dpUGC: Learn Differentially Private Representation for User Generated Contents

Xuan-Son Vu[1], Son N. Tran[2], Lili Jiang[1]
[1]Database Data Mining Group, Umeå University, Sweden
[2]ICT Discipline, University of Tasmania, Australia.

CICLing Conference, La Rochelle, 2019-04-09

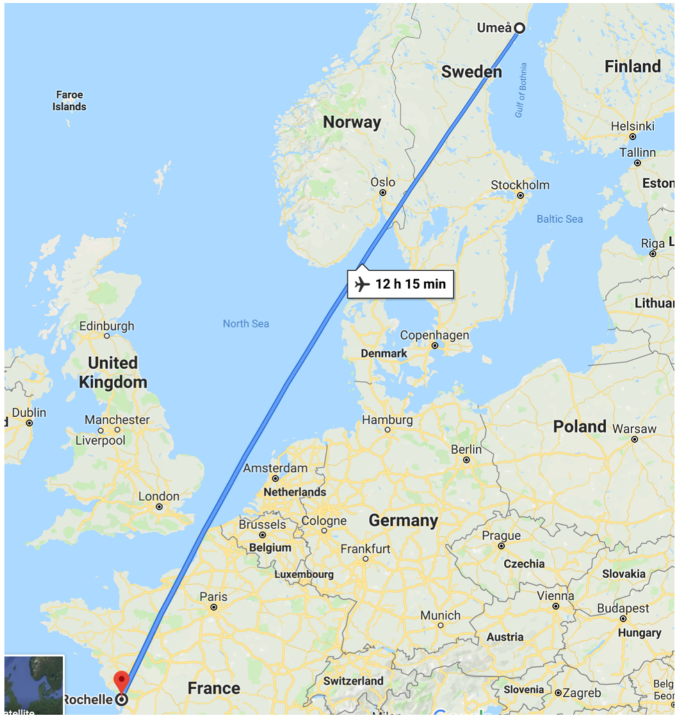
Outline

- Introduction
  - Motivation: UGC data, privacy
- Methodology:
  - Learn differential private embedding on UGC
  - User-level dpUGC
- Experiments, results and discussion
- Conclusions and Future Work

1. Introduction

Who are we?

- Umeå University, Sweden
  - Central north of Sweden
  - http://cs.umu.se



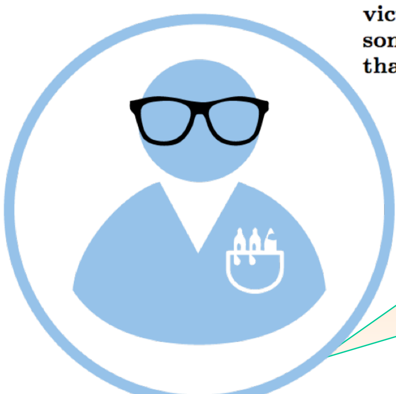
Introduction

- Privacy-leakage in data analysis
  - Narayanan et al. (2008): De-anonymize users of Netflix contest by matching to IMDB users
  - Fredrikson et al. (2015): reveal individual faces from the training data

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

I can attack this model to find who were involved in the study.

