

Improving Retrieval in Sponsored Search by Leveraging Query Context Signals

Akash Kumar Mohankumar[†] Gururaj K[†]
Gagan Madan[†] Amit Singh[†]

[†]Microsoft, India

{makashkumar, gururajk, gaganmadan, siamit}@microsoft.com

Abstract

Accurately retrieving relevant bid keywords for user queries is critical in Sponsored Search but remains challenging, particularly for short, ambiguous queries. Existing dense and generative retrieval models often fail to capture nuanced user intent in these cases. To address this, we propose an approach to enhance query understanding by augmenting queries with rich contextual signals derived from web search results and large language models, stored in an on-line cache. Specifically, we use web search titles and snippets to ground queries in real-world information and utilize GPT-4 to generate query rewrites and explanations that clarify user intent. These signals are efficiently integrated through a Fusion-in-Decoder based Unity architecture, enabling both dense and generative retrieval with serving costs on par with traditional context-free models. To address scenarios where context is unavailable in the cache, we introduce *context glancing*, a curriculum learning strategy that improves model robustness and performance even without contextual signals during inference. Extensive offline experiments demonstrate that our context-aware approach substantially outperforms context-free models. Furthermore, online A/B testing on a prominent search engine across 160+ countries shows significant improvements in user engagement and revenue.

1 Introduction

Sponsored search is a primary revenue model for many search engines, where advertisements are displayed alongside organic search results. In this model, advertisers bid on specific keywords to target user intents relevant to their business objectives. They can select various match types to determine how closely their keywords align with user queries. For instance, the exact match type restricts keyword matching to queries that precisely share the same intended meaning. In

Query: ad623armz reel
Web Results: Title 1: AD623ARMZ-REEL7 Analog Devices Inc. Integrated Circuits (ICs) DigiKey Snippet 1: AD623ARMZ-REEL7 – Instrumentation Amplifier 1 Circuit Rail-to-Rail 8-MSOP from Analog Devices Inc ...
Query Profile: Rewrites: ad623armz analog amplifier reel, analog devices reel of ad623 armz, ad623armz tape and reel, ... Intent: The user is looking for a specific integrated circuit (IC) chip, the AD623ARMZ, that is sold in a reel package. A reel package is a type of bulk packaging that contains many IC chips ...
Retrieved Keywords: Unity (Context-free): 1. fishing rods reels and gear 2. fishing reels with rod 3. fishing rods fishing reels Augmented Unity (with Context): 1. ad623armz reel analog devices 2. ad623armz reel7 analog devices 3. ad623armz microchip

Table 1: Table illustrates how incorporating web results and LLM-generated Query Profile enables our proposed Augmented Unity model to retrieve relevant keywords.

contrast, more flexible match types, such as phrase and smart match, enable advertisers to target a broader range of search intents with their keywords. Retrieving relevant bid keywords for a user query is a critical task in sponsored search, directly influencing both the revenue generated and the quality of ads served to users.

Prior Work and Challenges: Traditionally, keyword retrieval has been approached as a standard information retrieval task. Some methods utilize dual encoders to map queries and keywords into a shared semantic space, with optional use of a one-vs-all classifier to rerank the shortlisted keywords (Mittal et al., 2021; Dahiya et al., 2021, 2022). Another line of research considers this problem as a constrained Natural Language Generation (NLG) task, where language models transform queries into relevant keywords (Mohankumar et al., 2021; Lian et al., 2019; Qi et al.,

2020; Valluri et al., 2024). A recent approach, Unity (Mohankumar et al., 2022), integrates Dense Retrieval (DR) and NLG methodologies into a unified model, harnessing the strengths of both while requiring only a single model. Despite these advancements, existing methods struggle with short and ambiguous queries. As shown in Table 1, the query "*ad623armz reel*" - which actually refers to an integrated circuit by Analog Devices - is incorrectly associated with keywords related to fishing reels, leading to the retrieval of irrelevant ads. This issue largely stems from the use of shallow transformer models, which are necessary to meet strict online latency requirements but have limited capacity to encode complex world knowledge. Consequently, understanding specialized terms like "*ad623armz*" becomes challenging without any additional information.

