

Preserving Privacy in Spoken Language Databases

Min Tang*, Dilek Hakkani-Tür, and Gokhan Tur

AT&T Labs-Research,
Florham Park, NJ, 07932
{mtang, dtur, gtur}@research.att.com

Abstract. Goal-oriented spoken dialog systems aim to identify intents of humans, expressed in natural language, and take actions accordingly, to satisfy their requests. State-of-the-art data-driven spoken dialog systems are trained using large amounts of task data which is usually transcribed and then labeled by humans, a very expensive and laborious process. Hence sharing and reuse of this data has extreme importance for research and development of spoken language processing systems. On the other hand these utterances may include confidential personal information about the speakers, such as the social security numbers or credit card numbers. In this paper, we describe data sanitization approaches for natural language utterances to protect the privacy of the speakers. The challenge in sanitization is ensuring that the performance of the spoken dialog system models trained using the sanitized data is as good as the ones before the sanitization. We show that, by hiding task-dependent named entities we can preserve the privacy of the speakers, and still achieve a comparable accuracy.

1 Introduction

Goal-oriented spoken dialog systems aim to identify intents of humans, expressed in natural language, and take actions accordingly, to satisfy their requests. In a spoken dialog system, typically, first the speaker’s utterance is recognized using an automatic speech recognizer (ASR). Then, the intent of the speaker is identified from the recognized sequence, using a spoken language understanding (SLU) component. This step can be framed as a classification problem for goal-oriented call routing systems [10]. Figure 1 presents an example dialog between an automated call center agent and a user.¹ As it is clear from this example, these calls may include very sensitive information about the callers, such as names as well as the credit card and phone numbers.

State-of-the-art data-driven ASR and SLU systems are trained using large amounts of task data which is usually transcribed and then labeled by humans, a very expensive and laborious process. In the customer care domain, “labeling” means assigning one or more of the predefined intent(s) (call-type(s)) to each utterance. As an example, consider the utterance *I would like to pay my bill*, in a customer care application. Assuming that the utterance is recognized correctly, the corresponding intent or the call-type would be *Pay(Bill)* and the action would be learning the caller’s account number

* Currently at the Center for Spoken Language Research, University of Colorado, Boulder, CO.

¹ All names and numbers in the dialog are made up for illustration purposes.

-
- **System:** How may I help you?
 - **User:** Hello. This is John Smith. My phone number is 9 7 3 area code 1 2 3 9 6 8 4. I wish to have my bill, long distance bill, sent to my Discover card for payment.
 - **System:** OK, I can help you with that. What is your credit card number?
 - **User:** My Discover card number is 2 8 7 4 3 6 1 7 8 9 1 2 5 7 hundred and it expires on first month of next year.
 - **System:** ...
-

Fig. 1. An example natural language dialog

and credit card number and fulfill the request. The transcribed and labeled data is then used to train automatic speech recognition and call classification models.

The bottleneck in building an accurate statistical system is the time spent for high quality labeled data. Sharing of this data has extreme importance for machine learning, data mining, information extraction and retrieval, and natural language processing research. Reuse of the data from one application, while building another application is also crucial in reducing the development time and making the process scalable [9]. However, preserving privacy while sharing data is important since such data may contain confidential information. Outsourcing the data and tasks requiring private data is another example of information sharing that may jeopardize privacy of speakers. It is also possible to mine these natural language databases to gather aggregate information using statistical methods, and these may be confidential or sensitive. For example, in an application from the medical domain, using the caller utterances, one can extract statistical information such as the following:

y% of the US doctors prescribe <DRUG1> instead of <DRUG2>

which may also be need to be kept private due to business-related reasons.

Another similar example which includes natural language that has personal information is the medical reports of patients, which if made publicly available, can facilitate the medical research significantly. Ruch *et al.* have worked on hiding names in formatted medical reports [14].

The aim of sanitization is to hide personal information given the privacy requirements in order to disable data mining approaches to extract personal or other private business related information in spoken language databases. This can be considered as *privacy preserving text mining*. The security implications of data mining and threats to privacy that can occur through data mining are described in [6] and [16]. Methods to preserve privacy for data mining are well studied for numeric and categorical values [2, 8][among others], however preserving privacy for natural language databases and its effects has not been studied to the best of the authors' knowledge.

