A new Framework for Measuring Re-Identification Risk

CJ Carey
Google
cjcarey@google.com

Travis Dick
Google
tdick@google.com

Alessandro Epasto
Google
aepasto@google.com

Adel Javanmard
Google
adeljavanmard@google.com

Josh Karlin
Google
jkarlin@google.com

Shankar Kumar
Google
shankarkumar@google.com

Andres Muñoz Medina
Google
ammedina@google.com

Gabriel Henrique Nunes
Macquarie University and UFMG*
gabriel.nunes@mq.edu.au

Sergei Vassilvitskii
Google
sergeiv@google.com

Peilin Zhong
Google
peilinz@google.com

Abstract

Compact user representations (such as embeddings) form the backbone of personalization services. In this work, we present a new theoretical framework to measure re-identification risk in such user representations. Our framework, based on hypothesis testing, formally bounds the probability that an attacker may be able to obtain the identity of a user from their representation. As an application, we show how our framework is general enough to model important real-world applications such as the Chrome's Topics API for interest-based advertising. We complement our theoretical bounds by showing provably good attack algorithms for re-identification that we use to estimate the re-identification risk in the Topics API. We believe this work provides a rigorous and interpretable notion of re-identification risk and a framework to measure it that can be used to inform real-world applications.

1 Introduction

From curated travel suggestions to localized search results and relevant ads, personalization of online content has become increasingly important, as users expect online systems to be intelligent enough to anticipate their needs. The majority of these systems are now powered by so-called user representations — for instance dense vector embeddings in $\mathbb{R}^d$ or discrete tokens—that enable classifiers and recommendation systems to generate useful personalized content.

User representations form a compact description of a user profile that distill a user's interest into a short vector. For instance, a music streaming service may represent a user by a genre of songs the user likes. More sophisticated embeddings may be trained using neural networks and not be as readily explainable.

*Work done while at Google.

Workshop on Regulatable Machine Learning at the 37th Conference on Neural Information Processing Systems (RegML @ NeurIPS 2023).

The compact representation of user profiles may be seen as providing some privacy: instead of describing a user by all raw data (e.g. all songs they have listened to), one can summarize their profile in a few bits. Of course, such informal privacy arguments can be incorrect, and do not account for sophisticated attacks or non-obvious data leak vectors. In this work we study the question formally. Specifically, we ask to what extent user representations can be used to link back to and re-identify individuals. We provide a rigorous and interpretable notion of re-identification risk and a framework to measure it.

Assessing Privacy Since the seminal work of Dwork et al. [18] introducing differential privacy, significant effort has been devoted into developing differentially private algorithms for numerous problems and settings. Despite these efforts, far from all production systems in use currently employ differential privacy. Moreover, the ever evolving legal framework for privacy (such as GDPR [12] and other regulations) has introduced over time even more privacy definitions, while research has pointed out that these definitions are often not compatible with basic privacy properties like composability [12].

An immediate question is how to assess the privacy of such real-world systems. If the underlying method is randomized, one can try to establish its sensitivity to a single user's input and derive differential privacy bounds. However, just as in the GDPR example highlighted by [12], the bounds may be vacuous. In this work we propose a complementary approach. We develop a rigorous new framework, centered around re-identification risk, and prove two kinds of results. First, is a series of unconditional lower bounds, showing the risk present in the system. Second, we give a formalization of upper bounds under a "closed-world" assumption, where we can characterize all of the information an attacker has access to.

We remark that measuring re-identification risk complements the guarantees provided by other notion of privacy, such as (local) differential privacy and k-anonymity, by giving guarantees in a different domain. As part of our analysis, we derive re-id risk guarantees for differentially private algorithms, and k-anonymous datasets. Critically, however, this notion can provide a measure of risk for systems adopting other ad-hoc privacy approaches, an important contribution given the large number of products not designed with differential privacy guarantees.

Applications One concrete example we study is the the Topics API [26] proposed by Chrome as part of the Privacy Sandbox initiative [25]. In a nutshell, the Topics API creates a representation of a user corresponding to the top interests of the user in a taxonomy of about 350 interests. When a website calls the Topics API, it returns one of the top interests of the user uniformly at random (and with some small probability a random topic from the taxonomy not necessarily in the user representation). We present a formal study quantifying the extent to which the Topics API can be used to re-identify users across different domains. We couple this work with an experiment quantifying the risk in releasing samples of music listened by users in a large song dataset. Using our framework, we can quantify re-identification risk in this example, and show that with 4 independent samples per user, the re-id risk is around 1%.

Main contributions In summary, we make the following main contributions:

- We describe a hypothesis testing [9] framework for evaluating the risk of identifying a random user given access to a representation as well as knowledge of the (potentially randomized) process that generates the representation.
- We also conside a scenario where attacker also observes representations of all users and uses this information to match representations to user identities. We refer to this scenario as the matching setting.
- We measure the re-identification risk against an attacker that does not fully know the representation generation distribution (e.g., due to uncertainty on the underlying user data) but has a prior over the set of distributions that generated the representations.
- We specialize this scenario to the Topics API for which we provide an in-depth analysis of an attacker's re-identification ability under this model.
- We validate our results on two datasets. First a publicly available song dataset, and second a simulation of the Topics API based on proprietary data.

2 Related Work

Our work builds upon the rich literature on data anonymization [29], statistical privacy [28], representation learning [7], and web fingerprinting and tracking [31] and more generally on privacy preserving data mining and analysis [2, 23, 45]. Here we discuss only elements of these areas that most relate to the problem of user re-identification. In Section 5 we provide a more in-depth study of how our notion of re-identification relates to other privacy definitions.

Privacy preserving data releases This privacy framework traditionally models a data curator that wants to release data without leaking individual user information. In the context of sharing user representations, two of the most popular schemes for doing this are k-anonymity [42] and differential privacy [16]. In k-anonymity, the data curator ensures that each user representation is shared by at least another k users. This is normally achieved by using optimization techniques for minimizing error in redacting tabular data [30, 37, 1, 35], as well as clustering algorithms [21], and tree-based methods [32]. Differential privacy and in particular local differential privacy [48], introduce noise into the user representations to bound that information about a user that is leaked by the representation. There is a vast literature on differential privacy [18, 34, 10, 38, 33, 17] (we refer to [19] for a survey). Significant work in differential privacy has been devoted to private representation learning and machine learning. In Section 5, we show that these two notions are theoretically sufficient (but not necessary) to prove low re-identification risk.

Browser fingerprinting and web tracking It is well known that data brokers and ad tech providers can use technologies such as third-party cookies to build detailed browsing history profiles of users [6]. With the impending deprecation of third-party cookies by major browsers, significant attention has been focused by the research community on understanding covert ways of tracking. One such method is browser fingerprinting [31], which relies on characteristics of devices including the screen size and operating system versions to uniquely identify a user across the web. Several studies [20, 24, 4] have analyzed how much information (measured by entropy) trackers can gain from such APIs to create a user identifier. The more entropy an API has, the more likely is that it can be used to re-identifying a user. With some few exceptions [36] most of the research work in this area has focused on understanding the re-identification risk of an API over a fixed instant in time. Our work expands this area of research by modeling the information leakage across time as values returned by an API can change (see section 6). Moreover, we believe our notion of re-identification can have a more straight-forward semantic interpretation compared to the number of bits leaked by an API as it directly measures the re-identification risk.

Information theoretic notions of privacy Finally, there is a rich body of literature [47, 46, 15] devoted to studying privacy leakage of information release through the lens of channel capacity. In the context of this paper, a channel is an object that takes as input user information and outputs a user representation. The information theoretic view of privacy measures the ability of an attacker with access to the output of a channel (and with knowledge of the channel internal mechanism) to reconstruct the input to the channel. The quantification of this ability traditionally involves analyzing the mutual information [14] between the input and output of the channel and its generalizations [44]. In Section 5 we show how one can derive bounds on re-identification risk using mutual information. Another related work is that of Cohen et al. [12] that formalizes privacy as a protection against singling out a user. Unlike our paper, it makes certain assumptions on data generation not present in our work (e.g., that data must be i.i.d. from certain distributions).

Finally, we compare our framework to that of quantitative information flow (QIF) [3], a recent generalization of information theory which provides a flexible and semantically sound framework for analyzing the privacy risks of a channel. As we shall see later on, elements of our framework can be seen as a special case of the QIF framework. By specializing it however, we are able to provide tight characterizations of optimal re-identification attacks.

3 Model

We consider a universe of n users indexed by identities in the set $\mathcal{I} = [n]$ and an attacker whose goal is to re-identify users based on their representations. Our goal is to quantify the extent to which an attacker can re-identify users as a function of how the user representations are constructed. We study

this problem in two settings: First, in the random-user setting, one of the n users is chosen uniformly at random from $\mathcal{I}$, that user's representation is revealed to the attacker, and the attacker attempts to guess the user's identity. Next, in the matching setting, the attacker observes representations for all n users, and their goal is to match each of the n representations to one of the n user identities. Notice how this modeling can be seen as a form of hypothesis testing, where the hypothesis corresponds to the possible identities of the user. We begin by formally defining the processes that generate representations for each user. We then describe the formal attack model, which quantifies how attackers interact with these representations.

Representations We assume that user representations belong to a representation space $\mathcal{O}$ with finite cardinality $|\mathcal{O}| = m$. For instance, $\mathcal{O}$ may be the set of all music genres.[2] Note that we do not make any further structural assumptions on $\mathcal{O}$. For example, $\mathcal{O}$ may be a finite set of topics, songs, images, or points in $\mathbb{R}^d$.

An important part of our setup is that a user is not associated with a single representation, but rather with a *distribution* over representations. In other words, the process that assigns a user to a representation may be randomized. As a running example, suppose a user is equally interested in *classical* music and *alt-rock*. One way to represent this behavior is as a fractional assignment $\{0.5, 0.5\}$ to the two genres. Then, whenever a single genre is required, one can select it from the given distribution. (Looking ahead, this fractional assignment will be key for strengthening re-identification protections.)

Formally, we can encode the representation distributions assigned to all n users using a row-stochastic *representation matrix* $\mathbf{P} \in [0,1]^{n \times m}$ where $\mathbf{P}[i, o]$ is the probability that user $i \in \mathcal{I}$ has representation $o \in \mathcal{O}$ (where we slightly abuse notation and use elements of $\mathcal{O}$ to index columns of $\mathbf{P}$). We write $\mathbf{P}[i, :]$ to denote the ith row of $\mathbf{P}$, which is the user i's distribution over $\mathcal{O}$.[3]

Finally, in order to reason about potentially randomized algorithms that assign representations to users, we model the process that constructs the representation matrix $\mathbf{P}$ (i.e., that assigns representations to users) as a distribution $\mathcal{D}$ over representation matrices in $[0,1]^{n \times m}$. The distribution $\mathcal{D}$ models both the user behavior informing the representations and algorithms used to construct the assignment of representation distributions to users.

Attack Model We begin with nature sampling a single representation matrix $\mathbf{P}$ from $\mathcal{D}$. At that point, attackers will attempt to re-identify users based on representations sampled from $\mathbf{P}$. We detail this further below.

