Incorporating Item Frequency for Differentially Private Set Union

Ricardo Silva Carvalho, Ke Wang, Lovedeep Gondara

School of Computing Science, Simon Fraser University, Canada
rsilvaca@sfu.ca, wangk@cs.sfu.ca, lgondara@sfu.ca

Abstract

We study the problem of releasing the set union of users' items subject to differential privacy. Previous approaches consider only the set of items for each user as the input. We propose incorporating the item frequency, which is typically available in set union problems, to boost the utility of private mechanisms. However, using the global item frequency over all users would largely increase privacy loss. We propose to use the local item frequency of each user to approximate the global item frequency without incurring additional privacy loss. Local item frequency allows us to design greedy set union mechanisms that are differentially private, which is impossible for previous greedy proposals. Moreover, while all previous works have to use uniform sampling to limit the number of items each user would contribute to, our construction eliminates the sampling step completely and allows our mechanisms to consider all of the users' items. Finally, we propose to transfer the knowledge of the global item frequency from a public dataset into our mechanism, which further boosts utility even when the public and private datasets are from different domains. We evaluate the proposed methods on multiple real-life datasets.

1 Introduction

The *set union* is a basic operation that provides a global view of the data from all of the users involved in a study. It is one of the most fundamental building blocks in many data-related problems, such as vocabulary extraction, data integration and SQL queries (Kannan et al. 2016; Lenzerini 2002; Chen et al. 2019). However, when the data contains sensitive information of users, this operation may breach users' privacy. For example, releasing a vocabulary disclosing just a single word is enough to impact a user's privacy, as it may represent a password, disease diagnosis or political preference. Recent works (Carlini et al. 2019, 2020) have used extraction attacks to successfully reveal sensitive text. Therefore, algorithms that provide privacy guarantees are highly desirable. Differential Privacy (DP) (Dwork et al. 2006) is the most often used standard for privacy, implemented by Google (Fanti, Pihur, and Erlingsson 2016), Microsoft (Ding, Kulkarni, and Yekhanin 2017), the US Census Bureau (Abowd 2018; Kuo et al. 2018), etc. A randomized mechanism $\mathcal{M}$ is (ε, δ)-*differentially private*, or (ε, δ)-DP, if for all neighboring datasets D and D' that differ in the addition or removal of one user's data, and all sets S of possible outputs, we have:

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

Since neighboring datasets differ by the data of one user, the inequality above ensures that the outputs of a mechanism $\mathcal{M}$ satisfying DP will have only a small impact if we remove or add any single user from the dataset used to generate the output. Thus, such outputs will not be tightly related to any individual user, effectively protecting users' sensitive data.

1.1 Differentially Private Set Union

The problem of *Differentially Private Set Union* (DPSU) was formalized by (Gopi et al. 2020).

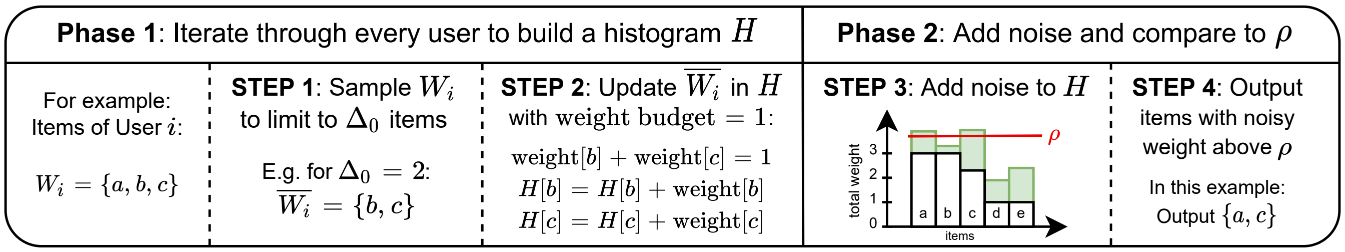
Problem 1 (Differentially Private Set Union (DPSU))**.** *Let U be some universe of items, possibly of unbounded size. Given a database D of users, each user i has a set of items $W_i \subseteq U$, we want an (ε, δ)-DP mechanism $\mathcal{M}$ that outputs a subset $S \subseteq \cup_i W_i$ such that the size of S is as large as possible.*

The goal is to output the largest possible subset of the union while meeting DP. Previous work related to DPSU (Korolova et al. 2009; Wilson et al. 2020) designed mechanisms with two phases, as shown in Figure 1. Phase 1 builds a histogram on $\cup_i W_i$ by iterating over the users and getting their contributions in terms of items. To provide DP, the contribution of every single user is bounded. That means the maximum number of items a user may contribute is fixed, denoted by Δ_0, and the updates to items' weights in the histogram given by each user sum to a total weight budget of 1. In Phase 2, only items with a noisy weight above the threshold ρ will be outputted. Basically, the threshold is used to limit the probability of outputting the items only contributed by the user present in just one of the neighboring data D or D'.

1.2 Contributions

This work addresses several limitations in the previous DPSU mechanisms, detailed below.

