DQAC: Detoxifying Query Auto-Completion with Adapters

Aishwarya Maheswaran*[1], Kaushal Kumar Maurya*[1]
Manish Gupta[2], Maunendra Sankar Desarkar[1]

[1]IIT Hyderabad, India, [2]Microsoft, India
ai21resch11002@iith.ac.in, cs18resch11003@iith.ac.in,
gmanish@microsoft.com, maunendra@cse.iith.ac.in

Abstract. Recent Query Auto-completion (QAC) systems leverage natural language generation or pre-trained language models (PLMs) to demonstrate remarkable performance. However, these systems also suffer from biased and toxic completions. Efforts have been made to address language detoxification within PLMs using controllable text generation (CTG) techniques, involving training with non-toxic data and employing decoding time approaches. As the completions for QAC systems are usually short, these existing CTG methods based on decoding and training are not directly transferable. Towards these concerns, we propose the first public QAC detoxification model, Detoxifying Query Auto-Completion (or DQAC), which utilizes adapters in a CTG framework. DQAC operates on latent representations with no additional overhead. It leverages two adapters for toxic and non-toxic cases. During inference, we fuse these representations in a controlled manner that guides the generation of query completions towards non-toxicity. We evaluate toxicity levels in the generated completions across two real-world datasets using two classifiers: a publicly available (Detoxify) and a search query-specific classifier which we develop (QDETOXIFY). DQAC consistently outperforms all existing baselines and emerges as a state-of-the-art model providing high quality and low toxicity. We make the code publicly available[1].

Keywords: Auto-completion· Query Detoxification· Controllable Text Generation· Language Generation· Pre-trained Models· Adapters

1 Introduction

Query auto-completion (QAC) systems have become an integral part of modern search engines, primarily enriching the user experience by providing potential query completions. Over the past several decades, there has been active research on QAC, encompassing traditional methodologies like log-based approaches [15], learning to rank-based approaches [22], and many more [1]. However, more recently QAC systems have demonstrated remarkable performance by leveraging state-of-the-art technologies such as natural language generation (NLG) [4] and pre-trained language models (PLMs) [13]. A notable challenge arises when these systems produce toxic completions, which can be unexpected and potentially detrimental. The ramifications of encountering such potentially harmful suggestions can be far-reaching, encompassing negative impacts on user experience, erosion of trust in the search engine, and perpetuation of biases

* These authors contributed equally to this work
[1] https://shorturl.at/zJ024

in the training data. *Toxicity in QAC refers to the presence of harmful, offensive, or inappropriate suggestions that may appear during the automated recommendation of completions of search queries [16].* This paper is a step towards mitigating/reducing toxicity in query completions for QAC systems based on PLMs.

Traditionally blocklist of toxic words were used to avoid generating toxic suggestions, the drawbacks of this approach are (1) the list needs to be constantly updated, (2) mere presence of toxic words does not necessarily classify a query as toxic. Ex: "deepfake daughter s*x" is toxic while "f*ck you knowledge lyrics" is non-toxic. Recently, active efforts have been made to detoxify text generated using large pre-trained language models. These efforts can be broadly categorized into three main approaches. (1) Controlled text generation (CTG) through fine-tuning with clean datasets [5]. The drawbacks of this approach relates to the difficulty in obtaining a clean dataset and the need to retrain these models. (2) Decoding time algorithms for CTG [2, 10]. These algorithms aim to modify the decoding process during generation to ensure that the output aligns with desired constraints. The drawback of this approach is the increased time during generations, which is not desirable for auto-complete settings. (3) Reinforcement learning (RL) techniques to *unlearn* toxic content [12], by providing feedback in the form of toxicity scores for generations. It is important to note that the aforementioned approaches have primarily been found to perform well in scenarios where the input/prompt and completions are well-formed and longer in nature. Specific characteristics of QAC datasets like comparatively shorter text length (due to nature of queries), spelling and grammatical errors, hinder the adaptation of existing detoxification models for QAC systems as is. In Section 5 of our paper, we provide experimental and quantitative evidence to support these claims, including Tables 1 and 2 for reference.

Towards these concerns, we propose a novel approach called *DQAC: Detoxifying Query Auto-Completion*, which utilizes Adapters [8] in a CTG framework to reduce toxicity in query auto-completion. It utilizes toxicity-aware adapter that steers the latent state to generate non-toxic completions with lower parameters compared to fine-tuning the entire model. Overall, our main contributions are as follows. (1) We introduce DQAC which, to the best of our knowledge, is the first publicly available query detoxification model designed explicitly for Query Auto-completion systems. We compare its performance on Bing and AOL datasets against several strong baselines. (2) We develop a novel toxicity classifier model called QDETOXIFY to assess the toxicity level of complete queries from QAC systems. It demonstrates a high accuracy rate of 96%. (3) We introduce the first toxicity evaluation benchmark for QAC models, i.e., *DQAC-Benchmark*, to stimulate further research within this domain. (4) We make the code and models for AOL dataset publicly available[1].