Our Contribution: To address these limitations, we introduce Augmented Unity, a framework for context-aware retrieval in sponsored search. Our approach utilizes a large dynamic cache to enrich queries with contextual signals. This cache includes organic search results, such as titles and snippets from the top-k web documents for each query. Additionally, we create a *Query Profile* for each query, containing multiple rewrites and a description of the user’s potential intent, generated using GPT-4, and store them in the cache. If a query is absent from the cache, an offline pipeline is triggered to generate this context, ensuring its availability for future instances of the same query within a short timeframe. Our Augmented Unity model employs a Fusion-in-Decoder architecture, which enables efficient processing of diverse contexts. We train this model using a curriculum learning strategy termed *context glancing*, which progressively introduces more challenging scenarios with varying levels of context availability. Our evaluations show that Augmented Unity significantly outperforms the context-free Unity model by 19.9% in exact match Precision at 100, while maintaining a comparable online GPU serving cost (within 7-9%). Moreover, Augmented Unity, trained with context glancing, demonstrates robust performance even when context signals are absent, matching the performance of the Unity model. Through extensive online A/B testing, we show that Augmented Unity achieves a 1% and 1.4% increase in ad revenue for English and non-English queries, respectively, without any statistical change in ad defects.

2 Proposed Method: Augmented Unity

Figure 1 provides an overview of our Augmented Unity workflow. We use a context cache to store and retrieve query context signals. For incoming queries, we first check the cache. In case of a cache miss, an asynchronous offline pipeline is triggered to generate the context signals. These signals are then used to update the cache, ensuring their availability for future occurrences of the same query. This section is structured as follows: Section 2.1 details the various context signals employed. Section 2.2 outlines our efficient model architecture for integrating these signals. Section 2.3 introduces *context glancing*, our curriculum learning strategy designed to enhance model performance and robustness in scenarios with missing context signals.

2.1 Query Context Signals

We utilize two sources of query-level signals to provide richer context and disambiguate user intent:

Web Search: Organic search results, often highly relevant to user intent, provide valuable contextual information. We utilize the title and snippet of each web result, offering a concise summary of the webpage content. For example, as demonstrated in Table 1, web titles and snippets help identify "*ad623armz*" as an electronic IC chip from Analog Devices Inc., leading to the retrieval of keywords with spans such as "*analog devices*" and "*microchip*". We mine web results from the logs of a prominent search engine. To account for location-based variations in search results, we utilize the country where a query is most frequently searched. In cases of multiple occurrences, the most recent result is prioritized to ensure up-to-date information. We cache the top 10 web results per query and periodically refresh them to incorporate new queries and update existing ones.

LLM-generated Query Profile: We leverage the reasoning abilities and extensive "world knowledge" of large language models like GPT-4 to generate Query Profile. These profiles comprise of query rephrases and explanations of potential user intents, aiding in disambiguation. Table 1 illustrates this where Query Profile includes clarifying rewrites like "*ad623armz analog amplifier reel*" and "*analog devices reel of ad623 armz*". Further, it includes a concise explanation of the possible user intent, offering crucial background information for query understanding.

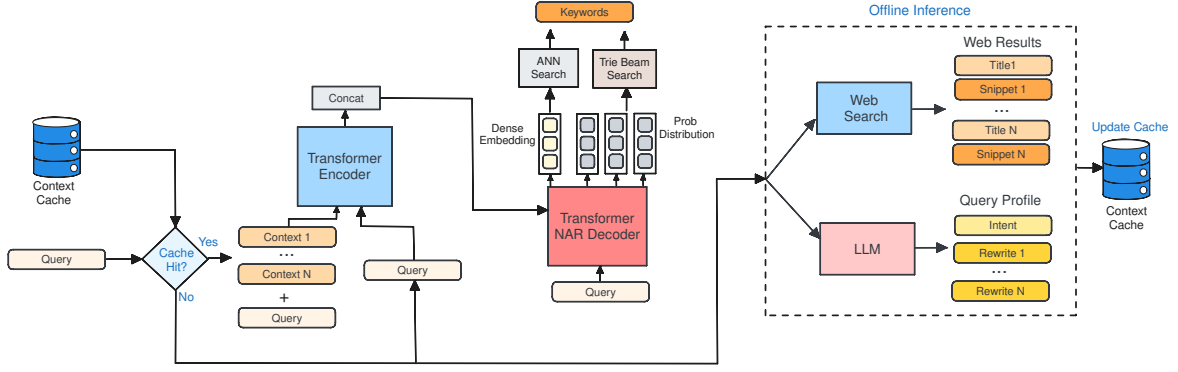


Figure 1: Overview of the Augmented Unity architecture for context-aware retrieval. The system leverages a context cache to store and retrieve pre-computed query context signals, employing an offline pipeline for cache misses. The Augmented Unity model, utilizing a Fusion-in-Decoder approach, effectively combines query representations

We utilize various query normalization techniques, including spell correction, to map minor query variations to a canonical form for cache storage and lookups. With these normalizations, our cache hit rate is approximately 70% of all user search requests.