The attacker's goal is to re-identify users based on their representations (instead of using them for the intended use case—e.g., personalization of content). We categorize attackers by the varying degrees of knowledge about the sampled representation matrix $\mathbf{P}$.

In the *full-information* setting, we assume that the attacker observes the sampled representation matrix $\mathbf{P}$. This corresponds to a powerful attacker with detailed knowledge of all users and the representation generation process $\mathcal{D}$. In the partial-information setting, we assume a weaker attacker: one that receives a vector containing one representation sampled from each user's representation distribution: $W \in \mathcal{O}^n$ where $W_i \sim \mathbf{P}[i, :]$. This corresponds to a situation where an attacker only learns about $\mathbf{P}$ as a client consuming representations.

In both scenarios, the attacker uses their knowledge of $\mathbf{P}$ (either the actual $\mathbf{P}$ itself or the vector of representations W) in order to construct a prediction rule φ that they will use to re-identify or match users. In the full-information setting, we require that φ be $\mathbf{P}$-measurable, while in the partial-information setting, φ must be W-measurable.

Success Metrics In the random-user setting, the attacker attempts to re-identify a single randomly chosen user based on a sample of their representation.

[2] We make the finite cardinality assumption for simplicity, as it is sufficient to elucidate the main results of the paper and to model the Topics API. We believe however our theory can be extended to dense embeddings as well.

[3] We stress that the set O is arbitrary and this allows representing arbitrary discrete distributions, including over high dimensional spaces. We leave generalizing our framework to continuous distributions as a future work.

Formally, the attacker uses their knowledge of $\mathbf{P}$ to construct a possibly randomized prediction rule $\varphi_R : \mathcal{O} \to \mathcal{I}$. We define the accuracy random variable as follows: let I be a uniformly random sample from $[n]$, O be sampled from $\mathbf{P}[I, :]$, and define

$$\mathrm{Acc_R}(\varphi_R) = \mathbb{P}(\varphi_R(O) = I \,|\, \mathbf{P}),$$

which is the probability that the attacker correctly re-identifies the random user conditioned on the representation matrix $\mathbf{P}$.

In the matching setting, the attacker receives a set of representations, one for each user, and their goal is to match them to the user identities.

Formally, the attacker uses their knowledge of $\mathbf{P}$ to construct a matching rule $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$. We define the matching accuracy random variable as follows: Sample a permutation $\pi : [n] \to [n]$ uniformly at random, a vector of independent representations $O_{1:n} \in \mathcal{O}^n$, where $O_i \sim \mathbf{P}[\pi(i), :]$ is a representation sampled for user $\pi(i)$, and define

$$\mathrm{Acc_M}(\varphi_M) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\varphi_M^i(O_{1:n}) = \pi(i)\} \,\middle|\, \mathbf{P}\right],$$

where $\varphi_M^i(O_{1:n})$ denotes the ith component of $\varphi_M(O_{1:n})$, which is the attacker's prediction for the user identity that produced observation O_i. This is the expected fraction of users the attacker correctly re-identifies, conditioned on the representation matrix $\mathbf{P}$. We do not require that $\varphi_M(O_{1:n})$ is a permutation of $\mathcal{I}$. Note that the random permutation π is used to ensure that the indices in the representation vector $O_{1:n}$ are not useful for re-identification. In particular, any constant prediction rule φ_M has $\mathrm{Acc_M}(\varphi_M) = 1/n$.

4 Random-user Accuracy Bounds

In this section we provide information theoretic bounds on the accuracy that any attacker can achieve in the random-user setting. Recall that, in the random user setting, we sample a representation matrix $\mathbf{P}$ from the distribution $\mathcal{D}$ and the attacker formulates a (possibly randomized) prediction rule $\varphi_R : \mathcal{O} \to \mathcal{I}$ based on their knowledge of $\mathbf{P}$. We let I be a user drawn uniformly at random from $\mathcal{I}$ and $O \in \mathcal{O}$ be a representation sampled from $\mathbf{P}[I, :]$, and the accuracy random variable, $\mathrm{Acc_R}(\varphi_R)$, is the probability that $\varphi(O) = I$, conditioned on the representation matrix $\mathbf{P}$. Our main result holds with probability one over the draw of the representation matrix $\mathbf{P}$ from $\mathcal{D}$. First, for any prediction rule $\varphi_R : \mathcal{O} \to \mathcal{I}$, we provide an exact expression for $\mathrm{Acc_R}(\varphi_R)$. Next, we prove an upper bound on $\mathrm{Acc_R}(\varphi_R)$ that depends only on $\mathbf{P}$. Our upper bound is tight in the full information setting.

Before stating our results, we introduce a matrix representation for any (possibly randomized) prediction rule $\varphi_R : \mathcal{O} \to \mathcal{I}$. Define the matrix $\mathbf{A} \in [0, 1]^{n \times m}$ to have entries $\mathbf{A}[i, o] = \mathbb{P}(\varphi_R(o) = i \,|\, \mathbf{P})$. That is, conditioned on $\mathbf{P}$, $\mathbf{A}[i, o]$ is the probability that the attacker's rule predicts user $i \in \mathcal{I}$ after receiving representation $o \in \mathcal{O}$. With this, we are ready to state our main results. (All omitted proofs are available in the Appendix.)

Lemma 1. *Let $\mathbf{P} \sim \mathcal{D}$ be a random representation matrix and $\mathbf{A}$ be the matrix representation of an attacker's prediction rule $\varphi_R : \mathcal{O} \to \mathcal{I}$. Then, with probability one, we have that*

$$\mathrm{Acc}_R(\varphi_R) = \frac{1}{n}\,\mathrm{tr}(\mathbf{P}\mathbf{A}^\top).$$

Next, we provide upper bounds on $\mathrm{Acc_R}(\varphi_R)$ in terms of the representation matrix $\mathbf{P}$.

Corollary 1. *Let $\mathbf{P} \sim \mathcal{D}$ be a random representation matrix and let φ_R be any prediction rule. Then, with probability one, we have that*

$$\mathrm{Acc}_R(\varphi_R) \leq \frac{1}{n}\sum_{o \in \mathcal{O}} \max_{i \in \mathcal{I}} P[i, o] =: \frac{1}{n}\|\mathbf{P}\|_{\infty,1}.$$

Additionally, there exists an attacker in the full-information setting capable of constructing φ_R^* such that $\mathrm{Acc}_R(\varphi_R^*) = \frac{1}{n}\|\mathbf{P}\|_{\infty,1}$.

Next, we prove a bound on the accuracy of an attacker with partial-information. In particular, we bound their expected accuracy conditioned on the partial information contained in $W \in \mathcal{O}^n$. This is the probability that the attacker will correctly predict the identity of a random user based on a sample representation, conditioned on vector W.

Lemma 2. *Let $\mathbf{P} \sim \mathcal{D}$ be a random representation matrix, $W \in \mathcal{O}^n$ be a vector of independent representations where $W_i \sim \mathbf{P}[i,:]$, and let $\varphi_R : \mathcal{O} \to \mathcal{I}$ be a prediction rule that is W-measurable. Then we have that*

$$\mathbb{E}\big[\mathrm{Acc}_R(\varphi_R) \mid W\big] \leq \frac{1}{n}\big\|\mathbb{E}[\mathbf{P} \mid W]\big\|_{\infty,1}.$$

Moreover there exists a prediction rule φ_R^* for which the upper bound is achieved.

For lack of space we refer to Appendix A for a discussion of the matching model.

5 Relation to other privacy notions

In this section discuss the re-identification risk introduced in Section 3 in relation to two prior notions of algorithmic privacy: local differential privacy and k-anonymity. We show that (for appropriate parameters) both of these privacy notions are *sufficient* to imply low re-identification risk, but neither condition is *necessary* to obtain low re-identification risk in our framework. (We refer however to the discussion in Section 8 on why they may still be needed for other privacy risks.) This shows that the re-identification risk outlined in Section 3 is not entirely captured by either of these concepts. For lack of space, in this section we only report the results for local DP and defer to the Appendix B for the other results.

Local differential privacy Local differential privacy (LDP) [13] is a strong privacy notion applicable to publishing user representations constructed from private information. Intuitively, it should be hard to derive the identity of a user from the output of a differentially private mechanism. In this section we prove this implication, while, at the same time, showing that local differential privacy is *not necessary* for low re-identification risk. This result highlights the ability of our framework to characterize directly and sharply re-identification risks.

Definition 1 (Local differential privacy)**.** *Let $\mathcal{X}$ be an arbitrary space encoding user information. A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{O}$ for mapping user data to a representation satisfies (ϵ, δ)-LDP if the following holds: for all $x, x' \in \mathcal{X}$ and any set of representations $E \subset \mathcal{O}$, we have that*

$$\mathbb{P}(\mathcal{A}(x) \in E) \leq e^{\epsilon} \cdot \mathbb{P}(\mathcal{A}(x') \in E) + \delta.$$

LDP representations as described above can be modeled under our framework as follows: Let $\mathcal{D}$ be a distribution over representation matrices that samples $\mathbf{P}$ in two steps: First, the n users generate their data $x_1, \ldots, x_n \in \mathcal{X}$. Second, we define $\mathbf{P} \in [0,1]^{n \times m}$ to have entries $\mathbf{P}[i,o] = \mathbb{P}(\mathcal{A}(x_i) = o)$, where the probability is only over the randomness of the mechanism $\mathcal{A}$. Sampling $\mathbf{P}$ from $\mathcal{D}$ corresponds to the process generating the users data, while the matrix $\mathbf{P}$ encodes the mechanism $\mathcal{A}$'s output distribution for each user. Since $\mathcal{A}$ is (ϵ, δ)-LDP, with probability one over the draw of $\mathbf{P}$, we have that for any users i and j, and any representation subset $E \subset \mathcal{O}$, the following holds:

$$\sum_{o \in E} \mathbf{P}[i,o] \leq \delta + e^{\epsilon} \cdot \sum_{o \in E} \mathbf{P}[j,o]. \tag{1}$$

More generally, we say that any distribution $\mathcal{D}$ over representation matrices $\mathbf{P}$ that satisfy (1) with probability one is (ϵ, δ)-LDP.

The following result shows that (ϵ, δ)-LDP implies low re-identification accuracy in the random-user setting.

Lemma 3. *Let $\mathcal{D}$ be any distribution that satisfies (ϵ, δ)-LDP, let $\mathbf{P} \sim \mathcal{D}$, and φ_R be an attacker's prediction rule in the random user setting. Then $\mathrm{Acc}_R(\varphi_R) \leq \frac{e^{\epsilon} + \min(n,m) \cdot \delta}{n}$.*

6 Case Study: The Topics API

As mentioned in the introduction, we will use our framework to provide a theoretical and empirical analysis of the re-identification risk in context of the Topics API [26] of the Privacy Sandbox [25]. For lack of space, here, we only briefly introduce the Topics API and present a sketch of our empirical results. In Appendix D we discuss a formal analysis of the Topics API in the theoritical framework framework of Section 3. Then, in Section 7, we present our empirical analysis of the Topics API.