2 Related Work

Detoxification in QAC: Existing models for detoxifying QAC are limited to discovery and detection approaches. Leading search engines typically manage toxicity by maintaining a blocklist of offensive terms, engaging in red teaming, or soliciting users to report objectionable completions. However, these methods need constant monitoring and maintenance. Other techniques such as maintaining common query templates were introduced to reduce these overheads, yet their coverage remains limited. Conversely, learning-based approaches were explored using query embedding, active learning and

machine learning for the detection and removal of toxic completions. Additionally, N-strike rules were proposed to generate multiple completions and eliminate the toxic ones. These approaches do not provide safe alternatives for blocking toxic content. we address this drawback via our proposed DQAC model.

Detoxification with CTG: Initial methods for CTG were based on word filtering where a specific set of words are disallowed during generation [3] which has scalability and maintenance constraints. Fine-tuning the NLG (PLM) models with desirable attribute datasets (i.e., non-toxic) [5] can steer generation towards desirable attributes, but the model does not learn how to handle toxic cases. Another popular approach is to alter the generation strategy called *Decoding time* approaches. Dathathri et al. [2] propose PPLM which uses an attribute model to get gradients with respect to the desired class and updates the hidden representations of the PLM. This method is computationally expensive, as shown in [3], which makes its deployment unfavorable. Close to our work, Liu et al. [10] proposed the DExperts model, which uses a base PLM along with two additional fine-tuned LMs, to learn desirable and undesirable attributes. This is again computationally expensive and requires larger memory footprint. Unlike this, DQAC is efficient by having $3\times$ less number of parameters and fine-tuning latency. Recently, Lu et al. [12] proposed an RL-based approach for CTG called *Quark*. It is trained using an RL approach with iteration sampling, quantization steps and the reward function as toxicity score. However, for QAC detoxification task, we found it to be ineffective due to the unstructured and short nature of queries. Our proposed model is specifically designed to operate at the latent representation level, employing adapters, and exhibits improved performance when handling short prefixes and completions.

Text Generation with Adapters: Adapters [8] are lightweight (consisting of a small number of parameters) modules inserted into each layer of the PLM to adapt it to downstream task/domain/language. While training an adapter, all the parameters of the original pre-trained LM are frozen to mitigate the effect of *catastrophic forgetting* [14]. These light-weight modules enable parameter-efficient training and significantly reduce the fine-tuning computation cost [20]. Ustun et al. [21] used language-specific denoising adapters for unsupervised machine translation tasks. We take inspiration from previous studies and explore the application of adapters in CTG framework for QAC tasks.

3 Methodology

In this section, we first introduce the QAC detoxification problem, and subsequently delve into the specifics of the proposed toxicity classifier model, i.e., QDETOXIFY. Lastly, we furnish architectural details of the proposed DQAC model.

Problem Statement: The task of detoxifying QAC can be formulated as a *controlled text generation* problem. A QAC system comprises of a triplet: $\langle$*session, prefix, complete query*[2]$\rangle$. Here, the session s consists of the previous n queries (ordered from earliest to latest) searched by the user. The current query being typed by the user is represented as a complete query q, and p is the query prefix entered so far. Formally, for a given input s and p, the goal is to generate m (we set m=10) completions that are close to the actual human-generated queries and should be relevant with respect to the

[2] also called as completion or query

session. The completions should have desired behaviors (i.e., non-toxicity) and should not have undesired behaviors (i.e., toxicity). The incorporation of session information lends a personalization aspect to the model.

3.1 QDETOXIFY: Toxicity Classifier for Search Queries

The primary prerequisite for evaluating any detoxification model is a reliable evaluation model that provides a numerical value capable of determining the toxicity level of the generated text. Some well-known models are Perspective API [9], Detoxify [6] and ToxiGen [7]. However, these models possess their own limitations and exhibit biases [17], which restrict their usage, casting doubts on their reliability. None of these models have been trained using any QAC datasets, which typically feature short and structurally distinct text, highlighting the disparity. Further, since we work with a proprietary dataset (Bing), we require an offline tool for evaluating toxicity. In response to these concerns, we train a toxicity classifier specifically designed for QAC systems called QDETOXIFY. It generates a score ranging from 0 to 1, where a score $\geq$0.5 is considered toxic. QDETOXIFY is developed by leveraging the publicly available Detoxify model. Detoxify model uses RoBERTa as the base pretrained model which is fine-tuned with the Jigsaw dataset[3], QDETOXIFY was trained using a labeled query log dataset from Bing where each query is labeled as "toxic" or "non-toxic" using their proprietary classifiers.