Introducing item frequency: All previous works treat W_i as a set of distinct items, ignoring items' frequencies for the user i when available. For example, for vocabulary extraction the data for each user can be a text document where terms are allowed to occur multiple times, but previous works represent each user by a set of distinct terms in the user document. Our



Phase 1: Iterate through every user to build a histogram H
Phase 2: Add noise and compare to ρ
For example: Items of User i:
$W_i = \{a, b, c\}$
STEP 1: Sample W_i to limit to Δ_0 items
E.g. for $\Delta_0 = 2$: $\overline{W_i} = \{b, c\}$
STEP 2: Update $\overline{W_i}$ in H with weight budget $= 1$:
$\text{weight}[b] + \text{weight}[c] = 1$
$H[b] = H[b] + \text{weight}[b]$
$H[c] = H[c] + \text{weight}[c]$
STEP 3: Add noise to H
ρ
total weight
0
1
2
3
a
b
c
d
e
items
STEP 4: Output items with noisy weight above ρ
In this example: Output $\{a, c\}$

Figure 1: Overview of all previous DPSU mechanisms. Phase 1 builds a histogram user by user. STEP 1 samples items from W_i into $\overline{W_i}$ to limit user contribution to at most Δ_0 items. To further limit contributions, in STEP 2 a total weight budget of 1 is distributed among the items in $\overline{W_i}$ and the histogram is updated for each item in $\overline{W_i}$ with the corresponding weights defined by the DPSU mechanism in use. These first two steps are done for all existing users and a final histogram is built. Finally, Phase 2 adds noise to the total weights of the items in the final histogram, and outputs items with noisy weight above the threshold ρ.

first contribution is introducing an item frequency array C_i associated with the items in W_i and using this information to return more items in the set union output while meeting DP.

Eliminating sampling: All previous works implement STEP 1 in Figure 1 using uniform sampling, which tends to waste the weight budget on updating items that occur among many users, as such items do not require many updates to reach the threshold ρ. We introduce a mechanism that eliminates the need for sampling in STEP 1, so each user i moves on to STEP 2 with *all* items W_i. This allows us to tailor the updates by choosing the most promising items available for each user. Another positive consequence from this is that, unlike previous work, we do not need to set the Δ_0 parameter, which would cost additional privacy budget to choose privately. To the best of our knowledge, this is the first work that eliminates the sampling step.

DP greedy update: (Gopi et al. 2020) shows that some well thought greedy update for STEP-2 does **not** satisfy DP. We propose an update algorithm for STEP 2 that follows a *greedy* approach guided by the item frequency C_i for each user i, but our *greedy* update provably satisfies DP and enables a smaller threshold ρ compared to (Gopi et al. 2020). The reduced threshold increases the probability of outputting more items.

Knowledge transfer: Looking at the data of each individual user i to get the frequency array C_i may not reflect the global frequency distribution for all users. On the other hand, getting C_i by aggregating all users' sensitive data would not satisfy DP. To address this issue, we propose a variant of our mechanism that uses the frequency of items in the histogram built from a *public* dataset, leveraging the fact that even datasets from different domains could have a significant overlap of items. For example, such overlap happens often in English documents across different domains (Vodrahalli 2015).

Empirical evaluation: Extensive empirical evaluation shows our method with a utility increase of up to 10% compared to previous works. For knowledge transfer we see utility improvements of up to 25%, showing benefits of the approach even when the public dataset is from a completely different domain than the sensitive data.

In the rest of the paper, we first review related work in Section 2. Our *Greedy* mechanism without sampling is detailed in Section 3, *Knowledge transfer* is described in Section 4, and the empirical evaluation is reported in Section 5.

2 Related Work

In general, a DPSU mechanism follows the flow in Algorithm 1 and proves DP guarantees for carefully chosen Δ_0, η (sampling technique), $\mathcal{F}$ (weight update algorithm), ξ (noise distribution) and ρ (threshold for outputting items).

Algorithm 1: Meta framework for DP Set Union

```
Input: $D$: Database of $n$ users s.t. each user $i$ has $W_i \subseteq U$.
$\Delta_0$: Maximum allowed contribution of items per user.
$\eta$: Sampling technique.
$\mathcal{F}$: Algorithm to determine weights of items in a user's set.
$\xi$: Noise distribution ($\texttt{Lap}(0, \lambda)$ or $\mathcal{N}(0, \sigma^2)$).
$\rho$: threshold for outputting items.
Output: Set of items.
1:  Let $H =$ be an empty histogram
2:  Consider the users follow a global order $\{1, ..., n\}$.
    – Phase 1: Iterate over the users, to build $H$
3:  for each $i = 1$ to $n$ do
4:      Let $W_i$ be the set of items of user $i$;
5:      Let $C_i$ be a dictionary with items only from $W_i$,
        where $C_i[w]$ is the count of item $w$ for $w \in W_i$;
        STEP 1: Random sampling to limit number of items
6:      $\overline{W_i} \leftarrow$ Sample at most $\Delta_0$ items from $W_i$ using $\eta$
        STEP 2: Update histogram for selected items
7:      Update $H[u]$ for each $u \in \overline{W_i}$ using $\mathcal{F}$
8:  Let $S = \emptyset$ be an empty set.
    – Phase 2: Iterate through the items, to output $S$
9:  for each $u \in \cup_i \overline{W_i}$ do
        STEP 3: Add noise to the items in the histogram
10:     $\hat{H}[u] \leftarrow H[u] + \xi$
        STEP 4: Output items with noisy weight above $\rho$
11:     if $\hat{H}[u] > \rho$ then
12:         $S = S \cup u$
13: return $S$
```

All previous works employed uniform sampling[1] η to select items $\overline{W_i}$ from W_i in STEP 1 of Algorithm 1. In terms of the weight update strategy $\mathcal{F}$, there exist three approaches, denoted as `COUNT`, `WEIGHTED` and `POLICY` mechanisms.

[1]We also highlight the work of (Desfontaines, Voss, and Gipson 2020) that gives optimal results for the special case of fixed $\Delta_0 = 1$.

Table 1: Differences between our mechanisms and previous work. We are the first to remove the need for sampling Δ_0 items, add greedy updates and Knowledge Transfer (KT).

