PrIA: A Private Intelligent Assistant

Aruna Balasubramanian, Niranjan Balasubramanian, Supriyo Chakraborty*,
Shashank Jain, and Vivek Tiwari
Stony Brook University, *IBM T. J. Watson Research Center

ABSTRACT

Personalized services such as news recommendations are becoming an integral part of our digital lives. The problem is that they extract a steep cost in terms of privacy. The service providers collect and analyze user's personal data to provide the service, but can infer sensitive information about the user in the process. In this work we ask the question "How can we provide personalized news recommendation without sharing sensitive data with the provider?"

We propose a local private intelligence assistance framework (PrIA), which collects user data and builds a profile about the user and provides recommendations, *all* on the user's personal device. It decouples aggregation and personalization: it uses the existing aggregation services on the cloud to obtain candidate articles but makes the personalized recommendations locally. Our proof-of-concept implementation and small scale user study shows the feasibility of a local news recommendation system. In building a private profile, PrIA avoids sharing sensitive information with the cloud-based recommendation service. However, the trade-off is that unlike cloud-based services, PrIA cannot leverage collective knowledge from large number of users. We quantify this trade-off by comparing PrIA with Google's cloud-based recommendation service. We find that the average precision of PrIA's recommendation is only 14% lower than that of Google's service. Rather than choose between privacy or personalization, this result motivates further study of systems that can provide both with acceptable trade-offs.

1. INTRODUCTION

Personalized intelligence assistance comes at a steep cost to user privacy. Today, there are a range of personalized services including recommending news articles, recommending music, setting up appointments, answering questions, and much more. Most require the user to fork over their data such as click history, search queries, voice commands. While essential for effective personalization, this data can reveal sensitive information about the users which many would like to keep private. Unfortunately, there is no viable alternative today. A privacy-conscious user has to choose between getting these IA services and privacy. In this work, we propose a privacy focused solution for one particular IA service, personalizing news recommendation and discuss issues in scaling the approach to a broader class of IA services.

To illustrate the privacy issues in news recommendation consider Alice, a typical news consumer. She has many varied interests and uses a cloud-based news recommendations service, such as Google News, to follow news from different sources. Alice logs onto Google's service; the service in-turn tracks news articles read by Alice and builds a user profile for her. The cloud-based service also aggregates articles from large number of news sources, and then uses a recommendation system to selects articles for Alice according to her user profile. The user profile built for Alice could reveal sensitive information about her, for example, pertaining to her political preference.

We envision PrIA a *Private* Intelligent Assistant which provides personalized news recommendation similar to services such as Google News. However, different from these cloud-based services, the main idea in PrIA is to obtain aggregated news from the cloud but personalize it locally. PrIA tracks the user activities, builds a profile, and make recommendations to the user all from within the user's personal device and thus does not divulge the user's interactions to the aggregation service. In effect, PrIA decouples news aggregation from personalization.

PrIA trades off knowledge of collective user behavior for more personalized local knowledge. Typically, intelligent assistant services gather data from a large number of users to build global models on the population at large. These global models are known to significantly improve personalization using techniques such as collaborative filtering [6]. By virtue of being private, PrIA does not have access to the global user model. However, we posit that PrIA can build a richer profile about the individual user, by combining textual data across the user's email, social networks, browsing activities, and other contextual information. Cloud providers are limited to a smaller slice pertaining to interactions on their specific service or on their ecosystem. For example, the user's Facebook posts on the Facebook application cannot be used by Google for their recommendation. By building a better user profile with more data, we hypothesize that PrIA can improve its personalization despite not having access to global user behavior.

We show the feasibility of a local news recommendation system through a proof-of-concept implementation. This

HotMobile '17, February 21-22, 2017, Sonoma, CA, USA
© 2017 ACM. ISBN 978-1-4503-4907-9/17/02...$15.00
DOI: http://dx.doi.org/10.1145/3032970.3032988

PrIA implementation logs information about the user from their browser history and their Facebook and Twitter feeds to create a user profile that represents the user's core interests. The user profile is represented as a graph which contains the articles, and the key entities and topics extracted from the articles. Given a set of candidate articles from which to recommend, PrIA ranks the articles based on their similarity to the user profile graph by measuring the centrality of each candidate article within the profile graph. PrIA obtains publicly available candidate news articles, a generic collection without providing login credentials and does not reveal which of these articles that the user likes or reads. Thus, no sensitive information is revealed to the aggregation site.