The dataset comprises of $\sim$7.59M training, 100K validation, and 100K test examples. Each of these splits includes an equal number of both toxic and non-toxic samples. The model was fine-tuned for 128 epochs using a learning rate of 2e-4. SGD optimizer and cross-entropy loss were employed in the training process. This training strategy allows QDETOXIFY to leverage the knowledge learned from Detoxify and RoBERTa, through transfer learning on a diverse range of toxic and non-toxic texts.

Results: The proposed QDETOXIFY model achieves a high accuracy of 95.96% on the test set, while the corresponding score for Detoxify is just 82.82%. There exists a high correlation of 0.797 between QDETOXIFY and Detoxify, supporting the hypothesis that QDETOXIFY effectively leverages the learning acquired from Detoxify. A query 'm.i.c.r.o.s.o.f.t.' is rated as toxic by Detoxify (score=0.58) where QDETOXIFY correctly classified it as non-toxic (score=0.23). Based on these findings, we conclude that QDETOXIFY is an accurate and reliable evaluation model for measuring the toxicity of search queries.

3.2 The DQAC Model

The proposed DQAC model is based on natural language generation using Transformer-based pre-trained language models.

DQAC Model Architecture: Fig. 1 shows the details of the proposed DQAC model. The model architecture is based on *personalized pre-trained language models* which is obtained by fine-tuning the base PLM using a large personalized auto-completion dataset. We will refer to this model as PrsGPT2. The personalized auto-completion dataset consists of session and prefixes as input and completions as the target. Further, two trainable adapters, i.e., *non-toxic* (A^+) and *toxic* (A^-), are added at each transformer layer (after feed-forward neural network sub-layer) of the personalized PLM in parallel. The representation from the feed-forward neural network sub-layer output is

[3] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification

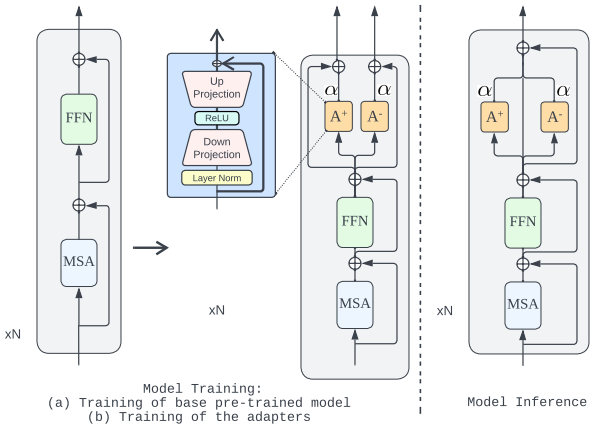

Fig. 1: DQAC model details: (left) training (right) inference. Here MSA is Multi-head Self-attention and FFN is Feed Forward Network.

passed through non-toxic and toxic adapters in parallel to shift the hidden representations towards specific desirable and undesirable behaviors, respectively. Finally, the hidden representations from two adapters and base LM are fused in a controlled manner such that the final fused representation is inclined towards expert attribute behaviors, which is fed as input to the next layer.

Formally, an adapter A_i (A_i^+/A_i^-) at layer i consists of layer-normalization (LN), followed by down-projection $W_{down} \in \mathbb{R}^{k \times d}$ with bottleneck dimension d, non-linear activation ReLU and up projection $W_{up} \in \mathbb{R}^{d \times k}$ combined with input $h^i \in \mathbb{R}^k$ through residual connection, where k is the transformer's hidden layer dimension. Overall, adapter A_i outputs $A_i(h^i) = W_{up}^T \text{ReLU}(W_{down}^T LN(h_i)) + h_i$. Bias terms are omitted for clarity. As adapters add only a small number of additional parameters, the generated text consists of desirable behavior and at the same time has comparable number of parameters to the personalized PLM. This modeling also enables an easy adaptation to different domains and languages.

Text Generation with DQAC: At decoding time step t, given an input session+prefix X_t, the personalized PLM computes hidden representation h_t^i at i-th layer. DQAC alters this representation by passing it through the adapter modules, and then performing *representation fusion*, to obtain the final representation z_t^i. In particular, h_t^i is passed through non-toxic and toxic adapters to get output representations $r_t^{i+} = A_i^+(h_t^i)$ and $r_t^{i-} = A_i^-(h_t^i)$, respectively. These outputs are then fused to obtain the controlled output representation as given in Equation 1. The fusion tries to steer the representation towards the output of the non-toxic adapter, and away from the representation generated by the toxic adapter, thereby attempting to bias the model towards non-toxic generation.

$$z_t^i = h_t^i + \alpha(r_t^{i+} - r_t^{i-}) \tag{1}$$

where α is a hyper-parameter that controls the amount of steering over the base language model. The next token x_{t+1} is obtained with the standard language model decoding approach. We use beam search decoding to generate 10 completions.