2.2 Model Architecture & Training

We now discuss our proposed architecture that effectively combines the aforementioned context signals with the original query. Inspired by the Unity framework (Mohankumar et al., 2022), we use a shared model to perform both dense and non-autoregressive (NAR) generative retrieval, leveraging the complementary benefits of the two approaches at the cost of one. Our model consists of an encoder \mathcal{E} with L_e transformer encoder layers and a NAR decoder \mathcal{D} with L_d transformer decoder layers. We first encode the query $Q = \{w_1^0, \dots, w_{l_0}^0\}$ of length l^0 and each context signal $C^i = \{w_1^i, \dots, w_{l_i}^i\}$ of length l^i , independently using the encoder, obtaining hidden states $\mathbf{H} = \{\mathbf{H}^i\}_{i=0}^n$, where $\mathbf{H}^i = \{\mathbf{h}_1^i, \dots, \mathbf{h}_{l_i}^i\} \in \mathbb{R}^{l^i \times d}$, \mathbf{H}^0 corresponds to the query’s hidden states, $\{\mathbf{H}^i\}_{i=1}^n$ represents the hidden states of the n context signals, and d represents the hidden size. Following the Fusion-in-Decoder (FiD) approach (Izacard and Grave, 2020), we concatenate these hidden states into $\tilde{\mathbf{H}} = [\mathbf{H}^0, \dots, \mathbf{H}^n] \in \mathbb{R}^{l \times d}$, where $l = \sum_{i=0}^n l_i$, and leverage them within our NAR decoder \mathcal{D} . This decoder utilizes bidirectional attention without a causal mask and receives the original query Q as input, as opposed to right-shifted target tokens in autoregressive models. After processing through the L_d decoder layers, we obtain final hidden states $\mathbf{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_{l^0}\} \in \mathbb{R}^{l^0 \times d}$. These

are used to compute the dense retrieval embedding $\mathbf{e}(Q)$ and the NAR token probabilities $P(k_t|Q)$:

$$\mathbf{e}(Q) = \text{Attention}(\tilde{\mathbf{g}}, \mathbf{G}\mathbf{W}^K, \mathbf{G}\mathbf{W}^V)$$

$$P(k_t|Q) = \text{Softmax}(\mathbf{W}^O \mathbf{g}_t)$$

where $\mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d \times d'}$ are attention key and value matrices, $\tilde{\mathbf{g}} \in \mathbb{R}^{d'}$ is a learnable query vector, $\mathbf{W}^O \in \mathbb{R}^{V \times d}$ represents the language modeling head’s weight matrix, and d' and V correspond to the dense embedding and vocabulary sizes, respectively. We train the model using a combination of the contrastive loss with in-batch negatives for DR and the negative log-likelihood loss for NLG:

$$\mathcal{L}(\theta, \mathcal{B}) = \frac{-1}{|\mathcal{B}|} (\mathcal{L}^N(\theta, \mathcal{B}) + \lambda \mathcal{L}^D(\theta, \mathcal{B}))$$

$$\mathcal{L}^N(\theta, \mathcal{B}) = \sum_{Q, K \in \mathcal{B}} \sum_{k_t \in K} \log P(k_t|Q)$$

$$\mathcal{L}^D(\theta, \mathcal{B}) = \sum_{Q, K \in \mathcal{B}} \log \frac{\exp(\text{Sim}(Q, K))}{\sum_{K' \in \mathcal{B}} \exp(\text{Sim}(Q, K'))}$$

where \mathcal{B} represents a training batch of query-keyword pairs, θ represents the model’s learnable parameters, λ is a hyperparameter for weighting the two losses, and $\text{Sim}(Q, K)$ is the cosine similarity between their dense embeddings: $\frac{\mathbf{e}(Q)^T \mathbf{e}(K)}{\|\mathbf{e}(Q)\| \cdot \|\mathbf{e}(K)\|}$.