We conduct a small scale user study with six subjects over a period of around ten days to show that such a local recommendation can be provided from laptop-class devices. Based on user feedback, we compare PrIA's local recommendation with Google's news recommendation service. Unsurprisingly, PrIA's recommendations have lower precision in terms of the number of recommendations that were useful to the user. However, despite not having access to collective user knowledge, the average precision of PrIA is only 14% lower compared to Google's service. In other words, providing the news recommendation service with privacy guarantees only requires a modest cost in terms of personalization effectiveness. We believe that PrIA's recommendations can be made even more precise by tuning our NLP algorithms.

We envision that the PrIA framework can be used to support a large number of IA services starting from news/event recommendation systems to question answering to context aware services. We discuss the privacy, NLP, and systems challenges in supporting these services through PrIA.

2. THREAT MODEL

News aggregation sites which personalize recommendations do so by recording sensitive data that a user may want to keep private. Every time the user signs in, the sites log which articles she reads, what she shares, and how she rates the suggested articles. All this information is used to build a user profile which is then used to make new recommendations. While useful for personalization, this profile contains many bits of sensitive information about the user such as her political inclinations, social habits, and medical conditions [9], all of which is now available to the recommendation service provider in their cloud storage. A privacy focused user would like to avoid sharing this information with the news aggregation sites.

Formally, we define the adversary as news aggregation services that provide personalized news recommendations to logged-in users. For example, Google's news recommendation service provides news recommendations and require the user to sign-in to receive the service. Prior work on linkage attacks have demonstrated that personally identifiable attributes (e.g., user IDs) and quasi-identifiable attributes (e.g., age, zip code) can be exploited to uniquely identify individuals [11]. There are two additional scenarios to consider: (i) Individual news sites may also provide personalization requiring the use of a login. If the content they provide is available publicly otherwise, the individual sites are also adversaries that PrIA targets. (ii) News sites, individual or aggregation, may not require a login but can still track user interests through fingerprinting or other similar techniques. We do not consider these as adversaries. We expect that one can use standard approaches to avoid fingerprinting (e.g.., Prevaricator [12]) or use `tor` [1] to avoid tracking.

2.1 Utility and Privacy metrics

We now define the utility and the privacy metrics for the news recommendation system as follows. Let $\mathbb{X}$ be the set of the top-k news articles that are recommended by PrIA and $\widetilde{\mathbb{X}}$ be the subset of articles in $\mathbb{X}$ that the user found *useful*. Also, let Ω_s be the set of sensitive behaviors that the user would like to keep private. For example, a possible instantiation of $\Omega_s = \{$Political view, Medical condition, Stock market plan$\}$. Finally, let M denote the information shared by the user via PrIA. For example, in case of the news recommendation system, M is the request for downloading all the generic RSS feeds from Google news page.

- Utility: We define the utility metric Ψ_{util} as the fraction of the recommended articles that the user found useful. While for now we consider a binary rating of articles, the system can be easily extended to incorporate a more fine-grained measure of rating. Formally,

$$\Psi_{util} = \frac{|\widetilde{\mathbb{X}}|}{|\mathbb{X}|} \tag{1}$$

  Ψ_{util} is same as the *Precision at k* metric (§4).
- Privacy: Intuitively, after observing the messages M from PrIA, we do not want the adversarial belief about a particular sensitive behavior of the user, as defined in Ω_s, to increase beyond a particular user-specified threshold, δ. In other words, let the prior adversarial belief about a sensitive behavior $\omega \in \Omega_s$ be given by $\mathbb{P}(\omega)$. The posterior probability after observing the messages M is given by $\mathbb{P}(\omega|M)$. We want the difference of these two probabilities to be bounded by δ for every element in Ω_s. Formally,

$$\Psi_{priv} = \mathbb{P}(\omega|M) - \mathbb{P}(\omega) \tag{2}$$

Eqn. 2, ensures that the *knowledge gain* of an adversary, is bounded by $0 \leq \delta \leq 1$ even after observing the messages M output from PrIA.