Overall DQAC Model Training: The DQAC model undergoes training in three distinct stages. First, the model is trained to incorporate personalized context by fine-tuning the PLM known as the personalized PLM (PrsGPT2). The second stage involves training the toxic adapter while freezing the rest of the model parameters including non-toxic adapters with an annotated toxic QAC dataset. Finally, the third stage focuses on training the non-toxic adapter while freezing the rest of the model parameters including toxic adapters with an annotated non-toxic QAC dataset. For our experiments, we use GPT2 as the base PLM, while noting that the proposed framework is agnostic of the PLM choice. The order of the adapter training does not have a major impact on model performance as both the adapters work in parallel. Formally, we train the adapters A (A^+/A^-) to minimize the following loss.

$$L^A = -\sum_{S \in D_A} \log P(q_c|s; p; A) \tag{2}$$

where S is a sample in adapter-specific dataset D_A which consists of session s, prefix p and completion q_c.

Although the proposed approach seems similar to DExperts, it differs in the following novel ways: (1) It operates in the latent representation space, which is more suitable for reducing toxicity for QAC systems (more discussion in result section 5), (2) It does not have additional latency overhead like DExperts during generation which is crucial for the QAC systems and (3) The proposed model more efficient as DExperts takes $\sim$3x more RAM ($\sim$3x number of model parameters) compared to DQAC.

4 Experimental Setup

We seek to answer the following set of questions: (1) How to create a reliable evaluation benchmark for QAC toxicity evaluation? (2) What is the performance of existing state-of-the-art models for the QAC detoxification task? (3) How does the performance of the proposed DQAC model compare to these state-of-the-art baselines? (4) Does the performance of the DQAC model persist across different datasets and test set types?

Details of Datasets: We use two datasets to train and evaluate the model performance, i.e., *Bing* proprietary query log and *AOL* public query log datasets. The raw Bing data consists of three week worth user query log from October 2022. It was preprocessed to resemble AOL data format. Unlike AOL, the session and prefix were part of the dataset and hence no additional preparation was done. This makes the Bing dataset recent and a real user query log dataset. The training of the DQAC model requires two types of data: (1) **PQAC-Data:** a large-scale personalized query auto-complete training dataset to train base PLM (GPT2) to obtain personalized PLM (PrsGPT2) as discussed in Sec. 3.2 and (2) **Adapter-Data:** small toxic and non-toxic labeled datasets to train toxic and non-toxic adapters of DQAC model, respectively. PQAC-Data is obtained from *Bing*/*AOL*; however, Adapter-Data is obtained from Bing only. For both Bing and AOL, personalized QAC data is split temporally into train, validation and test such that train data is oldest and test data is the most recent. We call the train and validation parts together as PQAC-Data. The test part is referred to as DQAC-Benchmark (which is discussed in detail in Sec. 4). PQAC-Data and Adapter-Data are disjoint.

The raw AOL query log consists of a sequence of queries entered by users along with time-stamp details. Following previous studies [23], we split sequence of queries

into sessions with at least 30 minutes of idle time between two consecutive queries while ensuring each session has at least two queries (in earliest to latest order), i.e., $s = (q_1, q_2, \ldots, q_n q_{n+1})$. The prefix p_{n+1} is sampled from the last query q_{n+1} using exponential distribution to create triplet $\langle (q_1, q_2, \ldots q_n), p_{n+1}, q_{n+1} \rangle$ for each of the PQAC-Data example. Unlike the AOL dataset, where the prefix-to-query information is not explicitly available, and the prefixes are synthetically created by splitting a full query, the Bing dataset consists of real prefixes typed by users. We perform three pre-processing steps while preparing Bing PQAC-Data: (1) Restricting the maximum prefix length to 25 characters so that the model learns to predict for short queries. (2) We ensure the complete query is prefix-preserving by removing non-prefix-preserving examples. (2) We also verify that the query does not start with punctuation or numbers and has only ASCII characters. Adapter-Data is prepared from the Bing query log and has a similar triplet format as PQAC-Data. It consists of two labeled datasets, toxic and non-toxic, to train toxic and non-toxic adapters, respectively. For the Bing dataset PQAC-Data consists of 20M for training and 101K for validation while AOL datasets had 4M for training and 100K for validation. For training Adapters, 40K toxic and non toxic sets from Bing dataset were used.

Creation of Toxicity Evaluation Benchmark: To evaluate any detoxification model a reliable *evaluation benchmark* is required. Due to the lack of a public evaluation benchmark, we have created the first toxicity evaluation benchmark for QAC task: *DQAC-Benchmark*. The benchmark consists of two types of evaluation datasets: *non-toxic prefix and non-toxic query completions (`NPNQ`)* and *non-toxic prefix and toxic query completions (`NPTQ`)*. To construct these sets, we obtained toxicity scores using both QDETOXIFY and Detoxify for prefixes and queries (excluding sessions). We use average of both classifier scores to enhance the reliability of the dataset. **`NPNQ`** is a set of all examples where the toxicity score for prefix and query is $<$0.5 separately. On this set, we hypothesize that the QAC detoxification model should preserve exact completions while steering towards non-toxicity. **`NPTQ`** is a set of all examples where the toxicity score for prefix $<$0.5 and the score for query is $\geq$0.5. On this set, we hypothesize that the detoxification model should steer towards non-toxic completion while ensuring that the completions remain contextually aligned with the session and prefix, rather than necessarily matching the correct completions. For Bing datasets the size for both NPNQ and NPTQ is 30K, and for AOL NPNQ is 10K while NPTQ is 8.6K.