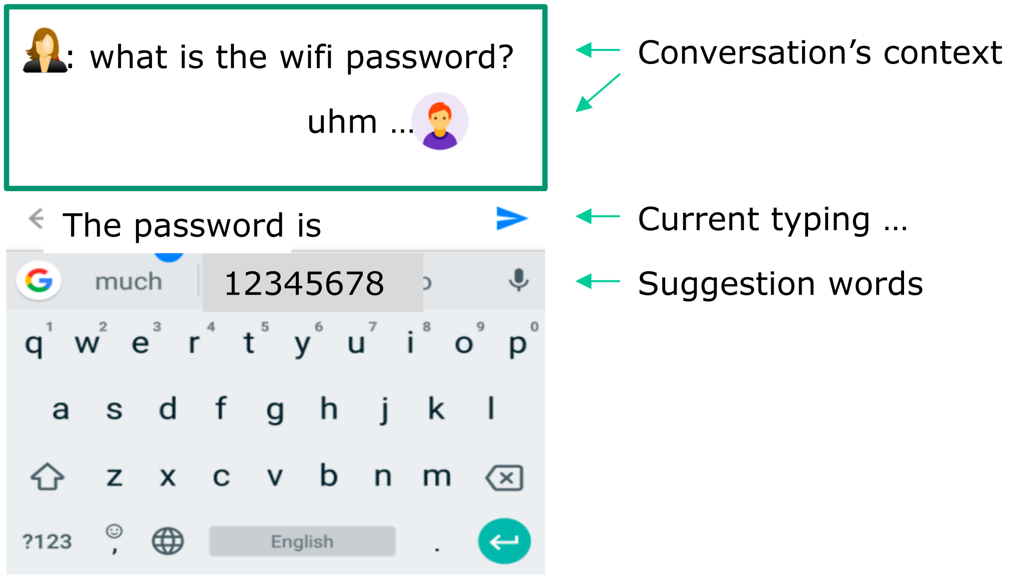
Researcher: publish a model to predict cancer based on genome data.



adverary

Introduction

- Privacy Issues in Text (1/2):
  - *Auto Suggestion* learns from what you **typed**?



  - Medical Text Data:
    - Patient Medical Journals: medical history/logs

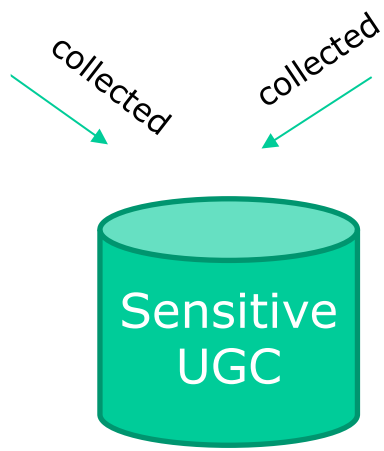
Introduction

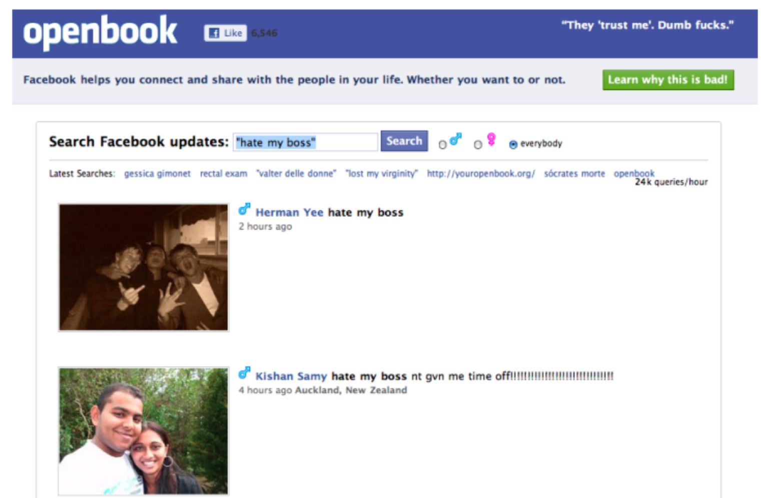
- Privacy Issues in Text (2/2):
  - User Generated Contents (UGC)
    - Any form of **content**: video, blogs, posts, digital images, audio files, and other forms of media
      - Created by consumers or end-**users**
- This work**:**
  - Applied and tested on UGC
  - But works seamlessly on any user-level text data:
    - Personal medical records
    - Personal Longitudinal Dialog (FB messages, Emails, …)
      - E.g., Welch et al., @ CICLing 2019.

Introduction

- Privacy in UGC:
  - Contains so much sensitive information
  - No one dares to share their UGC data

A prank to get sensitive information





Motivation (1/4)

- Sharing pre-trained embeddings:
  - On public text data: e.g., Google News, common crawl
    - Word2Vec, Glove, FastText, Elmo, BERT etc.
  - On private text data?
    - Can we do the same for private pre-trained embeddings?
    - Representation of private-words would otherwise not possible without privacy-guarantee:
      - e.g., disease names, dna2vec, etc.