Thus, the design objective of PrIA is to maximize Ψ_{util} while ensuring that $\Psi_{priv} \leq \delta$ for all $\omega \in \Omega_s$.

3. PrIA RECOMMENDATION MODEL

The PrIA news recommendation system involves: (i) building a user profile to model user interest, (ii) obtaining a candidate set of news articles from which to recommend and (ii) designing a ranking algorithm to score and rank the candidate articles based on how well they match the user profile.

3.1 Architecture

Figure 1 shows PrIA's news recommendation architecture. We assume that user's personal device (PD) performs most of the heavy lifting. The personal device is within the user's trust zone, and could be a personal computer or even a device in the user's private cloud. In our user study, the user's personal laptops act as the PD.

The PD builds a profile about the user's interests based on different sources of information obtained from the *user's digital history* starting from the user's interactions with news articles, general browsing data, social media interactions, and other contextual information. The PD obtains the candidate news article from an aggregation site and recom-

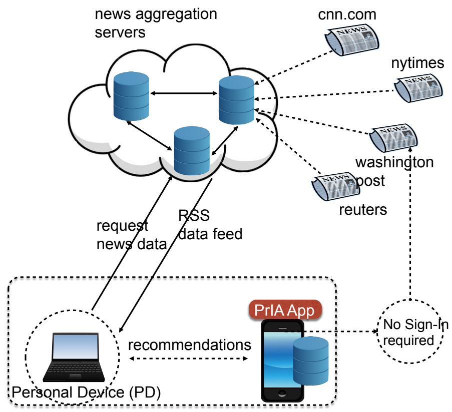

Figure 1: The PrIA News Recommendation System

mends news articles. recommended news articles are sent to the recommendation app on the user's phone.

We build the model on a laptop device rather than the smartphone because news recommendation is essentially a push-based service and is not latency sensitive. We later discuss how more latency-sensitive applications can be supported in the PrIA framework (§5).

3.2 Creating User Profiles

A basic content-based recommendation algorithm uses a simple bag-of-words [10] representation of the user's digital history. A given candidate article is recommended if the pairwise similarity between the candidate article and every article in the user's digital history is high [10].

The problem is that often the bag-of-words model alone does not effectively capture the key semantics of the domain. It does not adequately capture important concepts and tends to include a large number of words to represent articles. In the news domain, the topics of the articles (e.g. finance, sports) and the entities mentioned in them (e.g., people, places, and organizations) reflect the important aspects of user interest. For example, if the user reads several articles about *Roger Federer*, a tennis player, the user may be interested in the entity Roger Federer, and the topic of tennis. Therefore, we first process the news articles to annotate them with the topics and entities mentioned in the article, to augment the bag-of-words representation.

For extracting topics, PrIA uses Latent Dirichlet Allocation (LDA) [3], a widely used topic modeling technique to automatically discover the topics in the user history. For each article, LDA assigns a probability distribution over a set of K topics i.e., the probability of the article belonging to each of the K topics. To extract entities, PrIA uses an off-the-shelf named entity tagger [5], which identifies mentions of people, places, and organizations in each article.

PrIA uses these extracted topics and entities to create a graph based user profile. The nodes in the graph consist of the articles in the user's history, and the entities and topics representing the user's interests. The edges in the graph connect each article with the corresponding entities and topics extracted from the article. This graph-based representation scales easily to large number of nodes and is well-suited to run efficient ranking algorithms.

3.3 Obtaining a candidate set of news articles

In the current instantiation of PrIA, we download all news articles from existing news aggregation services (in our implementation, we use Google's news aggregation service). In effect, PrIA decouples news aggregation from personalization. News aggregation services such as Google aggregate news articles, but also provide personalized recommendation to signed-in users. In contrast, PrIA does not require the user to be signed-in, and the news articles we download are generic and not personalized to a user.