Baselines: This section provides an overview of the baseline models considered for comparison with the DQAC model. As our target is to develop a detoxification model for QAC, a comparison with regular QAC models is not required. Due to a lack of public detoxification models for QAC, to ensure fairness, we have selected state-of-the-art language detoxification NLG models from the natural language processing (NLP) community. For fair comparison, all the baselines are developed on top of the Personalized GPT2 model.

- **Personalized GPT2 (PrsGPT2)**: We fine-tune the GPT2 model with *PQAC-Data* for 3 epochs to obtain PrsGPT2 base model. We separately fine-tune for Bing and AOL PQAC-Data datasets.
- **DAPT [5]**: We continued fine-tuning PrsGPT2 with $\sim$4M non-toxic queries for which QDETOXIFY classifier scores are $<$0.5.

		Bing consolidated (NPNQ ∪ NPTQ)							
	Model	**Δ**MRR (%)↑	**Δ**SBMRR (%)↑	QDETOXIFY		Detoxify		**Δ**RR-BLEU (%)↑	**Δ**BLEU (%)↑
				ΔAmaxT (%)↓	**Δ**Prob(%)↓	**Δ**AmaxT(%)↓	**Δ**Prob(%)↓		
Baselines	PrsGPT2	-	-	-	-	-	-	-	-
	PPLM*	0.45	10.87	144.37	152.15	91.46	89.40	40.23	15.01
	DAPT	77.07	70.24	81.28	80.42	60.92	52.37	68.97	57.14
	Quark	10.68	21.89	68.77	65.13	29.39	20.57	40.23	24.13
	DExpert	16.13	18.60	33.33	28.67	19.21	13.61	46.26	23.11
Ours	T Adapter	21.20	27.47	38.70	29.26	25.62	13.92	43.39	41.75
	NT Adapter	**95.87**	**91.85**	91.90	89.55	72.58	71.36	**84.77**	**85.24**
	DQAC	43.03	39.91	**30.34**	**21.19**	**9.36**	**3.28**	48.28	39.55
		AOL consolidated (NPNQ ∪ NPTQ)							
	Model	**MRR** ↑	**SBMRR** ↑	QDETOXIFY		Detoxify		**RR-BLEU**↑	**BLEU** ↑
				AmaxT↓	**Prob**↓	**AmaxT**↓	**Prob**↓		
Baselines	PrsGPT2	**0.34**	**0.40**	0.54	0.53	0.31	0.33	**0.14**	**46.63**
	PPLM*	0.00	0.05	0.70	0.71	0.26	0.26	0.06	8.45
	GeDi	0.00	0.02	0.44	0.43	0.16	0.14	0.07	20.20
	DAPT	0.13	0.19	0.37	0.35	0.20	0.19	0.09	32.40
	Quark	0.32	0.39	0.54	0.54	0.28	0.30	0.14	46.21
	DExpert	0.00	0.02	0.28	0.25	0.08	0.04	0.07	22.25
Ours	T Adapter	0.06	0.13	0.28	0.24	0.09	0.05	0.08	27.64
	NT Adapter	0.01	0.09	0.52	0.51	0.26	0.27	0.06	7.44
	DQAC	0.08	0.14	**0.21**	**0.18**	**0.07**	**0.04**	0.08	30.67

Table 1: The consolidated average evaluation scores for Bing and AOL datasets, averaged across NPNQ and NPTQ test sets. "AmaxT" represents average maximum toxicity, and "Prob" denotes the toxicity probability. *Similar to [10], PPLM model was tested on 10% data. PrsGPT2 scores are not shown for Bing since the relative percentage ($S_{Model} * 100/S_{PrsGPT2}$) is computed with PrsGPT2. For AOL, we have reported raw scores.