Mechanism	Sampling	User View	Updates	KT
COUNT	Yes	Independent	$1/\Delta_0$	No
WEIGHTED	Yes	Independent	$1/\|W_i\|$	No
POLICY	Yes	Dependent	Gap to Γ	No
This work	**No**	**Dependent**	**Greedy**	**Yes**

In COUNT mechanisms (Korolova et al. 2009; Wilson et al. 2020), the histogram is updated using a weight[2] of $\frac{1}{\Delta_0}$ for each item. The drawback is that, when a user i has less than Δ_0 items, i.e. $|W_i| < \Delta_0$, the budget of 1 is not fully utilized for the user. WEIGHTED mechanisms (Gopi et al. 2020) address this issue by weighting each item in W_i uniformly by the factor[3] of $\frac{1}{|W_i|}$. However, in both COUNT and WEIGHTED mechanisms, each user's update is independent of the other users' updates. This may waste the budget if some items are already above the threshold ρ.

With this in mind, POLICY mechanisms (Gopi et al. 2020) consider that the user view of an item is *dependent* on the previous updates. In other words, they define the update weight of an item based on the previous updates to the item already made by other users. These algorithms use a cutoff, denoted as Γ, above the threshold ρ, and if an item has reached a total weight of Γ in the histogram, the item will not be updated anymore, effectively leaving more budget to the other items below Γ. Additionally, POLICY mechanisms increase the weights of the items in $\overline{W_i}$ proportionally to each item's gap to reach Γ in the histogram. Thus, POLICY mechanisms may take longer to reach Γ due to distributing the weight among all the items in $\overline{W_i}$.

Our mechanisms address the limitations of previous works with the differences summarized in Table 1. We eliminate the need for sampling users' items, instead of using uniform sampling, to consider all of the items for each user i in Algorithm 1, i.e. $\overline{W_i} = W_i$. As for the update strategy $\mathcal{F}$, we select one item at a time guided by the local item frequency C_i for the user i and greedily spend the weight budget as needed on the item, instead of distributing the weight among all items in $\overline{W_i}$. This effectively lowers the threshold ρ for outputting items, compared to previous work.

Finally, our Knowledge Transfer (KT) approach boosts utility by incorporating public data distribution into DPSU mechanisms. Even though there is a growing line of research that leverages public data to support private computations (Liu et al. 2021; Bassily et al. 2020; Bassily, Moran, and Nandi 2020; Ji and Elkan 2013), to the best of our knowledge this is the first work in this area for DPSU. Similar to the most recent works in this direction, we do not rely on the public and private datasets coming from the same distribution.

[2]For Laplace-based COUNT mechanisms the updates are $1/\Delta_0$, but for Gaussian-based they are $\sqrt{1/\Delta_0}$ (Korolova et al. 2009).

[3]For Laplace-based WEIGHTED mechanisms the updates are $1/|W_i|$, but for Gaussian-based they are $\sqrt{1/|W_i|}$ (Gopi et al. 2020).

3 Main Algorithms

Following the framework in Algorithm 1, here we present two of our main contributions: eliminating the need for sampling η on STEP 1, which allows all items to be considered without costing additional privacy, and the *greedy update* $\mathcal{F}$ on STEP 2, which lowers the threshold ρ for outputting items.

3.1 Eliminating Sampling

For the first time we have STEP 1 of Algorithm 1 returning all items instead of sampling Δ_0. In other words, we have $\overline{W_i} = W_i$ in line 6 of Algorithm 1, and we do **not** use Δ_0.

By taking all items into STEP 2, we avoid randomly losing promising items, giving the following update step more items to consider in order to increase utility. The need for the sampling step in previous work came from the threshold ρ definition. The threshold is used to avoid outputting items that come **only** from the differing user between neighboring D and D', as that would affect privacy. For this, previous work would distribute the weight among a fixed number of items, Δ_0, which needed to be known, and used uniform sampling to select Δ_0 items when users had more than Δ_0 items.

On the other hand, in our update algorithm, which we detail next in Section 3.2, for any set of items **only** coming from the differing user, it chooses just a *single* item to spend all of the weight budget on. The reason is that, before the update, the histogram had no previous contribution to such items – because only the differing user has them – and the cutoff Γ to be reached is ≥ 1. Thus our *greedy* approach will use all of the weight budget of 1 on just one of them to get closer to the cutoff Γ. This way, irrespective of how many items are **only** seen in the differing user, we will update just **one**. Thus we already know the number of items we have to limit outputting with the threshold ρ, not needing to set Δ_0.

3.2 Greedy Update

Our algorithm GU, used to update the items in the histogram following a *greedy* approach, is described in Algorithm 2. Recall that $C_i : W_i \rightarrow \mathbb{Z}$ is an associative item frequency array for each individual user i with a set of items $\overline{W_i}$.

GU starts by selecting the items from $\overline{W_i}$ that have not reached the threshold Γ in the histogram, denoted by S. Using Γ has been introduced by (Gopi et al. 2020) to allow user dependency when building the histogram H of items. GU selects the item with the largest C_i in S (i.e., line 8), hoping other users have a similar distribution, so that such items have the highest chances of reaching ρ sooner. This is the first greedy aspect of GU. After that, the second greedy aspect of GU comes from using the weight budget on the selected item until either spending *all* of the budget or reaching Γ. If there is still budget left, the process is repeated for the remaining items (i.e., line 7).

The key aspect of any greedy update is how to define which item is updated first from the set of possible items. Previous work (Gopi et al. 2020) designed a greedy mechanism that updates the item being closest to the cutoff Γ. However, that leads to a potentially huge difference in the resulting histograms for neighboring datasets, thus **not** satisfying DP, as

Algorithm 2: GU: Greedy update algorithm for each user i.

