Defensive Low Authority Host Predictor

Ashish Gupta, Microsoft R&D Center, ashishgupta@microsoft.com
Sunakshi Gupta, Microsoft R&D Center, sungupta@microsoft.com
Somi Reddy Satti, Microsoft R&D Center, satti.somireddy@microsoft.com

Abstract

Responsible AI is becoming critical since AI is used in everyday lives. Search, recommender, and ranking techniques widely used in multiple industries are using ML models heavily. Not only do we need to improve the accuracy of the models but also need to guarantee fairness, resiliency to noise, explainability, and authoritative results. These objectives are not only relevant for ML model training but also, we need to ensure that we are showing fair as well as authoritative results. In this paper, we propose a host score prediction technique via which we try to demote the unsatisfactory hosts based on the integrity, quality, and authority (where is the information from, and is the information credible) of the hosts. Based on multiple features extracted for the host from context and other stats we publish scores for the host which indicate if they are good to show up on the landing page. We demote down hosts having low scores (i.e. unsatisfactory hosts) below a threshold and thus, reduce leakage of Bing via this technique making sure that there is no impact on the relevance of results. Finally, we show state-of-the-art results on our dataset built around billions of hosts. We show that this technique around responsible AI is highly robust and easy to deploy. We believe to have scratched the niche area of responsible AI and suggest further research challenges around this work.

Keywords: Responsible AI, Ranking, AI ethics

1. Introduction

In recent times, we have seen ample ML systems deployed to production, failing without adequately analyzing the risk of ML models. Companies like Google, Microsoft, Meta, and IBM have time and again said out loud of AI systems need not only to be accurate, but also to be developed, evaluated, and monitored for trust. The major objectives around which Responsible AI revolves are explainability, interpretability, ethics, fairness, robustness, and security.

We need responsible AI since the data is highly biased due to human assumptions, system bias, algorithm bias, measurement bias, etc. So, let's say that the training data contains a bias towards a specific population and is highly skewed towards those, then there is a limit on the trained model to avoid these irrespective of the training model with the best algorithms. Hence, we need to address these root causes by removing bias initially and constructing data in an unbiased manner.

Most datasets currently used to train *defensive*(area on racial hate, climate change, elections, coronavirus, vaccines, opioids, etc. especially spreading misinformation) query classifiers were collected through crowdsourced annotations [1, 2], despite the risk of annotator bias. More specifically, annotators are more likely to label comments as abusive if they are written in African American English (AAE). We have multiple techniques to deal with annotator bias as defined in[3, 4]. But instead of using these complex techniques, we follow a rigorous process for the annotation task, wherein a set of hosts were annotated by a minimum of 5 different human judges. Judges were given multiple guidelines for annotation, and hidden quality control measures were employed to remove judgments from incompetent judges. We performed post-filtering of judgments based on the significant disagreement of the judge with other judges, which resulted in an iterative process following which judgments from over a few judges were removed from the pool of judges. The final label was calculated on the majority judgment on the URL, and therefore, bias if any, was amortized. While taking the judgments, there were a lot of checks done in terms of training the judges, putting up SPAMs to constantly monitor the quality of judgment and any spammy annotation identified by such judges.

Since the queries around defensive areas are not that frequent, compared to the head queries or the generic queries that most populations trigger, we need to specifically analyze these queries' results and act carefully around such areas. A lot of malicious user-generated content also revolves around such areas, which has a bad impact on the people using the search engine. To mitigate these issues, we need to penalize these hosts and demote them. Moreover, we face challenges around the maintenance of the ML models since they degrade over time. We have a dashboard monitoring the performance of our model. When the performance drops beyond a certain threshold, we re-train the model on fresh data.

Finally, we propose the major contributions made in this paper:

- We propose an ML model to classify if a query is defensive or not.
- We propose a second model to identify if a host is of low authority or not and finally reduce leakage around the defensive area.
- We propose a few approaches focusing on data pre-processing to reduce the noise and predict precise host scores.

The above contributions show that our work scratches the surface of responsible AI for removing low authority hosts from the SERP (Search Engine Results Page). The leakage of Bing is 20% and Google is 13%. Our main goal is to reduce this gap between Bing and Google in terms of leakage.