		Bing - NPTQ							
	Model	**Δ**MRR (%)↑	**Δ**SBMRR (%)↑	QDETOXIFY		Detoxify		**Δ**RR-BLEU (%)↑	**Δ**BLEU (%)↑
				ΔAmaxT (%)↓	**Δ**Prob(%)↓	**Δ**AmaxT(%)↓	**Δ**Prob(%)↓		
Baselines	PrsGPT2	-	-	-	-	-	-	-	-
	PPLM*	0.19	13.64	113.99	115.95	89.67	87.78	34.55	18.09
	GeDi	0.05	0.65	85.88	84.83	95.27	86.01	27.27	13.83
	DAPT	34.11	35.71	80.44	80.80	56.39	50.64	55.91	44.85
	Quark	3.27	6.82	56.09	53.57	24.69	19.13	34.55	22.94
	DExpert	0.37	1.62	35.49	32.24	18.56	13.67	30.00	15.43
Ours	T Adapter	**70.56**	**70.46**	85.62	83.92	72.68	71.54	**73.64**	**70.09**
	NT Adapter	0.37	0.33	38.73	31.65	19.44	12.38	35.00	26.68
	DQAC	0.05	1.62	**29.73**	**22.78**	**5.43**	**3.01**	30.00	16.35
		Bing - NPNQ							
Baselines	PrsGPT2	-	-	-	-	-	-	-	-
	PPLM*	0.63	8.70	275.42	354.35	118.42	190.00	50.00	10.93
	GeDi	16.04	25.06	110.62	132.61	47.37	20.00	69.53	35.33
	DAPT	105.98	97.44	84.92	78.26	128.95	160.00	91.41	73.44
	Quark	15.66	33.76	123.46	129.71	100.00	110.00	50.00	25.70
	DExpert	26.73	31.97	**24.02**	**8.70**	**28.95**	**10.00**	74.22	33.28
Ours	T Adapter	**112.89**	**108.70**	118.99	121.01	71.05	60.00	**103.91**	**105.32**
	NT Adapter	35.22	48.85	38.55	15.94	118.42	110.00	57.81	61.72
	DQAC	71.95	70.08	32.96	12.32	68.42	20.00	79.69	70.31

Table 2: Model performance for NPTQ and NPNQ testset for Bing dataset. Rest of the notations are similar to Table 1.

- **PPLM [2]**: As implemented in the paper, we train a discriminator that learns to classify the hidden representation of the base PrsGPT2 model as toxic or non-toxic, using the 80K Adapter-Data.
- **DExperts [10]**: We use the base model as the PrsGPT2 checkpoint and train the expert and anti-expert models on the 40K toxic and non-toxic data splits.
- **Quark [12]**: We use QDETOXIFY score as a reward and base PLM as PrsGPT2. Similar to the official implementation, we train the model for 3M episodes.
- **T-Adapter and NT-Adapter:** These are ablation baselines, where we only consider the toxic and non toxic adapter seperately which is trained using the toxic and non toxic split of the Adapter-Data respectively.

Evaluation Metrics We consider the following metrics to evaluate the performance of the baseline and the proposed DQAC models.

- **Mean Reciprocal Rank (MRR):** Computed as MRR $= \frac{1}{D_{ts}} \sum_{i=1}^{D_{ts}} \frac{1}{r_i}$. Here, D_{ts} is the size of the test data and r_i is the rank of the ground-truth query in the generation (∞ if not found).
- **Semantic BERT Mean Reciprocal Rank (SBMRR):** It is a variant of MRR where an exact match is replaced by a semantic match between reference and complete query. We consider a match if cosine similarity is $\geq$0.9. We use the Sentence-BERT (*all-MiniLM-L6-v2*) [19] transformer model to obtain the query representations.
- **Bilingual Evaluation Understudy (BLEU):** We evaluate the degree of lexical match between the reference query and the first generated completion.
- **BLEU Reciprocal Rank (RR-BLEU) [23]**: It is defined as the reciprocal rank weighted average where weights are BLEU scores between reference and complete query.
- **Average Max Toxicity (AmaxT):** Following [3], we define it as the average of the maximum toxicity over 10 generations for a test example. It helps set an upper-bound on the toxic generations.
- **Empirical Toxicity Probability (Prob):** Following [3], we define it as the probability of at least one of any 10 generations being toxic (toxicity score $\geq$0.5).

The toxic scores were obtained from the MSDetoxify and Detoxify classifiers. Queries with toxicity score $>$0.5 were considered toxic otherwise non-toxic. We use GPT2 (*small*) as the base PLM. Different values of (α, d) influence the trade-off between toxicity and MRR. After hyperparameter tuning on the validation set, we find $(\alpha = 2.6, d = 8)$ leads to low toxicity scores while preserving semantic relevance. While generation, we use Beam search with beam size 10 to get 10 generations with a max generation length of 80.

5 Results and Analyses

All the evaluation scores for the baselines and the proposed DQAC model are presented in Table 1 which displays the consolidated average score of both NPNQ and NPTQ sets. Table 2 shows results separately for NPNQ and NPTQ on Bing dataset. In accordance with the confidential nature of the Bing dataset, the exact metric values cannot be disclosed, a practice that has been observed in previous studies as well [18]. Consequently, in Tables 1 and 2, and throughout the rest of the paper, the percentage improvement scores over the PrsGPT2 baseline are reported. Due to this, PrsGPT2 scores for Bing

are not shown. As AOL is a public dataset, we have reported exact evaluation scores for this dataset. The evaluation scores for MRR, SBMRR, RRBLEU and BLEU should be preferred high while scores for AmaxT and prob should be preferred low.