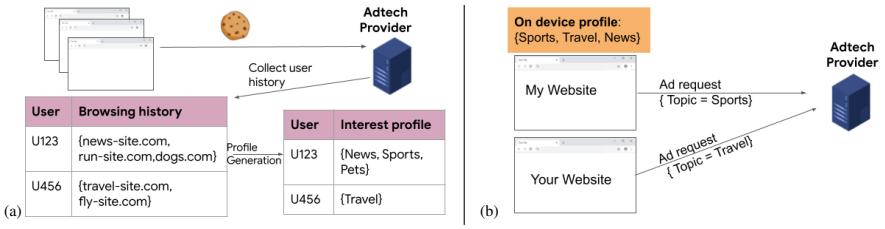

Figure 1: Comparison of IBA under (a) third party cookies and (b) the Topics API. In the former, the ad tech provider gets to build the browsing history of a user. In the latter, the ad tech provider only observes a single topic for this user.

The Privacy Sandbox [25] is a series of proposals to enable online advertising while limiting cross-site tracking on the web. We will focus on Interest Based Advertising (IBA) use case of the Privacy Sandbox. IBA is a sector of online advertising in which ad-tech providers build models of the users' interests in an effort to show them relevant ads. For instance, people interested in a car may be served car ads even on unrelated pages. Historically, IBA has been enabled through third-party cookies. These serve as a cross-site user identifier, allowing ad techs to keep track of the sites a user has visited and build an interest profile based on their browsing history. This cross-site tracking is in direct contrast with the goals of the Privacy Sandbox, which has led Chrome to announce the Topics API to support IBA without relying on cross-site tracking.

The Topics API works as follows (we refer to the specifications in [26]): every week the *browser* builds an interest profile of the user, in the form of selecting top five topics from a fixed topics taxonomy, $\mathcal{T}$. Importantly, this profile is kept on the browsers and is not shared with others.

Whenever a website wants to show an ad, the browser shares a topic selected uniformly at random from one of the top 5 topics in the profile of the previous week with the ad tech provider (additionally with some probability p, it may simply return a uniformly at random topic from $\mathcal{T}$). Crucially, for every user, the topic sampled for a website is fixed for an entire week, and the samples on two different websites are independent.[4] See Figure 1 for an example of the Topics API.

Re-identification risks of the Topics API Compared to third-party cookies, the Topics API has a significantly lower risk of re-identifiability (the former guarantees re-identifiability by nature of being a persistent cross-site identifier). The goal of our work is to formally and empirically measure the risk in the case of misuse of the API.

Consider a user that visits two sites every week. Over time, even with the randomization, the sequence of topics observed on website 1 will be similar to the sequence of topics observed on website 2 (for instance, we expect an exact match in $(1-p)/5$ fraction of the weeks). Thus two sites could try to collude to use the Topics API to link the identity of a user across them.[5]

Figure 3 shows how this attack may happen.

For lack of space we defer to Appendix D for more details on our theoretical analysis of the Topics API. In such section we first provide a mathematical formalization of this attack in our framework. Then, we present novel algorithms to perform re-identification attacks that we later use in our empirical analysis. Finally, we also present, under certain assumptions, proof of optimality for some attack algorithm.

[4]The API actually returns, on top of the current sampled topic, a cached result for the output of the previous 2 weeks for the caller. We omit this detail from the modeling as observing r consecutive weeks of Topics, simply corresponds to performing calls for $r+2$ weeks in our model.

[5]We restrict our analysis to two sites only for simplicity of exposition, as two sites are sufficient to elucidate the re-identification risk. We refer to section 8 for the limitations of our work.

7 Empirical analysis

Based on our theoretical modelling (see Appendix D), we now discuss an empirical evaluations of our model for re-identification risk on a proprietary dataset[6] containing de-identified user data from a simulation of the Chrome Topics API, In order to foster the reproducibility of our results we released our code open source.[7]

First, we describe the data used.

Chrome Topics API data In our empirical analysis we simulate the output observed by an adtech from the Topics API for a set of users over a period of time. This is achieved using a proprietary dataset of de-identified user browsing histories. Starting from this dataset, we run the Topics API algorithm for a sample of such users and simulate the output observed for two sites by the adtech from the Topics API sampling distribution. Our observation period consists of 8 intervals of 7 days of traffic shifted by 3 days each – i.e. we consider intervals $[1, 7], [4, 10], [7, 13], \ldots$. Each interval (hence-forth epoch) is used to establish the top k topics of the user using the API topic model.[8] We restrict our analysis to the set of users that are observed in every epoch. To be consistent with the current Topics API specification [26], for users with fewer than k topics in an epoch, we pad the top topics with random topics. Moreover, we set $k = 5$ for the number of topics and use $p = 5\%$ for the probability of releasing a random topic instead of an organic one, as currently implemented in Chrome.

Methods We now present the methodology used to establish the accuracy of empirical re-identification attacks on the Topics API. We focus on the random-user model and consider different attack algorithms (See Appendix E for more details). We extract a target dataset of 10 million users chosen uniformly at random that are used in the re-identification attack analysis as the set $\mathcal{I}$ over which to re-identify the user and simulate the observation on website 1 of all their topics sequence for r epochs ($r \in [1, 8]$). Then we repeat $10,000$ times a uniformly random draw of a user from the set $\mathcal{I}$, generate the r-length sequence on website 2 for such user and verify if the attack algorithm matches it correctly to the sample in website 1.

Results for the re-identification attack We report the estimation of the probability of each attack algorithm to correctly re-identify the random user over the 10 million target set of users. In our analysis we study $r = 1, 2, 4, 6, 8$ epochs. The main result is shown in Figure 2.

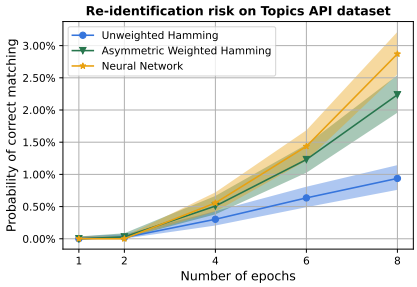

Figure 2: Probability of a correct cross-site match depending on the number of epochs observed. The 95% confidence intervals are reported. Notice how, even after 8 epochs, the probability of correct re-identification is below 3%.

As expected, we observe that as the number of epochs increases there is an increased probability of correctly matching the user across the two sites. Notice also how the more sophisticated supervised

[6]For reproducibility, in Appendix E.2 we report results also on the the public available Million Song Dataset [8].

[7]The code is available online at `https://github.com/google-research/google-research/tree/master/re_identification_risk`.

[8]The real Topics API has disjoint epochs of 7 days, here we simulate training over overlapping periods because our analysis is limited for privacy reasons to 4 weeks of data. We do not observe a significant difference in the results.

Neural Network attack outperforms all methods but has performance close to our implementation of the theory driven attack algorithm (Asymmetric Weighted Hamming). This is a further confirmation of the validity of the simplifying assumptions made in our theoretical study. We also observe the gap between algorithms increases with the number of epochs. This is expected as more epochs of observation allow the advanced neural network algorithm to learn more correlations. Overall, we observe that even after 8 epochs, the probability of correct re-identification is below 3%. We defer to the appendix for the results for the million song dataset.

8 Discussion and limitations

In this paper we have presented a framework for quantifying re-identification risk. Our theoretical formulation is general enough to capture real-world examples such as the Topics API and to derive results for common notions of privacy like k-anonymity and differential privacy, Our experiments show an empirical estimation of re-id risk even on very large representation spaces. We conclude this paper highlighting some of the limitations of our work and future directions of research in this area.

Scope of the privacy risk measurement First, it is important to observe that in our modeling we have focused exclusively, and purposefully, on re-identification risk. Preventing re-identification is only one of the many privacy and safety requirements of real-world applications. As real systems often require to bound additional risks (such as membership inference attacks [39]), we believe that privacy protections from differential privacy (DP) may be needed to complement re-identification risk analyses in many real systems.

Theoretical limitations From a theoretical point of view, we make mild modeling assumptions on the generation process for the representations. Specifically, we assume that each user sample from $\mathbf{P}$ is drawn independently and we restrict our modeling to discrete representations. We remark that the partial information scenario we consider, and the distribution family $\mathbb{D}$ we model, are just one of the many ways in which we could model an attacker with limited information on $\mathbf{P}$. We omit for instance, modeling the presence of more than 2 colluding clients.

More crucially, for the upper bound results, we make a closed-world assumption: we measure the re-identification risk under the assumption that this is the only information available to an attacker. In other words, all of the side information is built into the representation matrix $\mathbf{P}$. While limiting in some situations, this assumption is reasonable in other cases (as we detail below). Moreover, our lower bounds hold even without this assumption (more information can only increase the risk).

Limitations of the Topics API study In generating the data for the Topics API study, we assumed that users visit both websites in all of the epochs; thus we specifically avoided dealing with gaps in the data. This is unlikely to be the case in a real world deployment of such an API. An additional concern is that we used overlapping time periods to simulate longer time horizons, to deal with limitations of the data we had on hand. On the modeling side, while we have tried to develop powerful re-identification methods, future researchers may be able to find more powerful approaches, increasing the empirical re-identification risk. Finally, in our analysis we have focused exclusively on the Topics API output in isolation, ignoring other sources of side information that an attacker might have. Considering additional sources of information is beyond the scope of our work and would require incorporating this information into either the matrix $\mathbf{P}$ or the distribution family $\mathbb{D}$.

Applicability of the framework In summary, our framework provides a lower bound on re-identification risk of data release that complements other privacy metrics that may be implemented by system designers, such as (differential) privacy budgets and aggregation (k-anonymity) guarantees.

In cases where the 'closed-world' assumption holds, we can use the methods developed in this work to provide even stronger upper bounds on re-identification risk. One potential such scenario of special interest is bounding *insider risk*: the risk of an employee maliciously re-identifying user information across a company's systems. Given that such systems have strict controls on the flow of information (e.g. ACLs and auditing systems), we believe this framework can allow data protection officers to measure possibility of re-identification in such cases. Overall, we expect that the tools we have developed could be used by privacy advocates for a data-driven verification of the privacy claims by data brokers and technology companies.

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. In *Proceedings of the International Conference on Database Theory (ICDT 2005)*, November 2005.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 439–450, 2000.

[3] M. S. Alvim, K. Chatzikokolakis, A. McIver, C. Morgan, C. Palamidessi, and G. Smith. *The Science of Quantitative Information Flow*. Information Security and Cryptography. Springer, 2020.

[4] A. I. Awad. Machine learning techniques for fingerprint identification: A short review. In A. E. Hassanien, A. M. Salem, R. A. Ramadan, and T. Kim, editors, *Advanced Machine Learning Technologies and Applications - First International Conference, AMLTA 2012, Cairo, Egypt, December 8-10, 2012. Proceedings*, volume 322 of *Communications in Computer and Information Science*, pages 524–531. Springer, 2012.

[5] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, Jan. 2015.

[6] M. A. Bashir and C. Wilson. Diffusion of user tracking data in the online advertising ecosystem. *Proc. Priv. Enhancing Technol.*, 2018(4):85–103, 2018.

[7] Y. Bengio, A. C. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828, 2013.

[8] T. Bertin-Mahieux, D. P.W. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, 2011.