PrIA can work with any publicly available news source to obtain candidate news articles. In the absence of aggregations sites, one could use other sources such as RSS feed subscriptions or even download news articles from various news sources using techniques to avoid fingerprinting [12].

3.4 Recommendation Algorithm

Given a candidate pool of news articles, the recommendation task is to score each candidate article based on how well it matches the user profile. To this end, PrIA uses a variant of the popular page rank technique, known as personalized page rank [7].

The key intuition in PrIA is that a candidate article that is well connected to the user profile graph is likely to be of interest to the user. An article that mentions many entities and topics of interest to the user is going to be well connected within the user profile graph. The Personalized Page Rank algorithm measures this connectedness of candidate articles.

Figure 2 shows how the personalized page rank algorithm is applied to the user profile graph. The graph has nodes representing articles, topics, and entities, and edges representing the connection between the articles and the topics/entities. For recommendation, the candidate articles are first directly added as nodes to the user profile graph. Then, the articles are processed to extract topics and entities. If the topics and entities in the candidate article are also present in the original user profile graph, a new edge is drawn between the article and corresponding topic/entity.

Once the candidate articles are added, we score each candidate article by its centrality in the graph (described in detail below). We note that such graph based ideas have been explored in the context of collaborative search, but existing work only considers the direct neighbors of the candidate articles in the user profile graph [8]. In PrIA, we propose a more general approach that uses all nodes in the user profile graph via random walks between the candidate article and the rest of the graph. This technique ensures that we are able to take into account for the global structure of the graph and the long distance influences when using the recommendation.

Personalized Page Rank (PPR): Our goal is to score news articles based on how well they are connected to the user profile graph. Page Rank is a measure that formalizes the notion of connectedness on a graph. It measures connectedness of each node as the probability that a random walker would land on that node after taking long random walk over the graph following edges with a probability proportional to their weights.

PrIA uses the Personalized Page Rank [7][1] variant that allows us to explicitly model a notion of *core interests* for the user, rather than treat all nodes in the graph uniformly. Given a set of nodes that represent these core interests, PPR

[1]The word *Personalized* in the name here refers to search personalization task used in the original paper and is not related to the news personalization we are concerned with.

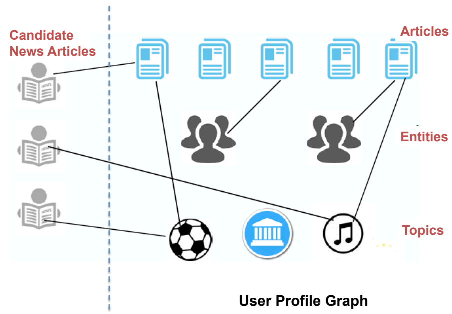

Figure 2: User profile graph with topic, entities and articles. The candidate news articles are augmented to the existing user profile graph and the personalized page rank is run on the augmented graph.

modifies the page rank computation such that nodes that are well connected to these seed nodes score higher. From the random walk perspective, the walker randomly restarts from a seed node with some *seed probability* based on the importance of the nodes. The effect is that the candidate articles that more easily reachable from the seed nodes receive higher scores. The formal PPR scoring function is shown below.

$$PPR(v) = \alpha_v + (1 - \alpha_v) \sum_{v:e(u,v)\in E} P(v|u)PPR(u)$$

where (u, v) is an edge in the graph, v is a vertex, $P(v|u)$ is the transition probability, i.e., the probability with which a random walker at node u will move to node v, and α_v is the seed probability.

Transition Probabilities The transition probabilities $P(v|u)$ are estimated based on the similarity or affinity between two nodes: (i) Article node -Topic node edges are scored by the fraction of overlap between representative words in the topic and the article. (ii) Article node -Entity node edges are scored by the number of mentions of the entity within the article. (iii) Article node -Article node edges are scored by the fraction of overlapping words in the article pair.

Seed Probabilities In the random walk, α_v is the probability with which the surfer restarts a random walk from node v. We estimate these seed probabilities iteratively at the end of each day to track the user's changing interests. Initially, we set the seed weights uniformly. At the end of each day, we run PPR until convergence and select the top scoring nodes as *core* nodes and renormalize their weights to assign seed probabilities. The rest of the nodes get a small default probability.

