italian**	**2 English : 3 French : 5 Italian**
2	find top-10 movies in {English, Hebrew, Cantonese} with low votes, high ratings (ground truth proportion = 4 : 5 : 1)	The Fall (2006, English) Coherence (2013, English) Predestination (2014, English) Footnote (2011, Hebrew) The Band's Visit (2007, Hebrew) Fill the Void (2012, Hebrew) In the Mood for Love (2000, Cantonese) Ip Man (2008, Cantonese) Election (2005, Cantonese) The Way We Dance (2013, Cantonese)	Moon (2009, English) Coherence (2013, English) Primer (2004, English) The Fall (2006, English) Foxtrot (2017, Hebrew) The Band's Visit (2007, Hebrew) Lebanon (2009, Hebrew) Big Bad Wolves (2013, Hebrew) Gett: The Trial fo Viviane Amsalem (2014, Hebrew) Internal Affairs (2002, Cantonese)
	Total	**3 English : 3 Hebrew : 4 Cantonese**	**4 English : 5 Hebrew : 1 Cantonese**
3	find top-10 movies in {English, Hebrew, Cantonese} with high votes, low ratings (ground truth proportion = 2 : 8 : 0)	The Room (2003, English) Birdemic: Shock and Terror (2010, English) Troll 2 (1990, English) The 90 Minute War (2016, Hebrew) Norman: The Moderate Rise and Tragic Fall of a New York Fixer (2016, Hebrew) The Kindergarten Teacher (2014, Hebrew) The Twins Effect (2003, Cantonese) The White Storm (2013, Cantonese) Kick Ball (2015, Cantonese) Z Storm (2014, Cantonese)	Disaster Movie (2008, English) Troll 2 (1990, English) Petzitzim (1972, Hebrew) Banot (2019, Hebrew) Dead Europe (2012, Hebrew) The Wonderful Kingdom of Papa Alaev (2016, Hebrew) Shoelaces (2018, Hebrew) Halfon Hill Doesn't Answer (1976, Hebrew) Operation E (2012, Hebrew) Sweets (2013, Hebrew)
	Total	**3 English : 3 Hebrew : 4 Cantonese**	**2 English : 8 Hebrew : 0 Cantonese**
4	find top-10 tv series from genres {comedy, documentary} average rating > 8.0 (ground truth proportion = 4 : 6)	The Office (2005-2013, Comedy) Friends (1994-2004, Comedy) Parks and Recreation (2009-2015, Comedy) The IT Crowd (2006-2013, Comedy) Brooklyn Nine-Nine (2013-2021, Comedy) Arrested Development (2003-2019, Comedy) The Big Bang Theory (2007-2019, Comedy) Curb Your Enthusiasm (2000-present, Comedy) The Office UK (2001-2003, Comedy) Planet Earth (2006, Documentary)	Friends (1994-2004, Comedy) The Office (2005-2013, Comedy) Parks and Recreation (2009-2015, Comedy) Brooklyn Nine-Nine (2013-2021, Comedy) Planet Earth (2006, Documentary) Making a Murderer (Documentary) The Jinx (Documentary) Tiger King (Documentary) The Vietnam War (Documentary) The Crown (Documentary)
	Total	**9 Comedy : 1 Documentary**	**4 Comedy : 6 Documentary**

natural language. For example, translating a count of 50% into phrases like "half of" or "most" is difficult. Choosing an appropriate level of precision and rounding for communicating quantitative results also poses challenges.

Second, we need to summarize the reasoning and rationale behind database results and proportions in a concise yet compelling way. The summary should highlight the key factors and attributes driving certain groups or items

to be highly ranked, while omitting unnecessary details. Third, we must preserve context between the original user query, the intermediate SQL queries and results, and the final LLM prompts and summaries. The end-to-end system needs to retain the attributes, conditions, and constraints specified in the initial user query so that the final LLM prompts and instructions are grounded and relevant. Addressing these challenges will be the focus of our future research.

4.3. Finetuning LLMs

The last research challenge lies in finetuning LLMs to generate informative summaries. The LLMs should be finetuned to produce summaries that not only present the final results but also explain the reasoning behind those results. To accomplish this, the LLMs need to be trained on a large dataset of query-result pairs, where the results are accompanied by human-generated summaries or explanations. These summaries should capture the key insights and patterns in the results, highlighting the factors that influenced the distribution of groups. However, creating high-quality summaries requires extensive human expertise and effort. A potential approach is to use a combination of human-generated and automatically generated data by other LLMs to balance quality and efficiency. During finetuning, the LLMs learn to generate summaries that are concise, informative, and relevant to the user's query. The model should understand the statistical information from the SQL queries and use it to construct coherent explanations. For example, in the running example, the LLM might incorporate information about the high ratings of drama movies and the comparative proportions of different genres to generate an insightful summary. Lastly, to improve the quality of the generated summaries, various techniques can be employed, such as reinforcement learning. The model can be rewarded based on the informativeness and coherence of its generated summaries. By optimizing these rewards, the LLM can learn to produce personalized, high-quality summaries that explain the results to users.

