International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 07, Issue – 03, March 2022, PP – 20-26

Tactics, Techniques and Procedure to Avoid Harmful AI: Exploring Web Search Queries

Ayse Kok Arslan

Oxford Alumni of Nortehrn Califgornia, Fremont, CA, USA

Abstract: AI models are now capable of performing a very wide range of tasks, often out of the box. In order to prevent malicious mis-use of AI models, developers can think in terms of traditional security risk assessment frameworks, which outline key steps such as identifying threats and potential impacts, assessing likelihood, and determining risk as a combination of likelihood and impact. This study provides a synthesis of different AI-centric perspectives towards the capabilities for a world with AI and explores the use-cases for an API user to achieve respectful behavior by focusing on Web search queries.It explores trends towards pre-trained data representations in AI model, to be applied in increasingly flexible and task-agnostic ways. The paper concludes that in order to inform appropriate policy interventions, the mindset should be expanded from code generation to other modalities.

Review of Existing Work

Driven by recent developments in AI, Web search query autosuggestion (often also referred to as autocompletion) is a feature enabled within the search bar of -many Web search engines such as Google (Sullivan, 2018) to predict the most likely intended queries given the user's partially typed query, where the predictions are primarily based on frequent queries mined from the search engines past query logs (Cai and Rijke, 2016).

Search query suggestions are part of a broader class of applications that provide *text prediction* or *text re-writing* suggestions to help users write faster, better, or in a more inclusive manner (Arnold, *et al.*, 2018; Kannan, *et al.*, 2016; Cai and Rijke, 2016; R.E. Robertson, *et al.*, 2021). While the systems generating these suggestions often leverage richer and cleaner data, they were also found to surface problematic suggestions (R.E. Robertson, *et al.*, 2021) that might *e.g.*, misgendered users (Vincent, 2018), or surfaced offensive associations (Larson, 2017).

The recent advances in large generative neural language models, such as GPT-3 (Brown, *et al.*, 2020), and their use in a wide variety of text prediction tasks has also drawn particular attention of late. Because of the vast amount of data from a wide variety of sources used to train these models, they are also likely to reproduce biases, stereotypes and other social or cultural viewpoints that might be problematic.

Despite its advantages, there are also pitfalls that can accompany this search feature. Since the suggestions are often derived from search logs, they can, as a result, be directly influenced by the search activities of the search engine's users. The dependence on query logs also leaves the autosuggestion feature susceptible to manipulation through adversarial attacks.

Understanding which suggestions should be construed as problematic and how to efficiently detect them also requires examining possible dimensions including

1) *content* (Olteanu, *et al.*, 2020; Miller and Record, 2017; Yenala, *et al.*, 2017);
2) *targets* (Olteanu, *et al.*, 2020; Olteanu, *et al.*, 2018; UN Women, 2013);
3) *structure* (Santos, *et al.*, 2017); and,
4) *harms* (Miller and Record, 2017).

A query suggestion may constitute harmful speech if the query could be perceived as hateful, as it offends, shows a derogatory attitude, intimidates, or promotes violence towards individuals or groups; or if the query appears related to *e.g.*, defamatory content promoting negative, unproven associations, or statements about individuals, groups, organizations.

Other query suggestions might be problematic if they could nudge users towards conspiracy theories or if they seem manipulated in order to promote certain viewpoints or content, typically content about ideological viewpoints, businesses or Web sites.

When determining whether a suggestion is problematic, the potential for various **harmful effects** — along with their severity, frequency or impact (Boyarskaya, *et al.*, 2020) — should also be factored in (*e.g.*, discomfort versus physical harm).

The implementation of a system to suppress problematic queries typically involves two key stages:

- First, there is a *discovery* stage where example queries are reported by users, or mined from search logs and assessed as problematic or not. This process typically generates a collection of annotated examples of problematic queries. Using this collection of queries, the second stage often includes the design and training of ML models for *detecting* problematic queries in order to suppress them from appearing as suggestions.