2.3 Context Glancing

A key challenge in our proposed workflow is ensuring the retrieval model functions effectively both with and without available context. To address this, we introduce *context glancing*, a curriculum learning based strategy that gradually accustoms the model to scenarios where context might be absent. Initially, the model is trained for several epochs

with context provided for all training examples. Subsequently, we progressively drop context from a subset of training examples, increasing the drop rate throughout the training process. We employ a combination of random and structured context dropping methods. For d_{rand} fraction of queries, we randomly drop k contexts, with k sampled uniformly from 1 to n . For d_{web} and d_{qp} fractions of queries, we drop all web-based and query profile-based contexts, respectively. Finally, for d_{all} fraction of queries, we remove all context signals. All drop rate parameters (d_{rand} , d_{web} , d_{qp} , and d_{all}) are gradually increased during training based on a linear schedule following an initial warm-up phase with no context dropping. Our approach follows the principles of curriculum learning by gradually introducing increasingly difficult scenarios. The model initially learns in a context-rich environment, becoming progressively accustomed to handling cases with partial or even complete absence of context signals. Notably, the Augmented Unity architecture adapts to varying numbers of input contexts.

3 Results & Discussion

In this section, we begin by outlining the offline experimental setup, including the datasets used and evaluation metrics in Section 3.1. Sections 3.2 and 3.3 discuss the core offline results, showcasing the performance gains achieved by Augmented Unity. We further dissect the impact of different components within our approach through detailed ablation studies in Section 3.4. Finally, in Section 3.5, we discuss our online experiments and results.

3.1 Experimental Setup

Dataset: We construct a dataset of high-quality query-keyword pairs extracted from the search logs of a prominent search engine, encompassing 40 languages globally. The training set consists of approximately 60M unique queries, 240M unique keywords, and 900M query-keyword pairs. The keywords are chosen to be either exact, phrase, or smart match variants of the query. Our test set consists of 1M queries sampled across all languages. Retrieval is performed against a corpus of 1B keywords sampled from the full bid keyword corpus.

Evaluation Metrics: Evaluating the quality of retrieved keywords for a given query often necessitates nuanced understanding beyond simple n-gram matching (Mohankumar et al., 2022). While recent studies (He et al., 2023) demonstrate the

superior accuracy of LLMs like GPT-4 compared to crowd-sourced human annotators for this task, using such models for large-scale evaluation over billions of query-keyword pairs remains computationally expensive. To address this, we first curate a large-scale dataset annotated with query-keyword match type quality (exact, phrase, smart) using GPT-4. We then use this dataset to train a smaller, computationally efficient student model capable of accurately predicting match type quality. We utilize this student model’s predictions to compute Precision@K, separately for each match type.

Baselines: We compare Augmented Unity against several competitive context-free baselines: CLOVERv2 (Mohankumar et al., 2022), PIXAR (Valluri et al., 2024), SimCSE (Gao et al., 2021), NGAME (Dahiya et al., 2022), and Unity (Mohankumar et al., 2022). Due to space constraints, we provide further details on the baselines and the implementation details in Appendix A.

3.2 Offline Results

Table 2 presents a comparative analysis of Augmented Unity’s retrieval performance against prominent context-free NLG and DR methods. Our results demonstrate that Augmented Unity consistently outperforms the best-performing context-free baselines across all match types. Specifically, the NLG component of Augmented Unity surpasses Unity by 12-20% and the state-of-the-art PIXAR method by 5-13% in P@100. Note that our approach of leveraging query context signals is complementary to the idea of scaling up the vocabulary in PIXAR. Similarly, the DR component of Augmented Unity showcases substantial gains, with a 7-20% improvement in P@100 compared to the baseline Unity DR model. These findings underscore the effectiveness of our approach in leveraging additional context to improve query understanding. Further, despite processing 13x more tokens due to the inclusion of query context, the GPU serving cost of Augmented Unity remains comparable to Unity, within 7-9%. This efficiency stems from our use of the Fusion-in-Decoder architecture, where inference complexity scales as $O(NL_{max}^2)$, in contrast to $O(N^2L_{max}^2)$ when concatenating and encoding all N contexts together (L_{max} is the max sequence length of the contexts).

3.3 Evaluation with GPT-4 as Judge

We also conducted additional evaluations with GPT-4 as the evaluation model. We randomly sampled

Model	Exact Match		Phrase Match		Smart Match		GPU Cost
	P@100	P@200	P@100	P@200	P@100	P@200	
NLG							
CLOVERv2	6.78	4.97	16.58	14.08	31.83	28.04	1.17x
Unity NLG	7.00	5.14	16.93	14.36	32.69	28.81	1.17x
PIXAR	7.50	5.48	18.12	15.30	34.87	30.54	1.41x
Aug Unity NLG (Ours)	8.43	6.21	19.06	16.26	36.88	32.35	1.26x
DR							
NGAME	9.66	6.77	16.21	12.48	42.86	39.47	1.00x
SimCSE	9.70	6.82	16.50	12.78	43.88	40.26	1.00x
Unity DR	10.58	7.52	18.34	14.20	48.29	44.30	1.00x
Aug Unity DR (Ours)	12.74	8.87	20.86	15.82	52.05	45.66	1.09x