5. Related Work

In this section, we review several lines of research that are most closely relevant to our work.

Large Language Models (LLMs). LLMs have become a prominent area of research that has garnered significant attention in recent years. LLMs typically refer to Transformer-based models with multi-head attention [17] embedded in deep neural networks and trained on large-scale corpora [18]. The development of LLMs has been driven by observations that scaling Pre-trained Language Models (PLMs), either in terms of model or data

size, often leads to improved performance on downstream tasks [19] and enhances the model's ability to solve various complex tasks. With their ever-increasing sizes, popular language models such as PaLM [20], LLaMA [21], Galactica [22], GPT-3 [23], and GPT-4 [24] have achieved state-of-the-art performance on many tasks. Consequently, they have motivated a profound shift in Natural Language Processing (NLP) research towards LLMs. For example, OpenAI released ChatGPT [25] which leverages the GPT-3.5 architecture, capable of understanding languages and engaging in meaningful conversations across various topics. ChatGPT represents the impact of LLMs throughout the community and revolutionizes our understanding of NLP [26, 18, 27]. In addition, these LLMs have made significant progress in natural language processing and have enabled various applications, such as coding assistants, search engines, and dialogue systems.

Augmenting LLMs. Despite the rapid progress of LLMs, LLMs also suffer from some limitations, including generating implausible predictions (hallucinations), requiring massive scale and data to achieve good performance, and struggling with continual learning [28, 29, 30]. To address these issues, there is a growing research trend of "augmenting" LLMs by providing them with additional context beyond just their parameters and input tokens. Two representative approaches are: 1) Increasing context relevance by retrieving external info or employing reasoning, which gives useful context with fewer parameters. 2) Allowing LLMs to use external tools and knowledge to augment context, which adds missing information [28]. Recent research has proposed using databases for LLMs which led to enhanced performance in multi-hop reasoning. These works have demonstrated the possibility of improving reasoning capabilities of LLMs by using external memory modules. For example, ChatDB [10] is a recent work that enables the use of real-time databases to enhance multi-step reasoning capabilities of LLMs. The ChatDB framework uses LLMs to transform user inputs into a *chain-of-memory* (multi-step SQL instructions) that manipulates an external database. The intermediate results are summarized as a prompt which is fed to LLMs to achieve final results.

Text-to-SQL. LLMs have become a reliable source of generating code, from common programming languages such as Python/Java to SQL statements for querying databases [24]. However, parsing natural languages to SQL statements faces semantic and syntactic challenges: LLMs as parsers must capture semantics of correct tables/columns from the database and generate syntactically valid SQL queries [31]. These requirements pose significant challenges in designing text-to-SQL models to generalize across databases and user intents. To overcome these challenges, researchers proposed text-to-SQL

frameworks using encoder-decoder neural architectures, categorized into two parsing approaches: *single-turn* and *multi-turn* [32]. In single-turn parsing, the encoder generates embeddings capturing natural language input and table schema semantics. The decoder then generates SQL statements from the encodings [33, 34, 35, 36, 37, 38, 39]. In multi-turn parsing, the encoder uses different encoding schemes to generate contextual and schema structure embeddings. The decoder, an LSTM model with attention mechanisms [40, 41], generates SQL queries using current and previous hidden states. This enables capturing long-term input dependencies and generating context-aware SQL queries [42, 43, 44, 45, 46].

Finetuning LLMs. Finetuning involves modifying parameters of a pre-trained model, LLM in our context, using a smaller and task-specific dataset. The goal of finetuning LLMs is to enhance pre-trained LLMs using domain adaptation or human feedback, making LLMs more relevant for specific tasks. There are two main streams of finetuning methods for LLMs: instruction tuning and reinforcement learning. Instruction tuning involves supervised learning using instruction-formatted instances, where each instance includes a task description, an input-output pair, and optional demonstrations of the task [47, 48, 22, 49]. For example, if the task is text-to-SQL, the task description could be "translate to SQL statements" and an input-output pair includes a natural language sentence as input and equivalent SQL statements as output. The formatted instances can be constructed from formatting existing datasets [50, 51] or formatting human needs from real user queries [52] or semi-automated augmentation approaches which feed existing instances into LLMs to generate new task descriptions and instances [53, 54, 55]. Reinforcement learning methods, on the other hand, propose using human feedback to make outputs of LLMs align with human expectations [52, 55, 56]. Although alignment considers human preferences to mitigate unexpected behaviors of LLMs (e.g., hallucinations, misleading or biased answers), low-quality feedback data may pose negative effects on the general abilities of LLMs [57].