Input: H: Current histogram.
$\overline{W_i}$: Selected set of items for user i.
C_i: Dictionary with items only from user i where $C_i[w]$ is the count of item w for $w \in \overline{W_i}$.
Γ: Cutoff parameter.
Output: Updated histogram H.

```
1: budget = 1          ▷ Each user has a total weight budget of 1
2: $S = \emptyset$          ▷ Set for items with weight below $\Gamma$.
3: $O = \emptyset$          ▷ Set for items updated in a previous iteration.
   Filtering: Will only update items below cutoff $\Gamma$.
4: for $u \in \overline{W_i}$ do
5:     if $H[u] < \Gamma$ then
6:         $S \leftarrow S \cup u$
7: while budget > 0 do
       Item Selection: Choose the best item to update
8:     $u^\star = \operatorname{argmax}_{u \in S \setminus O} C_i[u]$
       Greedy update: Allocate as much as possible to $u^\star$
9:     cost = $\Gamma - H[u^\star]$
10:    if cost ≤ budget then   ▷ Use budget until reaching $\Gamma$
11:        $H[u^\star] = H[u^\star]$+ cost
12:        budget = budget − cost
13:        $O = O \cup \{u^\star\}$
14:    else          ▷ Budget not enough to reach $\Gamma$, use it all
15:        $H[u^\star] = H[u^\star]$+ budget
16:        break
17: return $H$
```

explained in their paper. We noticed that this ordering step being based on all previous updates of multiple users was the factor breaking the privacy. Our greedy update eliminates this problem because the order of items to be updated for a user i is defined by this user's own internal item distribution, i.e., C_i, hence not being impacted by other users' items.

Comparison to POLICY mechanisms. POLICY updates (Gopi et al. 2020) have to sample Δ_0 items, being subject to more randomness, and update the items' weights proportionally to the gap to the cutoff Γ. This means they distribute the weight budget (thus, not greedy) and give more of the budget to the items further from the cutoff, not closer. (Gopi et al. 2020) suggests various methods to select a value for Δ_0 that either discard some of the private data or cost additional privacy budget, such as (Liu and Talwar 2019). Even though in our experiments we select the best Δ_0 for the POLICY mechanisms without charging extra privacy, in real-world applications that cost needs to be taken into account. In contrast, our update algorithm GU does not need to charge extra privacy budget in this case, as it does not need to set Δ_0.

3.3 Putting All Together: GW

Our DPSU mechanism, **GW** for **G**reedy updates **W**ithout sampling, is obtained from the meta framework in Algorithm 1 without any sampling method η, i.e. $\overline{W_i} = W_i$ in line 6, with the update method $\mathcal{F} = \texttt{GU}(\overline{W_i}, C_i, \Gamma)$ given by Algorithm 2, noise distribution[4] $\xi = \texttt{Lap}(0, \lambda)$ with $\lambda = 1/\varepsilon$, $\varepsilon > 0$, $\Gamma \geq 1$, $0 < \delta < 1$, and the threshold

$$\rho = 1 - \lambda \cdot \log 2\delta \tag{1}$$

To prove the (ε, δ)-DP guarantees of **GW**, which will be stated in Theorem 1, we first prove two related Lemmas.

Let D and D' be neighboring datasets, and let O be any set of possible outputs returned by **GW** on D where an output is a set of items. Let H and H' be the histograms built by the first phase (see Algorithm 1) of **GW** respectively on D and D'. W.l.o.g. let D' be D plus one user's data. Additionally, let E be the event that $O \in supp(H)$ for any possible output O of **GW**.

Lemma 1. *For any pair of neighboring datasets D and D':*

$$\Pr[\boldsymbol{GW}(D) \in O] \leq e^{\varepsilon} \Pr[\boldsymbol{GW}(D') \in O \mid E] \tag{2}$$

Proof. To show Equation (2) we will need to show that $||H' - H||_{\ell_1} \leq 1$. Since O conditioned by E only contains outputs with items that can be returned by both **GW**(D) and **GW**(D'), and with $||H' - H||_{\ell_1} \leq 1$, by adding Laplace noise with $\lambda = 1/\varepsilon$ to the histograms, we get the result in Equation (2) as a consequence of the Laplace mechanism with sensitivity of 1 (see Theorem 3.6 in (Dwork, Roth et al. 2014)). Now, in the rest of proof, we show $||H' - H||_{\ell_1} \leq 1$.

First, note that, every user i present in both D and D' has the same greedy order of items in both datasets, which in our case is deterministic and based on C_i. That is because we do **not** have random sampling of Δ_0 items and C_i is the same on D and D' for any such user i. Similar to (Gopi et al. 2020), we assume that users follow a global order. W.l.o.g. let D' be D plus one user's data, denoted as user x, at position p in the global order. As we have the same first $p-1$ users on D and D', at time $p-1$ we have $H'_{p-1} = H_{p-1}$. After that, as the new user x adds a weight budget of at most 1 to **only** H', we have $||H'_p - H_{p-1}||_{\ell_1} \leq 1$. Now we need to show that this inequality remains true after processing the remaining users after x. This can be done by induction on the number of remaining users f processed after x. $||H'_p - H_{p-1}||_{\ell_1} \leq 1$ is our base case. Let H'_f and H_f be the histograms after the updates from f users after x respectively on H'_p and H_{p-1}. For the inductive step, we assume, for $f \geq 1$, $||H'_{f-1} - H_{f-1}||_{\ell_1} \leq 1$, and we prove that, after the updates from one additional user y, $||H'_f - H_f||_{\ell_1} \leq 1$.