2. Related Works

Trustworthiness Several authors agree upon the search for trustworthiness as the primary aim of an XAI (Explainable AI) model[5, 6]. However, declaring a model as explainable as per its capabilities of inducing trust might not be fully compliant with the requirement of model explainability. Trustworthiness might be considered as the confidence of whether a model will act as intended when facing a given problem. Although it should most certainly be a property of any explainable model, it does not imply that every trustworthy model can be considered explainable on its own, nor is trustworthiness a property easy to quantify. The trust might be far from being the only purpose of an explainable model since the relation between the two if agreed upon, is not reciprocal. Part of the reviewed papers mentions the concept of trust when stating their purpose for achieving explainability.

3. Methodology

Our goal is to train a model that can predict whether a target host is an authoritative (satisfactory and neutral) or low authority (unsatisfactory) host. To achieve this, we use multiple features from the index platform team like stats of hosts (impression Count, access Count, isNavQuery, Clicks) and a few other features like impactScore, spamScore, etc.

Model Architecture: Our demotion model consists of three parts:

1. An encoder H that encodes the query text into a high dimensional space.
2. A binary classifier C that predicts whether the query is a defensive query or not.
3. A secondary classifier (low authority) which says that for the particular query marked as defensive whether the host appearing in SERP(Search Engine Results Page) is defensive(area on racial hate, climate change, elections, coronavirus, vaccines, opioids, etc.) or not.

Training data: Each data point in our training set is a pair $(x_i, y_i); i \in 1...N$, where x_i is the input query text, y_i is the label for the query being defensive or not. The (x_i, y_i) tuples are used to train the classifier C. We adopt a two-phase training procedure from [7]. We use this procedure because [7] shows that their model is more effective than alternatives in a setting similar. The classifier supports multi-lingual text classification.

3.1. Binary Classifier

Training procedure: The key idea of the model in Fig.1 is to deeply mimic the teacher's self-attention module, which draws dependencies between words and is the vital component of Transformer. To introduce more fine-grained self-attention knowledge and avoid using teacher's self-attention distributions, we introduce multi-head self-attention relations of pairs of queries, keys, and values to train the student. Our method eliminates the restriction on the number of attention heads of student models. Moreover, using more relation heads in computing self-attention brings more fine-grained self-attention knowledge and improves the performance of the student model.

We use A1, A2, and A3 to denote queries, keys, and values of multiple relation heads. The training objective between teacher and student having multi-head self-attention relation is described via KL divergence:

$$L = \sum_{i=1}^{3} \sum_{j=1}^{3} \alpha_{ij} L_i \quad (1)$$

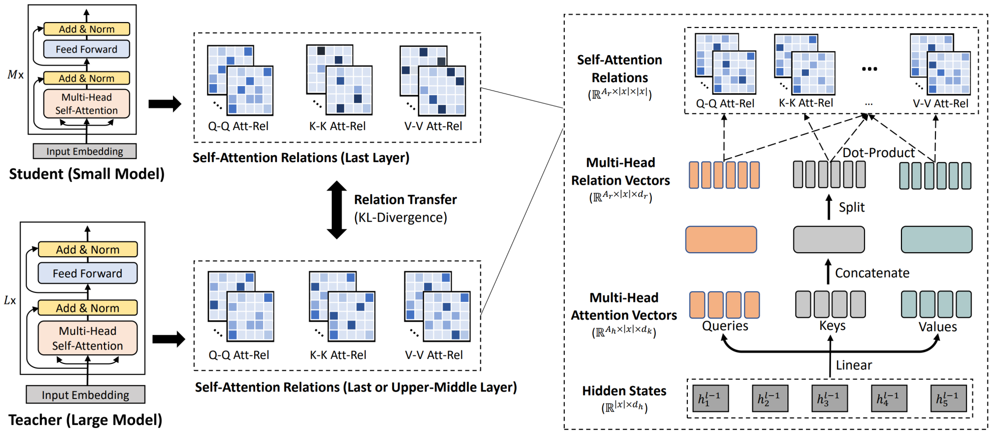
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self-Attention
Mx
Input Embedding
Student (Small Model)
Q-Q Att-Rel
K-K Att-Rel
V-V Att-Rel
Self-Attention Relations (Last Layer)
Relation Transfer
(KL-Divergence)
Add & Norm
Feed Forward
Add & Norm
Multi-Head Self-Attention
Lx
Input Embedding
Teacher (Large Model)
Self-Attention Relations (Last or Upper-Middle Layer)
Self-Attention Relations
$(\mathbb{R}^{A_r \times |x| \times |x|})$
Multi-Head Relation Vectors
$(\mathbb{R}^{A_r \times |x| \times d_r})$
Dot-Product
Split
Concatenate
Multi-Head Attention Vectors
$(\mathbb{R}^{A_h \times |x| \times d_k})$
Queries
Keys
Values
Linear
Hidden States
$(\mathbb{R}^{|x| \times d_h})$
h_1^{l-1}
h_2^{l-1}
h_3^{l-1}
h_4^{l-1}
h_5^{l-1}