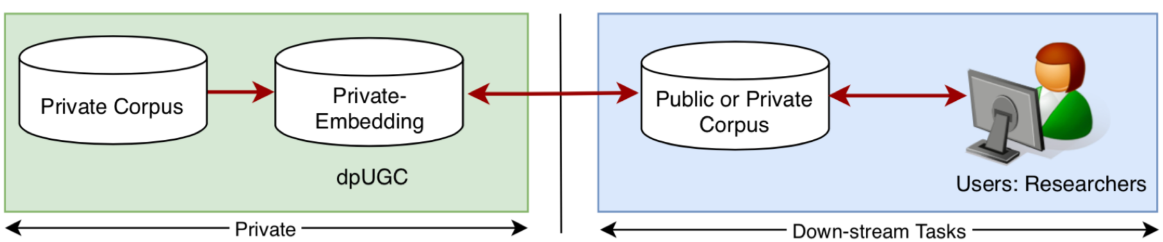


Fig. 2: Overview of our safe-to-share embedding model that can be used to facilitate research on sensitive data with privacy-guarantee.

Motivation (2/4)

- Privacy issues in pre-trained embeddings:
  - "You shall know a word by the company it keeps" (J. R. Firth 1957:11)
    - One of the most successful ideas of modern statistical NLP

Query	Top#1	Top#2	Top#3	Top#4
???	Prof.	NLP	Mexico	CICLing
???	Prof.	NLP	France	CICLing
???	Prof.	NLP	UK	Speakers

"You shall know a person by the company it keeps"

Motivation (2/4)

- Privacy issues in pre-trained embeddings:
  - "You shall know a word by the company it keeps" (J. R. Firth 1957:11)
    - One of the most successful ideas of modern statistical NLP

	Query	Top#1	Top#2	Top#3	Top#4
Alexander	???	Prof.	NLP	Mexico	CICLing
Antoine	???	Prof.	NLP	France	CICLing
Lucia	???	Prof.	NLP	UK	Speakers

"You shall know a person by the company it keeps"

Motivation (3/4)

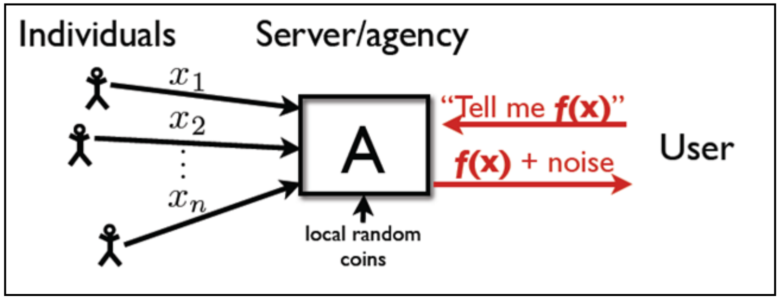
- UGC is good for science:
  - 660 publications work on **myPersonality**, the popular UGC dataset for personality prediction
    - Machine learning model can predict personality better than human.
  - Tons of research work on Twitter/Facebook data on many important topics:
    - Sentiment classification, recommendation, privacy detection, social behavior etc.
    - In fact:
      - **6.7M results** from google scholar mentioned Twitter
      - **6.17M results** from google scholar mentioned Facebook

Motivation (4/4)

- Research Questions?
  - How to learn representation from UGC data while protect user's privacy?
  - How to share embedding models trained on UGC data for other researchers?
  - Will normal differential privacy is enough for embedding models?