Comparison with state-of-the-art models: We compare with state-of-the-art baselines such as Quark and DExpert. As presented in Table 1, overall, the proposed DQAC model demonstrates superior performance by effectively reducing toxicity (with both classifiers scores), while simultaneously achieving acceptable scores in ranking and generation evaluation metrics (MRR, SBMRR, RR-BLEU and BLEU). The reduction in these metrics is expected as there is always a trade-off between performance and safe generation [11]. Increase in parameter α leads to decrease in toxicity scores as expected. A decrease in MRR and SBMRR scores is also observed. The drop in SBMRR scores is relatively lowere than the drop in MRR scores which indicates the model tries to maintain some semantic relevance while detoxifying. Additionally, we have performed two ablations: *T-Adapter* and *NT-Adapter*, which use only one adapter at a time - either toxic or non-toxic.

Performance for NPTQ testset: Table 2 compares performance of various models for the NPTQ dataset. In this subsubsection, we will focus on discussing the lower scores observed for MRR, SBMRR, and other ranking and generation metrics. It has been frequently observed that when a model is fine-tuned for safe and detoxification tasks, there is a decrease in overall model performance or potential semantic disturbances, which introduces a subjective trade-off between the acceptable level of toxicity and performance [11]. In the QAC detoxification task, toxicity is mitigated by avoiding toxic tokens in the completion by producing non-toxic tokens, resulting in no match with the ground truth. This leads to MRR scores close to 0 (especially for the NPTQ dataset, where the ground truth completions are toxic in nature). Additionally, since completions contain only a few tokens, any alteration to a word can significantly affect the semantics of the completion, leading to a low SBMRR score. Similar effects can be observed for other ranking and generation metrics. However, the reasonable BLEU scores indicate that the generated completions remain somewhat lexically relevant to the given prefix and session, maintaining topical alignment.

We further analyze with a specific example. Given the prefix “piece of a” a toxic completion in the dataset is “piece of a*s”. In contrast, the DQAC model generates a non-toxic completion of “piece of analysis” which has a different semantic meaning. Moreover, in the context of QAC systems, “piece of analysis” is generally considered more acceptable. Since all the completions in the NPTQ dataset are toxic, low scores of MRR and SBMRR are expected. Similar observations can be made for other strong baseline models like Quark and DExperts, which also exhibit low MRR and SBMRR scores. In addition, we randomly selected 50 examples from the DQAC model for **human evaluation** to quantify *semantic difference* and *contextual alignment*. Our findings indicated that 47 (94%) examples displayed semantic differences from the reference, while 42 (84%) examples maintained contextual alignment (lexical overlap) with prefix and session. Considering all these points, we can conclude that the low MRR scores are acceptable when the ground truth query is toxic in nature. Overall, the DQAC model, which consistently generates the lowest toxic completions, has emerged as the best-performing model for the NPTQ test set.

Session: braces || teen anal brutal braces || young teen anal || sally mann 11 **Prefix:** teen **Reference Completion:** teen braces fucked sex	
Generation with Baselines:	**Generation with DQAC:**
GPT2: teen anal	1. teen braces
DAPT: teen anal	2. teen braces white
GeDi: teen ugly scorpion get cock	3. teen browse youtube
PPLM: teen nude	4. teen browse youtube app
DExpert: teeneachy get my fat ugly wife pregnant	5. teen browse facebook
Quark: teen n instagram porn	

Table 3: Sample generation from baseline and proposed DQAC model from NPTQ testset. Top generations from the baselines and top 5 generations from DQAC are shown.

Performance for NPNQ testset: Table 2 compares performance of various models for the NPNQ dataset. The NPNQ test set is specifically designed to evaluate the capability of CTG models in generating non-toxic completions for non-toxic prefixes. We observe that several baseline models, as well as the DQAC model, achieve low toxicity scores while simultaneously maintaining satisfactory scores in terms of MRR and other ranking and generation metrics, across both datasets. These results highlight the effectiveness of the DQAC model in generating non-toxic completions while preserving the quality and relevance of the generated completions. These results further reinforce the model's efficacy and reliability in the QAC domain.

Sample Generation: Table 3 illustrates sample generations from the baselines and the proposed DQAC model, specifically considering samples from the NPTQ test set. From the observations, we can infer two key points: (1) Generations from the baseline models exhibit a tendency towards toxicity, while the proposed DQAC model successfully avoids generating toxic content. (2) The generated outputs differ semantically from the reference completions, leading to lower MRR and SBMRR scores. The previous subsection provides a detailed discussion on this observation.