Table 2: **Performance and Efficiency Comparison of Augmented Unity with Context-Free Methods.** Precision (P) at 100 and 200 for different match types are reported, along with relative GPU serving cost compared to NGAME

Lang	EM		PM		SM	
	Unity	Aug	Unity	Aug	Unity	Aug
NLG						
English	12.05	14.29	24.77	26.74	44.28	48.40
French	9.91	11.90	25.27	27.43	47.33	51.95
German	9.90	11.93	24.38	26.43	41.39	45.54
DR						
English	19.09	23.21	28.44	32.99	58.15	66.23
French	17.55	21.75	29.28	34.69	61.86	69.65
German	17.34	21.07	29.18	34.36	58.31	66.32

Table 3: **GPT-4 as Judge:** Precision@50 for Unity and Augmented Unity with GPT-4 as the evaluation model

1000 queries each from English, French, and German from our test set, and retrieved the top 50 keywords from Unity and Augmented Unity. These query-keyword pairs were then evaluated by GPT-4, which provided binary judgments for exact, phrase, and smart match quality. Table 3 presents the results, demonstrating that Augmented Unity consistently outperforms Unity across all three languages and match types. We observe an average relative improvement of 12.6% for NLG and 17.7% for DR, further validating the effectiveness of our proposed approach in improving keywords retrieval.

3.4 Ablation Studies

Augmented Unity incorporates three key components: (i) leveraging various query contexts from both web search results and LLM-generated Query Profiles, (ii) utilizing multiple instances of each context type, (iii) using context glancing to enhance model robustness to scenarios with missing context signals. To understand the contribution of each component, we conducted a series of ablation studies, which are detailed below:

Context (Num)	EM	PM	SM
None (0)	10.45	17.35	45.70
Web Title (4)	11.61	19.13	49.58
Web Snippet (4)	11.16	18.45	48.07
QProfile Rewrites (4)	11.60	19.32	49.47
QProfile Intent (1)	12.07	20.06	49.86
Web Title (4) + Snippet (4)	11.63	19.18	49.70
QProfile Rewrites (4) + Intent (1)	12.17	20.15	49.96
All (13)	12.74	20.86	52.05

Table 4: Precision@100 for Augmented Unity DR with different types of query contexts used during inference

Context Type: Augmented Unity leverages four distinct types of query context: web titles, web snippets, Query Profile rewrites, and Query Profile intent. Table 4 shows the impact of using these different context types on the performance of Augmented Unity DR, as measured by P@100. The results reveal several key insights: (1) Irrespective of the type, incorporating any context leads to a substantial performance gain over the context-free scenario. This highlights the inherent value of each signal in enhancing query understanding. (2) Among the four types, Query Profile intent yields the largest performance gains. This could be because the intent derived from GPT-4 often provides a concise and accurate explanation of the user’s underlying intent, directly aiding in disambiguation. (3) Utilizing both web results and Query Profile context outperforms using either source alone. This indicates that these sources provide complementary information, highlighting the importance of leveraging them jointly.

Number of Context Instances: While utilizing various context types proves beneficial, determining the optimal number of contexts to use is crucial.

# Context	Context	EM	PM	SM
4	1 Title + 1 Snippet + 1 Rewrite + Intent	12.03	19.89	49.20
7	2 Title + 2 Snippet + 2 Rewrite + Intent	12.32	20.08	50.04
13	4 Title + 4 Snippet + 4 Rewrite + Intent	12.74	20.86	52.05
31	10 Title + 10 Snippet + 10 Rewrite + Intent	12.55	20.39	50.57

Table 5: Precision@100 for Augmented Unity DR with different numbers of context instances used

Context	w/o GG	w CG	Δ
None	14.36	17.35	20.8%
Web Title	17.04	19.13	12.3%
Web Snippet	16.17	18.45	14.1%
QProfile Rewrites	17.06	19.32	13.3%
Qprofile Intent	19.30	20.15	4.4%
Web Title + Snippet	17.14	19.18	11.9%
Qprofile Rewrites + Intent	19.30	20.15	4.4%
All	19.80	20.86	5.3%

Table 6: Phrase Match P@100 for Augmented Unity DR trained with and without Context Glancing (CG)

To investigate this, we varied the number of web titles, web snippets, and query profile rewrites. Table 5 displays the P@100 scores for different number of contexts used per type. As evident from the results, increasing the number of contexts per type from 1 to 4 consistently enhances retrieval performance. However, further increasing the number of contexts to 10 leads to a performance decline. This suggests that while incorporating multiple contexts per type can be advantageous up to a certain point, including an excessive number of potentially noisy contexts can negatively impact retrieval accuracy.