As the user interacts with new data the user profile graph grows and the user interests change with current events and happenings [6]. Processing all data to extract topics and entities will consume resources. At the end of each day we uniformly sample a subset of articles visited that day and add it to the profile graph. To avoid recalculating entire topic distributions every day, we use an online version of LDA which incrementally updates its topic distributions. Further, we adopt a simple hard decay strategy that drops articles that have low page rank scores and are more than k days old.

3.5 Utility and privacy

The PPR algorithm above ensures that the articles are ranked based on changing interests of the user. Thus, the top

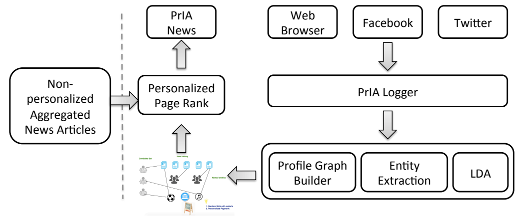

Figure 3: The applications in PrIA for news recommendation.

$k-$ candidate articles selected, are more likely to cover the topics that are of interest to the user, maximizing the utility metric Ψ_{util}. Further, since every PrIA user downloads the same set of articles from the aggregation site, the users do not give any personal information to the aggregation site. In the absence of fingerprinting and any collusion between the aggregation site and other individual news sources, PrIA provides strong privacy guarantees.

4. IMPLEMENTATION AND USER STUDY

We implemented PrIA as described in the previous section and conduct a user study with 6 subjects to compare the performance of PrIA with existing cloud-based news recommendation systems. The user study was approved by our Institutional Review Board.

Figure 3 shows the components in the implementation. The PrIA news recommendation runs on the user's laptop (which acts as their personal device) and the user's smartphone. We use a server to facilitate communication between the user's laptop and the smartphone over a secure channel.

The implementation consists of four applications:

1. Logger – The logger collects the web browsing history from Chrome, and the user's Twitter and Facebook feeds through the respective APIs. The logger runs on both the laptop and smartphone and the user profile graph is built over this collected history. Logged data from the smartphone is encrypted and sent securely over to the user's laptop via a server. The encryption key is known only to the laptop/smartphone user so that the data cannot be decrypted by the interim server. The server deletes any data once the transfer is complete.
2. Profile Builder – The user profile graph is built at the user's laptop. The Stanford Named Entity tagger [5] is used to extract entities and the well known LDA algorithm [3] is used to extract the topics. We choose the top 20 topics to add to the profile graph.
3. PPR Recommendation – The recommendation algorithm periodically downloads all headline articles (40 to 50 articles) from Google News, a popular news aggregation site. These articles are obtained without sign-in; therefore the news articles are not personalized for the user. In other words, we use the Google News service in-lieu of a generic news aggregation service. PrIA runs the recommendation algorithm over the candidate articles and pushes the recommended articles to the smartphone.

All computation over the user's personal data is performed within the trust domain of the user and the interactions outside the trust domain relates to either retrieving the articles from the individual providers or obtaining non-personalized candidate articles from aggregation sites.

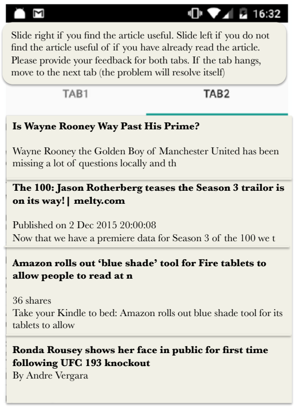

Figure 4: The PrIA app that shows news recommendations by Google and PrIA in two tabs.

4.1 User Study results

We conduct a small scale user study with six subjects summarized in Table 1. PrIA is downloaded on each users laptop and Android phone. The users provided feedback on the news recommendation provided by PrIA and an alternative cloud-based news recommendation service. The user feedback was collected for an average of 10 days for each subject, ranging from 6 to 15 days.