Sanitization depends on the corresponding task. We need to make sure that the data quality is still preserved after the sanitization. Data quality could be measured in terms of readability and the ability to use the sanitized text for the corresponding task. For example, if the data is going to be used for text classification, it is necessary to perform sanitization without hurting the classification accuracy. If the task is information

retrieval, sanitization methods should not interfere with the indexing and document matching methods.

In this paper, we describe data sanitization approaches for natural language utterances to protect the privacy of the speakers. Our method is based on detecting named entities such as person, location and organization names and numeric values, such as date, credit card number, in the spoken utterances. We then sanitize them using various means with the purpose of hiding the personal information. The challenge in the sanitization of these utterances is ensuring that the accuracy of the spoken dialog system models trained using the new sanitized data and used for automatic speech recognition and spoken language understanding is as good as the ones before the sanitization. We show that, by hiding task-dependent named entities we can preserve the privacy of the speakers, and still achieve a comparable accuracy.

In the next section, we describe our approach. In Section 3, we present our method of named entity extraction. We show that this method does not result in significant degradation in the accuracy of the individual spoken dialog system components in Section 4. Then, in Section 5, we discuss further issues and present the future work.

2 Approach

There are 3 known methods for partial access to databases [7], which can also be used for sanitization of personal information in spoken language utterances:

- *Value distortion* alters the confidential values that needs to be hidden with random values.
- *Value dissociation* keeps the true distribution of the values but replaces each value in a record with the value of the same field from another record. This can be achieved, for example, by exchanging the values across sentences.
- *Value-class membership* exchanges the individual values with disjoint, mutually exhaustive classes. For example, all the person names can be changed to a single token <NAME>.

The simplest form of text sanitization is modifying the values of named entities or replacing them with generic tokens. If the named entities are not already marked during transcription or labeling, we can utilize automatic named entity extraction methods, which are well studied in the computational linguistics community.

The generic tokens can also preserve the non-sensitive information to ensure readability of the text. For example, the gender or identification of the people can be marked in the token for the resolution of further (pronominal) references (i.e., <PERSON> versus <PERSON, GENDER=MALE>). An even harder task would be associating references during sanitization, for example <DATE2> is extended as <DATE2=DATE1+3 days>.

Our approach consists of finding the named entities in the given utterance database of calls to a customer care center, and hiding these by using the three ways of sanitization. In distortion, we replace the named entity values with random values from the same named entity category. In dissociation, we exchange the value of the named entity with another named entity of the same category in the utterance database. In value-class

membership, we replace the named entity values with generic named entity category tokens, such as <NAME> and <PHONE_NUMBER>. This approach is likely to improve accuracy of call-type classification due to better generalization of word n -gram features, since call-types are expected to have strong associations with named entity categories, but not necessarily with their values. In speech recognition, it is traditional to use *language models* to statistically model the word sequences [12]. In this work, we trained class-based language models for the value-class membership approach, where each named entity category corresponds to one class [4]. For call classification, we have not changed the training algorithm, only the feature values in the train and test sets are changed accordingly.

3 Named Entity Extraction

The purpose of named entity extraction is to identify the sub-string of the input utterance that contains a named entity, and extract its type. For example in the utterance “my phone number is 1 2 3 4 5 6 7 8 9 0”, the sub-string “1 2 3 4 5 6 7 8 9 0” contains the named entity of type <PHONE_NUMBER>. After named entity extraction, this entity can be marked in the utterance using XML tags: “my phone number is <PHONE_NUMBER> 1 2 3 4 5 6 7 8 9 0 </PHONE_NUMBER>” for sanitization purposes.

Named entity extraction task has been studied extensively in the framework of the DARPA-funded programs, like ACE [1], and discussed and evaluated in many natural language processing conferences, such as the Message Understanding Conferences (MUCs) ² and CoNLL Workshop ³ as the shared task. A very detailed definition of the named entity extraction task has been developed in the framework of these programs [5]. Similar to most other language processing systems, developers have approached the named entity extraction problem as one of building a hand-crafted rule-based system [11], an automatically trained system [3], or a combination of these two approaches.

We employ a rule-based approach for named entity extraction. In this case, for each entity, a grammar in Backus Naur Form (BNF) is created manually. The creation could involve the reuse and extension of a library of application-independent named entity grammars (“phone numbers”, “dates”, etc.) or a set of named entity grammars can be created for the current application.