A key step in mitigating problematic query suggestion — though surprisingly under-explored by prior investigations and discussions about tackling various types of problematic suggestions — is their *discovery*. Although known methods for discovering problematic queries are varied, they are often *ad hoc* in nature, usually involving a combination of human intuition — typically of system designers about what is important and should be detected — with some sort of automated detection methods. This combination of intuition and automation can however leave them prone to important blind spots.

To help discover novel query forms expressing the same semantics as known problematic queries, deep-learned semantic embeddings can also be used. For instance, query embeddings can be learned from Web search click data to create a vector embedding space in which queries with similar click patterns are placed close together in the embedding space (Huang, *et al.*, 2013; Shen, *et al.*, 2014).

To take advantage of both machine learning models and human effort, an active learning approach could also be taken. Standard learning theory assumes that training and inference data is drawn from the same distribution. In transfer learning, the domain adaptation problem deals with transferring knowledge from the source domain (used for learning) to the target domain (the ultimate inference distribution). In order to prevent malicious mis-use of AI models, developers can think in terms of traditional security risk assessment frameworks, which provide evaluation metrics for measuring toxicity in model outputs and also in-house classifiers for detecting content that violates content policy, such as hate speech, violence, harassment, and self-harm.

Active learning is a human-in-the-loop method where machine learned detection models can examine large volumes of unannotated queries and propose the queries most likely to improve these model if annotated (Settles, 2009).

Because of the open nature of language, the complexity of the problem, and the potential harm of surfacing problematic queries, search engines typically employ a mix of multiple manual and automated methods for detecting and suppressing problematic queries in order to improve recall and robustness (Santos, *et al.*, 2017). Some of the most common approaches are discussed below.

Block lists

It is common for systems to maintain manually curated block lists of highly offensive terms that will trigger the suppression of a suggestion that contain those terms.

Block lists for whole queries can also be employed, and can be implemented efficiently for use in a run-time system. Block lists also ensure previously flagged queries remain permanently suppressed even if other modeling techniques unexpectedly leak the query after a model update. However, block lists cannot be generalized easily and (as with common templates) are not a scalable solution if used alone.

Query templates and grammars

Because search queries tend to be short and the likelihood of query uniqueness increases with query length, the collection of common query candidates used by an autosuggest feature will be dominated by short queries.

For example, to suppress derogatory suggestions about named individuals a simple approach is to use an entity extractor to identify people's names within queries, curate a list of derogatory expressions, and then identify the most common query templates that use a combination of a person's name and a derogatory expression.

To cover the wide range of template variations, a more efficient approach is to encode them into a rule-based grammar using a regular expression based finite-state tool (*e.g.*, Foma [Hulden, 2009]).

Machine learning models

To achieve greater generalization and avoid the use of hand-written rules, ML approaches can also be used to detect problematic queries. Techniques that have been explored for this purpose include gradient boosting decision trees (Chuklin and Lavrentyeva, 2013), long short-term memory networks (Yenala, *et al.*, 2017), and the deep structured semantic model (P. Gupta and Santos, 2017). Relative to grammar-based models, ML models have been found useful in improving the detection rate of problematic queries but at the expense of increased false positive rates (P. Gupta and Santos, 2017). While ML models avoid the effort required to hand-

craft rules, they require human effort to annotate collections of queries (both problematic and non-problematic) in order to train models.

When it comes to predicting harmful behaviors on platforms, one can employ classical statistical methods. Generally, regression analysis, discriminant analysis, and cluster analysis are examples of classical statistics approaches used. In addition to this, AI methods such as Backpropagation, Support Vector Regression, Gradient Boosting Classifier, Bayesian Classifier, Artificial Neural Network, and Decision Tree can also be employed. The latter involves a mix of advanced statistical methods and AI heuristics.

Classification is the most applied approach while the Decision Tree classifier is the most common algorithm (Mohamad and Tasir, 2019). Decision Tree classifiers are frequently used as they are easy to understand and have high predictive accuracy. Furthermore, Neural Network, Bayesian Networks, Rule Induction, and Support Vector Machines are examples of classification techniques that are frequently used to perform prediction.