Let α' and α be the updates from user y respectively on H'_{f-1} and H_{f-1}. Let S be the set of all items updated so far, including those from user y. Since updates are always positive and items in H'_{f-1} can only be closer to the cutoff than the items in H_{f-1} (due to the extra differing user x), $H'_f[u] \geq H_f[u]$ (**Property P1**) for any $u \in S$. Also $\sum_{u \in S} \alpha'[u] \leq \sum_{u \in S} \alpha[u]$ (**Property P2**), i.e. the total updates from user y in H'_{f-1} is no more than in H_{f-1}, as some item may reach the cutoff in H'_{f-1} before this happens in H_{f-1}. Therefore:
$||H'_f - H_f||_{\ell_1} = \sum_{u \in S} H'_f[u] - \sum_{u \in S} H_f[u]$ **(From P1)**

[4]Our greedy updates do not allow Gaussian noise due to the additive property on histogram distances not being satisfied by the L2 distance used by the Gaussian mechanism (Balle and Wang 2018).

$$= \sum_{u \in S}(H'_{f-1}[u] + \alpha'[u]) - \sum_{u \in S}(H_{f-1}[u] + \alpha[u])$$
$$\leq ||H'_{f-1} - H_{f-1}||_{\ell_1} + \sum_{u \in S} \alpha'[u] - \sum_{u \in S} \alpha[u]$$
$$\leq ||H'_{f-1} - H_{f-1}||_{\ell_1} \qquad \textbf{(From P2)}$$

Then, as we assumed $||H'_{f-1} - H_{f-1}||_{\ell_1} \leq 1$ in our inductive step, replacing this above completes the proof. □

Lemma 2 below bounds the probability of $\bar{E}$.

Lemma 2. *For any neighboring D and D', $\lambda = 1/\varepsilon$, $\varepsilon > 0$, $\Gamma \geq 1$, $0 < \delta < 1$ and ρ from Equation* (1)*:* $\Pr[\bar{E}] \leq \delta$.

Proof. To have $\bar{E}$ we need outputs with at least one item that can be returned by **GW**(D) but not **GW**(D'). For an item like that to exist, it should be added **only** by the single differing user that exists in one of the neighboring datasets but not the other. And if such an item is added by a single user, it only has a single/first contribution. Also note that even if the differing user has multiple such items, it will only update **one of them**, as they all have zero total weight previously to the update and our greedy mechanism uses all of the weight budget of 1 to get closer to the cutoff $\Gamma \geq 1$.

So to prove the δ bound above, on the event $\bar{E}$, the output must have at least one item u that comes from the differing user and passes the threshold ρ. The discussion above implies that there is only one such item u in the output. Thus, for $X_u \sim Lap(1/\varepsilon)$ and ρ from Equation (1):

$$\Pr[\bar{E}] = \Pr[\hat{H}[u] > \rho]$$
$$= 1 - \Pr[\hat{H}[u] \leq \rho] = 1 - \Pr[H[u] + X_u \leq \rho]$$
$$= 1 - \Pr[X_u \leq \rho - H[u]] \leq 1 - \Pr[X_u \leq \rho - 1]$$
$$= 1 - \big(1 - \frac{1}{2}\exp\big(-\varepsilon(\rho - 1)\big)\big)$$

The last step is a consequence of the Laplace distribution. Simplifying the above we get $0.5 \cdot \exp(\varepsilon - \varepsilon\rho)$, which for ρ in Equation (1) becomes $\Pr[\bar{E}] \leq \delta$. □

Finally, the DP guarantees for **GW** are stated as follows.

Theorem 1. *The DPSU mechanism **GW** is (ε, δ)-DP for $\lambda = 1/\varepsilon$, $\varepsilon > 0$, $\Gamma \geq 1$, $0 < \delta < 1$ and ρ from Equation* (1).

Proof. Let O be any set of possible outputs from **GW** on D where an output is a set of items. For neighboring D and D' differing in one user's data, we want to show that:

$$\Pr[\mathbf{GW}(D) \in O] \leq e^{\varepsilon} \cdot \Pr[\mathbf{GW}(D') \in O] + \delta \tag{3}$$

To achieve this we start from Lemma 1:

$$\Pr[\mathbf{GW}(D) \in O] \leq e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O \mid E] \qquad \textbf{(Lemma 1)}$$
$$\leq e^{\varepsilon} \frac{\Pr[\mathbf{GW}(D') \in O]}{\Pr[E]} \leq e^{\varepsilon} \frac{\Pr[\mathbf{GW}(D') \in O]}{1 - \delta} \textbf{(Lemma 2)}$$
$$= e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O] + \delta \left(\frac{e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O]}{1 - \delta} \right)$$
$$\leq e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O] + \delta$$

If the expression in parentheses is ≤ 1 the result above is direct, otherwise the opposite implies $e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O] > 1 - \delta$, giving $1 < e^{\varepsilon} \Pr[\mathbf{GW}(D') \in O] + \delta$, that with $\Pr[\mathbf{GW}(D) \in O] \leq 1$ and transitivity gives the same result. □

3.4 Threshold Comparison

Here we show that our threshold formulation in Equation (1) gives a ρ never larger than those used by previous work (Gopi et al. 2020). Since only items with noisy weights above a threshold ρ are outputted, smaller values are better for DPSU.