Comparisons with a Cloud-based news recommendation service: We compare PrIA with personalized news recommendation service provided by Google. The subjects in the user study sign-in to obtain this personalized recommendation. Note that this is for comparisons alone. In the real deployment, the user will not have to be logged in, since PrIA does not require cloud support. We choose the top 20 Google's suggested news articles everyday. The recommendations from Google's service and PrIA are presented in 2 tabs to the user as shown in Figure 4. The tabs randomly alternate between showing Google's news recommendations and PrIA news recommendation.

User feedback: The recommendation feedback provided by the subjects is binary: the subject marks if the recommended article is useful or not. The users ratings are sent to our server for evaluation. The rating only consists of the ranks of each articles that the user marked as useful, and does not contain any details about the article itself. In our current instantiation, we do not keep track of how long and how often the user *engages* with the news articles. Tracking the real utility of the recommended news article is beyond the scope of this work.

4.2 User study results

First we analyze the feasibility of running PrIA on user devices. Table below shows the amount of data collected for a single user.

Even though the actual size of articles that the user browses can be hundreds of MBs, PrIA prunes the graph periodically by removing older articles and topics/entities that occur less often in the user's history. Building the model the first time is compute intensive, but on subsequent days, PrIA spends less than 10 minutes processing the data and updating the user profile graph. Together these numbers suggest promise for the feasibility of implementing PrIA on commodity personal user devices.

Study Participant Details	
# Users	6
# Days	6-15
# Articles rated	1956
Statistics from a sample user	
Average Profile size	28.7MB
Average # Nodes	3546
Avgerage # Edges	29652
Average time per day (after first day)	8.7 mins

Table 1: User study statistics.

The users rated a total of 1956 articles in total and found 754 articles useful. Table below compares the news recommendation provided by PrIA and Google's news recommendation service in terms of three metrics: Fraction of relevant articles (i.e. those the users marked as useful) in the top five (precision@5), and top ten (precision@10) recommendations and average precision (AP). AP is the average of precisions at ranks with relevant articles. Given two rankings with same number of relevant articles, AP is higher for the ranking in which the relevant articles are ranked higher than non-relevant ones .

Recomm. Alg.	Precision@10	Precision@5	AP
Google	0.45	0.44	0.51
PrIA	0.38	0.35	0.44

Table 2: Comparing the precision of Google's news recommendation and PrIA news recommendation across three different metrics.

Since this is a small scale study, it is difficult to draw strong conclusions. However, the average precision of PrIA drops only by 14% relative to Google's news recommendation, which is surprising considering that there is no collective knowledge in PrIA. Further, our implementation is without any extensive tuning of the out-of-the-box components, a common necessity when using NLP algorithms.

5. DISCUSSION

Our work targets cloud applications that build models on user's textual data and use this model to provide personalized services.

Personalizing from large publicly available resources: News aggregation sites have access to large amounts of articles from which they recommend. In this work, we assumed that PrIA only downloads a snapshot of the articles that Google News displays once every day, which may miss some relevant articles. Downloading all aggregated news articles to local devices is not a scalable approach due to various reasons including data size, storage and bandwidth constraints, or some limits set by the public data source. One strategy to address this is to download subsets of articles that are likely to be of interest to the user and then make recommendations from this subset.

A key challenge of course is: what strategies can be used to query a public data source to download a slice of the articles both efficiently and without leaking sensitive information? Building the user profile graph is a first step: the profile can help guide the selection of articles to download. Further, Private Information Retrieval (PIR) [4] techniques have been used to maintain the privacy of a client that queries a database, without letting the database infer sensitive information about the client. Similar to PIR techniques, we can devise strategies, that for a given word query would generate multiple synonymous word queries to mask the intended query from the news provider.

Personalization over private data: Another class of services are intelligent assistance over user's private data. Users today collect a large amount of personal data from several sources and require assistance in querying and understanding the data. Imagine Alice read an email earlier about an event. Later, she is trying to recollect the event but does not remember many details about the event. She should be able to use simple natural language to ask questions to an IA system to retrieve this information.