These grammars are typically regular expressions written in a grammar rule notation. Figure 1 shows a simplified example grammar for extracting phone numbers. The grammars are compiled into finite-state transducers whose arcs are labeled with the terminals of the grammars. The two components of the arc labels are then interpreted as the input and the output symbols leading to a finite-state transducer representation.

² Message Understanding Conference (MUC-7), 1998, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/

³ Sixth Conference on Natural Language Learning (CoNLL), 2002, <http://cns.uia.ac.be/signll/conll02/>

```

PHONE_NUMBER ::=  $\epsilon$ :<PHONE_NUMBER>
                $\epsilon$ :<AREA_CODE>AREA_CODE  $\epsilon$ :</AREA_CODE>
                $\epsilon$ :<LOCAL> LOCAL  $\epsilon$ :</LOCAL>
                $\epsilon$ :</PHONE_NUMBER>
AREA_CODE     ::= DIGIT DIGIT DIGIT
LOCAL        ::= DIGIT DIGIT DIGIT DIGIT DIGIT DIGIT DIGIT
DIGIT        ::= 0:0 | 1:1 | ... | 9:9

```

Fig. 2. A simplified fragment of a ‘phone number’ grammar.

For named entity extraction, each utterance FSM (U_i) is composed with each entity grammar F_j sequentially resulting in an FSM (M_i) representing the utterances with the named entities marked. The grammar rules can also specify the context in which they can apply, to prevent false acceptances.

4 Experiments and Results

In this section we first describe the data set that we used in the evaluation. Then we present the named entity (NE) extraction evaluation metrics and experimental results. After that we show the automatic speech recognition (ASR) and natural language understanding (SLU) experimental results after applying three sanitization approaches to training set.

4.1 Data

Our experiments are conducted on the human-to-machine dialogs collected from the AT&T VoiceTone[®] Spoken Dialog System for an application in the pharmaceutical domain. In named entity evaluation, we used a training set of 5,415 utterances all of which contain one or more named entities (average length per utterance is 15.7 words) and a test set which contains 8,932 utterances which may or may not contain a named entity (average length per utterance is 8.8 words). Both the training and test sets are transcribed text and have been tagged with named entities by human labelers.

In the ASR and SLU evaluation, we used a training set with 29,561 utterances (average length per utterance is 10.1 words) and a test set with 5,537 utterances (average length per utterance is 8.6 words). For ASR evaluation, the language model is trained from transcribed training set. For SLU evaluation, both the training set and the test set have been labeled with call types, and we have tried different combinations of ASR output and manually transcribed versions of the training set and test set.

In this study, we have used trigram language models based on Variable Ngram Stochastic Automata (VNSA) [13] for ASR. The acoustic model is kept fixed and is subword unit based, with triphone context modeling. For call classification, we have used a boosting-style classification algorithm using word n -grams as features [15].

4.2 Named Entity Extraction

During the NE extraction evaluation, we used mainly three metrics. The first one is the *recall*, defined as the proportion of all the true NEs that are correctly tagged. The second

one is the *precision*, defined as the proportion of all the tagged NEs that are also true. The third metric, which is called *F-Measure*, combines the first two metrics as [12]:

$$F\text{-Measure} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (1)$$

With these evaluation metrics, we consider a named entity as correctly extracted, only if the extraction finds the *exact* named entity. However, in reality, this only is a lower bound, as one can find only portions of a named entity and still be able to sanitize it. For example, if only the surname in a person name is marked as a name, changing the surname may still hide the identity of the person. So we define an upper bound by considering a partial extraction to be a correct extraction and compute the performance accordingly. The real sanitization coverage will be somewhere between these lower and upper bounds.

Table 1 presents the named entities we have selected for this application along with their extraction performances using regular grammars. In the table, the exact match precision, recall and F-Measure are obtained by considering only exact named entity extraction results as correct. The partial match results are the upper bounds of recall, precision and F-Measure, respectively, and they are obtained by considering partial extractions of named entities as correct, as explained above. Since our goal is to hide private information, we want to achieve a high recall rate. As seen, the exact match recall rates for most of the NEs are higher than 80% except for Person Name, which has only 51% recall rate (however, the sanitization coverage rate for Person Name is somewhere between 51% and 82% since a partial extraction may also lead to a successful sanitization). Actually, it is difficult to detect person names using regular grammars. So we did another experiment which used a “heuristic” scheme to detect person names. We used grammars to detect location and organization names, and we marked all other words that start with an upper-case letter as a person name. This heuristic resulted in a significant improvement in the performance of extracting the person names as shown in Table 2. Our aim is to sanitize the human transcriptions of the utterances in the database, and all the location, organization and person names are already marked with an upper-case initial letter in the transcriptions. Thus this “heuristics” is reasonable and its condition can be easily guaranteed during the transcription process.

