LLM-Modulo-Rec: Leveraging Approximate World-Knowledge of LLMs to Improve eCommerce Search Ranking Under Data Paucity

Ali El Sayed
aelsaye1@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey, USA

Sathappan Muthiah
smuthiah@ebay.com
eBay Inc.
San Jose, California, USA

Nikhil Muralidhar
nmurali1@stevens.edu
Stevens Institute of Technology
Hoboken, New Jersey, USA

Abstract

Effective ranking of products relevant to a user's query and interest is the main goal of e-commerce product ranking. In this context, ranking irrelevant products or those mismatched with the intent of the user query results in sub-optimal user experience. Providing high-quality, relevant search rankings requires large labelled datasets for training powerful deep learning (DL) based ranking pipelines. However, such large datasets are costly and time-consuming to obtain. Another important facet that influences search ranking quality is the intent and ambiguity in the user's search query. Hence, data paucity and query ambiguity are two ever-present challenges impeding the success of modern deep learning (DL) based e-commerce ranking models. In this work, we present the first ever investigation of employing large-language models (LLMs) as *approximate knowledge sources* to counter these challenges and improve the performance of off-the-shelf ranking models, under data paucity and query ambiguity. Specifically, we undertake the first ever investigation of developing an *LLM-Modulo* method to improve the search ranking performance of off-the-shelf ranking models. Our experiments demonstrate notable performance improvements in ranking quality of these off-the-shelf models, when employed in an LLM-Modulo manner.

Keywords

e-commerce Search Ranking; Large Language Models; LLM-Modulo;

ACM Reference Format:
Ali El Sayed, Sathappan Muthiah, and Nikhil Muralidhar. 2024. LLM-Modulo-Rec: Leveraging Approximate World-Knowledge of LLMs to Improve eCommerce Search Ranking Under Data Paucity. In *Proceedings of the first workshop on Generative AI for E-Commerce 2024, October 25, 2024.* ACM, New York, NY, USA, 6 pages.

1 Introduction

Deep learning models have demonstrated great prowess in natural language processing (NLP) [11] on complex tasks like neural machine translation [2, 13] and text summarization [23]. The recent emergence of language understanding benchmarks like GLUE [32] has also seen the successful application of DL models also demonstrating their (rudimentary) ability for language understanding. In light of these successes in the NLP domain, research efforts have also investigated the effectiveness of these DL architectures for information retrieval tasks [21] and specifically for popular commercial applications like content search and product ranking [33]. However, unlike in traditional NLP contexts, a common failure mode of many such efforts has been the unavailability of large labelled datasets (*training data paucity*) to train DL pipelines employed for information retrieval tasks like content ranking as highlighted by many popular works [24]. Another key challenge for the sub-par performance of traditional NLP based pipelines for content ranking (especially in e-commerce applications) is the fact that most modern NLP pipelines are grounded in distributed representation learning based on word embeddings [20] which are ineffective at understanding query semantics and more importantly *ineffective at understanding query intent* of a user query. Modeling user intent of a query has been found to be crucial for effective product ranking, thus serving as a core component of several research efforts [7, 17, 28] in product ranking. An effective search ranking paradigm is hence one with the ability to effectively perform under (i) training data paucity and (ii) query intent ambiguity during model training.

Data augmentation is an effective counter to training data paucity and has been widely employed successfully in various computer vision [9, 36] tasks and also to improve performance on NLP tasks like text classification [34], sentiment analysis [1] and conditional text generation [18]. Similarly, query intent estimation has been significantly influenced by the ambiguity of the query [14, 29] and hence, obtaining a representation of *query ambiguity* and conducting ambiguity conditioned training of the ranking model can significantly affect the learned representation.

Recently, large language models (LLMs) like GPT [6], T5 [25], and BERT [11] revolutionized many aspects of the NLP pipeline. Some of the most recent dialogue models like ChatGPT from OpenAI and Llama 2 [30] from Meta research have gone a giant step further, demonstrating significant performance improvements across variegated text generation, summarization, information retrieval and question answering tasks. They have also been used for text data augmentation [10]. Although it is easy to consider these extremely large LLM dialogue models as a panacea, they are mostly *approximate knowledge sources* [16] that can be employed in a variety of constructive roles in conjunction with down-stream task models towards improving performance

Genaiecom '24, October 25, 2024, Boise, ID

on a target task of interest. Such frameworks have been aptly termed 'LLM-Modulo' frameworks by recent works [16].