Although (Gopi et al. 2020) proposed two `POLICY` mechanisms, here we compare our threshold with their mechanism that has the smallest ρ formulation: the *Policy Laplace*, which has the threshold given by:

$$\rho_{\texttt{Lap}} = \max_{1 \leq t \leq \Delta_0} \frac{1}{t} + \lambda \log \left(\frac{1}{2(1 - (1 - \delta)^{1/t})} \right) \tag{4}$$

Theorem 2. *Denoting ρ_{GW} as the threshold from Equation* (1)*, and $\rho_{\texttt{Lap}}$ from Equation* (4)*, we have:*

$$\rho_{GW} \leq \rho_{Lap} \tag{5}$$

Proof. If we replace $t = 1$ in Equation (4), we get:

$$1 + \lambda \log \left(\frac{1}{2(1 - (1 - \delta))} \right)$$
$$= 1 + \lambda \log \frac{1}{2\delta} \quad = \quad 1 - \lambda \cdot \log 2\delta \quad = \quad \rho_{GW}$$

Thus, with $t = 1$, $\rho_{\texttt{Lap}}$ becomes ρ_{GW}. Since Equation (4) is a *max* statement for all $1 \leq t \leq \Delta_0$, the result of $\rho_{\texttt{Lap}}$ can never be smaller than that of ρ_{GW}. □

Theorem 2 formally shows that ρ_{GW} is never larger than the threshold from the `POLICY` mechanisms (Gopi et al. 2020). To further illustrate the comparison, in Figure 2 we show the concrete threshold values for the mechanisms analyzed. Basically we see **GW** with considerably smaller threshold than `POLICY` mechanisms, especially for smaller ε.

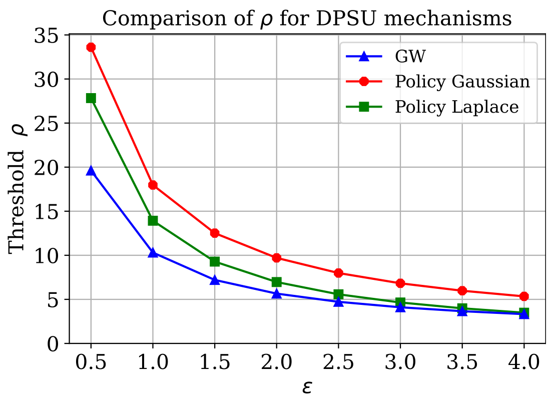


Figure 2: Threshold ρ comparison between DPSU mechanisms, for various ε and $\delta = \exp(-10)$. Smaller is better.

4 Knowledge Transfer

As discussed in Section 1.2, looking only at the data from a single user i to get C_i gives a limited view of the most frequent items in D, as it may not reflect the global distribution for all users. However getting C_i by aggregating all users' sensitive data would not satisfy DP, due to the increasingly different probabilities of outputting items on neighboring datasets. So we propose to use C_i built from the global distribution of a *public* dataset to get more robust weight updates.

The public data is not required to have the same distribution as the sensitive data; as long as the items in the public data have some overlap with those of the sensitive data, C_i built using the public data will be useful for increasing the size of the DP set union. Importantly, note that the usage of the public data is free in terms of privacy budget.

More precisely, we first build the public histogram H^{pub} of item counts using a public dataset. Then, for each user i, for every item u in W_i, if u is in H^{pub} we set $C_i[u]$ to the count of u in H^{pub}, otherwise, we set $C_i[u] = 1$. We use this new C_i, denoted by C_i^{pub}, to replace the input C_i in Algorithm 2 for $\mathcal{F}$. This means that the most frequent items considering the counts from the public dataset will tend to be updated sooner, increasing their chances of being part of the output. Note that the scale of the frequency from C_i^{pub} does not impact our algorithms, as only the *order* given by the item frequency vector matters. Finally, note that our goal still is to optimize the set union coverage of private data: we use public data to gauge the common items in the private data and reach the cutoff quicker for such items, leaving more budget to output the remaining items in the private data.

Our resulting mechanism is **GW-KT**, for **GW** with **K**nowledge **T**ransfer.

Theorem 3. *Let **GW-KT** be **GW** with C_i being replaced with C_i^{pub} in Algorithm 2. **GW-KT** is (ε, δ)-DP.*

Proof. **GW-KT** uses **GW** plus public data. Thus, it is also (ε, δ)-DP, as public data is the same on D and D'. $\square$

5 Experiments

In the empirical evaluation we focus on natural language data, due to their large availability and the frequent use of knowledge transfer in the text domain. In this context we consider the ubiquitous problem of building a vocabulary, which is equivalent to releasing the set union of n-grams for $n = 1$ (unigrams). Our goal is to output the largest vocabulary possible while satisfying user-level DP.

5.1 Datasets

We use three sources as sensitive datasets for the DPSU, as shown in Table 2. Each user i has one or more observations in a dataset, and W_i is the set of unique words from the aggregated observations from each user, with C_i as the corresponding word frequency array. Reddit is a text dataset collected from the subreddit r/AskReddit, and was the only dataset used by (Gopi et al. 2020) for DPSU. We include two other general NLP datasets available on Kaggle: 1) Twitter, with natural conversations between major companies and costumers on Twitter; and 2) Finance, with daily financial news headlines. We use the same pre-processing of (Gopi et al. 2020) for all datasets, such as tokenizing and cleaning.

We also include other five datasets as public datasets for knowledge transfer, with various vocabulary sizes, as shown in Table 3. To emphasize the generality of our approach, we select public datasets that are from different domains. "imdb" comes from IMBD reviews, "covid" from Covid-19 medical papers, "songs" from lyrics of English songs, "wiki" from Wikipedia abstracts, and "enron" from Enron internal e-mails. Thus note that we have domains such as medical, musical, movie and e-mails. Table 3 also shows the intersection (column "∩") between the sensitive and public datasets. The percentage shows how many items of each sensitive dataset are present in the public datasets, also adding the correlation (column "Corr.") of the counts of such intersection items.

Dataset	Users	Observations	Vocabulary
Reddit	223,388	373,983	153,701
Twitter	702,682	2,811,774	1,300,123
Finance	1,400,465	1,400,465	267,256

Table 2: Overview of the sensitive datasets: the number of users, observations and vocabulary size.