2. Methodology

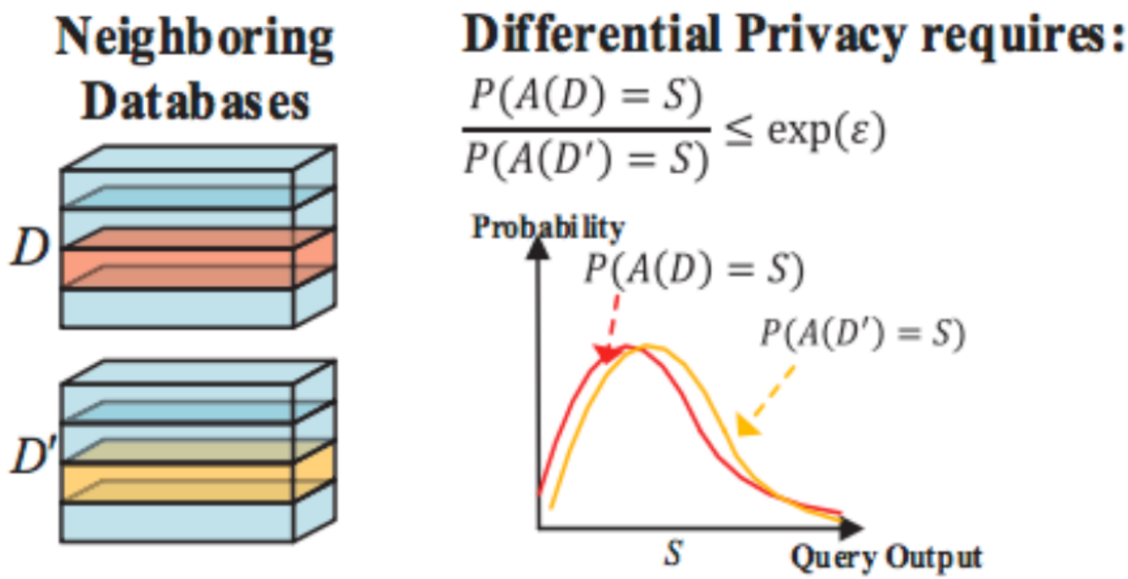
Background (1)



- Privacy-guarantee data analysis
  - Injecting scientific-noise into results [Dwork06]
    - State-of-the-art method by definition
    - Called: differential privacy (DP)
    - Amount of noise controlled by ε ($\downarrow\varepsilon$,$\uparrow$noise)
- Deciding amount of noise
  - Global noise (DP) vs personalized noise (S-PDP)

Background (2)

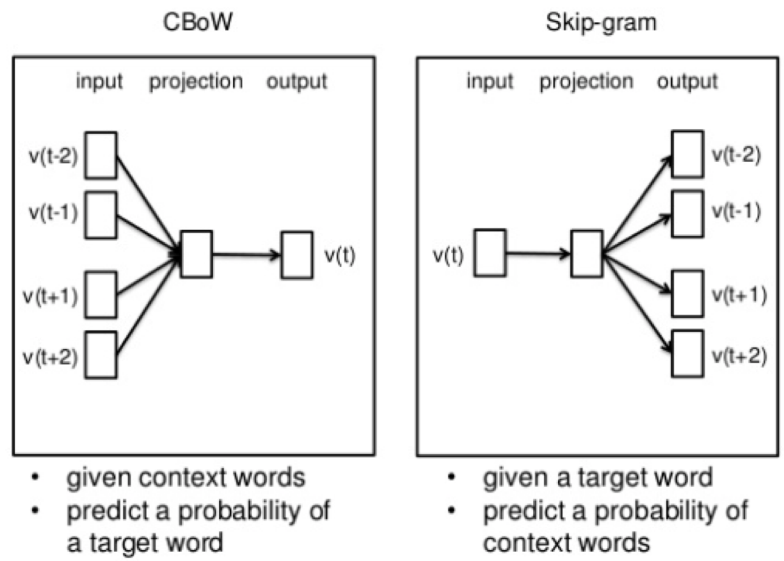
- ε-Differential Privacy (DP):



  – The adversary’s ability to infer the individual’s information is bounded!
    - More or less as a random guess [Stephen Tu ‘13].