In line with this, we demonstrate the effectiveness of LLM models for improving the task of product ranking when employed in an 'LLM-Modulo' fashion. Specifically, our contributions are as follows:

- We are the first to explore and successfully demonstrate the effectiveness of LLM-Modulo frameworks for e-commerce product ranking.
- We specifically investigate novel methods of countering the challenges of training data paucity through LLM-based training data augmentation.
- We also investigate a novel LLM-Modulo training curriculum based on query intent ambiguity, to train a downstream product ranking model.
- We perform our experiments in the context of a recent state-of-the-art, search ranking model and demonstrate performance improvements according to well accepted (MRR, nDCG) metrics resulting from our LLM-Modulo framework.

2 LLM-Modulo Search Ranking

Query ambiguity and data paucity are two important and ever present problems that challenge ranking models. Overcoming these challenges is crucial for high quality and relevant information retrieval and ranking systems.

Essentially, difficulty of the ranking task increases with the ambiguity of the query thereby warranting explicit treatment of query ambiguity quantification mechanisms [8]. Other works [19] have demonstrated that *query ambiguity* is a key factor in determining query concepts which are a significant indicator of the intention of the user. There have also been large-scale classification approaches [27] involving expensive human annotation to develop models that identify (harder) ambiguous queries enabling them to be treated differently from (easier) specific queries.

Query ambiguity, when coupled with data scarcity, exacerbates learning challenges. One effective technique to improve model performance of machine learning models is *curriculum learning* (CL) [4]. Humans tend to learn more effectively when examples are presented in a meaningful sequence that progressively introduces more concepts and increasing complexity, and that is predominantly the inspiration behind CL [5]. The application of CL has also been proven to enhance neural network performance in natural language processing tasks [31].

A primary challenge in developing CL approaches, however, is the notion of designing a *hardness score* for each training instance. To address this challenge, previous works have leveraged transfer learning [15] [35] to score examples on "difficulty" or "complexity". In our work, we address this challenge by directly querying the approximate knowledge of pre-trained LLM models with a query and a corresponding prompt requesting a *query intent ambiguity score*.

Availability of large volumes of training data remains a crucial factor governing the effective performance of DL models [22]. Consequently, DL models typically struggle to learn generalizable representations under data paucity (i.e., under low volumes of training data). Hence, the challenge of data paucity often necessitates innovative approaches, like curriculum learning (CL), to improve the DL training process. Another approach to counter the bane of training data paucity is *data augmentation*. The data augmentation approach requires diversifying the existing data by applying transformations, such as rotations and flips to images, or syntactic modifications in NLP such as synonym substitution, random word insertion/deletion, and, in our case, rephrasing of textual data.

To tackle the challenges of query intent ambiguity and training data paucity, we employ an LLM-Modulo framework, composed of Meta's Llama 3 (a transformer-based LLM model). In our case, we propose two potential use cases for these models in ranking systems: 1) generating additional and helpful data to enhance model performance and 2) Augment existing data to improve the diversity and quantity of training inputs, simulating real-world variations and noise without the need to collect new data. The framework also includes a Cross Encoder that serves as the ranker that determines the relative ranking of each product by their relevance to the query. The use of LLMs for query rephrasing and query intent ambiguity scoring offers distinct advantages, especially in zero-shot prediction scenarios. LLMs possess broad world knowledge and the ability to generalize across various tasks without explicit task-specific data, thanks to their pre-training on diverse text corpora. This makes them ideal for generating diverse and meaningful query rephrasings and ambiguity scores. Moreover, LLMs eliminate the need for costly data curation and manual labeling, which can be resource-intensive, whereas the marginal increase in computational cost to query LLMs is significantly outweighed by the efficiency and robustness of the resulting predictions.