[9] R. Blahut. Hypothesis testing and information theory. *IEEE Transactions on Information Theory*, 20(4):405–417, 1974.

[10] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

[11] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[12] A. Cohen and K. Nissim. Towards formalizing the gdpr's notion of singling out. *Proceedings of the National Academy of Sciences*, 117(15):8344–8352, 2020.

[13] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang. Privacy at scale: Local differential privacy in practice. In *Proceedings of the 2018 International Conference on Management of Data*, pages 1655–1658, 2018.

[14] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2001.

[15] M. Diaz, H. Wang, F. P. Calmon, and L. Sankar. On the robustness of information-theoretic privacy measures and mechanisms. *IEEE Transactions on Information Theory*, 66(4):1949–1978, 2020.

[16] C. Dwork. Differential privacy: A survey of results. In M. Agrawal, D. Du, Z. Duan, and A. Li, editors, *Theory and Applications of Models of Computation, 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings*, volume 4978 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2008.

[17] C. Dwork. Differential privacy and the us census. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI symposium on principles of database systems*, pages 1–1, 2019.

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

[19] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[20] P. Eckersley. How unique is your web browser? In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, page 1–18, Berlin, Heidelberg, 2010. Springer-Verlag.

[21] A. Epasto, A. M. Medina, S. Avery, Y. Bai, R. Busa-Fekete, C. J. Carey, Y. Gao, D. Guthrie, S. Ghosh, J. Ioannidis, J. Jiao, J. Lacki, J. Lee, A. Mauser, B. Milch, V. S. Mirrokni, D. Ravichandran, W. Shi, M. Spero, Y. Sun, U. Syed, S. Vassilvitskii, and S. Wang. Clustering for private interest-based advertising. In F. Zhu, B. C. Ooi, and C. Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2802–2810. ACM, 2021.

[22] A. Epasto, A. Munoz Medina, C. Ilvento, and J. Karlin. Measures of Cross-Site Re-Identification Risk: an Analysis of the Topics API Proposal. `https://github.com/patcg-individual-drafts/topics/blob/main/topics_analysis.pdf`, 2022.

[23] A. Evfimievski, J. Gehrke, and R. Srikant. Limiting privacy breaches in privacy preserving data mining. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 211–222, 2003.

[24] A. Gómez-Boix, P. Laperdrix, and B. Baudry. Hiding in the crowd: an analysis of the effectiveness of browser fingerprinting at large scale. In P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, editors, *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, pages 309–318. ACM, 2018.

[25] Google. The Privacy Sandbox. Available at `https://privacysandbox.com/intl/en_us/`.

[26] Google. The Topics API. Available at `https://github.com/patcg-individual-drafts/topics`.

[27] J. Halcrow, A. Mosoi, S. Ruth, and B. Perozzi. Grale: Designing networks for graph learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2523–2532, 2020.

[28] H. Hsu, N. Martinez, M. Bertran, G. Sapiro, and F. P. Calmon. A survey on statistical, information, and estimation—theoretic views on privacy. *IEEE BITS the Information Theory Magazine*, 1(1):45–56, 2021.

[29] S. Ji, P. Mittal, and R. Beyah. Graph data anonymization, de-anonymization attacks, and de-anonymizability quantification: A survey. *IEEE Communications Surveys & Tutorials*, 19(2):1305–1326, 2016.

[30] B. Kenig and T. Tassa. A practical approximation algorithm for optimal k-anonymity. *Data Mining and Knowledge Discovery*, 25(1):134–168, 2012.

[31] P. Laperdrix, N. Bielova, B. Baudry, and G. Avoine. Browser fingerprinting: A survey. *ACM Transactions on the Web*, 14(2), 2020.

[32] K. LeFevre, D. J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *22nd International conference on data engineering (ICDE'06)*, pages 25–25. IEEE, 2006.

[33] A. Machanavajjhala, X. He, and M. Hay. Differential privacy in the wild: A tutorial on current practices & open challenges. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1727–1730, 2017.

[34] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[35] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 223–228, 2004.

[36] Ł. Olejnik, G. Acar, C. Castelluccia, and C. Diaz. The leaking battery. In *Data Privacy Management, and Security Assurance*, pages 254–263. Springer, 2015.

[37] H. Park and K. Shim. Approximate algorithms with generalizing attribute values for k-anonymity. *Information Systems*, 35(8):933–955, 2010.

[38] A. Roy Chowdhury, C. Wang, X. He, A. Machanavajjhala, and S. Jha. Crypte: Crypto-assisted differential privacy on untrusted servers. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 603–619, 2020.

[39] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[40] G. Smith. On the foundations of quantitative information flow. In *International Conference on Foundations of Software Science and Computational Structures*, pages 288–302. Springer, 2009.

[41] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[42] L. Sweeney. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl. Based Syst.*, 10(5):557–570, 2002.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

[44] S. Verdú. α-mutual information. In *2015 Information Theory and Applications Workshop (ITA)*, pages 1–6, 2015.

[45] V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in privacy preserving data mining. *ACM Sigmod Record*, 33(1):50–57, 2004.

[46] W. Wang, L. Ying, and J. Zhang. On the relation between identifiability, differential privacy, and mutual-information privacy. *IEEE Transactions on Information Theory*, 62(9):5018–5029, 2016.

[47] G. Wu, X. Xia, and Y. He. Extending differential privacy for treating dependent records via information theory. *CoRR*, abs/1703.07474, 2017.

[48] M. Yang, L. Lyu, J. Zhao, T. Zhu, and K. Lam. Local differential privacy and its applications: A comprehensive survey. *CoRR*, abs/2008.03686, 2020.

A Matching model accuracy bounds

In this section, we provide an information theoretic upper bound on the accuracy that any attacker can achieve in the full-information setting. Recall that in the random-user setting, we sample a representation matrix $\mathbf{P}$ from the distribution $\mathcal{D}$, the attacker formulates a (possibly randomized) prediction rule $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$ based on their knowledge of $\mathbf{P}$. Then we let $\pi : [n] \to [n]$ be a permutation of $[n]$ chosen uniformly at random, $O_{1:n} \in O^n$ be a vector of independent observations with $O_i \sim \mathbf{P}[\pi(i), :]$, and define the accuracy random variable by $\mathrm{Acc}_{\mathrm{M}}(\varphi_M) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\{\varphi_M^i(O_{1:n}) = \pi(i) \,\middle|\, \mathbf{P}\right]$. Our accuracy bound for the matching setting is given below:

Lemma 4. *Let $\mathbf{P} \sim \mathcal{D}$ be a random representation matrix and let $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$ be the matching rule constructed by an attacker. Then with probability one over $\mathbf{P} \sim \mathcal{D}$, we have*

$$\mathrm{Acc}_M(\varphi_M) \leq \frac{m}{n} - \frac{1}{n}\sum_{o\in\mathcal{O}}\prod_{i\in\mathcal{I}}(1 - \mathbf{P}[i, o]).$$

Proof. Let $\mathbf{P}$ be sampled from $\mathcal{D}$ and $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$ be the matching rule constructed by the attacker based on their knowledge of $\mathbf{P}$. Next, let $O^*_{1:n}$ be independent representations with $O^*_i \sim \mathbf{P}[i,:]$, let $\pi : [n] \to [n]$ be a permutation of $[n]$ chosen uniformly at random, and define $O_{1:n}$ by $O_i = O^*_{\pi(i)}$. Next, for each observation $o \in \mathcal{O}$, let $S_o = \{i : O_i = o\}$ denote the set of indices i for which $O_i = o$. Since π is uniformly random and $O_1, \ldots, O_n$ are independent with $O_i \sim \mathbf{P}[\pi(i),:]$, we have that

$$\begin{aligned}
&\mathrm{Acc}_{\mathrm{M}}(\varphi_M) \\
&= \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\{\varphi^i_M(O_{1:n}) = \pi(i)\} \,\middle|\, \mathbf{P}\right] \\
&= \mathbb{E}\left[\frac{1}{n}\sum_{o\in\mathcal{O}}\sum_{i\in S_o}\mathbb{P}(\varphi^i_M(O_{1:n}) = \pi(i) \,|\, O_{1:n}, O^*_{1:n}, \mathbf{P}) \,\middle|\, \mathbf{P}\right],
\end{aligned}$$

where the second equality follows from breaking the sum over i into a sum over $o \in \mathcal{O}$ and $i \in S_o$, and adding an inner expectation conditioned on $O_{1:n}$, $O^*_{1:n}$, and $\mathbf{P}$. The key idea is that, conditioned on $O_{1:n}$ and $O^*_{1:n}$, the permutation π is still random, but $\varphi^i_M(O_{1:n})$ is fixed, which implies that $\varphi^i_M(O_{1:n})$ cannot be correct with too large of a probability. For any $i \in S_o$, we have that

$$\mathbb{P}(\pi(i) = j \,|\, O_{1:n}, O^*_{1:n}, \mathbf{P}) = \frac{\mathbb{I}\{O^*_j = o\}}{|S_o|}.$$

With this, we have

$$\begin{aligned}
\mathrm{Acc}_{\mathrm{M}}(\varphi_M) &= \mathbb{E}\left[\frac{1}{n}\sum_{o\in\mathcal{O}:S_o\neq\emptyset}\sum_{i\in S_o}\frac{\mathbb{I}\{O^*_{\varphi^i_M(O_{1:n})} = o\}}{|S_o|} \,\middle|\, \mathbf{P}\right] \\
&\leq \mathbb{E}\left[\frac{1}{n}\sum_{o\in\mathcal{O}:|S_o|\neq 0} 1 \,\middle|\, \mathbf{P}\right] \\
&= \mathbb{E}\left[\frac{1}{n}\sum_{o}\mathbb{I}\{S_o \neq \emptyset\} \,\middle|\, \mathbf{P}\right].
\end{aligned}$$

This final expression is the number of unique representations that the attacker observed, divided by the number of users n. Intuitively, this bound follows from the fact that for all the users that generated the same observation, the expected number of correct guesses of the attacker is at most one.

To finish the proof, we compute the expected number of distinct representations that the attacker will observe. We have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{o}\mathbb{I}\{S_o \neq \emptyset\} \,\middle|\, \mathbf{P}\right] \\
&\qquad = \sum_{o}\mathbb{P}(S_o \neq \emptyset \,|\, \mathbf{P}) \\
&\qquad = \sum_{o}\left(1 - \prod_{i=1}^{n}(1 - \mathbf{P}[i,o])\right).
\end{aligned}$$

which implies the statement of the Lemma. □

A.1 Connections between the random-user and matching models

In this section we study connections between the random-user and matching settings. In particular, we show that the matching setting is no harder for the attacker than the random-user setting: we prove that any attacker in the random-user setting can be modified to achieve the same accuracy in the matching setting. Next, we show that there exist representation probability matrices $\mathbf{P}$ such that an optimal attacker in the matching setting can do strictly better than the optimal attacker in the random-user setting.