6 Conclusions

This paper proposed a novel DQAC (Detoxifying Query Auto-Completion) model, which aims to mitigate toxicity in query auto-completions. To the best of our knowledge, this is the first publicly available model to detoxify QAC. DQAC operates in the latent representation space, employing a controllable text generation framework to effectively steer away toxic content from query completions and present related non-toxic alternatives. Additionally, we developed the QDETOXIFY model, specifically designed to evaluate the degree of toxicity for a given query completion. We conducted comprehensive comparisons of the model performance across multiple baselines using two real-world large-scale datasets. The results consistently demonstrate that our proposed DQAC model outperforms all the baselines and has emerged as a state-of-the-art model for the task of detoxifying query completions. In future, we will try more recent models as the base LM and extend the proposed framework to more generic language detoxification tasks and other CTG applications.

References

1. Cai, F., De Rijke, M., et al.: A survey of query auto completion in information retrieval. Foundations and Trends® in Information Retrieval **10**(4), 273–363 (2016)
2. Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., Yosinski, J., Liu, R.: Plug and play language models: A simple approach to controlled text generation. In: ICLR (2020)

3. Gehman, S., Gururangan, S., Sap, M., Choi, Y., Smith, N.A.: RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In: EMNLP Findings. pp. 3356–3369 (2020)
4. Gupta, M., Joshi, M., Agrawal, P.: Deep learning methods for query auto completion. In: ECIR. pp. 341–348. Springer (2023)
5. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., Smith, N.A.: Don't stop pretraining: Adapt language models to domains and tasks. In: ACL. pp. 8342–8360. Association for Computational Linguistics (2020)
6. Hanu, L., Unitary team: Detoxify. Github. https://github.com/unitaryai/detoxify (2020)
7. Hartvigsen, T., Gabriel, S., Palangi, H., Sap, M., Ray, D., Kamar, E.: ToxiGen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. In: ACL. pp. 3309–3326 (May 2022)
8. Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., Gelly, S.: Parameter-efficient transfer learning for nlp. In: ICML. pp. 2790–2799. PMLR (2019)
9. Lees, A., Tran, V.Q., Tay, Y., Sorensen, J.S., Gupta, J., Metzler, D., Vasserman, L.: A new generation of perspective api: Efficient multilingual character-level transformers. KDD (2022)
10. Liu, A., Sap, M., Lu, X., Swayamdipta, S., Bhagavatula, C., Smith, N.A., Choi, Y.: DExperts: Decoding-time controlled text generation with experts and anti-experts. In: ACL-IJCNLP. pp. 6691–6706 (Aug 2021)
11. Logacheva, V., Dementieva, D., Ustyantsev, S., Moskovskiy, D., Dale, D., Krotova, I., Semenov, N., Panchenko, A.: Paradetox: Detoxification with parallel data. In: ACL. pp. 6804–6818 (2022)
12. Lu, X., Welleck, S., Hessel, J., Jiang, L., Qin, L., West, P., Ammanabrolu, P., Choi, Y.: Quark: Controllable text generation with reinforced unlearning. NeurIPS **35**, 27591–27609 (2022)
13. Maurya, K.K., Desarkar, M.S., Gupta, M., Agrawal, P.: Trie-nlg: Trie context augmentation to improve personalized query auto-completion for short and unseen prefixes. In: DMKD. vol. 1573-756X. ECML-PKDD 2023 (2023)
14. Maurya, K.K., Desarkar, M.S., Kano, Y., Deepshikha, K.: ZmBART: An unsupervised cross-lingual transfer framework for language generation. In: ACL-IJCNLP Findings. pp. 2804–2818 (Aug 2021)
15. Mitra, B., Craswell, N.: Query auto-completion for rare prefixes. In: CIKM. pp. 1755–1758 (2015)
16. Olteanu, A., Diaz, F., Kazai, G.: When are search completion suggestions problematic? Proceedings of the ACM on human-computer interaction **4**(CSCW2), 1–25 (2020)
17. Pozzobon, L.A., Ermis, B., Lewis, P., Hooker, S.: On the challenges of using black-box apis for toxicity evaluation in research. ArXiv **abs/2304.12397** (2023)
18. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. JMLR **21**(140), 1–67 (2020), http://jmlr.org/papers/v21/20-074.html
19. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: EMNLP (11 2019)
20. Stickland, A.C., Murray, I.: Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In: ICML. pp. 5986–5995. PMLR (2019)
21. Üstün, A., Bérard, A., Besacier, L., Gallé, M.: Multilingual unsupervised neural machine translation with denoising adapters. In: EMNLP (2021)
22. Wu, Q., Burges, C.J., Svore, K.M., Gao, J.: Adapting boosting for information retrieval measures. Information Retrieval **13**, 254–270 (2010)
23. Yadav, N., Sen, R., Hill, D.N., Mazumdar, A., Dhillon, I.S.: Session-aware query auto-completion using extreme multi-label ranking. In: KDD. pp. 3835–3844 (2021)