As shown in Fig. 1(a) and Fig. 1(b), the modeling paradigm of the proposed use cases can be divided into the two following points:

- **LLM Intent Ambiguity Characterization + Sequential Curriculum Design.** This modeling paradigm employs the prompt-able LLM to characterize intent ambiguity scores of each user query in the training dataset. The application of curriculum learning here includes sorting the queries in decreasing order of intent ambiguity, and then presenting the dataset to the ranking model in 25th percentiles, introducing more examples each time.
- **LLM Query Data Augmentation/Generation.** Similarly to how the LLM is used in generating intent ambiguity scores, the LLM is also used to generate variants of each user query, used to train the ranking model. This is done to augment the data and thereby counter the data paucity problem. By generating one augmented variant of each query in our training dataset, we are able to essentially double the size of the available training data simply by employing pre-trained LLMs as approximate knowledge sources for query augmentation.

3 Experimental Setup

Dataset Description. In our experiments, we employ the well studied Amazon ESCI large dataset [26] with over 170,000 unique queries. The dataset provides search query-product pairs where each query has up to 40 product results, with each result having an ESCI relevance judgment (Exacts, Substitutes, Complements, and Irrelevants) that indicates

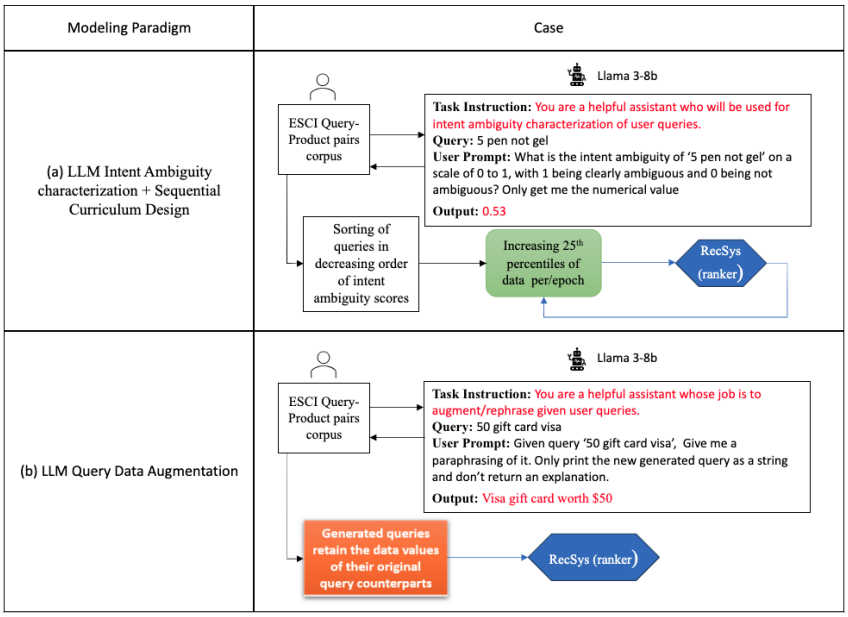

Figure 1: Two representative modeling paradigms employing generative large language models in ranking systems.

the relevance of each product to the query. We employ this dataset to investigate the 'Query Product Ranking' task. To simulate the training data paucity experiment, we only consider a small subset of 5,000 unique queries from this dataset. . Specifically, we employ a randomly sampled subset of 4,000 unique queries as the training data to train our ranking model and the remaining 1,000 unique queries are employed for testing and performance evaluation.

Model Description. *MSMCE*. As our baseline model, we employ a state-of-the-art MS MARCO Cross-Encoder (MSMCE) Information retrieval model[1], pre-trained on the MS MARCO passage ranking dataset [3]. Our choice is motivated by the fact that this was the model employed for evaluation on the ESCI dataset in the original paper [26]. We fine-tune the randomly initialized MSMCE model on our training dataset comprising of 4,000 queries from the ESCI dataset. MSMCE uses the following hyper-parameters for the CrossEncoder model: maximum length=512, activation function=identity, and number of labels=1 (binary task).