Context Glancing: Table 6 demonstrates the effectiveness of our context glancing strategy in enhancing model robustness. Without context glancing, the model exhibits a significant performance drop of 27.5% (in terms of phrase match P@100) when context signals are unavailable during inference, highlighting an over-reliance on availability of context. However, incorporating context glancing consistently improves performance across all scenarios, regardless of context availability. For instance, in the complete absence of context, context glancing leads to a 20.8% improvement in phrase match P@100. Remarkably, even when full context is provided, context glancing still yields a 5.3% performance gain. This suggests that our approach of gradually exposing the model to increasingly challenging scenarios not only enhances robustness to

Language	Δ Revenue	Δ Clicks	Δ QBR	Δ Defect
English	1.00%	0.33%	0.01%	0.02%
Non-english	1.44%	0.72%	0.18%	0.19%

Table 7: Online A/B results on a commercial search engine. Gray color indicates p-value > 0.01

Decile	Δ Query Coverage		Δ Ad Impressions	
	English	Non-English	English	Non-English
1	0.66%	1.49%	0.84%	1.71%
2	0.09%	0.70%	0.01%	0.35%
3	1.16%	0.48%	0.37%	0.86%
4	0.49%	0.51%	0.26%	0.48%
5	0.41%	0.51%	0.59%	0.61%
6	0.35%	0.56%	0.55%	0.70%
7	0.61%	0.71%	0.66%	0.54%
8	0.76%	0.69%	0.98%	0.74%
9	0.51%	1.12%	1.47%	1.09%
10	1.07%	0.76%	1.66%	0.76%

Table 8: Percentage Change in query coverage and Ad impression for different deciles in online A/B tests on sponsored search

missing context but also leads to a more generalizable model with improved overall performance.

3.5 Online A/B Testing

To validate the effectiveness of Augmented Unity in a real-world setting, we conducted extensive online A/B testing for 30 days on live traffic of a prominent commercial search engine, spanning over 160 countries. We deployed both the NLG and DR components of Augmented Unity and compared their performance against an ensemble of state-of-the-art retrieval techniques, including proprietary DR and NLG models, Unity, large language models, extreme classification, and graph-based methods. We measured the overall revenue, ad clicks, Quick Back Rate, and ad defect. Quick Back Rate (QBR) denotes the percentage of ad clicks with users quickly returning to the search results page. Ad defect, measured by offline relevance models, denotes the percentage of irrelevant ads shown to users. Table 7 summarizes the online A/B testing results, segmented by English and non-English queries. Augmented Unity improved user engagement, yielding statistically significant increases in overall clicks – a 0.72% lift for non-English queries and a 0.33% lift for English queries. Critically, these gains were not accompanied by any statistically significant degradation in QBR or ad defect. This suggests that Augmented Unity was able to retrieve keywords that aligned with the user

intent.

We also analyzed our online A/B experiment results by grouping queries into frequency-based deciles. Decile 1 contains highly frequent queries, while decile 10 consists of a large number of rare queries. Table 8 shows the query coverage (the fraction of queries for which any sponsored content was shown) and ad impressions (the total number of ads displayed). We observe an increase in both query coverage and ad impressions across all deciles, with particularly strong gains on tail queries, which are often longer and more ambiguous. As a result of the improved user engagement, we observed substantial revenue gains – 1.43% for non-English queries and 1.02% for English queries – underscoring the tangible business impact of our approach within a real-world sponsored search ecosystem.

4 Conclusion

In this paper, we introduced Augmented Unity, a novel approach that leverages rich query context to enhance sponsored search retrieval. By integrating web search results and GPT-4 generated Query Profiles, Augmented Unity effectively disambiguates user intent and retrieves more relevant bid keywords. Furthermore, our proposed context glancing strategy ensures robust performance even when contextual information is unavailable. Through extensive offline experiments and rigorous online A/B testing on a commercial search engine, we showed substantial improvements in key metrics such as retrieval accuracy, user engagement (ad clicks), and ad revenue. These findings underscore the significant potential of incorporating contextual information for achieving more effective and efficient sponsored search retrieval.