4.3 ASR Experiments and Results

In the literature, word and sentence accuracies are used to evaluate ASR performance. Sentence accuracy is the percentage of sentences that are recognized correctly (i.e., they exactly match the transcriptions). Word accuracy, is computed as follows [12]:

$$\text{Word Accuracy} = 1 - \frac{\# \text{ of insertions} + \# \text{ of substitutions} + \# \text{ of deletions}}{\text{Total } \# \text{ of words in the correct transcript}}$$

In the baseline system, the language model has a vocabulary of 5,345 words and was trained from the transcribed training set with all the personal information untouched. We tried three approaches to hide personal information in the training set as described in Section 2, namely value distortion, value dissociation and value-class membership.

NE	Count	Exact Match			Partial Match		
		Recall	Precision	F-Measure	Recall	Precision	F-Measure
Drug Name	367	0.796	0.896	0.843	0.861	0.969	0.912
Person Name	228	0.509	0.592	0.547	0.816	0.949	0.877
Organization	278	0.856	0.948	0.900	0.874	0.968	0.919
Location	51	0.980	0.962	0.971	0.980	0.962	0.971
Member ID	49	0.939	0.939	0.939	0.939	0.939	0.939
Prescription	58	0.845	0.845	0.845	0.914	0.914	0.914
Phone	13	0.923	0.667	0.774	0.923	0.667	0.774
Amount	38	0.842	1.000	0.914	0.842	1.000	0.914
Date	329	0.903	0.879	0.891	0.927	0.902	0.915
Overall	1411	0.803	0.858	0.829	0.881	0.942	0.910

Table 1. NE Extraction using regular grammars (General).

NE	Count	Exact Match			Partial Match		
		Recall	Precision	F-Measure	Recall	Precision	F-Measure
Person Name	228	0.969	0.676	0.796	0.978	0.682	0.804
Overall	1411	0.877	0.851	0.864	0.908	0.881	0.894

Table 2. NE Extraction using regular grammars, Person Name (PN) is detected using a heuristic scheme (General+PN).

As seen from Table 3, after finding the sensitive named entities in training set and hiding them by using the value distortion or value dissociation approach, the ASR accuracy did not degrade significantly and value dissociation approach has a better ASR accuracy than value distortion because it preserves the distribution of each value. The result also shows that with the assumption that class members are in uniform distribution, the value-class membership approach degraded ASR word accuracy from 73.8% to 72.6%. When the class member priors were kept in the class-based language models, the ASR word accuracy of this approach went up from 72.6% to 72.9%.

System	Word Accuracy	Sentence Accuracy
Baseline	73.8%	50.9%
Value Distortion	73.6%	50.6%
Value Dissociation	73.7%	50.7%
Value-Class Membership with uniform dist.	72.6%	50.3%
Value-Class Membership with keeping priors	72.9%	50.4%

Table 3. ASR experiments

4.4 SLU Experiments and Results

The SLU experiments were conducted using different combinations of ASR output or transcription of the training set, and ASR output of the test set and with or without sanitization. The total number of classes in this application is 97. The experimental results are shown in Table 4. In the table, “Top CE” stands for top scoring class error rate, which is the fraction of examples in which the call-type with maximum probability assigned by the classifier was not one of the true call-types. The baseline classification system does not use any named entities and sanitization. “General” is the system that extracts named entities by using regular grammars and then sanitizes data by using value-class membership approach. “General+PN” did the same thing as “General” except that person names, “PNs”, are extracted by using the heuristic scheme.