Dataset	Vocabulary	Reddit		Twitter		Finance	
		∩	Corr.	∩	Corr.	∩	Corr.
imdb	194,532	31%	0.87	5%	0.70	13%	0.39
covid	784,699	26%	0.42	6%	0.33	23%	0.24
songs	222,074	32%	0.70	6%	0.56	14%	0.27
wiki	631,866	33%	0.69	7%	0.52	24%	0.41
enron	989,560	35%	0.21	9%	0.15	30%	0.12

Table 3: Overview of the public datasets, with their vocabulary size. Column "∩" shows the percentage of the sensitive dataset that is present in the public dataset, while column "Corr." shows the correlation between such common items.

All of the code and datasets are publicly available[5].

5.2 Settings

We compare our algorithms **GW** and **GW-KT** with the current state-of-the-art in DPSU: Policy Laplace and Policy Gaussian (Gopi et al. 2020). To define the cutoff Γ we follow the approach from (Gopi et al. 2020), using a new parameter α. Given each corresponding ρ, λ and σ, for Laplace-based algorithms we set $\Gamma = \rho + \alpha\lambda$, for $\alpha \in [0, 6]$, similarly, for Gaussian-based algorithm we set $\Gamma = \rho + \alpha\sigma$, for $\alpha \in [0, 6]$. Note that λ and σ are defined from ε and δ by each individual mechanism. Moreover, unless otherwise stated, we use the base values of $\varepsilon = 3$, $\alpha = 3$ and $\delta = \exp(-10)$.

While **GW** is free from setting Δ_0 to bound users' contributions, the `POLICY` mechanisms given by (Gopi et al. 2020) still require choosing a value for Δ_0. In the experiments we try $\Delta_0 \in \{1, 10, 20, 30, 50, 100, 200, 300\}$ and report only the best result overall. Although in practice this costs privacy, we do not account for the privacy loss of choosing Δ_0 for the `POLICY` mechanisms, which gives them an advantage.

5.3 Main Conclusions

In the experiments utility is measured by the set union size, thus the larger the better. The general results are shown in Table 4. The main conclusions obtained from such results and the next sections can be summarized as follows:

[5]COMPLETE CODE SUBMITTED AS SUPPL. MATERIAL.

Dataset	Policy Lapl. (best Δ_0)	Policy Gaus. (best Δ_0)	GW	GW-KT (imdb)	GW-KT (covid)	GW-KT (songs)	GW-KT (wiki)	GW-KT (enron)
Reddit	15485 ± 63	16958 ± 37	$\mathbf{17051} \pm 51$	18968 ± 76	18811 ± 36	18979 ± 48	$\mathbf{19057} \pm 14$	19012 ± 41
Twitter	33757 ± 15	34332 ± 49	$\mathbf{37697} \pm 22$	40910 ± 41	41332 ± 32	41739 ± 7	42060 ± 63	$\mathbf{42808} \pm 26$
Finance	45323 ± 75	40724 ± 62	$\mathbf{49868} \pm 23$	50451 ± 98	$\mathbf{51059} \pm 43$	50409 ± 55	50934 ± 64	$\mathbf{51070} \pm 99$

Table 4: Average results with standard error of set union output size over 5 independent trials, for $\varepsilon = \alpha = 3$, $\delta = \exp(-10)$, and the best results for $\Delta_0 \in \{1, 10, 20, 30, 50, 100, 200, 300\}$ for the POLICY mechanisms. Our mechanism **GW** consistently outperforms previous work, and **GW-KT** shows considerable boost in utility by using public data even from diverse domains.

- **GW** consistently outperforms POLICY mechanisms (Gopi et al. 2020) by around 10% for Twitter and Finance datasets, whereas for Reddit it gets equivalent utility.
- **GW-KT** shows even larger improvements than **GW**, with the best results being 12%, 25% and 13% better than (Gopi et al. 2020) respectively for Reddit, Twitter and Finance.
- For knowledge transfer, all of the public datasets tested were beneficial to DPSU, even those from very diverse domains. Moreover, the larger the intersection between public and sensitive vocabularies, the better.

5.4 Impact of The Cutoff Γ

As explained above, to define the cutoff Γ we use the parameter $\alpha \in [0, 6]$. The first column of Figure 3 shows the results of the mechanisms compared for various values of α on the three sensitive datasets analyzed. We see that, for all of the mechanisms, the set union size increases sharply until $\alpha = 3$ and remains approximately constant after.

5.5 Impact of Privacy Budget

Here we analyze how the value of the privacy budget ε impacts the results, since in Table 4 we only used $\varepsilon = 3$. The second column in Figure 3 shows the utility improving as ε becomes larger, as expected. Additionally, overall **GW** performs better than previous work (Gopi et al. 2020). With the only exception being with small ε and specifically for the Reddit dataset. However, note that we did not charge privacy budget (ε) to choose Δ_0 for the POLICY mechanisms.

5.6 Impact of Public Dataset

From Table 4 we see that the public datasets used for knowledge transfer have slightly different results. However, they all consistently improve on previous work.

Cross-referencing the utility results with the overview of the public data from Table 3 shows that generally the largest the intersection of sensitive and public vocabularies, the better utility results. On the other hand, the correlation between the common items does not seem to directly impact the results. Therefore, as a general rule, we consider good practice to use a public dataset as large as possible. Nonetheless, even when the intersection is very small, e.g. 5% for Twitter dataset, using knowledge transfer is still beneficial. We note that even in these cases the absolute number of common items is reasonably large, which can help to improve DPSU.

Finally, we note that the domain of the dataset does not have a large impact on utility. For example, for the sensitive Finance dataset, we see the public dataset "covid" with medical text from Covid-19 papers as one of the best performing.