Other related algorithms based on an extensive study done by Mohamad and Tasir (2021) include, but are not limited to: Prism, Lasso, K-means kernel, CART, k-Star, Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), Information gain (IG), Quadratic SVM (QSVM), Cubic SVM (CSVM), Fine Gaussian SVM (FGSVM), Medium Gaussian SVM (MGSVM), Swarm Optimization Combined Neural Network, Back Propagation Neural Network (BP-NN), Deep learning (DL), and Self-Organising Map. The next table shows a classification of these algorithms.

Decision Tree classifiers
Naïve Bayes classifiers
Random Forest
SVM classifiers
K-Nearest Neighbours

Table 1. Classification of Algorithms

Other popular algorithms are Bayesian Networks and Random Forest. Bayesian Networks are graphical models with nodes and directed edges that are probabilistic in nature. Simple models that explain a certain form of Bayesian network with all attributes being class-conditionally independent are known as Naïve Bayes classifiers. Naïve Bayes classifiers comprise algorithms such as Gaussian, Multinomial, and Bernoulli. The key benefit of employing Naïve Bayes classifiers is that the outcome from the prediction model using Naïve Bayes can be easily translated into human language.

Moreover, it is also important to evaluate potential biases in the data and model by conducting a multi-pronged approach for bias assessment.For this purpose, one can conduct various explainability analyses aimed at discovering what parts of the input contribute most to the algorithm's predictions. To give a specific example, both saliency analyses (which examine which pixels most influenced the predictions) and ablation experiments (which examine the impact of removing various image regions) indicate where an algorithm is most influenced. Explainability analysis will show whether all predictions focus on different parts of the image, and whether occluding specific regions of the image such as the center of the image has a much greater effect than occluding another region such as the periphery. The "baseline" can be considered as a logistic regression model that takes self-reported metrics such as age, race and years as input.

N-strike rule

One scenario that can plague a query suppression mechanism occurs when a system successfully removes problematic queries from a suggestion list only to have them replaced by other problematic suggestions missed by the detection model. One way to combat this is to employ what is dubbed as the *N*-strike rule, *i.e.*, the detection of *N* or more problematic queries at the top of a pre-filtered suggestion list for a query prefix will trigger the suppression of all suggestions for that prefix.

Even with a clear definition of what constitutes *problematic query suggestions*, setting the boundary between problematic and non-problematic cases can still be difficult.

A typical way to determining whether query suggestions are problematic is through some form of crowd labelling or by using block lists (for both data collection and data annotation). Both approaches however have limitations when used to operationalize ambiguous, latent concepts like stereotyping (Blodgett, *et al.*, 2021).

Furthermore, how various types of problematic suggestions are expressed can vary across contexts and over time.

A key driver for suppressing suggestions is preventing defamatory statements from appearing as suggestions.

Mislabeling is particularly difficult to detect in autosuggestions because the terminology used is often not problematic in its own right, and would go undetected by typical filtering methods. Some challenges to detecting problematic query suggestions involve adversarial queries, data voids, and others; which we discuss next.

Adversarial queries

Adversaries may use a variety of tricks to mask the intent of a query from automated detection while still providing a clear intent to a typical user. A common circumvention strategy is to rewrite an offensive or problematic query to include misspellings, abbreviations, acronyms, or other types of text manipulations.For instance, to avoid being detected by automated hate-speech detection tools, some users have developed a code in which references to targeted communities are substituted by "benign" terms in order to seem out of context (Magu, *et al.*, 2017).

Data voids

The open ended nature of search allows users to look up anything and everything. Yet, some topics and their corresponding queries will be more popular than othersleads to topics and associated queries for which there is little to no Web content. Similarly, there will also be queries (and query prefixes) that are less frequent, making suggestions much more prone to manipulation and to errors — *e.g.*, such as misspelling of popular queries (Joslin, *et al.*, 2019).

In fact, each query tends to be associated with its own unique demographic fingerprint (Shokouhi, 2013), with rare queries being frequently run by niche groups of users, and some of these groups can be adversarial in nature. Leveraging this at run-time however can be computationally intensive, and could raise both privacy and bias concerns (if *e.g.*, queries from certain groups of users are more routinely suppressed).