CL-MSMCE. The first variant of the baseline MSMCE model, CL-MSMCE, introduces the use LLM for the intent ambiguity characterization of each of the 4000 training queries. We leveraged Llama 3 (8b version), one of Meta's LLama 3 [12] pre-trained generative text models, during the training of the model to obtain intent ambiguity of each query on a scale of 0 to 1, with 1 being completely ambiguous and 0 being completely un-ambiguous. Specifically, we prompt the LLM as follows: "what is the intent ambiguity of 'query' on a scale of 0 to 1, with 1 being clearly ambiguous and 0 being not ambiguous? only get me the numerical value" for each query. We demonstrate an example of the pretrained model's interaction with one of the queries in Fig. 1(a). In brief, the model should only print out a float value. However, it exhibited inconsistencies in scoring the queries, with a predominant clustering of scores in the range of 0.6 to 0.8. The lowest intent ambiguity score observed was 0.35. Notably, as anticipated, certain queries were assigned an intent ambiguity score approaching 1.0, which may not accurately reflect their true level of ambiguity. Once the intent ambiguity scores are generated, we utilize these scores to implement linear curriculum learning into the training of the baseline model. Specifically, we sort the data by intent ambiguity scores in descending order and then have sequential training with increasing data sizes for each epoch. In the first epoch in this approach, we train the model using only the first 25th percentile of the data (1000 queries). In the second epoch, we add the next 1000 queries (50th percentile), effectively training on the first 2000 data points. This process continues in 25-percentile increments until the final epoch, where the entire dataset is used.

QAug-MSMCE: The second variant of the baseline MSMCE model, QAug-MSMCE model, leverages another application of LLMs: generating artificial and simulated queries for model training. While the goal of CL-MSMCE model is to leverage LLM to allow a new setup to how the data is introduced, the objective of QAug-MSMCE is to augment the training dataset and rigorously evaluate model performance to determine if the incorporation of such synthetic queries leads to any enhancement in the model evaluation metrics compared to the other versions of the model. We use the same 4000 unique queries that are used previously, and feed the LLM with the prompt: "Given query 'query', Give me a rephrasing of it. Only print the new generated query as a string and don't return an explanation.", As shown in Fig 1(b). This process effectively doubles the amount of training queries where we have 7960 queries[2]. It is important to note that the augmented queries retain the data values of their original counterparts, ensuring consistency in the training process. We then feed the new dataset in the original MSMCE setup where we don't use curriculum learning.

[1]https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

[2]40 queries were removed during data cleaning of 8K query augmented data

(QAug+CL)-MSMCE: The third and final variant of the model, (QAug+CL)-MSMCE, follows the same setup as the first variant, using sequential **curriculum learning** and utilizing the LLM during the training of the model for intent ambiguity characterization. However, after demonstrating the effectiveness of the **query augmentation** in the second variant, we used the expanded dataset from the second variant to evaluate if this larger and augmented dataset could further enhance performance. We again run the same prompt as the one shown in Fig. 1(a) on all the unique queries, including the new augmented ones, and again sort them in decreasing order of those intent ambiguity scores. Once that is done, we use the same curricula as one in CL-MSMCE where we introduce the batches of increasing 25th percentiles, culminating in having the entire dataset (100th percentile) by the fourth and last epoch.

Training Setup. The training of all variants of the MSMCE ranking models includes employing MSE loss for training, batch size=32, learning rate=7e-6, training epochs=4.

Experimental Evaluation In line with the original paper introducing the dataset [26], the 4 relevance categories considered for each query and product pair are: Exacts, Substitutes, Complements, and Irrelevants, with respective gain scores, which are set to distinguish between the relevance categories, of 1.0, 0.1, 0.01, 0.0. As an extension of this, when testing the models, we use those target label/gain to compare the gain scores of the queries where we see queries that have scores greater than or equal to the gain values (i.e., >= 1.0 (E) on one instance, >= 0.1 (S) on another, etc.) Doing this allows us to consider only the products that have an E label relevant when setting the target gain to be greater than or equal to 1.0, and consider products with an E and S labels when the target gain is greater than or equal to 0.1, and consider E, S, and C labels when the target gain is greater than or equal to 0.01. However, there exists a corner case where nDCG and MRR are not well defined (i.e., when all results are irrelevant or all are relevant), where we compare the gain of each query to be greater than or equal to 0.0. In the testing run for each of the three instances of comparison, we consider products that have a gain score greater than or equal to the target gain to be relevant (positive) and those that have a gain score less than the target gain to be irrelevant (negative.) We then leverage a sentence transformer evaluator, 'CERerankingEvaluator', which evaluates the CrossEncoder ranking models, where it is given a search query, a list of positive 'relevant' documents, and a list of negative 'irrelevant' documents, and computes Normalized Discounted Cumulative Gain (nDCG@10) and Mean Reciprocal Rank (MRR@10) as shown in Table 1. The way the 'CERerankingEvaluator' calculates the MRR and nDCG scores is by considering the first and highest relevant product for each query, and then getting the average MRR and nDCG scores of those products. For consistency and comparability, we adhere to the same testing procedures for each variant as employed with the baseline model, allowing us to effectively compare the evaluation results.