Background (3): Word2Vec

- Continuous Bag-of-Words (CBOW) and Skip-gram
  - Similar in performance
- Thousand times faster than Bengio's model.



$$\frac{1}{T}\Sigma_{t=1}^{T}\Sigma_{j\in nb(t)}\ \log\, p(w_j|w_t)$$

Differentially Private (DP-) Embedding

- Adding noise to protect privacy

Require: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta)$, embed dimension k
Ensure: return optimized θ to calculate $W^{(k)}$ - a learned DP-Embedding.

```
// Algorithm 1-a: DP-Embedding
Initialize θ_0 randomly
for all round t = 0, 1, 2, ..., T do
    Take a random sample L_t with sampling probability L_t/N
    Compute gradient
    For each i ∈ L_t, compute g_t(x_i) ← ∇_{θ_0} 𝓛(θ_t, x_i)    // 𝓛 is from (2)
    Add noise
    g̃_t ← (1/L)(Σ_i g̃_t(x_i) + 𝒩(0, σ²C²I)
    Descent
    θ_{t+1} ← θ_t − η_t g̃_t
    𝓜.accum_priv_spending(z)
end for
```

Personalized DP-Embedding

Require: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta)$, embed dimension k
Ensure: return optimized θ to calculate $W^{(k)}$ - a learned DP-Embedding.
// **Algorithm 1-b**: **Personalized DP-Embedding**
Initialize θ_0 randomly
for all round $t = 0, 1, 2, \ldots, T$ **do**
  $K \leftarrow$ (get list of samples from valid users $\mathcal{U}$)
  Take a random sample $L_t \in K$ with sampling probability L_t/K.
  $\mathcal{U}_{L_t} \leftarrow$ the set of users where the sample L_t come from.
  Compute gradient
  For each $i \in L_t$, compute $g_t(x_i) \leftarrow \nabla_{\theta_0} \mathcal{L}(\theta_t, x_i)$ // $\mathcal{L}$ is from (2)
  Add noise
  $\tilde{g}_t \leftarrow \frac{1}{L}(\Sigma_i \tilde{g}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})$
  Descent
  $\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{g}_t$
  $(\epsilon_t, \delta_t) = \mathcal{M}.\text{get\_priv\_spending}(z)$
  Update privacy spending for each user
  for all user $u \in \mathcal{U}_{L_t}$ **do**
    $(\epsilon, \delta)_u \leftarrow (\epsilon, \delta)_u + \frac{(\epsilon_t, \delta_t)}{L}$
    If user u gets out of privacy-budget: $\mathcal{U} \leftarrow \mathcal{U} \setminus \{u\}$
  end for
end for

3. Evaluations

Experimental Settings

- On two criteria:
  - **Word similarity:** a standard measurement for evaluating word embedding models [15].
  - **Data utilities:** preserve privacy when sharing the model for other scholars.

- Datasets:

Table 1: A simple statistics of the myPersonality dataset and Text8 corpus.

Dataset	#users	#documents	#words
myPer (private)	153,727	22,043,394	416,862,367
myPer (public)	250	9,917	144,616
Tex8 corpus	-	-	17,005,207

$$MAP = \frac{\Sigma_{q=1}^{Q} AvgP(q)}{Q}$$

Experiment Design

- Changes in semantic space:
  - Evaluation metric, we used MAP (mean-average-precision):
    - MAP-Word: evaluates the top similar words at word level
    - MAP-Char: evaluates the top similar words at character level
- Regression task (downstream task):
  - E(public): None DP-Embedding
  - E(private): DP-Embedding

$$R_{E(Private)+E(Public)} \geq R_{E(public)}$$

Results #1a: semantic space

Query	Gold model	DP-Embedding (top 4)	MAP (W,C)	Topic
three	four:two:five:seven	zero:one:feeder:nine	(0, 3.814)	Numbers
eight	seven:nine:six:four	cornerback:four:stockholders:zero	(0.5, 0.1347)	Numbers
they	we:there:you:he	morgan:century:contentious:ferroelectric	(0, 0.4237)	Pronouns