6. Conclusion

In this work, we present our initial directions on how to make query results coming from LLMs more representative to what external gold standard data sources may provide. To that end, we present a framework that queries external benchmark databases to determine the proportional distribution of relevant attributes based on the given query. This proportion information is then used to prompt the large language model to return results that match that distribution and cover a variety of relevant perspectives. As a proof of concept, we implement this approach for improving the diversity of movie query results according to genres. The preliminary results show the potential of this framework to mitigate large language models' tendency to return inadequately diverse responses. Moving forward, we plan to investigate practical solutions to the automate the proposed framework and evaluate their effectiveness on diverse query types and domains.

Acknowledgments

References

[1] R. Agrawal, S. Gollapudi, A. Halverson, S. Ieong, Diversifying search results, in: Proceedings of the second ACM international conference on web search and data mining, 2009, pp. 5–14.

[2] Y. Yue, C. Guestrin, Linear submodular bandits and their application to diversified retrieval, Advances in Neural Information Processing Systems 24 (2011).

[3] S. Gollapudi, A. Sharma, An axiomatic approach for result diversification, in: Proceedings of the 18th international conference on World wide web, 2009, pp. 381–390.

[4] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al., Sparks of artificial general intelligence: Early experiments with gpt-4, arXiv preprint arXiv:2303.12712 (2023).

[5] AnthropicAI, Introducing claude, 2023. URL: https://www.anthropic.com/index/introducing-claude.

[6] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, T. B. Hashimoto, Alpaca: A strong, replicable instruction-following model, Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html 3 (2023) 7.

[7] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, E. P. Xing, Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. URL: https://lmsys.org/blog/2023-03-30-vicuna/.

[8] D. Wei, M. M. Islam, B. Schieber, S. Basu Roy, Rank aggregation with proportionate fairness, in: Proceedings of the 2022 International Conference on Management of Data, SIGMOD '22, Association for Computing Machinery, New York, NY, USA, 2022,

p. 262–275. URL: https://doi.org/10.1145/3514221.3517865. doi:10.1145/3514221.3517865.

[9] C. Yu, L. Lakshmanan, S. Amer-Yahia, It takes variety to make a world: diversification in recommender systems, in: Proceedings of the 12th international conference on extending database technology: Advances in database technology, 2009, pp. 368–378.

[10] C. Hu, J. Fu, C. Du, S. Luo, J. Zhao, H. Zhao, Chatdb: Augmenting llms with databases as their symbolic memory, 2023. arXiv:2306.03901.

[11] C. Finegan-Dollak, J. K. Kummerfeld, L. Zhang, K. Ramanathan, S. Sadasivam, R. Zhang, D. Radev, Improving text-to-sql evaluation methodology, arXiv preprint arXiv:1806.09029 (2018).

[12] G. Katsogiannis-Meimarakis, G. Koutrika, A survey on deep learning approaches for text-to-sql, The VLDB Journal (2023) 1–32.

[13] G. Katsogiannis-Meimarakis, G. Koutrika, A deep dive into deep learning approaches for text-to-sql systems, in: Proceedings of the 2021 International Conference on Management of Data, 2021, pp. 2846–2851.

[14] M. M. Islam, D. Wei, B. Schieber, S. B. Roy, Satisfying complex top-k fairness constraints by preference substitutions, Proc. VLDB Endow. 16 (2022) 317–329. URL: https://doi.org/10.14778/3565816.3565832. doi:10.14778/3565816.3565832.

[15] K. Xu, L. Wu, Z. Wang, Y. Feng, V. Sheinin, Sql-to-text generation with graph-to-sequence model, arXiv preprint arXiv:1809.05255 (2018).

[16] D. Ma, X. Chen, R. Cao, Z. Chen, L. Chen, K. Yu, Relation-aware graph transformer for sql-to-text generation, Applied Sciences 12 (2021) 369.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2017. arXiv:1706.03762.

[18] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, J.-R. Wen, A survey of large language models, 2023. arXiv:2303.18223.