References

- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, E. Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.
- Kunal Dahiya, Ananye Agarwal, Deepak Saini, K Gururaj, Jian Jiao, Amit Singh, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Siamese: Siamese networks meet extreme classifiers with 100m labels. In *ICML*.
- Kunal Dahiya, Nilesh Gupta, Deepak Saini, Akshay Soni, Yajun Wang, Kushal Dave, Jian Jiao, K Gururaj, Prasenjit Dey, Amit Singh, Deepesh V. Hada, Vidit Jain, Bhawna Paliwal, Anshul Mittal, Sonu Mehta, Ramachandran Ramjee, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2022. [Ngame: Negative mining-aware mini-batching for extreme classification](#). *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. [Simcse: Simple contrastive learning of sentence embeddings](#). In *Conference on Empirical Methods in Natural Language Processing*.
- Xingwei He, Zheng-Wen Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2023. [Annollm: Making large language models to be better crowdsourced annotators](#). *ArXiv*, abs/2303.16854.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Gautier Izacard and Edouard Grave. 2020. [Leveraging passage retrieval with generative models for open domain question answering](#). *ArXiv*, abs/2007.01282.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L el io Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth ee Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Yijiang Lian, Z. Chen, J. Hu, Ke feng Zhang, Chunwei Yan, Muchenxuan Tong, W. Han, Hanju Guan, Y. Li, Y. Cao, Yang Yu, Z. Li, X. Liu, and Y. Wang. 2019. An end-to-end generative retrieval method for sponsored search engine -decoding efficiently into a closed target domain. *ArXiv*, abs/1902.00592.
- Anshul Mittal, Kunal Dahiya, Sheshansh Agrawal, Deepak Saini, Sumeet Agarwal, Purushottam Kar, and Manik Varma. 2021. Decaf: Deep extreme classification with label features. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*.
- Akash Kumar Mohankumar, Nikit Begwani, and Amit Singh. 2021. Diversity driven query rewriting in search advertising. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*.
- Akash Kumar Mohankumar, Bhargav Dodla, K Gururaj, and Amit Singh. 2022. [Unified generative & dense retrieval for query rewriting in sponsored search](#). *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*.
- Weizhen Qi, Yeyun Gong, Yu Yan, Jian Jiao, B. Shao, R. Zhang, H. Li, N. Duan, and M. Zhou. 2020. Prophetnet-ads: A looking ahead strategy for generative retrieval models in sponsored search engine. *ArXiv*, abs/2010.10789.

Ravisri Valluri, Akash Kumar Mohankumar, Kushal Dave, Amit Singh, Jian Jiao, Manik Varma, and Gaurav Sinha. 2024. [Scaling the vocabulary of non-autoregressive models for efficient generative retrieval](#). *ArXiv*, abs/2406.06739.

A More Details on Experimental Setup

This section provides further details about the experimental setup, including baseline descriptions and implementation details.

A.1 Baselines

We compare Augmented Unity against strong baselines in both dense retrieval (DR) and natural language generation (NLG) for keyword retrieval:

Dense Retrieval: We utilize the same baselines as reported in (Mohankumar et al., 2022): NGAME (Dahiya et al., 2022) and SimCSE (Gao et al., 2021). Both NGAME and SimCSE employ a siamese dual encoder architecture to represent queries and keywords in a dense vector space. However, they differ in their approaches to curating negatives and their training objectives. SimCSE uses a contrastive InfoNCE-style loss with in-batch random negatives, while NGAME adopts a triplet margin loss. Moreover, NGAME uses a clustering-based strategy to curate batches, ensuring that the batches themselves contain hard negatives. In addition to these methods, we also compare against the DR component of the Unity model, which utilizes a dual encoder architecture similar to NGAME.

NLG: We compare our model against CLOVERv2, the NLG component of Unity (Mohankumar et al., 2022), and PIXAR (Valluri et al., 2024) as NLG baselines. All these models are non-autoregressive, predicting the keyword token distribution independently and in parallel. Autoregressive models, though potentially effective, are impractical for online deployment due to significantly higher latency and inference costs (Mohankumar et al., 2022). CLOVERv2 utilizes an encoder-based architecture with a language modeling head to map the hidden states to the vocabulary space, thereby obtaining the final token distributions. It then performs a constrained beam search using a Trie to produce the predicted keywords. Unity leverages a similar approach while sharing its model with the DR component. PIXAR scales up the target vocabulary in non-autoregressive models to include phrases.