Figure 1. MiniLM Architecture

THREAT	THRESHOLD	PRECISION	RECALL	F1
ANTISEMITIC HATE	0.9	0.72	0.733	0.726
CORONAVIRUS	0.7	0.987	0.929	0.979
BLM	0.85	0.898	0.93	0.914
OPIOIDS	0.9	0.943	0.929	0.936

Table 1. Classification accuracies for MultiLM model.

$$L_{ij} = \frac{1}{A_r|x|} \sum_{a=1}^{A_r} \sum_{t=1}^{|x|} D_{KL}(R^T_{ij,l,a,t} || R^S_{ij,m,a,t}) \qquad (2)$$

where $A^T_{i,l,a} \in R^{|x|Xd_r}$ and $A^S_{i,m,a} \in R^{|x|Xd'_r}$ are the queries, keys, and values of a relation head of l-th teacher layer and m-th student layer. d_r and d'_r are the relation head size of the teacher and student model.$R^T_{ij,l}$ is the self-attention relation of $A^T_{i,l}$ and $A^T_{j,l}$ of teacher model.

The student models are initialized randomly. For models distilled from RoBERTa[8], we use similar pre-training datasets as in[9].

3.2. Secondary Classifier(Low auth)

We use the above architecture for downstream tasks like classification, extractive QA, cross-lingual Natural Language Inference, and classifying which queries are defensive. Moreover, with the help of defensive queries, we can extract corresponding hosts. But all these hosts need not necessarily be defensive(bad) since there is leakage in query results. Hence, we have a second classifier to identify defensive hosts and demote them to reduce leakage. Formally *leakage*, is defined as the documents(in SERP results) adding misinformation to the Bing results.

For these hosts, we obtain features from the index team like impression Count, access Count, Clicks, URL Count, Trust score, etc. We use these to classify whether a particular host is defensive or not. Apart from these features, we add Authority Score, and Domain Count as a feature to further refine these scores. Since the defensive low authority hosts are very few in numbers thus, we sampled the data accordingly to make the ML model unbiased.

The model we use here is a mixture of Ridge regression and RandomForest since the data is highly skewed towards SAT(Satisfactory) hosts and thus is highly likely to overfit towards these hosts. Hence, we apply regularization to reduce the overfitting issue.

4. Experiments

4.1. Dataset

The data is collected from the past year (Oct2020-Jul2021) slapi(interface to access all the information in Search Merge Log on Cosmos) query logs. We use OneDCG extraction which uses human judges to label the queries. On these, we use our first classifier Sec3.1 to find out which queries are defensive.

4.2. Setup

We use the uncased version for BERT teacher models. We train student models using 256 as the batch size and 6e-4 as the peak learning rate for 400k steps. We use linear warm-up over the first 4000 steps and linear decay. We use Adam[10] with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The maximum sequence length is set to 512. The dropout rate and weight decay are 0.1 and 0.01. The number of attention heads is 12 for all student models. The number of relation heads is 48 and 64 for the base-size and large-size teacher models, respectively. The student models are initialized randomly.

$$Defensive\ Document\ Defect\ Rate = \frac{\sum \frac{\#\ of\ US\ \&\ VUS\ URLs}{Total\ \#\ of\ Results}}{Total\ \#\ of\ Queries}$$

$$Defensive\ Query\ Defect\ Rate = \sum \frac{(1\ if\ \#\ of\ US\ \&\ VUS\ URLs\ > 0\ in\ a\ query\ else\ 0)}{Total\ \#\ of\ Queries}$$

Method	Control Score	Treatment Score	Score
Document Defect Rate	7.15	6.32	**-0.83**$^{*\&}$
Query Defect Rate	29.85	26.41	**-3.44**$^{*\&}$

Table 1. Metrics. * and & signify statistically significant difference between the method and two best performing baselines using χ^2 test with $p \leq 0.05$

Figure 2. Defect Rate Metrics

For multilingual student models distilled from XLM-R, we perform training using the same datasets as in [11] for 1000k steps.