Lemma 5. *Let $\mathbf{P} \sim \mathcal{D}$ and $\varphi_R : \mathcal{O} \to \mathcal{I}$ be any (possibly randomized) attacker prediction rule for the random-user setting. Define $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$ by $\varphi_M(O_{1:n}) = (\varphi_R(O_1), \ldots, \varphi_R(O_n))$. Then with probability one over $\mathbf{P}$, we have $\mathrm{Acc}_M(\varphi_M) = \mathrm{Acc}_R(\varphi_R)$.*

Proof. Intuitively, the matching rule φ_M applies the random-user rule independently for each representation vector in $O_{1:n}$, and the expected fraction of entries it will predict correctly is equal to the expected accuracy of φ_R in the random-user setting. Formally, let $\pi : [n] \to [n]$ be the random permutation used in the matching setting. Then we have

$$\begin{aligned}
&\mathrm{Acc_M}(\varphi_M) = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\left\{\varphi_M^i(O_{1:n}) = \pi(i)\right\} \,\middle|\, \mathbf{P}\right] \\
&= \sum_{i=1}^{n} \frac{1}{n}\mathbb{P}(\varphi_M^i(O_{1:n}) = \pi(i) \,|\, \mathbf{P}) \\
&= \sum_{i=1}^{n} \frac{1}{n}\mathbb{P}(\varphi_R(O_i) = \pi(i) \,|\, \mathbf{P}) \\
&= \mathrm{Acc_R}(\varphi_R),
\end{aligned}$$

where the final equality follows from the fact that the pair (I, O) with $I = \pi(i)$ and $O = O_i$ is distributed identically to $I \sim \mathrm{Uniform}(n)$ and $O \sim \mathbf{P}[I, :]$, since when π is a random permutation, $\pi(i)$ is randomly chosen uniformly at random from $[n]$. □

Next, we construct a distribution $\mathcal{D}$ over representation matrices $\mathbf{P}$ such that an attacker in the matching setting can have a constant factor higher accuracy than the best attacker in the random-user setting.

Lemma 6. *For any even number of users n, there exists a representation space $\mathcal{O}$ of size $m = \frac{3n}{2}$ and a distribution $\mathcal{D}$ over representation matrices $\mathbf{P} \in \mathbb{R}^{n \times m}$ such that: with probability one, every $\varphi_R : \mathcal{O} \to \mathcal{I}$ has $\mathrm{Acc}_R(\varphi_R) \leq \frac{3}{4}$ in the random-user setting, and there exists a rule $\varphi_M : \mathcal{O}^n \to \mathcal{I}^n$ such that $\mathrm{Acc}_M(\varphi_M) = \frac{7}{8}$ in the matching setting.*

Proof. For simplicity, we construct $\mathcal{D}$ as a distribution supported on a single representation matrix $\mathbf{P}$. First, consider the case where we have only $n = 2$ users, the representation space is $\mathcal{O} = \{u_1, u_2, a\}$, and the representation probability matrix is defined by

$$\mathbf{P} = \begin{array}{c} \\ \text{User 1} \\ \text{User 2} \end{array} \begin{array}{c} \begin{array}{ccc} u_1 & u_2 & a \end{array} \\ \begin{pmatrix} 1/2 & 0 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix} \end{array}.$$

We have $\frac{1}{n}\|\mathbf{P}\|_{\infty,1} = 3/4$, and it follows that no prediction rule φ_R can achieve accuracy higher than $3/4$ in the random-user setting. However, in the matching setting, if the attacker observes at least one of $\{u_1, u_2\}$, this is sufficient for perfectly matching the users, since u_i is only ever generated by user i. When both users generate the ambiguous representation a, the attacker still has a $1/2$ chance to correctly identify the users (e.g., by predicting a random permutation). The probability that both users generate representation a is $1/4$, and the probability that at least one of u_1 or u_2 is generated is $3/4$. It follows an attacker in the matching setting can achieve:

$$\begin{aligned}
\mathrm{Acc_M}(\varphi_M) &= 1 \cdot \mathbb{P}(u_1 \text{ or } u_2 \text{ observed}) + \frac{1}{2} \cdot \mathbb{P}(\text{only } a \text{ observed}) \\
&= 1 \cdot \frac{3}{4} + \frac{1}{2} \cdot \frac{1}{4} = \frac{7}{8}.
\end{aligned}$$

To extend this example to any even number of users, we create $n/2$ copies of the 2-user problem as follows: Let $\mathcal{O} = \{u_1, \ldots, u_n\} \cup \{a_1, \ldots, a_{n/2}\}$ and define the representation probability matrix by

$$\mathbf{P}[i, o] = \begin{cases} 1/2 & \text{if } o = u_i \text{ or } o = a_{\lceil i/2 \rceil}. \\ 0 & \text{otherwise.} \end{cases}$$

Then $\mathbf{P}$ has $\frac{3n}{2}$ columns and the maximum value in each column is $1/2$. It follows that $\frac{1}{n}\|\mathbf{P}\|_{\infty,1} = \frac{1}{n} \cdot \frac{3n}{2} \cdot \frac{1}{2} = \frac{3}{4}$. On the other hand, for any even index i, we know that there are exactly two entries in $O_{1:n}$ in the set $\{u_{i-1}, u_i, a_{i/2}\}$, and that these entries must correspond to users $i-1$ and i (but we do not know the order). When the attacker attempts to identify users $i-1$ and i, they are faced exactly with the two-user problem described above, and their expected accuracy for users $i-1$ and i is $7/8$. Averaging over the $n/2$ pairs of users, their overall accuracy is also $7/8$. □

B Relationships between re-id risk and other notions of privacy

In this section we continue the discussion initiated on section 5 discussing the relationship to other privacy notions.

B.1 k-anonymity

A process that releases anonymized data about a collection of users is said to be k-anonymous if the information released for each user cannot be distinguished from at least $k-1$ other users who also appear in the release. We say that a distribution $\mathcal{D}$ over representation matrices is k-anonymous if, with probability one over $\mathbf{P} \sim \mathcal{D}$, every row of $\mathbf{P}$ is a one-hot vector and appears at least k times. That is, each user i is assigned a representation $o_i \in \mathcal{O}$ that is shared with at least $k-1$ other users, and their row of $\mathbf{P}$ is given by $\mathbf{P}[i, o] = \mathbb{I}\{o = o_i\}$.

The following result shows that k-anonymity is sufficient to limit an attacker's accuracy to $1/k$ in the random-user setting.

Lemma 7. *Let $\mathcal{D}$ be any distribution that satisfies k-anonymity, let $\mathbf{P} \sim \mathcal{D}$, and φ_R be an attacker's prediction rule in the random user setting. Then*

$$\mathrm{Acc}_R(\varphi_R) \leq \frac{1}{k}.$$

Proof. Let $\mathbf{P} \sim \mathcal{D}$ and $o_1, \ldots, o_n \in \mathcal{O}$ be the corresponding representations assigned to each user. For each representation $o \in \mathcal{O}$, let $\mathcal{I}_o = \{i \in \mathcal{I} \mid o_i = o\}$ denote the set of users assigned that representation. From the k-anonymity condition, we are guaranteed that either $|\mathcal{I}_o| = 0$ or $|\mathcal{I}_o| \geq k$. It follows that there are at most n/k observations o for which $|\mathcal{I}_o| > 0$. This implies that

$$\|\mathbf{P}\|_{\infty,1} = \sum_{o \in \mathcal{O}} \max_{i \in \mathcal{I}} \mathbf{P}[i, o] = \sum_{o \in \mathcal{O}} \mathbb{I}\{|\mathcal{I}_o| > 0\} \leq \frac{n}{k}.$$

By Corollary 1, it follows that $\mathrm{Acc}(\varphi_R) \leq \frac{1}{k}$, as required. □

B.2 LDP and k-anonymity are not necessary conditions for low re-identification risk

In the previous two subsections we showed that, for appropriate parameter settings, (ϵ, δ)-LDP and k-anonymity both imply that an attacker in the random-user setting has low accuracy. In this section, we show that neither condition is necessary.

Lemma 8. *There exist distributions $\mathcal{D}$ such that with probability one, every attacker has $\mathrm{Acc}_R(\varphi_R) \leq \frac{2}{n}$ and $\mathcal{D}$ is not (ϵ, δ)-LDP unless $\delta = 1$ and not k-anonymous for any k.*

Proof. Let $\mathcal{O} = \{1, 2\}$ and let $\mathbf{P}$ have entries given as follows: for each user $i \in [n]$, define

$$\mathbf{P}[i, 1] = 1 - \frac{(i-1)}{n-1} \qquad \text{and} \qquad \mathbf{P}[i, 2] = \frac{(i-1)}{n-1}.$$

Not all of the rows of $\mathbf{P}$ are one-hot, so it does not satisfy the k-anonymity requirement. Next, we have that $\mathbf{P}[1, 2] = 0$ while $\mathbf{P}[n, 2] = 1$, which implies that $\mathbf{P}$ only satisfies the (ϵ, δ)-LDP constraint when $\delta = 1$. Finally, we have that $\|\mathbf{P}\|_{\infty,1} = 2$ and by Corollary 1 it follows that $\mathrm{Acc_R}(\varphi_R) \leq \frac{2}{n}$. □

B.3 Mutual Information

Another view of re-identification risk can be obtained from the field of information theory. Given a joint pair of random variables (X, Y), we are interested in measuring how much *information* does Y encode about X. For our random user model, this can be translated to measuring how much information the representation O provides about the identity random variable I. This concept is formalized by the conditional mutual information [14] $\text{MI}(I; O|\mathbf{P})$ for the full information scenario and by $\text{MI}(I; O|W)$ for the partial-information scenario. This metric was in fact used in prior work [22] to quantify the re-identification risk of the Topics API.

Lemma 9. *Under the random-user model we have*

$$\text{Acc}_R(\varphi_R) \leq \frac{1 + MI(I; O|\mathbf{P})}{\log(n)}\,, \tag{2}$$

It is worth noticing that the dependency on the number of users n here is logarithmic as opposed to that of Lemma 1 where the dependency is linear. This is an exponential improvement and demonstrates that our framework can better capture re-identification risks.

B.4 Quantitative information flow

Quantitative information flow [40, 3] (QIF) is a different framework for analyzing the privacy vulnerability of a system. QIF is specified by a space of secrets $\mathcal{S}$, an output space $\mathcal{O}$, a (possibly randomized) channel $C\colon \mathcal{S} \to \mathcal{O}$ assumed to be known to an adversary, and a gain function $g\colon \mathcal{W} \times \mathcal{S} \to \mathbb{R}$, where $\mathcal{W}$ is an adversary's space of strategies that may coincide with $\mathcal{S}$ depending on the scenario.