4 Results & Discussion

In this section, we report performance comparisons of the baseline MSMCE model and its three LLM-Modulo variants, for the query product ranking task, as evaluated on the popular ESCI dataset [26]. Through our evaluation, we strive to demonstrate the effectiveness of LLM-Modulo solutions for ranking systems.

4.1 Improving Ranking Under Data-Paucity with LLM-Modulo Design

In Table 1, we report the overall performance according to the popular mean reciprocal rank (MRR) and normalized discounted cumulative gain (nDCG) ranking metrics.

Variant	ESC Label	MRR@10	NDCG@10
MSMCE	E	0.8153	0.7225
	E | S	0.9061	0.8351
	E | S | C	0.9203	0.8496
CL-MSMCE	E	0.8241	0.7233
	E | S	0.9135	0.8343
	E | S | C	0.9242	0.8504
QAug-MSMCE	E	0.8647	0.7696
	E | S	0.9333	0.8610
	E | S | C	0.9436	0.8708
(QAug+CL)-MSMCE	E	**0.8832**	**0.7995**
	E | S	**0.9374**	**0.8640**
	E | S | C	**0.9466**	**0.8727**

Table 1: Evaluation scores for the ESC labels for the MSMCE model and three LLM-Modulo variants, evaluated using mean reciprocal rank (MRR) and normalized discounted cumulative gain (NDCG) metrics.

We notice that all the LLM-Modulo variants (i.e., CL-MSMCE; QAug-MSMCE; (QAug+CL)-MSMCE) outperform the vanilla MSMCE ranking model. Further we breakdown the performance by different product relevance categories and notice that this performance improvement trend is consistent across all relevance categories. The best performing model, i.e., *(QAug+CL)-MSMCE* is the LLM-Modulo variant of MSMCE trained on an augmented training dataset (augmented queries are obtained from an LLM) where training is guided by a query intent ambiguity score based curriculum to allow the model to learn unambiguous queries before encountering challenging ambiguous ones.

According to the MRR@10 metric, (QAug+CL)-MSMCE model achieves an **8.32**% performance improvement over vanilla MSMCE, on ranking products that are exact matches to the query (category 'E'), **3.45**% on ranking products that are either exact matches or substitutes to exact matches of a query (i.e., category 'E|S') and **2.86**% on ranking products that are exact matches, substitutes or complements (i.e., category 'E|S|C'). We further verify similar results across the

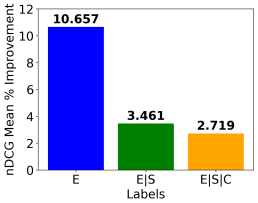

Figure 2: nDCG Mean Percentage Improvement of (QAug+CL)-MSMCE over MSMCE

same product category splits (i.e., exact matches E, exact matches or substitutes E|S, exact matches, or substitutes

Query	LLM-augmented query
the white shadow complete series dvd	Complete series of 'The White Shadow' on DVD
samsung galaxy j7v covers pink	Samsung Galaxy J7V phone cases in pink color
5d mark iii	Canon EOS 5D Mark III
75 carat diamond rings	Large diamond engagement rings with 75 carats
gpu gtx 750	NVIDIA GeForce GTX 750
fujifilm insta mini	Fujifilm Instax Mini 9 Camera
disney roller skates	Disney-themed roller skates

Table 2: Examples of original queries and the corresponding LLM augmented variant.