A.2 Implementation Details

Model details: For Augmented Unity, We employ a 4-layer encoder and 4-layer decoder architecture ($L_e = 4, L_d = 4$) with a hidden size (d) of 512, utilizing the multilingual 250k vocabulary of XLM-R (Conneau et al., 2020). The model is trained from

Language	Unity	Aug	Δ
English	22.69	26.88	18.5%
German	23.00	27.88	21.2%
French	22.83	27.45	20.2%
Maltese	5.86	8.44	44.1%
Icelandic	7.62	10.81	41.8%
Lithuanian	9.55	13.53	41.8%
Latvian	10.38	14.46	39.3%
Albanian	7.95	10.93	37.5%
Slovenian	11.58	15.63	35.0%

Table 9: Phrase Match P@100 for Augmented Unity and Unity DR across different languages

scratch. For NGAME, SimCSE, Unity, and PIXAR, we utilize the XLM-R base encoder model initialized with pretrained weights. The dense retrieval embedding size (d') is set to 128.

Training details: All models are trained with a learning rate of 5×10^{-5} , 1000 warmup steps, and an effective batch size of 16384 for 10 epochs. When using context glancing, we train for first 3 epoch without any context dropping and then linearly increase each dropping rate parameter (d_{rand} , d_{web} , d_{qp} , and d_{all}) to a maximum value of 10%. We utilize the Adam optimizer with a linear decay learning rate scheduler. The dense retrieval loss weight hyperparameter (λ) is set to 1. We train all our models on 16x AMD Mi200 GPUs with DeepSpeed stage 1.

B Multilingual Analysis

Table 9 presents a language-wise breakdown of phrase match P@100 for Augmented Unity DR and Unity DR. Augmented Unity demonstrates consistent improvements across all languages, including high-resource languages like English, German, and French. Notably, we observe a substantial 18.5% relative gain in phrase match P@100 for English. The gains are even more pronounced for low-resource languages like Maltese and Icelandic, highlighting the benefits of incorporating additional context, particularly when training data is limited. These results suggest that incorporating additional context is particularly beneficial for languages with limited training data. We also note that the Query Profile intent is generated in English regardless of the query language and hence provides valuable cross-lingual context, aiding in improved query understanding for tail languages.

C Error Analysis

While Augmented Unity leads to significantly better model performance overall, there are a few cases where adding context can lead to errors. Table 10 shows a few selected failure modes for the model.

D Alternatives to GPT-4 for Query Profile

Running GPT-4 for a large number of queries to build a sufficiently large Query Profile cache can get fairly expensive. In order to mitigate this, we experiment with a finetuning based approach with Mistral 7B (Jiang et al., 2023) with LoRA (Hu et al., 2022) for generating QProfile Rewrites. We evaluate the generated rewrites on three metrics precision, novelty and diversity. Precision is calculated as described in 3.3, while novelty and diversity are calculated by GPT-4, where we get a score between 1-5 for novelty for each of rewrites and a score between 1-5 for diversity for all 10 rewrites together. Table 11 shows a comparison of results between rewrites generated by GPT-4 and Mistral-7b. Our results show that fine-tuned Mistral-7B achieves comparable performance to GPT-4 in terms of query rewrite precision, at the cost of novelty and diversity, while significantly reducing inference costs. Improving novelty and diversity without losing out on precision could be a fruitful direction for a future work.

Example: Noise from Web Snippets

Query	compensation injury claim
Web Snippet	Report an occupational injury to the Compensation Fund Report injuries sustained by your employees in the course of their work to the Compensation Fund within seven days of the occurrence. The fund covers permanent, casual workers, trainees and apprentices who are injured in the course of their work and lose income or are impaired as a result
Predictions	
Augmented Unity	what to claim on income tax
Unity	compensation for injury
Analysis	
The model incorrectly picked up the token “income” from the context, leading to an irrelevant keyword related to income tax.	

Example: Location Bias

Query	cruises singles
Web Title	Singles Cruise Deals Marella Cruises TUI.co.uk
Predictions	
Augmented Unity	cruises holidays uk
Unity	cruises for singles only
Analysis	
The cached web result was location-specific (UK), leading to a geographically biased keyword retrieval.	

Table 10: **Some qualitative examples of failure modes for Augmented Unity compared to Unity.**

Model	Precision@10	Avg Novelty (1-5)	Avg Diversity (1-5)
GPT-4	6.9	2.19	2.53
Finetuned Mistral-7b	7.62	2.05	2.36

Table 11: Performance of Finetuned Mistral-7b vs GPT-4 for QProfile Rewrites