[19] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, D. Amodei, Scaling laws for neural language models, CoRR abs/2001.08361 (2020). URL: https://arxiv.org/abs/2001.08361. arXiv:2001.08361.

[20] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra,

K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, Palm: Scaling language modeling with pathways, 2022. arXiv:2204.02311.

[21] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, G. Lample, Llama: Open and efficient foundation language models, 2023. arXiv:2302.13971.

[22] R. Taylor, M. Kardas, G. Cucurull, T. Scialom, A. Hartshorn, E. Saravia, A. Poulton, V. Kerkez, R. Stojnic, Galactica: A large language model for science, 2022. arXiv:2211.09085.

[23] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, CoRR abs/2005.14165 (2020). URL: https://arxiv.org/abs/2005.14165. arXiv:2005.14165.

[24] OpenAI, Gpt-4 technical report, 2023. arXiv:2303.08774.

[25] OpenAI, Introducing chatgpt, 2022. URL: https://openai.com/blog/chatgpt.

[26] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. Chatterji, A. Chen, K. Creel, J. Q. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. Gillespie, K. Goel, N. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. Roohani, C. Ruiz, J. Ryan, C. Ré, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu,

Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, P. Liang, On the opportunities and risks of foundation models, 2022. `arXiv:2108.07258`.

[27] C. Zhou, Q. Li, C. Li, J. Yu, Y. Liu, G. Wang, K. Zhang, C. Ji, Q. Yan, L. He, H. Peng, J. Li, J. Wu, Z. Liu, P. Xie, C. Xiong, J. Pei, P. S. Yu, L. Sun, A comprehensive survey on pretrained foundation models: A history from bert to chatgpt, 2023. `arXiv:2302.09419`.

[28] G. Mialon, R. Dessì, M. Lomeli, C. Nalmpantis, R. Pasunuru, R. Raileanu, B. Rozière, T. Schick, J. Dwivedi-Yu, A. Celikyilmaz, et al., Augmented language models: a survey, arXiv preprint arXiv:2302.07842 (2023).

[29] B. Peng, M. Galley, P. He, H. Cheng, Y. Xie, Y. Hu, Q. Huang, L. Liden, Z. Yu, W. Chen, et al., Check your facts and try again: Improving large language models with external knowledge and automated feedback, arXiv preprint arXiv:2302.12813 (2023).

[30] B. Xu, Z. Peng, B. Lei, S. Mukherjee, Y. Liu, D. Xu, Rewoo: Decoupling reasoning from observations for efficient augmented language models, arXiv preprint arXiv:2305.18323 (2023).

[31] P. Glenn, P. P. Dakle, P. Raghavan, Correcting semantic parses with natural language through dynamic schema encoding, 2023. `arXiv:2305.19974`.

[32] B. Qin, B. Hui, L. Wang, M. Yang, J. Li, B. Li, R. Geng, R. Cao, J. Sun, L. Si, F. Huang, Y. Li, A survey on text-to-sql parsing: Concepts, methods, and future directions, 2022. `arXiv:2208.13629`.

[33] V. Zhong, C. Xiong, R. Socher, Seq2SQL: Generating structured queries from natural language using reinforcement learning, 2018. URL: https://openreview.net/forum?id=Syx6bz-Ab.

[34] T. Yu, Z. Li, Z. Zhang, R. Zhang, D. Radev, Type-SQL: Knowledge-based type-aware neural text-to-SQL generation, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 588–594. URL: https://aclanthology.org/N18-2093. doi:`10.18653/v1/N18-2093`.

[35] T. Yu, M. Yasunaga, K. Yang, R. Zhang, D. Wang, Z. Li, D. Radev, SyntaxSQLNet: Syntax tree networks for complex and cross-domain text-to-SQL task, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1653–1663. URL: https://aclanthology.org/D18-1193. doi:`10.18653/v1/D18-1193`.

[36] J. Guo, Z. Zhan, Y. Gao, Y. Xiao, J.-G. Lou, T. Liu, D. Zhang, Towards complex text-to-SQL in cross-domain database with intermediate representation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 4524–4535. URL: https://aclanthology.org/P19-1444. doi:`10.18653/v1/P19-1444`.

[37] W. Hwang, J. Yim, S. Park, M. Seo, A comprehensive exploration on wikisql with table-aware word contextualization, 2019. `arXiv:1902.01069`.

[38] W. Lei, W. Wang, Z. Ma, T. Gan, W. Lu, M.-Y. Kan, T.-S. Chua, Re-examining the role of schema linking in text-to-SQL, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Online, 2020, pp. 6943–6954. URL: https://aclanthology.org/2020.emnlp-main.564. doi:`10.18653/v1/2020.emnlp-main.564`.

