BLENDER: Enabling Local Search with a Hybrid Differential Privacy Model

Brendan Avent¹, Aleksandra Korolova¹, David Zeber², Torgeir Hovden², Benjamin Livshits³

¹University of Southern California ²Mozilla ³Imperial College London

Full paper available here.
Local Search

Goal
To make popular queries and their corresponding URLs available locally on users’ devices

Why its needed?
Caching popular search data avoids many round-trips to a server
• Reduces latency in web-browsing
• Useful for temporary network disruptions
• Enables new browser features
Local Search with Privacy

Why is privacy needed?

- Local search is generated from user data
- Want differential privacy guarantees
Local Search with Privacy

Why

Algorithm $\mathcal{A}$ is $(\epsilon, \delta)$-differentially private iff for all neighboring databases $D$ and $D'$ differing in the value of precisely one user’s data, the following inequality is satisfied for all possible sets of outputs $Y \subseteq \text{Range}(\mathcal{A})$:

$$\Pr[\mathcal{A}(D) \in Y] \leq e^\epsilon \Pr[\mathcal{A}(D') \in Y] + \delta$$
Local Search with Privacy

Why is privacy needed?

• Local search is generated from user data

• Want differential privacy guarantees

Why is differentially private local search hard?
## Differential Privacy Models

<table>
<thead>
<tr>
<th>trusted curator model</th>
<th>local model</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Central curator collects the data from all users, then performs privatization</td>
<td>• Each user privatizes their own data, then sends it to a central curator</td>
</tr>
<tr>
<td>• Most differentially private algorithms are in this model</td>
<td>• Requires less trust from users</td>
</tr>
<tr>
<td>Requires the users to trust the curator with their private data</td>
<td>Harsh utility trade-offs compared to trusted curator model algorithms [Chan et al 2012; Duchi et al 2013; Kairouz et al 2014, 2016]</td>
</tr>
</tbody>
</table>
Hybrid Model

a more realistic privacy model
Users Have Heterogeneous Privacy Preferences

Firefox Browser Privacy Notice
Our pre-release versions (Beta/Developer Edition, Nightly, and TestFlight) may have different privacy characteristics. Pre-release versions automatically send Telemetry data to Mozilla.

Chrome Release Channels

- Chromium
- Google Chrome Canary
- Google Chrome Dev
- Google Chrome Beta
- Google Chrome Stable

32-bit/64-bit 32-bit 32-bit 32-bit 32-bit

Microsoft reminds privacy-concerned Windows 10 beta testers that they're volunteers
If you don't like it, don't participate
Hybrid Model for Differential Privacy

Hybrid model

- Allows some users to contribute in the Trusted Curator Model; others in the Local Model

Trusted curator model

- Beta users we call “Opt-in” users

Local model

- Regular users we call “Clients”
Why a Hybrid Model?
Why a Hybrid Model?
Why a Hybrid Model?
Why a Hybrid Model?
BLENDER

local search in the hybrid model
BLENDER Architecture
BLENDER Architecture

**Opt-in Group**
- Curator
- Privacy barrier
- Query/URL:
  - Head list
  - Probability
  - Variance

**Client Group**
- Curator
- Privacy barriers (3)
- Query/URL:
  - Probability
  - Variance
BLENDER Architecture

Opt-in Group

Client Group

Curator
privacy barrier

query/url:
head list
probability
variance

privacy barrier
privacy barrier
privacy barrier

Curator

query/url:
probability
variance

Outputs

head list
probability

Blending Stage
Opt-in Group Algorithm

Two-phase approach: Discovery and Estimation

Partition users into two disjoint groups

Group A – Discovery phase

Group B – Estimation phase
Opt-in Group Data: Discovery of Head List

For each distinct <query, URL> record from Group A’s data:

• Compute empirical probability

• Add Laplace noise to form noisy empirical probability

• If noisy empirical probability exceeds threshold, add record to the head list

[Korolova et al, 2009]
Opt-in Group Data Usage: Estimation

For each distinct <query, URL> record from Group B’s data and using the privatized head list:

• Compute empirical probability

• Add Laplace noise to form noisy probability estimate

• Compute the sample variance of the probability estimate

[Dwork et al, 2006]
Client Data Reporting

2-stage k-randomized response [Warner 1965]

1. Report the query truthfully with probability $t$, otherwise, report a query at random.

2. Report the URL truthfully with probability $t_q$, otherwise, report a URL at random.
Server Aggregating Client Data

- Collects privatized reports from all users
- Aggregates the privatized reports into empirical probability estimates for each record
- Performs denoising procedure to generate unbiased probability estimates and variance estimates
BLENDER: Blending Stage

Opt-in Group

Client Group

(\epsilon, \delta)-differentially private

Outputs

head list
probability

query/url: head list
probability
variance

Curator

privacy barrier

query/url: probability
variance

Curator

privacy barrier

(\epsilon, \delta)-differentially private
Evaluation

Measuring the utility of BLENDER
# Experimental Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th># Users</th>
<th># Unique Queries</th>
<th># Unique URLs</th>
<th>$\delta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOL (2006)</td>
<td>0.5M</td>
<td>4.8M</td>
<td>1.6M</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>Yandex (2013)</td>
<td>4.9M</td>
<td>13.2M</td>
<td>12.7M</td>
<td>$10^{-7}$</td>
</tr>
</tbody>
</table>
Measuring Utility

Normalized Discounted Cumulative Gain (NDCG)

- Standard measure of ranking quality

\[ DCG = \sum_i \frac{2^{rel_i - 1}}{\log(i+1)} \]

- \[ NDCG = \frac{DCG}{\text{Ideal } DCG} \]

NDCG of NDCGs

1. Compute the NDCG for each query’s URL list, \( NDCG_{q_i} \)

2. Generalized DCG for the query list:

\[ \sum_i \frac{2^{rel_i - 1}}{\log(i+1)} \cdot NDCG_{q_i} \]

3. Normalize by analogous Ideal DCG
Comparison with Local Model  [Qin et al, CCS 2016]

How does BLENDER compare to having all users use the Local Model?

AOL dataset
Head list size: 10
Comparison with Local Model  [Qin et al, CCS 2016]

How does BLENDER compare to having all users use the Local Model?

AOL dataset
Head list size: 10

BLENDER
- 5% “opt-in” users
- 95% “client” users

Caveat: Slightly different versions of NDCG. See paper.
Effect of Opt-in User Percentage on NDCG

How does BLENDER’s utility depend on the size of the opt-in user group?

Yandex dataset
$\epsilon = 4$
Head list sizes: 50, 100, 500
How does BLENDER’s utility depend on the privacy budget $\epsilon$?

Yandex dataset
2.5% opt-in, 97.5% client
Head list sizes: 10, 50, 100, 500
Conclusions
Conclusions

- Proposed a hybrid model for differential privacy
- Constructed a blended approach within the hybrid model for local search
- Achieved significant improvement on real world datasets with the blended approach
Future Work

• Improve on the sub-components of BLENDER to utilize state-of-the-art privatization methods

• Derive theoretical guarantees for the utility of BLENDER

• Reduce BLENDER’s reliance on distributional assumptions

• Develop algorithms in the hybrid model for other applications
BLENDER: Enabling Local Search with a Hybrid Differential Privacy Model

Brendan Avent\textsuperscript{1}, Aleksandra Korolova\textsuperscript{1}, David Zeber\textsuperscript{2}, Torgeir Hovden\textsuperscript{2}, Benjamin Livshits\textsuperscript{3}

\textsuperscript{1}University of Southern California
\textsuperscript{2}Mozilla
\textsuperscript{3}Imperial College London

Full paper available [here](#).