The first step towards performing these tasks to build question-answering capabilities on the user's own personal data. In our earlier work, we show how search can be performed locally on the phone [2]. We envision a step further towards personalization that not only searches local content but also answers natural language questions. Systems such as Alexa and Siri provide a similar service, but require the cloud infrastructure.

A key challenge in supporting services such as question-answering on local data is in running the models and storing data locally on the phone. For push-based applications such as recommendation services, it is sufficient for PrIA to model the service in a trusted domain and push the recommendation to the smartphone periodically. However, question-answering services are pull-based where responsiveness is key and smartphone may not always be connected to the user's laptop or private cloud. Recent works have shown the feasibility of running machine learning models on phones, for similar privacy concerns [13]. Such approaches can alleviate the challenges in supporting personalization of private data.

Personalization and Context based services (public + private data): Finally, a class of IA applications can be thought of as using a hybrid model, that use both private and public data. To illustrate a use case for a hybrid IA system, let says Alice likes the Harry Potter book series. She is visiting London where J.K.Rowling (the author of the Harry Potter series) is giving a talk. The IA system should notify Alice that she might be interested in going to Rowling's talk. Notice that several pieces of information is required for this notification. The IA service needs to know that Alice will be visiting London on certain days, and the service needs to know the public information about where J.K.Rowling is giving a talk.

Such applications again fit within the PrIA framework, and the challenges in enabling these services are the sum of the problems in providing personalization over public and private data.

6. CONCLUSIONS

As the scope and performance of personalized IA services improve, they become more indispensible, and it becomes ever more critical to characterize the privacy trade-offs and investigate privacy preserving solutions. As a case in point, news recommendation services reveal news browsing history, which can be used to infer many sensitive personal information about the users. A local private news recommendation provides an interesting trade off: it trades collective knowledge for a much bigger slice of personal data, which is kept private. This work shows the feasibility of building a local recommendation system, which minimizes leaking personal information to the service provider. Looking ahead, for scaling to the larger class of intelligence assistance problems, we identify significant challenges in modeling privacy, and addressing algorithmic and system challenges in getting these services to run on constrained devices, all ripe areas for future investigation.

Acknowledgements

We thank our shepherd Landon Cox for his insightful comments that helped improve the paper. We thank the anonymous reviewers for their constructive feedback. We thank our 6 user study participants for downloading PrIA and providing feedback on the recommendations. This work was partially supported by CEWIT (Center for Excellence in Wireless and Information Technology), Stony Brook.

7. REFERENCES

[1] Tor Project. https://www.torproject.org/.

[2] A. Balasubramanian, N. Balasubramanian, S. Huston, D. Metlzer, and D. Wetherall. FindAll: A Local Search Engine for Mobile Phones. In *Proc. ACM CoNext*, December 2012.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[4] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *J. ACM*, 45(6), Nov. 1998.

[5] J. R. Finkel, T. Grenager, and C. D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[6] F. Garcin, K. Zhou, B. Faltings, and V. Schickel. Personalized news recommendation based on collaborative filtering. In *Web Intelligence*, 2012.

[7] T. H. Haveliwala. Topic-sensitive pagerank. In *WWW*, 2002.

[8] L. Li and T. Li. News recommendation via hypergraph learning: encapsulation of user behavior and news content. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 305–314. ACM, 2013.

[9] J. Liu, P. Dolan, and E. R. Pedersen. Personalized news recommendation based on click behavior. In *IUI*, 2010.

[10] Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang. Learning to model relatedness for news recommendation. In *Proceedings of the 20th international conference on World wide web*, pages 57–66. ACM, 2011.

[11] A. Narayanan and V. Shmatikov. Myths and fallacies of personally identifiable information. *Communications of the ACM*, 53(6):24–26, 2010.

[12] N. Nikiforakis, W. Joosen, and B. Livshits. Privaricator: Deceiving fingerprinters with little white lies. In *Proceedings of the 24th International Conference on World Wide Web*, pages 820–830. ACM, 2015.

[13] O. Riva, S. Nath, and E. Fernandes. Appstract: On-the-fly app content semantics with better privacy. In *Proc. ACM Mobicom*, 2016.