Stereotyping

Common ML algorithms learn features and frequent associations about words and phrases, but they are often not able to capture deeper insights about the social or cultural aspects of a query. It may thus be difficult for ML algorithms to make nuanced decisions about which queries might reinforce harmful stereotypes.

Research into discovering common gender stereotypes present in, for example, learned word embedding vectors examines this issue (Bolukbasi, *et al.*, 2016), but these solutions are often sensitive to various implementation parameters, and solving this problem for the wide range of social and cultural stereotypes possible in autosuggestions remains an open, difficult problem.

False suppression

It is sometimes difficult to distinguish a derogatory query from a legitimate non-offensive query based only on the query's words.

Additional mechanisms that take advantage of knowledge gleaned from prior search results or other query understanding models can help mitigate such erroneous suppressions. For example, the non-offensive nature of the query examples mentioned above could perhaps be identified through the use of Web search entity linking and detection tools that match these queries against entities present in a knowledge base (*e.g.*, titles of works of art, business names).

More broadly, dealing with ambiguity has been found challenging in other similar settings. For instance, sarcasm can be confused with abusive language (Nobata, *et al.*, 2016), while ambiguity may originate from both the use of language and how various types of problematic suggestions might have been defined.

Implementation Framework

Regardless of the methods and techniques used, the deployment approach should emphasize continuous iteration, and make use of the following strategies aimed at maximizing the benefits of deployment while reducing associated risks:

- Pre-deployment risk analysis, leveraging a growing set of safety evaluations and red teaming tools

- Starting with a small user base
- Studying the results of pilots of novel use cases (e.g., exploring the conditions under which we could safely enable longform content generation, working with a small number of customers)
- Implementing processes that help keep a pulse on usage (e.g., review of use cases, token quotas, and rate limits)
- Conducting detailed retrospective reviews (e.g., of safety incidents and major deployments)

A process based on the use-cases for an API user to achieve respectful behavior would entail in general following steps:

Step One: Sensitive Topic Categories and Outlining Desirable Behavior

Engineers should select categories that they prioritize as having direct impact on human well-being and describe desired behavior in the form of following categories. It should be noted that the following list is not exhaustive and prioritization depends on context.

- *Abuse, Violence, and Threat (including self-harm)*: Oppose violence or threats; encouraged seeking help from relevant authorities.
- *Health, Physical and Mental*: Do not diagnose conditions or prescribe treatment; oppose non-conventional medicines as scientific alternatives to medical treatment.
- *Human Characteristics and Behavior*: Oppose unhealthy beauty or likeability standards; support goodness and likeability being subjective.
- *Injustice and Inequality (including discrimination against social groups)*: Oppose human injustices and inequalities, or work that exacerbates either. This includes harmful stereotypes and prejudices, especially against social groups according to international law.

Step Two: Crafting the Dataset and Fine-Tuning

Developers can craft a values-targeted dataset of various samples; in a question-answer format and then fine-tuned their models on this dataset using standard fine-tuning tools.

In order to inform appropriate policy interventions, the dataset should also take other modalities into account. For example, developers initially focused on long form text generation as a threat vector, given prior cases of influence operations that involved people manually writing long form misleading content. Given that emphasis, they set maximum output lengths for generated text. Yet, output restrictions had little effect on policy violations—so short-form content amplifying or increasing engagement on misleading content could offer a greater risk.

Examples of limitations in existing datasets include the following:

- an overly narrow focus (e.g., just measuring occupational bias),
- an overly broad focus (e.g., measuring all under the umbrella of "toxicity"),
- a tendency to abstract away the specifics of use and context,
- a failure to measure the *generative* dimension of real world model use (e.g., using multiple choice style),
- prompts that differ stylistically from those typically used in real world model use cases,
- not capturing dimensions of safety that are important in practice (e.g., an output following or ignoring a safety-motivated constraint in the instruction), or
- not capturing types of outputs to be correlated with misuse (e.g., harmful content).

Step Three: Evaluating Models

Developers then can use quantitative and qualitative metrics: human evaluations to rate adherence to predetermined values; toxicity scoring which could also not capture all nuance in toxicity and host their own biases.