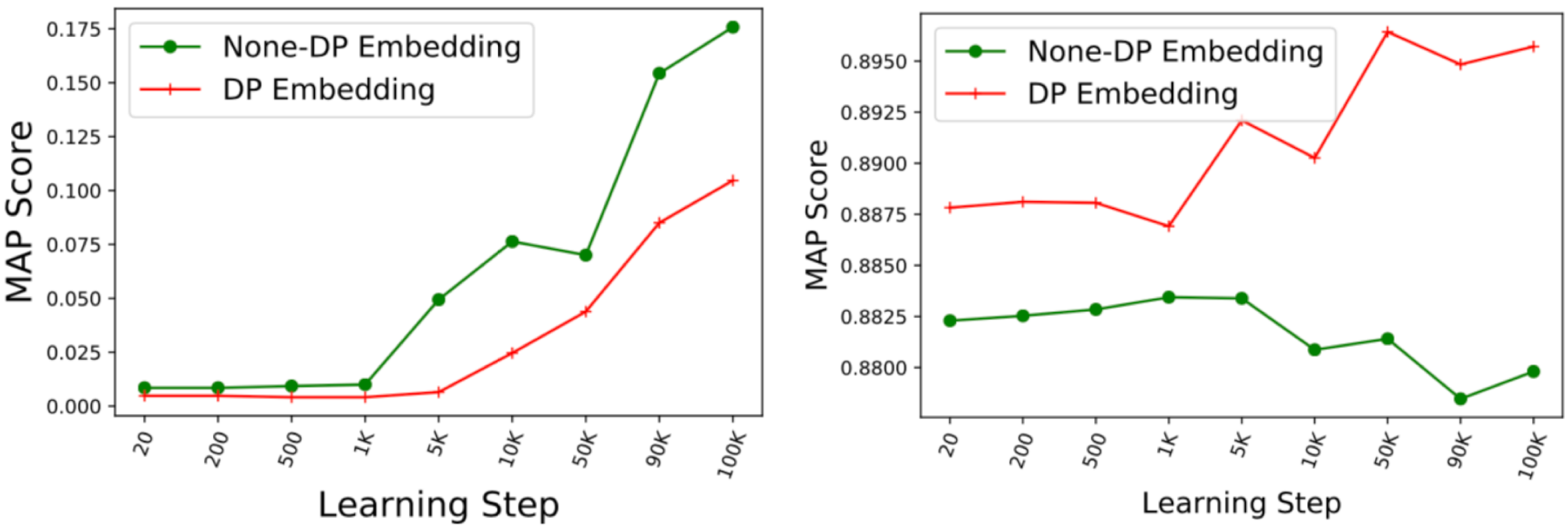
(a) Top 4 on DP-Embedding model

Query	Gold model	Non-DP Embedding (top 4)	MAP (W, C)	Topic
three	four:two:five:seven	one:in:UNK:zero	(0, 0.1288)	Numbers
eight	seven:nine:six:four	integrator:transfection:four:one	(0.33, 0.3561)	Numbers
they	we:there:you:he	that:monorail:it:lesbian	(0, 0.2341)	Pronouns

(b) Top 4 on Non-DP Embedding model

Table 2: Top similar words of DP-Embedding (a), and Non-DP Embedding (b) models given three queries "three", "eight", and "they" at 100K learning step. The second column shows the best results from the Gold model. MAP(W,C) denotes (MAP-Word,MAP-Char).

Results #1b: semantic space



(a) MAP at word level

(b) MAP at character level

Fig. 3: Semantic space changes when learning embedding model with and without differential privacy compared to the *Gold model*. Learning step is number of minibatch steps

Results #2: Downstream tasks

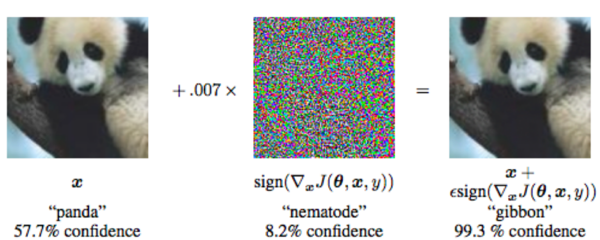
- Results:
  - DP-Embedding gets better or slightly different results than the None-DP Embedding
  - Best at learning step 20 and 500:
    - Better performance with privacy-guarantee (win-win)