QIF assumes there is a secret s sampled from a known distribution π and that the adversary observes $o = C(s)$. The goal of the adversary is, given o, to learn about s. The gain function may then capture the reward of an adversary predicting secret s', here a reward function $r(s', s)$. Note that an adversary with access to the channel, given an output o, can always predict the secret s' that maximizes their posterior reward

$$R(o) = \max_{s'} \sum_{s \in \mathcal{S}} r(s', s) P(s|o).$$

The privacy vulnerability of a channel may be seen in QIF as the expected posterior reward $\mathbb{E}_\pi[R(o)]$. For our full-information setting, the known representation matrix $\mathbf{P}$ corresponds to the channel, the identity space $\mathcal{I}$ is the secret space and the reward function $r(s', s) = 1$ if $s' = s$ and it is 0 otherwise. That is, the adversary is only rewarded if they predict the correct user in one try. It is known [3] that for this scenario the expected posterior reward corresponds to the so-called Bayes vulnerability and it satisfies:

$$\mathbb{E}[R(o)] = \frac{1}{n} \|\mathbf{P}\|_{\infty,1}$$

That is, our full-information setting is an alternative formulation of QIF as a hypothesis testing framework. To the best of our knowledge, we are not aware of a partial information QIF formulation that fully matches the random-user or matching scenarios although we are actively exploring ways to use advanced concepts in QIF such as the internal fixed-probability choice model [3] to establish a similar connection.

C Other omitted proofs

C.1 Proof of Lemma 1

Proof of Lemma 1. Let I be a user chosen uniformly at random from $\mathcal{I}$ and $O \in \mathcal{O}$ be a representation sampled from $\mathbf{P}[I, :]$. From the law of total probability, we have that

$$\begin{aligned}
&\mathbb{P}(\varphi(O) = I \mid \mathbf{P}) \\
&\quad = \sum_{o \in \mathcal{O}} \sum_{i=1}^{n} \mathbb{P}(\varphi_R(O) = I \mid O = o, I = i, \mathbf{P}) \\
&\quad \cdot \mathbb{P}(O = o, I = i \mid \mathbf{P}) \\
&\quad = \frac{1}{n} \sum_{o \in \mathcal{O}} \sum_{i=1}^{n} \mathbf{A}[i, o] \cdot \mathbf{P}[i, o],
\end{aligned}$$

where the double sum is equal to the Frobenius inner product of $\mathbf{P}$ and $\mathbf{A}$, which can also be written as $\operatorname{tr}(\mathbf{P}\mathbf{A}^\top)$. □

C.2 Proof of Corollary 1

Proof of Corollary 1. Let $\varphi_R : \mathcal{O} \to \mathcal{I}$ be any attacker prediction rule and let $\mathbf{A} \in [0, 1]^{n \times m}$ be its matrix representation. By Lemma 1, the following holds with probability one:

$$\begin{aligned}
&\mathrm{Acc_R}(\varphi_R) \\
&= \frac{1}{n} \sum_{o \in \mathcal{O}} \sum_{i=1}^{n} \mathbf{A}[i, o]\mathbf{P}[i, o] \leq \frac{1}{n} \sum_{o \in \mathcal{O}} \sum_{i=1}^{n} \mathbf{A}[i, o] \cdot \max_{j \in [n]} \mathbf{P}[j, o] \\
&= \frac{1}{n} \sum_{o \in \mathcal{O}} \max_{i \in [n]} \mathbf{P}[i, o] = \frac{1}{n} \|\mathbf{P}\|_{\infty,1},
\end{aligned}$$

where the second last equality follows from the fact that for any $o \in \mathcal{O}$, we have $\sum_{i=1}^{n} \mathbf{A}[i, o] = 1$.

Finally, the inequality in the above derivation holds with equality whenever, for each representation $o \in \mathcal{O}$ and user $i \in \mathcal{I}$, $\mathbf{A}[i, o] > 0$ implies that $\mathbf{P}[i, o] = \max_j \mathbf{P}[j, o]$. An attacker in the full-information setting can design $\mathbf{A}$ so that this holds, which implies they are able to achieve this accuracy exactly. In other words, if the attacker designs φ_R^* so that $\varphi_R^*(o)$ is always some user i with the maximum probability of generating representation o, then $\mathrm{Acc_R}(\varphi_R) = \frac{1}{n}\|\mathbf{P}\|_{\infty,1}$. □

C.3 Proof of Lemma 2

Proof of Lemma 2. Let $\mathbf{A} \in [0, 1]^{n \times m}$ be the matrix-representation of the attacker's W-measurable prediction rule φ_R. With probability one, we have that $\mathrm{Acc_R}(\varphi_R) = \frac{1}{n} \operatorname{tr}(\mathbf{P}\mathbf{A}^\top)$. From this, it follows that

$$\mathbb{E}\big[\mathrm{Acc_R}(\varphi_R) \mid W\big] = \mathbb{E}\left[\frac{1}{n} \operatorname{tr}(\mathbf{P}\mathbf{A}^\top) \,\middle|\, W\right] = \frac{1}{n} \operatorname{tr}(\mathbb{E}[\mathbf{P} \mid W]\mathbf{A}^\top),$$

where the final equality follows from the fact that $\mathbf{A}$ is W-measurable. Next, since $\mathbb{E}[\mathbf{P} \mid W]$ is a row-stochastic matrix, the same argument as in Corollary 1, it follows that $\frac{1}{n} \operatorname{tr}(\mathbb{E}[\mathbf{P} \mid W]\mathbf{A}^\top) \leq \|\mathbb{E}[\mathbf{P} \mid W]\|_{\infty,1}$, as required. To find the prediction rule φ_R^* we use the same argument as in Corollary 1 and define a matrix $\mathbf{A}$ such that $\mathbf{A}[i, o] > 0$ implies $i \in \operatorname{argmax}_j \mathbb{E}[\mathbf{P} \mid W]$. □

C.4 Proof of Lemma 3

Proof of Lemma 3. Let $\mathbf{P} \sim \mathcal{D}$ be a sampled representation matrix and partition $\mathcal{O}$ into sets $\mathcal{O}_1, \ldots, \mathcal{O}_n$, where $\mathcal{O}_i$ contains all representations that user i generates with higher probability than any other user with ties broken in favor of the user with lower index. That is,

$$\begin{aligned}
\mathcal{O}_i = \{o \in \mathcal{O} \ &: \text{for all } j \neq i, \mathbf{P}[i, o] \geq \mathbf{P}[j, o] \\
&\text{and if } \mathbf{P}[i, o] = \mathbf{P}[j, o] \text{ then } i < j\}.
\end{aligned}$$

Then we have that

$$\|\mathbf{P}\|_{\infty,1} = \sum_{o\in\mathcal{O}} \max_{i\in\mathcal{I}} \mathbf{P}[i,o] = \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}_i} \mathbf{P}[i,o]. \tag{3}$$

Now suppose that $\mathcal{D}$ is (ϵ,δ)-LDP. Then we have that:

$$\|\mathbf{P}\|_{\infty,1} = \sum_{o\in\mathcal{O}} \max_{i\in\mathcal{I}} \mathbf{P}[i,o] \leq \sum_{o\in\mathcal{O}} (e^{\epsilon}\mathbf{P}[1,o] + \delta) = e^{\epsilon} + m\delta.$$

At the same time, from (3) we have that

$$\|\mathbf{P}\|_{\infty,1} = \sum_{i\in\mathcal{I}} \sum_{o\in\mathcal{O}_i} \mathbf{P}[i,o] \leq \sum_{i\in\mathcal{I}} \left(\delta + e^{\epsilon} \sum_{o\in\mathcal{O}_i} \mathbf{P}[1,o]\right) = e^{\epsilon} + n\delta.$$

The above arguments show that $\|\mathbf{P}\|_{\infty,1} \leq e^{\epsilon} + \min(n,m)\cdot\delta$. From Corollary 1, it follows that $\mathrm{Acc_R}(\varphi_R) \leq \frac{e^{\epsilon}+\min(n,m)\delta}{n}$. □

C.5 Proof of Theorem 1

Proof. By Lemma 2 we know that, given a representation $o = (o_1,\ldots,o_r)$, the optimal attacker selects the user that maximizes

$$\max_i \mathbb{E}_{\mathcal{D}}\left[P[i,o]|W\right] = \max_i \prod \mathbb{E}_{\mathcal{D}_s}\left[P_s[i,o^s]|W_i^s\right], \tag{4}$$

where the last equality follows by the independence of representations across time and users. Let us calculate one of the terms in the above expression. Note that since $P_s[i,o^s]$ can only take two values q_{in} and q_{out}, each factor in the above expectation is given by:

$$\begin{aligned}
&\mathbb{E}_{\mathcal{D}_s}\left[P_s[i,o^s]|W_i^s\right] \\
&= \mathcal{D}_s(P_s[i,o^s] = q_{\text{in}}|W_i^s)q_{\text{in}} + \mathcal{D}_s(P_s[i,o^s] = q_{\text{out}}|W_i^s)q_{\text{out}} \\
&= q_{\text{out}} + \mathcal{D}_s(P_s[i,o^s] = q_{\text{in}} \mid W_i^s)(q_{\text{in}} - q_{\text{out}})
\end{aligned} \tag{5}$$

Note that if $W_i^s = o^s$, a straightforward application of Bayes rule and the fact that $\mathbb{P}(W_i^s = o^s|P_s[i,o^s] = q_{\text{in}}) = q_{\text{in}}$ yields

$$\mathcal{D}_s(P_s[i,o^s] = q_{\text{in}} \mid W_i^s) = \frac{q_{\text{in}}p_s[o^s]}{q_{\text{in}}p_s[o^s] + q_{\text{out}}(1-p_s[o^s])}$$

When $W_i^s \neq o^s$ we simply rewrite the conditional expectation by the more explainable $\mathbb{P}(o^s \in S_i^s|W_i^s \in S_i^s)$. Plugging this expression in (5) we see that

$$\begin{aligned}
&E_{\mathcal{D}_s}\left[P_s[i,o^s]|W_i^s\right] \\
&= \left(q_{\text{out}} + \frac{(q_{\text{in}} - q_{\text{out}})q_{\text{in}}p_s[o^s]}{q_{\text{out}} + (q_{\text{in}} - q_{\text{out}})p_s[o^s]}\right)\mathbb{I}_{W_i^s = o^s} \\
&+ (q_{\text{out}} + (q_{\text{in}} - q_{\text{out}})\mathbb{P}(o^s \in S_i^s|W_i^s \in S_i^s))\,\mathbb{I}_{W_i^s \neq o^s}
\end{aligned}$$

The result follows by replacing this expression in (4) and taking the logarithm of the product. □

Proof Of lemma 10. Let $\mathbf{W}$ be a random variable sampled as follows. Sample a set of top topics $S \sim \mathcal{D}_s$. Then sample $\mathbf{W}$ according to

$$\mathbb{P}(\mathbf{W} = o|S) = \begin{cases} q_{\text{in}} & o \in S \\ q_{\text{out}} & o \notin S \end{cases}$$

It is very easy to see that the collection (W_i^s) is an i.i.d. sample from random variable $\mathbf{W}$. Moreover we have that

$$\begin{aligned}
\mathbb{P}(\mathbf{W} = o) &= \sum_{S\,:\,o\in S} \mathbb{P}(\mathbf{W} = o|S)\mathcal{D}_s(S) + \sum_{S\,:\,o\notin S} \mathbb{P}(\mathbf{W} = o|S)\mathcal{D}_S(s) \\
&= q_{\text{in}}p_s[o] + (1-p_s[o])q_{\text{out}}
\end{aligned}$$

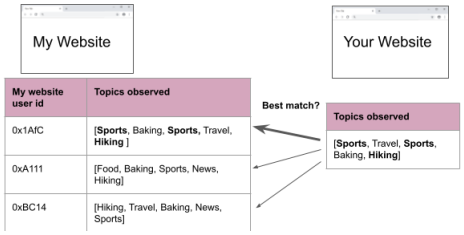

Figure 3: Example of two sites using the sequence of topics returned by the API to potentially re-identify a user across two sites. In the example the user on website 2 shares 3 topics with a user on website 1. This has the most matches across all three users.