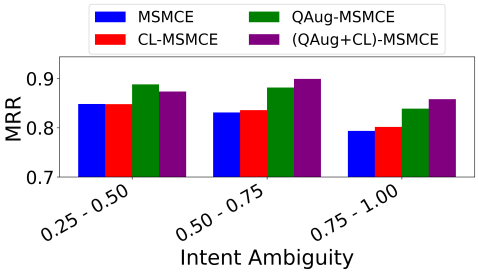

Figure 3: Intent Ambiguity vs MRR score for E label for four models

or complements E|S|C) using the NDCG metric. Fig. 2 depicts the performance improvements of the (QAug+CL)-MSMCE model over vanilla MSMCE. Table 1 and Fig. 2 both indicate significant relative performance improvements of (QAug+CL)-MSMCE model (over vanilla MSMCE) in the strictest case of ranking of exact matches with lower performance degradation for easier contexts (namely E|S, E|S|C).

4.2 LLMs for Synthetic Query Generation

In Fig. 1(b), we observe an example where the LLM is able to capture the query in the prompt and correctly infer the intention of the user which is to *buy a Visa gift card worth $50*, and not the less likely scenario of the user wanting to buy 50 vias gift cards. The query variant generated by the LLM to this somewhat ambiguous query may be construed as less ambiguous and capturing the original intention of the user, thereby enriching our training data with another high quality, relatively un-ambiguous query.

We list other such interesting examples of query augmentations obtained from the LLM, in Table 2. We notice from the table, a convincing demonstration of the LLM's comprehension to a diverse set of queries of varied topics and ambiguity levels, all of which the LLM is able to faithfully augment owing to its vast approximate world knowledge. For instance, the LLM recognizes that "5d mark iii" is the "Canon EOS 5D Mark III." In another instance, the model was able to identify "samsung galaxy j7v covers pink" as "Samsung Galaxy J7V phone cases in pink color," where it knows the user is intending to search for pink phone cases for a Samsung Galaxy J7V phone. Separately, even in the context of short, coded queries, the augmentations generated (e.g."gpu gtx 750," is augmented to "NVIDIA GeForce GTX 750,") show the LLM's data augmentation capabilities. Another demonstration of the benefits of the approximate world knowledge of the LLM is the augmentation of the query "75 carat diamond rings" into "Large diamond engagement rings with 75 carats." This augmentation reflects the model's comprehension that diamond rings are often associated with engagements and that a 75-carat diamond is extraordinarily large, correlating carat size with the perception of grandeur and significance. Such transformations indicate the model's capacity to apply real-world knowledge and cultural associations, further enhancing its effectiveness in query refinement and data augmentation.

This ability of the LLM to intelligently generate rich and intent-aligned variants of user queries, contributes significantly to the QPR performance improvement of the QAug-MSMCE model over vanilla MSMCE. In Table 1, we notice that QAug-MSMCE (i.e., ,the model trained with the LLM augmented queries) achieves a mean performance improvement of **3.9**% over vanilla MSMCE across product categories.

4.3 LLMs for Ambiguity Conditioned Curriculum Design

Finally, we also characterize the effect of LLM-modulo CL training of the ranking model, conditioned on query ambiguity scores generated by the LLM. The characterization is performed in terms of MRR for various ambiguity bins. Specifically, we consider bins 0.25 - 0.5 (low ambiguity); 0.5 - 0.75 (medium ambiguity) and 0.75 - 1.0 (high ambiguity). Fig. 3 clearly shows that more ambiguous queries are better resolved by sequential curriculum learning-based models.

5 Conclusion & Future work

In this work, we undertake the first ever investigation of augmenting search ranking models in an LLM-modulo manner for improved performance under data paucity, for e-commerce product ranking. Specifically, our investigations are confined to the the query product ranking task on the well studied ESCI dataset. We demonstrate two LLM-Modulo mechanisms to improve ranking model performance for the QPR task. The first being developing a training curriculum conditioned on query ambiguity where the query ambiguity scores are derived from an LLM. We show that the curriculum learning variant of a standard ranking baseline achieves a mean performance improvement of **1.08**% over the baseline in the strictst evaluation case (i.e., exact) product relevance category as per the MRR metric. The second LLM-Modulo mechanism we have incorporated to counter data paucity and improve effectiveness of ranking is intelligent query augmentation via. the LLM to generate additional training data for our model at no additional labeling cost. The search ranking variant trained with this augmented dataset achieves a mean performance improvement of **6.06**% over the baseline across the strictest evaluation of exact match product category as per the MRR metric. Our results demonstrate that LLM-Modulo frameworks can have significant positive impact on ranking systems.