LS	SVR			LR			Privacy-Budget $(0.125, \delta)$
	Baseline-SVR	DP-SVR	NoneDP-SVR	Baseline-LR	DP-LR	NoneDP-LR	
20	2.6563	**1.7881**	3.5942	1.2903	**1.2616**	1.2642	0.0184 †
200	2.6563	2.4983	**2.0198**	1.2903	**1.2589**	1.2717	0.0189
500	2.6563	**2.7795**	3.6231	1.2903	**1.2514**	1.2909	0.0197 †
1K	2.6563	3.2146	**2.0206**	1.2903	**1.2611**	1.262	0.0211
5K	2.6563	6.1596	**2.7472**	1.2903	**1.2577**	1.2642	0.0372
10K	2.6563	**1.6396**	3.9155	1.2903	1.2768	**1.2574**	0.0755
50K	2.6563	2.9438	**2.5769**	1.2903	1.2574	**1.2556**	0.5929
90K	2.6563	**2.4033**	2.5175	1.2903	1.2585	**1.258**	0.7681
100K	2.6563	2.6043	**2.0215**	1.2903	**1.2548**	1.262	0.7926

4. Conclusions and Future Work

Conclusions and Future Work

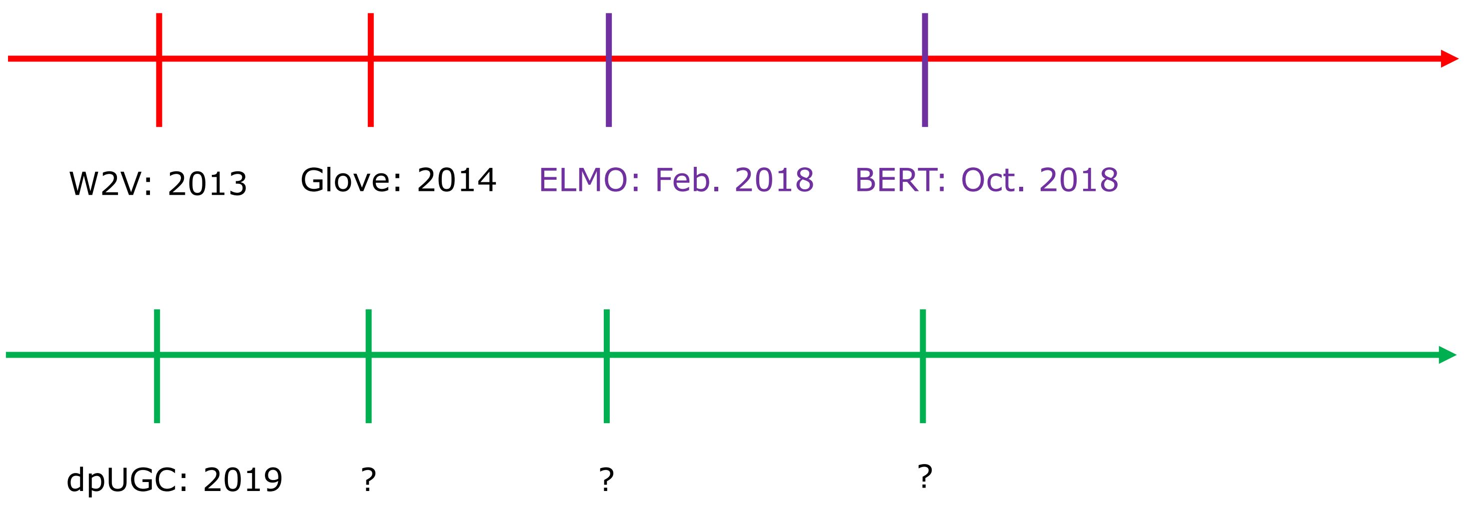
- Key findings:
  - Proposed algorithms for learning differentially private text representation for UGC sharing.
    - Works seamlessly on any personal text data
  - Evaluated the algorithms on a realistic UGC dataset
  - Adding noise to images:



  - Adding noise to word embeddings?
    - Similar to manipulate with different characters

Conclusions and Future Work

- Future Work:



Questions?

E.g., motivation, application, DP …

Code will be available at: https://github.com/sonvx/dpText/