Let $Y_i[o] = \frac{\mathbb{I}_{W_i^s = o} - q_{\text{out}}}{q_{\text{in}} - q_{\text{out}}}$ and $\widehat{p}_s[o] = \frac{1}{n}\sum_{i=1}^n Y_i$. Using the fact that $|Y_i[o]| \leq \frac{1}{q_{\text{in}} - q_{\text{out}}}$ and $\mathbb{E}[Y_i[o]] = p_s[o]$, by Hoeffding's inequality we have with probability $1 - \frac{\delta}{N}$:

$$|p_s[o] - \widehat{p}_s[o]| \leq \frac{1}{q_{\text{in}} - q_{\text{out}}}\sqrt{\frac{\log(2N/\delta)}{2n}}.$$

The result follows by using a union bound over all topics o. □

D Theoretical modelling of the Topics API

In this section we continue the analysis of the Topics API initiated in Section 6. First, we provide a detailed specification of the API in Algorithm 1.

Algorithm 1 Topics API.
Input: Topics $\mathcal{T}$, probability to return a random topic p.

On device: Select set S of top 5 most popular topics for this client.
On call GetTopic() from website w on week s and user u:
Seed the random number generator with w, s, u.
Flip coin with heads probability p.
if Heads **then**
  return Element of $\mathcal{T}$ chosen u.a.r.
else
  return Element of S chosen u.a.r.
end if

We now formalize the re-identification risks of the Topics API from the perspective of a website as an attacker colluding with another website to re-identify the user.

Let $N = |\mathcal{T}|$ and $\mathcal{O} = \mathcal{T}^r$. That is if $o \in \mathcal{O}$ then $o = (o^1, \ldots, o^r)$ where o^s corresponds to the topic returned by the API on week s. We begin by modeling a single representation of the Topics API. Based on Algorithm 1, for every user $i \in \mathcal{I}$, the topic returned by the API at round s depends on the set of top topics S_i^s associated with the user. Having fixed this set, the topic selection of the API at round s can be modeled by a matrix $\mathbf{P}_s$ given by:

$$\mathbf{P}_s[i, o^s] = \begin{cases} q_{\text{in}} := (1-p)/5 + p/N & o^s \in S_i^s \\ q_{\text{out}} := p/N & o^s \notin S_i^s \end{cases} \tag{6}$$

It is easy to see that the representation matrix for $o \in \mathcal{O}$ is given by

$$\mathbf{P}[i, o] = \prod_{s=1}^{r} \mathbf{P}[i, o^s]$$

Thus far we have assumed the top set of topics S_i^s to be fixed. However, notice that these are not observable by the websites directly, since they can only see the samples from the top sets. Moreover,

the sets S_i^s are determined by the behaviors of the users which we encode as a probabilistic process in the distribution $\mathcal{D}$.

Formally, the matrix $\mathbf{P}$ is a random variable sampled from a (latent) distribution $\mathcal{D}$ which encodes the way that the top set of topics S_i^s is generated. Clients of the Topics API (i.e., websites) learn about the sampled matrix $\mathbf{P}$ through the observation of samples of the representations in their site.

With this modeling in mind, we see that this re-identification risk can be defined in terms of the (partial) information random-user model:

1. A matrix $\mathbf{P}$ is sampled from $\mathcal{D}$.
2. A sample of representations $W = (W_1, \ldots, W_n) \sim \mathbf{P}$ (W_i corresponds to the representation of user i) is obtained from website 1 and shared with colluding website 2.
3. Given the representation O of a random user on website 2, the attacker must find a W-measurable function to predict the identity of that representation to match that user on website 1.

Modeling the distribution $\mathcal{D}$ A natural question is: how can we model the distribution $\mathcal{D}$? And, can we define the optimal action φ taken by an attacker given their knowledge of $\mathcal{D}$? Below we propose a natural parametric family of distributions $\mathbb{D}$, and an efficient way for the attacker to estimate their parameters based only on the samples known. We also describe a simple to implement optimal attack for this parametric family. In section F.1 we verify empirically that our assumptions on $\mathbb{D}$ closely match the observations from current web traffic.

Note that since $\mathbf{P}$ is fully determined by the top sets S_i^s, we can equivalently define $\mathbb{D}$ as a family of joint distributions over the sequence of random variables $(S_i^s)_{i\in[n],s\in[r]}$. We denote by $\mathbb{P}$ the probability measure induced by $\mathcal{D}$ on S_i^s. A distribution $\mathcal{D}$ belongs to $\mathbb{D}$ if it satisfies the following conditions:

1. All users have the same distribution for variable S_i^s for a given time s (i.e. the top topics of the user are sampled independent and identically distributed from the same distribution, but naturally users can have different sampled top topics.)
2. For every user, samples of top sets are independent across time (but not necessarily identically distributed).

It is not hard to see that the assumptions on $\mathbb{D}$ imply that distribution $\mathcal{D} \in \mathbb{D}$ if and only if there exist distributions $\mathcal{D}_1, \ldots, \mathcal{D}_r$ such that

$$\mathcal{D}(\mathbf{P}) = \prod_{s=1}^{r} \mathcal{D}_s(\mathbf{P}_s).$$

For each distribution $\mathcal{D}_s$ we will be interested in the following

$$\mathcal{D}_s(\mathbf{P}_s[i,o] = q_{\text{in}}) = \mathbb{P}(o \in S_i^s) = p_s[o]$$

These parameters represent the probability that topic o is part of the top 5 topics of a user. The following lemma, which is proved in the Appendix, shows how we can use observations from user representations to estimate these terms.

Lemma 10. *Let $W_1^s, \ldots, W_n^s$ be a sample of topics on website 1, let $N = |\mathcal{T}|$ and $\delta > 0$. Let also*

$$\widehat{p}_s[o] = \frac{1}{n}\sum_{i=1}^{n} \frac{\mathbb{I}_{W_i^s=o} - q_{out}}{q_{in} - q_{out}},$$

where $\mathbb{I}_{W_i^s=o}$ is 1 iff $W_i^s = o$. With probability $1-\delta$ uniformly across all topics $o \in \mathcal{T}$ the following inequality holds:

$$|\widehat{p}_s[o] - p_s[o]| \leq \frac{1}{q_{in} - q_{out}}\sqrt{\frac{\log(2N/\delta)}{2n}}. \tag{7}$$

Let us now proceed to identify the optimal attacker under the assumptions on $\mathcal{D}$. We begin by discussing a natural attacker to derive some intuition into the results of our main theorem of the section.

Example 1 (Hamming attack)**.** *Given a representation $o = (o^1, \ldots, o^r)$ and representations $W = (W_i^s)_{i\in[n],s\in[r]}$ a naïve attacker will assign representation o to a user i such that $|\{s : W_i^s \neq o^s\}$ is minimal. That is, it will naturally try to find the user that minimizes the Hamming distance between W_i and o.*

While the Hamming distance attack matches the intuition that the most likely user that generates a representation is that with the largest overlap on the topic sequence of website 1, one should keep in mind that not all topics are the same. Indeed, a match on a very unpopular topic should be worth more than a match on a popular topic. It is not hard to see that the parameters $p_s[o]$ are a proxy for the popularity of a topic across the population. Thus an optimal attacker should leverage this information. The following theorem, proved in the Appendix, formalizes this intuition.

Theorem 1 (Asymmetric Weighted Hamming Distance Attack)**.** *Given a representation $o = (o^1, \ldots, o^r)$ on website 2, and representations $W = (W_i^s)_{i\in[n],s\in[r]}$ on website 1. An attacker that wants to maximize its accuracy under the partial information setting selects the identity of the user that minimizes the following* asymmetric weighted Hamming distance

$$-\sum_{s:W_i^s=o^s} \log\left(q_{out} + \frac{(q_{in}-q_{out})q_{in}p_s[o^s]}{q_{out}+(q_{in}-q_{out})p_s[o^s]}\right)$$
$$-\sum_{s:W_i^s\neq o^s} \log\left(q_{out} + (q_{in}-q_{out})\mathbb{P}(o^s \in S_i^s | W_i^s \in S_i^s)\right).$$

Notice how the attack in Theorem 1 can be seen as an asymmetric variant of the simple Hamming distance attack described above. It is important to note that while there may likely be several heuristics for utilizing the Topics API signal for re-identification our framework has allowed us to derive — from first principles — a simple optimal algorithm under some basic assumptions. We expect that future work on understanding the privacy of the Topics API can be done by relaxing some of these assumptions.

E Empirical modelling of the Topics API

In this section we provide all material omitted from the main body concerning the empirical analysis.

First we present more detailed way the attack algorithms.

E.1 Attack algorithms

We simulate the following three attacks methods: the **Unweighted Hamming** attack and the **Asymmetric Weighted Hamming** attack as well as the **Neural Network** attack. We now describe each method before presenting our results in section 7.

E.1.1 Unweighted Hamming Attack

This method is an exact implementation of the simple attack presented in Example 1.

E.1.2 Asymmetric Weighted Hamming Attack

This method is a simplified implementation of the attack given in Theorem 1 which is optimal under the assumptions described above.

In our experiments, we further make the approximation of assuming that $\forall W_i^s \neq o_s, \mathbb{P}(o^s \in S_i^s | W_i^s \in S_i^s)$ is a function only depending on o^s. Then we show that $\forall W_i^s \neq o_s, \mathbb{P}(o^s \in S_i^s | W_i^s \in S_i^s) = \frac{4p_s[o^s]}{5-p_s[o^s]}$ (see Lemma 11 for more details). This reduces the parameters of the model to be estimated to only $p_s[o^s]$. Moreover, given that we empirically observe $p_s[o^s]$ to be very close in every period we further assume $p_s[o^s] = p[o^s]$.

E.1.3 Neural Network Attack

In addition to the previous methods, we also implement a heuristic attack method based on a deep neural network. Our method is general and works on an arbitrary embedding representation for the

topics sequence of a user. In section F.1.1 we show how we obtain such embedding from sequence to sequence models while in this section we focus on how to use any embedding for matching users.

We train a deep neural network which takes two sequence embeddings as inputs and outputs a similarity score in $[0, 1]$, indicating the predicted probability that the two sequences are from the same user. Our network structure is similar to that of the Grale infrastructure [27]. The detailed network structure is presented in Figure 5.

Training process. We sample 20 million random users (different from the target set $\mathcal{I}$ used in the re-identification task). For each sampled user u, we simulate a pair of topics sequences where both sequences are from u and we regard it as an example from the class of correct matching. We also create 10 pairs of sequences where the first sequence of each pair is the sequence from u and the second sequence is from a random user $v \neq u$, and we regard each pair as an example from the class of incorrect matching. Each training example is a pair of sequences generated by the above procedure and each sequence is embedded using a sequence model. The training objective of the neural network is to minimize the binary cross entropy loss.