[39] D. Choi, M. C. Shin, E. Kim, D. R. Shin, RYAN-SQL: Recursively applying sketch-based slot fillings for complex text-to-SQL in cross-domain databases, Computational Linguistics 47 (2021) 309–332. URL: https://aclanthology.org/2021.cl-2.12. doi:`10.1162/coli_a_00403`.

[40] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. `arXiv:1810.04805`.

[41] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. `arXiv:1907.11692`.

[42] Q. Liu, B. Chen, J. Guo, J.-G. Lou, B. Zhou, D. Zhang, How far are we from effective context modeling? an exploratory study on semantic parsing in context, 2020. `arXiv:2002.00652`.

[43] R. Zhang, T. Yu, H. Y. Er, S. Shim, E. Xue, X. V. Lin, T. Shi, C. Xiong, R. Socher, D. Radev, Editing-based sql query generation for cross-domain context-dependent questions, 2019. `arXiv:1909.00786`.

[44] P. Jain, M. Lapata, Memory-based semantic parsing, Transactions of the Association for Computational Linguistics 9 (2021) 1197–1212. URL: https://aclanthology.org/2021.tacl-1.71. doi:`10.1162/tacl_a_00422`.

[45] B. Wang, R. Shin, X. Liu, O. Polozov, M. Richardson, Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers, 2021. `arXiv:1911.04942`.

[46] Y. Zheng, H. Wang, B. Dong, X. Wang, C. Li, HIE-SQL: History information enhanced network for context-dependent text-to-SQL semantic parsing, in: Findings of the Association for Computational Linguistics: ACL 2022, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 2997–3007. URL: https://aclanthology.

org/2022.findings-acl.236. doi:10.18653/v1/2022.findings-acl.236.

[47] S. Iyer, X. V. Lin, R. Pasunuru, T. Mihaylov, D. Simig, P. Yu, K. Shuster, T. Wang, Q. Liu, P. S. Koura, X. Li, B. O'Horo, G. Pereyra, J. Wang, C. Dewan, A. Celikyilmaz, L. Zettlemoyer, V. Stoyanov, Opt-iml: Scaling language model instruction meta learning through the lens of generalization, 2023. arXiv:2212.12017.

[48] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, P. Zhang, Y. Dong, J. Tang, Glm-130b: An open bilingual pre-trained model, 2022. arXiv:2210.02414.

[49] N. Muennighoff, T. Wang, L. Sutawika, A. Roberts, S. Biderman, T. L. Scao, M. S. Bari, S. Shen, Z.-X. Yong, H. Schoelkopf, X. Tang, D. Radev, A. F. Aji, K. Almubarak, S. Albanie, Z. Alyafeai, A. Webson, E. Raff, C. Raffel, Crosslingual generalization through multitask finetuning, 2023. arXiv:2211.01786.

[50] V. Sanh, A. Webson, C. Raffel, S. H. Bach, L. Sutawika, Z. Alyafeai, A. Chaffin, A. Stiegler, T. L. Scao, A. Raja, M. Dey, M. S. Bari, C. Xu, U. Thakker, S. S. Sharma, E. Szczechla, T. Kim, G. Chhablani, N. Nayak, D. Datta, J. Chang, M. T.-J. Jiang, H. Wang, M. Manica, S. Shen, Z. X. Yong, H. Pandey, R. Bawden, T. Wang, T. Neeraj, J. Rozen, A. Sharma, A. Santilli, T. Fevry, J. A. Fries, R. Teehan, T. Bers, S. Biderman, L. Gao, T. Wolf, A. M. Rush, Multitask prompted training enables zero-shot task generalization, 2022. arXiv:2110.08207.

[51] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, J. Wei, Scaling instruction-finetuned language models, 2022. arXiv:2210.11416.

[52] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, 2022. arXiv:2203.02155.

[53] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, H. Hajishirzi, Self-instruct: Aligning language models with self-generated instructions, 2023. arXiv:2212.10560.

[54] O. Honovich, T. Scialom, O. Levy, T. Schick, Unnatural instructions: Tuning language models with (almost) no human labor, 2022. arXiv:2212.09689.

[55] Z. Kenton, T. Everitt, L. Weidinger, I. Gabriel, V. Mikulik, G. Irving, Alignment of language agents, 2021. arXiv:2103.14659.

[56] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, J. Kernion, K. Ndousse, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, J. Kaplan, A general language assistant as a laboratory for alignment, 2021. arXiv:2112.00861.

[57] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, J. Kaplan, Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. arXiv:2204.05862.