References

[1] Hugo Queiroz Abonizio, Emerson Cabrera Paraiso, and Sylvio Barbon. 2021. Toward text data augmentation for sentiment analysis. *IEEE Transactions on Artificial Intelligence* 3, 5 (2021), 657–668.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. arXiv:1611.09268 [cs.CL] https://arxiv.org/abs/1611.09268

[4] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.

[5] Y. Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. *Journal of the American Podiatry Association* 60, 6. https://doi.org/10.1145/1553374.1553380

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[8] Steve Cronen-Townsend, W Bruce Croft, et al. 2002. Quantifying query ambiguity. In *Proceedings of HLT*, Vol. 2. 94–98.

[9] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501* (2018).

[10] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. 2023. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007* (2023).

[11] Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[12] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[13] Mikel L Forcada. 2017. Making sense of neural machine translation. *Translation spaces* 6, 2 (2017), 291–309.

[14] Jing Gong, Vibhanshu Abhishek, and Beibei Li. 2018. Examining the impact of keyword ambiguity on search advertising performance. *MIS Quarterly* 42, 3 (2018), 805–A14.

[15] Guy Hacohen and Daphna Weinshall. 2019. On The Power of Curriculum Learning in Training Deep Networks. arXiv:1904.03626 [cs.LG] https://arxiv.org/abs/1904.03626

[16] Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. [n. d.]. Position: LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. In *Forty-first International Conference on Machine Learning*.

[17] Haoyang Li, Xin Wang, Ziwei Zhang, Jianxin Ma, Peng Cui, and Wenwu Zhu. 2021. Intention-aware sequential recommendation with structured intent transition. *IEEE Transactions on Knowledge and Data Engineering* 34, 11 (2021), 5403–5414.

[18] Ruibo Liu, Guangxuan Xu, Chenyan Jia, Weicheng Ma, Lili Wang, and Soroush Vosoughi. 2020. Data boost: Text data augmentation through reinforcement learning guided conditional generation. *arXiv preprint arXiv:2012.02952* (2020).

[19] Craig Macdonald, Rodrygo LT Santos, and Iadh Ounis. 2012. On the usefulness of query features for learning to rank. In *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2559–2562.

[20] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[21] Bhaskar Mitra, Fernando Diaz, and Nick Craswell. 2017. Learning to match using local and distributed representations of text for web search. In *Proceedings of the 26th international conference on world wide web*. 1291–1299.

[22] Alhassan Mumuni and Fuseini Mumuni. 2022. Data augmentation: A comprehensive survey of modern approaches. *Array* 16 (2022), 100258. https://doi.org/10.1016/j.array.2022.100258

[23] Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023* (2016).

[24] Thanh V Nguyen, Nikhil Rao, and Karthik Subbian. 2020. Learning robust models for e-commerce product search. *arXiv preprint arXiv:2005.03624* (2020).

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research* 21, 140 (2020), 1–67.

[26] Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A Large-Scale ESCI Benchmark for Improving Product Search. arXiv:2206.06588 [cs.IR] https://arxiv.org/abs/2206.06588

[27] Ruihua Song, Zhenxiao Luo, Jian-Yun Nie, Yong Yu, and Hsiao-Wuen Hon. 2009. Identification of ambiguous queries in web search. *Information Processing & Management* 45, 2 (2009), 216–229.

[28] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

[29] Jaime Teevan, Susan T Dumais, and Daniel J Liebling. 2008. To personalize or not to personalize: modeling queries with variation in user intent. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. 163–170.

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[31] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Brian MacWhinney, and Chris Dyer. 2016. Learning the Curriculum with Bayesian Optimization for Task-Specific Word Representation Learning. arXiv:1605.03852 [cs.CL] https://arxiv.org/abs/1605.03852

[32] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).

[33] Xinxi Wang and Ye Wang. 2014. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM international conference on Multimedia*. 627–636.

[34] Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. *arXiv preprint arXiv:1901.11196* (2019).

[35] Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum Learning by Transfer Learning: Theory and Experiments with Deep Networks. arXiv:1802.03796 [cs.LG] https://arxiv.org/abs/1802.03796

[36] Suorong Yang, Weikang Xiao, Mengchen Zhang, Suhan Guo, Jian Zhao, and Furao Shen. 2022. Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610* (2022).