There are some limitations to the methods in use such as classifier-based data filtration. For instance, operationally defining the content areas to detect via filtration is challenging and filtration itself can introduce harmful biases. Additionally, the labeling of toxic data is a critical component and ensuring the mental health of these labelers is an industry-wide challenge.

While aligned models have practical advantages such as reducing the need for "prompt engineering" (providing examples of the desired behavior to steer the model in the right direction), saving space in the model's context window which can be used for other purposes and other models in a way that reduces risks of harm has posed various technical and policy challenges.

Many of the issues covered throughout the paper are complex and difficult to mitigate. Some may even argue that these issues are so intractable that the safest approach is to disable the autosuggest feature entirely (despite its known benefits to users), or to at least allow users to opt-in to the feature with a warning about its potential problems. Alternatively, the feature could be used only to surface prior frequent queries made by the current user, which might preserve some benefit while eliminating problematic suggestions learned from other users. Others may also argue that the decision to moderate and the moderation processes are themselves problematic as they could reflect the biases of those tasked with moderating these systems (an observation sometimes used to argue for less rather than better moderation). Nevertheless, the goal of this paper was to review on-going existing issues posed by the search autosuggestion feature, with the hope that highlighting them will inspire new research and development efforts into the challenging aspects of the problems, both technical and social.

Conclusion

There is no silver bullet for responsible deployment, so everyone in the process should try to learn about and address models' limitations, and potential avenues for misuse, at every stage of development and deployment in order to learn as much as possible about safety and policy issues at small scale and to incorporate those insights prior to launching larger-scale deployments.

A human-centric AI framework can help in augmenting the standard scientist's toolkit with powerful pattern recognition and interpretation methods from ML and demonstrate its value and generality. Mature ML methodologies can be adapted and integrated into existing scientific workflows to achieve novel results. Guiding scientific intuition with a scientist's intuition plays an enormously important role in discovery given the powerful combination of both rigorous formalism and good intuition to tackle complex scientific problems.

Last, but not least, when it comes to designing for AI, the ultimate aim should be to promote human creativity, responsibility, sustainability, and social connectedness as well as to increase self-efficacy, bring joy, spread compassion, and respect human dignity.

References

[1]. Akhtar, 2016. "Google defends its search engine against charges it favors Clinton," *USA Today* (10 June) https://www.usatoday.com/story/tech/news/2016/06/10/google-says-search-isntbiased-toward-hillary-clinton/85725014/, accessed 14 July 2020.

[2]. W. Arentz and B. Olstad, 2004. "Classifying offensive sites based on image content," *Computer Vision and Image Understanding*, volume 94, numbers 1–3, pp 295–310.doi: https://doi.org/10.1016/j.cviu.2003.10.007, accessed 14 July 2020.

[3]. K. Arnold, K. Chauncey, and K. Gajos, 2018. "Sentiment bias in predictive text recommendations results in biased writing," *GI '18: Proceedings of the 44th Graphics Interface Conference*, pp. 42–49. doi: https://doi.org/10.20380/GI2018.07, accessed 14 July 2020.

[4]. BBC, 2019. "Microsoft Word AI 'to improve writing'" (7 May), at https://www.bbc.com/news/technology-48185607, accessed 14 July 2020.

[5]. F. Cai and M. de Rijke, 2016. "A survey of query auto completion in information retrieval," *Foundations and Trends in Information Retrieval*, volume 10, number 4, pp. 273–263, and at https://www.nowpublishers.com/article/Details/INR-055, accessed 14 July 2020. doi: http://dx.doi.org/10.1561/1500000055, accessed 30 January 2022.

[6]. N. Diakopoulos, 2014. "Algorithmic accountability reporting: On the investigation of black boxes" (3 December), at https://www.cjr.org/tow_center_reports/algorithmic_accountability_on_the_investigation_of_black_boxes.php, accessed 14 July 2020.

[7]. M. Golebiewski and d. boyd, 2018. "Data voids: Where missing data can easily be exploited," *Data & Society*, at https://datasociety.net/wp-content/uploads/2018/05/Data_Society_Data_Voids_Final_3.pdf, accessed 14 July 2020.