Re-identification inference. Finally, to match a user sample using the neural network, given a user sequence A, we enumerate every sequence B from the target dataset and feed (A, B) to the neural network. We choose the sequence B^* as the re-identification output where the sequence (A, B^*) maximizes the predicted probability given by the neural network.

E.2 Re-identification task for Million Song Dataset and Results

We now present our empirical study of re-identification attacks on the publicly available Million Song Dataset [8]. Similar to the study of Topics API, we focus on the random-user model.

In this dataset, a user is represented by all songs liked by them. The dataset contains 48 million entries on the listening activity of about a million users. The number of distinct songs is about a million.

We simulate a system that outputs a sample of r songs for a user, independently, to generate two different databases. Then, we measure the risk of re-identifying the user across the two datasets, depending on the number r of independent samples observed. The results are reported in Figure 4.

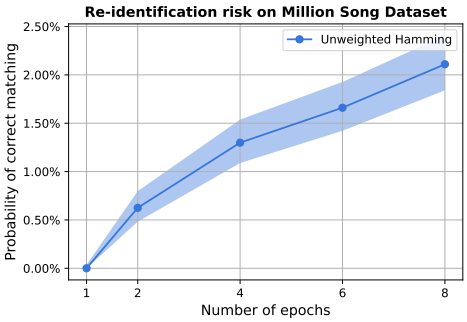

Figure 4: Probability of a correct cross-site match depending on the number of epochs observed. The 95% confidence intervals are reported.

Here we use the Unweighted Hamming Attack for guessing the match of the user (notice that the other two attacks were specifically designed for Topics, so they are not meaningful here). We note that observing 4 independent random songs results in 1% re-identification risk. This is an example of how our framework allows us to assess the risk in such data releases.

F Modeling assumption on topics

In this section we discuss the modeling assumption on the Topics API used in the experiments (Section 7). The results of this section show that we can model an attacker using only the parameters $p_s[o]$.

Assumption 1. *For a fixed round s we assume that the random variable S_i^s representing the top set of users satisfies:*

$$\mathbb{P}(o \in S_i^s | o' \in S_i^s) = \alpha_s(o)$$

for all topics $o' \neq o$.

The above assumption suggests some form of independence between the topics belonging to the top set. The following lemma shows that under this assumption $\alpha_s(o)$ is in fact a simple function of $p_s[o]$.

Lemma 11. *Let S_i^s satisfy Assumption 1. Then*

$$\alpha_s(o) = \frac{4p_s[o]}{5 - p_s[o]}$$

Proof. By proposition 1 we know that:

$$\begin{aligned} 4p_s[o] &= \sum_{o' \neq o} \mathbb{P}((o, o') \in S_i^s) = \sum_{o' \neq o} \mathbb{P}(o \in S_i^s | o' \in S_i^s) P(o' \in S_i^s) \\ &= \alpha_s(o) \sum_{o' \neq o} p_s[o'] = \alpha_s(o)(5 - p_s[o]), \end{aligned}$$

where we have used Bayes rule and Assumption 1 for the second and third equalities respectively. The statement of lemma follows by rearranging terms. □

Proposition 1. *Let o be a fixed topic. The following properties holds for any distribution over the top set S_i^s.*

$$\sum_{o' \neq o} p_s[o'] = 5 - p_s[o] \tag{8}$$

$$\sum_{o' \neq o} \mathbb{P}((o, o') \in S_i^s) = 4p_s[o] \tag{9}$$

Proof. Fix a top set S_i^s of 5 elements. It is then easy to see that

$$\mathbb{I}_{o \in S_i^s} + \sum_{o' \neq o} \mathbb{I}_{o \in S_i^s} = 5 \tag{10}$$

Similarly, we claim that

$$\sum_{o' \neq o} \mathbb{I}_{o \in S_i^s} \mathbb{I}_{o' \in S_i^s} = 4\mathbb{I}_{o \in S_i^s} \tag{11}$$

Indeed, if $o \notin S_i^s$ then the above expression is trivially true as both sides are 0. If, on the other hand, $o \in S_i^s$, then there are exactly 4 other elements in the set so $\sum_{o' \neq o} \mathbb{I}_{o' \in S_i^s} = 4$. The result follows by taking expectation of (11) and (9).

□

F.1 Validation of the Modelling Assumptions: Analysis of Mutual Information and Sequence2Sequence Model

In this section, our aim is to measure the validity of a key assumption made in Theorem 1 which we simulated: that of the cross-time independence of topics.

To do this we measure mutual information $M^* = I(A^1, A^2, \cdots A^r;\ B^1, B^2, \cdots B^r)$, for $A = A^1, A^2, \cdots A^r$ and $B = B^1, B^2, \cdots, B^r$ being the two sequences of topics observed from a given user on two websites. Notice that, as shown in [22] in case of cross-time independence this M^* reduces to the easy to compute $\sum_{s=1}^r I(A^s; B^s)$. By estimating M^* and showing that it is close to $\sum_{s=1}^r I(A^s; B^s)$ we verify the accuracy of the assumption.

In general estimation of M^* is a non-trivial task as the distribution of long sequences of topics has an exponentially growing support. In this section we use advanced ML techniques to tackle this challenge allowing us to extend the previously published work in [22] beyond the analysis of 2 epochs of data.

The rest of the section presents the ML model and how we used it to estimate the mutual information.

Hyperparameter	Transformer	Transformer-LSTM
Encoder	Transformer	Transformer
Decoder	Transformer	LSTM
Attention dropout rate	0.1	0.1
Attention layer size	1,024	1,024
Dropout rate	0.1	0.1
Embedding size	1,024	1,024
MLP dimension	4,096	4,096
Number of attention heads	8	8
Number of encoder layers	6	6
Number of decoder layers	6	8
Decoder Hidden dimension	1,024	1,024
Training batch size	1,536	1,536
Total number of parameters	-	300M

Table 1: Hyper-parameters of the S2S Model Architectures.

F.1.1 Sequence to Sequence Model

We use state-of-the-art sequence to sequence models (S2S) [41] that are currently being used in a variety of machine learning applications ranging from natural language processing [43] to computer vision. As a byproduct of this model we develop an embedding for a topics sequence that we used in the previous section for the Neural Network attack.

First, we introduce formally the concept of sequence-to-sequence model. A sequence-to-sequence (S2S) model [41] assigns a probability to a sequence of target symbols $B^1, B^2, \cdots B^Y$ given a sequence of source symbols $A^1, A^2, \cdots A^X$: $P(B^1 = b^1, B^2 = b^2, \cdots, B^Y = b^y | A^1 = a^1, A^2 = a^2, \cdots, A^X = a^x)$. The general architecture of an S2S model consists of an encoder network which generates an embedding of the source sequence and a decoder network that generates the target sequence conditional on the source sequence. A further refinement is an attention mechanism [5] that allows the decoder to attend to specific tokens of the source sequence when assigning probabilities to each token in the target sequence. A number of architectures have been proposed for the encoder and decoder networks. In this work, we will compare two popular architectures: Transformer [43] and a variant consisting of a Transformer encoder and a Long Short Term Memory (LSTM) decoder [11].

F.1.2 Estimation method

Now, we present how to use S2S models to estimate mutual information.

The mutual information between sequences A and B can be written as a difference of two entropies [14]:

$$I(A^1, A^2, \cdots, A^r; B^1, B^2, \cdots, B^r) = \\ H(B^1, B^2, \cdots, B^r) - H(B^1, B^2, \cdots, B^r | A^1, A^2, \cdots, A^r) \tag{12}$$

Thus if we train two S2S models to estimate $H(B^1, B^2, \cdots B^r)$ and $H(A^1, A^2, \cdots A^r | B^1, B^2, \cdots, B^r)$ respectively, we can compute an estimate of $I(A^1, A^2, \cdots A^r; B^1, B^2, \cdots B^r)$.

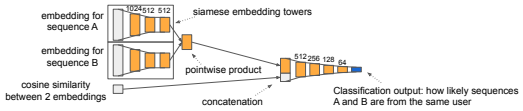

Figure 5: The structure of the network for re-identification attack. The embeddings of each sequence are computed by feeding the sequence to the trained S2S model (Transformer-Only) described in Section F.1 and concatenating the hidden representations of the final encoder layer at each week. The number in the figure indicates the size of each layer. If not specified, each layer is a Fully Connected layer + ReLU activation.

Given a training data set of tuples consisting of a user, and a pair of topics sequences associated with the user on two websites, $A = A^1, A^2, \cdots, A^r$, $B = B^1, B^2, \cdots, B^r$, we estimate a sequence-to-sequence model to predict B given A. Over an unseen test data set consisting of n tuples $\{A_i^1, A_i^2, \cdots A_i^r, B_i^1, B_i^2, \cdots B_i^r\}_{i=1}^n$, we can estimate the conditional entropy of B given A:

$$H(B^1, B^2, \cdots, B^r | A^1, A^2, \cdots, A^r) = \tag{13}$$

$$-\frac{1}{n}\sum_{i=1}^{r} \log P(B_i^1, B_i^2, \cdots B_i^R | A_i^1, A_i^2, \cdots A_i^r) \tag{14}$$

To estimate the unconditional entropy of the target sequence, we replace the source sequence within each tuple in the data set (both training and test) with a single special token \$, (i.e. $A^1, A^2, \cdots, A^r$ is replaced with \$). We can then use the above approach to train a model to estimate the unconditional entropy of the target sequence:

$$H(B^1, B^2, \cdots, B^r) = -\frac{1}{n}\sum_{i=1}^{r} \log P(B_i^1, B_i^2, \cdots B_i^r | \$) \tag{15}$$

F.1.3 Model Details

In our work we experiment with two separate S2S model architectures: either the vanilla Transformer [43] or a variant [11] consisting of a Transformer encoder and an LSTM decoder. (Exact details on the architecture may be found in the above references). For estimating mutual information using a given sequence-to-sequence model architecture, we use the same set of hyper-parameters for both the unconditional and the conditional model (Table 1).

F.1.4 Results

We now present the main set of results from the sequence-to-sequence model using the Transformer architecture. The hybrid architecture achieves a very close but slightly lower mutual information than the Transformer-only model so we omit its results and use the Transformer only embeddings in our re-identification analysis.

We measured the unconditional entropy of a sequence of topics for r epochs and as well as the conditional entropy (of the second sequence of the user) and their difference which is the mutual information. We report the results as bits/epoch. Using the Transformer model trained on $r = 8$ epochs, we observe 6.54, 5.45 and 1.09 bits/epoch of unconditional entropy, conditional entropy and mutual information, respectively. On a single epoch of data we observe 7.59, 6.66, 0.93 bits for unconditional entropy, conditional entropy and mutual information, respectively.

Notice how, on average, we observe about 1.1 bits of mutual information per epoch of observation vs 0.93 bits of a single epoch. This suggests that while there is indeed some information gained by looking at sequences of topics across time, previous observations of the Topics API do not provide significant information about the topic returned in the current epoch. This validates our hypothesis that topics are close to independent across time and the model for the distribution $\mathcal{D}$.