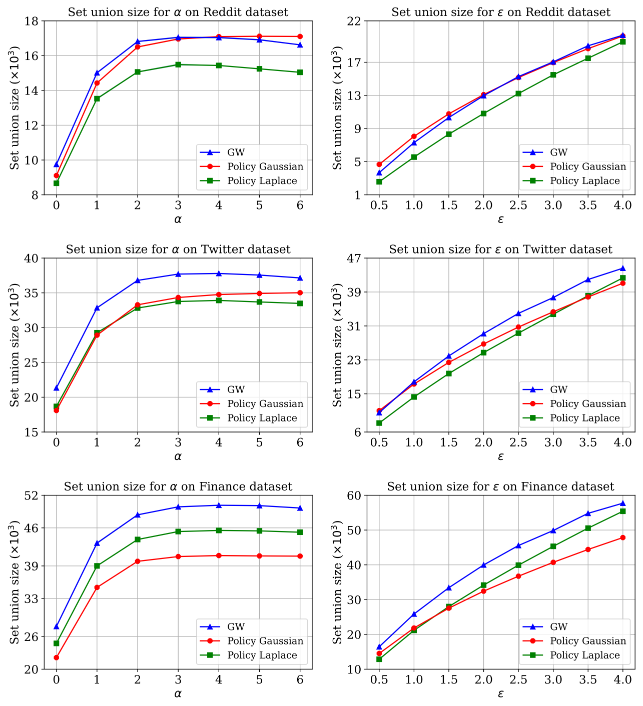


Figure 3: Comparison of mechanisms for varying α (first column) or ε (second column). Each row has results for a dataset. For α, results improve until reaching $\alpha = 3$ and remain nearly constant after. Across all of the ε tested, improvements are generally similar.

6 Conclusion

We proposed DPSU mechanisms that incorporate item frequency in a novel greedy update step. The first main advantage of our methods is eliminating the sampling step that was employed by all previous works to limit the number of items contributed by a single user in order to satisfy DP. Moreover, our proposed greedy mechanisms satisfy differential privacy and have a threshold for outputting items that is formally proved to never be larger than those of previous work. Finally, we include a version of our algorithm with knowledge transfer, which empirically shows an additional utility boost, even when using a public dataset from a very diverse domain.

Acknowledgements

This research is in part supported by a CGS-D award for Lovedeep Gondara and a discovery grant for Ke Wang from Natural Sciences and Engineering Research Council of Canada.

References

Abowd, J. M. 2018. The US Census Bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2867–2867.

Balle, B.; and Wang, Y.-X. 2018. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. arXiv:1805.06530.

Bassily, R.; Cheu, A.; Moran, S.; Nikolov, A.; Ullman, J.; and Wu, S. 2020. Private query release assisted by public data. In *International Conference on Machine Learning*, 695–703. PMLR.

Bassily, R.; Moran, S.; and Nandi, A. 2020. Learning from mixtures of private and public populations. *arXiv preprint arXiv:2008.00331*.

Carlini, N.; Liu, C.; Erlingsson, Ú.; Kos, J.; and Song, D. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 267–284.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. 2020. Extracting Training Data from Large Language Models. *arXiv preprint arXiv:2012.07805*.

Chen, M. X.; Lee, B. N.; Bansal, G.; Cao, Y.; Zhang, S.; Lu, J.; Tsay, J.; Wang, Y.; Dai, A. M.; Chen, Z.; et al. 2019. Gmail smart compose: Real-time assisted writing. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2287–2295.

Desfontaines, D.; Voss, J.; and Gipson, B. 2020. Differentially private partition selection. *arXiv preprint arXiv:2006.03684*.

Ding, B.; Kulkarni, J.; and Yekhanin, S. 2017. Collecting telemetry data privately. In *Neur IPS*, 3571–3580.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*.

Fanti, G.; Pihur, V.; and Erlingsson, Ú. 2016. Building a rappor with the unknown: Privacy-preserving learning of associations and data dictionaries. *Proceedings on Privacy Enhancing Technologies*, 2016(3): 41–61.

Gopi, S.; Gulhane, P.; Kulkarni, J.; Shen, J. H.; Shokouhi, M.; and Yekhanin, S. 2020. Differentially private set union. *International Conference on Machine Learning (ICML)*.

Ji, Z.; and Elkan, C. 2013. Differential privacy based on importance weighting. *Machine learning*, 93(1): 163–183.

Kannan, A.; Kurach, K.; Ravi, S.; Kaufmann, T.; Tomkins, A.; Miklos, B.; Corrado, G.; Lukacs, L.; Ganea, M.; Young, P.; et al. 2016. Smart reply: Automated response suggestion for email. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 955–964.

Korolova, A.; Kenthapadi, K.; Mishra, N.; and Ntoulas, A. 2009. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, 171–180.

Kuo, Y.-H.; Chiu, C.-C.; Kifer, D.; Hay, M.; and Machanavajjhala, A. 2018. Differentially private hierarchical count-of-counts histograms. *arXiv preprint arXiv:1804.00370*.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 233–246.

Liu, J.; and Talwar, K. 2019. Private selection from private candidates. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 298–309.

Liu, T.; Vietri, G.; Steinke, T.; Ullman, J.; and Wu, S. 2021. Leveraging Public Data for Practical Private Query Release. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 6968–6977. PMLR.

Vodrahalli, K. 2015. Deep Learning for NLP.

Wilson, R. J.; Zhang, C. Y.; Lam, W.; Desfontaines, D.; Simmons-Marengo, D.; and Gipson, B. 2020. Differentially private sql with bounded user contribution. *Proceedings on Privacy Enhancing Technologies*, 2020(2): 230–250.