[8]. A. Gulli, 2013. "A deeper look at Autosuggest," *Microsoft Bing Blogs* (25 March), at https://blogs.bing.com/search/2013/03/25/a-deeper-look-at-autosuggest/, accessed 14 July 2020.

[9]. P. Gupta and J. Santos, 2017. "Learning to classify inappropriate query-completions," In: In: J. Jose, C. Hauff, I. SengorAltngovde, D. Song, D. Albakour, S. Watt, and J. Tait (editors). *Advances in information retrieval. Lecture Notes in Computer Science*, volume 10193. Cham, Switzerland: Springer, pp 548–554. doi: https://doi.org/10.1007/978-3-319-56608-5_47, accessed 14 July 2020.

[10]. S. Karapapa and M. Borghi, 2015. "Search engine liability for autocomplete suggestions: Personality, privacy and the power of the algorithm," *International Journal of Law and Information Technology*,

volume 23, number 3, pp. 261–289. doi: https://doi.org/10.1093/ijlit/eav009, accessed 30 January 2022.

[11]. P. Lee, S. Hui, and A. Fong, 2002. "Neural networks for Web content filtering," *IEEE Intelligent Systems*, volume 17, number 5, pp. 48–57. doi: https://doi.org/10.1109/MIS.2002.1039832, accessed 14 July 2020.

[12]. R. Magu, R., K. Joshi, and J. Luo, 2017. "Detecting the hate code on social media." *arXiv*:1703.05443v1 (16 March), at https://arxiv.org/abs/1703.05443, accessed 14 July 2020.

[13]. K. McGuffie and A. Newhouse, 2020. "The radicalization risks of GPT-3 and advanced neural language models," *arXiv*:2009.06807v1 (15 September), at https://arxiv.org/abs/2009.06807, accessed 9 April 2021.

[14]. D. Metaxa-Kakavouli and N. Torres-Echeverry, 2017. "Googles role in spreading fake news and misinformation," *Stanford Law School, Law and Policy Lab, Fake News & Misinformation Policy Practicum* (31 October), at https://www-cdn.law.stanford.edu/wp-content/uploads/2017/11/SSRN-id3062984.pdf, accessed 30 January 2022.

[15]. B. Miller and I. Record, 2017. "Responsible epistemic technologies: A social-epistemological analysis of autocompleted Web search," *New Media & Society*, volume 19, number 12, pp. 1,945–1,963. doi: https://doi.org/10.1177/1461444816644805, accessed 14 July 2020.

[16]. A. Olteanu, C. Castillo, F. Diaz, and E. Kcman, 2019. "Social data: Biases, methodological pitfalls, and ethical boundaries," *Frontiers in Big Data* (11 July).doi: https://doi.org/10.3389/fdata.2019.00013, accessed 14 July 2020.

[17]. A. Olteanu, C. Castillo, J. Boy, and K. Varshey, 2018. "The effect of extremist violence on hateful speech online," *Proceedings of the Twelfth International AAAI Conference on Web and Social Media*, at https://www.aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17908/17013, accessed 14 July 2020.

[18]. A. Olteanu, K. Talamadupula, and K. Varshney, 2017. "The limits of abstract evaluation metrics: The case of hate speech detection," *WebSci '17: Proceedings of the 2017 ACM on Web Science Conference*, pp. 405–406.doi: https://doi.org/10.1145/3091478.3098871, accessed 30 January 2022.

[19]. Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, 2014. "Learning semantic representations using convolutional neural networks for Web search," *WWW '14 Companion: Proceedings of the 23rd International Conference on World Wide Web*, pp. 373–374.doi: https://doi.org/10.1145/2567948.2577348, accessed 14 July 2020.

[20]. H. Yenala, M. Chinnakotla, and J. Goyal, 2017. "Convolutional bi-directional LSTM for detecting inappropriate query suggestions in Web search," In: J. Kim, K. Shim, L. Cao, J.G. Lee, X. Lin, and Y.S. Moon (editors). *Advances in knowledge discovery and data mining. Lecture Notes in Computer Science*, volume 10234. Cham, Switzerland: Springer, pp. 3–16.doi: https://doi.org/10.1007/978-3-319-57454-7_1, accessed 14 July 2020.