4.3. Query Classifier Results

We cut out the dev set from our labeled data(3) and show the results of our binary classifier on a few threats like coronavirus, antisemitic hate, BLM (Black Lives Matter), and opioids in the Table 1. Moreover, we can see that our binary classifier works well in classifying queries for multiple threats.

4.4. Low authority predictor

We use the above model's result to find out hosts corresponding to these defensive threat areas. Various features of these hosts are used like impression Count, access Count, isNavQuery, clicks, etc. to classify them into defensive and non-defensive hosts.

We run the Low Authority classifier(3.2) on the detected Defensive hosts to classify them into low and high authority buckets. The results of the classifier on Defensive QuerySet are shown in table 2. The focus of the predictor is on the Unsatisfactory hosts for which the recall is increased from **3%** to **47%** compared to current production.

Moreover, we found out that a lot of UGC(User Generated Content) sites were top contributors to spreading a lot of misinformation. Hence, we demoted top UGC domains such as *wordpress, weebly, and blogspot* and found no change in the precision of our model but the recall went up to **75%**.

4.5. Results on Trustworthy Metrics

We measure the trustworthiness of Bing using two significant metrics. We define DefensiveQueryDefectRate@10(in Fig.2) to be the percentage of queries leaking in the top 10 documents about a query, and DefensiveDocumentDefectRate@10 as the percentage of leaking documents. So, if a query has at least one document predicted as unsatisfactory, then it is said to be leaking. Hence, using this low auth(3.2) model, we are able to reduce the DefensiveQueryDefectRate@10 by approximately **3.44%** as evident in Fig2(results on test set) and the DefensiveDocumentDefectRate@10 by approx. **0.83%** which shows there is not much hit on relevance on the Query results when triggered on Bing. Hence, with the help of this technique, we can reduce leakage by approx. 3%. Thus, we can reduce the gap from 7%(20%-13% as mentioned earlier) to 3.56%(20-13-3.44) currently, which is a huge win in terms of leakage reduction.

Labels	Precision	Recall	F1
Unsatisfactory(Production)	0.58	*0.03*	0.06
Satisfactory(Production)	0.61	0.98	0.76
Unsatisfactory(Low auth)	0.68	**0.47**$^{*\&}$	0.56
Satisfactory(Low auth)	0.71	0.85	0.77

Table 2. Comparing production and secondary classifier prediction results. * and & signify statistically significant difference between the method and two best performing baselines using χ^2 test with $p \leq 0.05$

5. Conclusion

In this work, we show how to classify defensive as well as non-defensive hosts via multiple features used in the secondary classifier(3.2). We plan to extend our work to query context-aware host predictors wherein we'll be using the context of the host to determine if it's unsatisfactory or not thereby, demoting the unsatisfactory hosts. Also, we plan to extend our work around the graph technique to determine unsatisfactory hosts wherein we'll expand from unsatisfactory hosts and will see how many in-links a particular host has to those sets of unsatisfactory hosts to conclude it as an unsatisfactory host.

References

[1] T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated hate speech detection and the problem of offensive language,” in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 11, pp. 512–515, 2017.

[2] A. M. Founta, C. Djouvas, D. Chatzakou, I. Leontiadis, J. Blackburn, G. Stringhini, A. Vakali, M. Sirivianos, and N. Kourtellis, “Large scale crowdsourcing and characterization of twitter abusive behavior,” in *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[3] Y. Chen and J. Joo, “Understanding and mitigating annotation bias in facial expression recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14980–14991, 2021.

[4] M. Xia, A. Field, and Y. Tsvetkov, “Demoting racial bias in hate speech detection,” *arXiv preprint arXiv:2005.12246*, 2020.

[5] B. Kim, E. Glassman, B. Johnson, and J. Shah, “ibcm: Interactive bayesian case model empowering humans via intuitive interaction,” 2015.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, “” why should i trust you?” explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

[7] W. Wang, H. Bao, S. Huang, L. Dong, and F. Wei, “Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers,” *arXiv preprint arXiv:2012.15828*, 2020.

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

[9] L. Zhuang, L. Wayne, S. Ya, and Z. Jun, “A robustly optimized bert pre-training approach with post-training,” in *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pp. 1218–1227, 2021.

[10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[11] G. Lample and A. Conneau, “Cross-lingual language model pretraining,” *arXiv preprint arXiv:1901.07291*, 2019.