<i>Training set</i>	<i>Sanitization</i>		<i>Top CE</i>	<i>F-Measure</i>
Transcribed	Baseline		0.2628	0.7252
	Value-Class	General	0.2619	0.7290
		General + PN	0.2612	0.7270
	Value Distortion		0.2658	0.7274
	Value Dissociation		0.2655	0.7283
ASR output	Baseline		0.2664	0.7181
	Value-Class	General	0.2658	0.7190
		General + PN	0.2640	0.7189
	Value Distortion		0.2655	0.7183
	Value Dissociation		0.2635	0.7190

Table 4. *SLU experiments (Test set is from ASR Output)*

As seen from Table 4, no matter whether transcribed training set or ASR output training set is used, the classification systems which sanitize data using value-class membership approach (replacing personal information by named entities) achieved lower top scoring class error rate (the lower the better) and higher F-Measure (the higher the better). This may be due to that there are stronger associations between call-types and named entities than the associations between call-types and values of named entities. Also there is no significant change of SLU performance after using value distortion approach or value dissociation approach. Therefore we are able to maintain or even slightly improve the SLU performance while hiding personal information in spoken language utterances.

5 Discussion and Future Work

One problem with data sanitization is that, the performance of the state of the art information extraction techniques is still far from being perfect (especially when used for previously unseen text or domains). In order not to miss any confidential information,

one may choose high recall for low precision, which may end up with more falsely sanitized portions of text.

Another problem is the domain dependency of the confidential information. For example, some named entities may be confidential for only some domains, but not in others. As an example consider the drug names in the pharmaceutical customer care center application vs. in medical reports. Right now, our approach is aimed to find and sanitize as many named entities as we can find with high recall.

The current approach also does not solve the problem of extraction of statistical information. For example, using the call-type distribution of the utterances in the database, one can extract business-related information, which may be sensitive or confidential, such as:

x% of the customers of company A call the customer service center to cancel their service

In order to prevent such cases, as a future work, we are working on ways of changing the distribution of customer intents, without losing any information necessary for building spoken dialog systems with the same performance. This is a harder task, since changing the prior distribution of the call-types randomly hurts the accuracy of most classifiers.

The approach presented here can also be used on sanitizing the ASR output utterances. The initial letters of the proper names can also be in upper-case in the ASR output, if the proper names are also capitalized in the training data. However, due to the noise introduced by ASR errors, instead of using regular grammars for NE extraction, the use of statistical approaches would be more appropriate [3].

6 Conclusions

We have presented a sanitization approach to preserve the privacy in spoken language databases. We have shown that, by hiding task-dependent named entities, we can preserve the privacy of the speakers, and still achieve comparable ASR and SLU performances with the ones before the sanitization. The same ideas are also applicable to any natural language and speech processing application given the privacy requirements.

7 Acknowledgments

We would like to thank Jay G. Wilpon for his support and comments on this work and the anonymous reviewers and the workshop chair, Yucel Saygin, for their valuable suggestions.

References

1. DARPA Automatic Content Extraction (ACE) Program.
<http://www.nist.gov/speech/tests/ace/>.

2. R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proc. of the ACM SIGMOD Conference on Management of Data*, pages 439–450, Dallas, TX, May 2000. ACM Press.
3. D. M. Bikel, R. Schwartz, and R. M. Weischedel. An Algorithm that Learns What’s in a Name. *Machine Learning*, 34:211–231, 1999. Special Issue on Natural Language Learning.
4. P. F. Brown, V. J. Della Pietra, P. V. deSouza, C. J. Lai, and R. L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479, 1992.
5. N. Chinchor and P. Robinson. MUC-7 Named Entity Task Definition (version 3.5). In *Proceedings of the MUC-7*, 1998.
6. C. Clifton and D. Marks. Security and privacy implications of data mining. In *Proceedings of the ACM SIGMOD Conference Workshop on Research Issues in Data Mining and Knowledge Discovery*, Montreal, Canada, June 1996.
7. R. Conway and D. Strip. Selective partial access to a database. In *Proceedings of the Annual ACM Conference*, 1976.
8. A. Evfimievski. Randomization in privacy preserving data mining. *SIGKDD Explorations*, 4(2):43–48, 2002.
9. G. Di Fabbrizio, G. Tur, and D. Hakkani-Tr. Bootstrapping spoken dialog systems with data reuse. In *Proceedings of 5th SigDial Workshop on Discourse and Dialogue*, Boston, MA, May 2004.
10. A. L. Gorin, A. Abella, T. Alonso, G. Riccardi, and J. H. Wright. Automated natural spoken dialog. *IEEE Computer Magazine*, 35(4):51–56, April 2002.
11. J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. FASTUS: A Cascaded Finite-State Transducer for Extracting Information from Natural-Language Text. In Roche and Shabes, editors, *Finite State Devices for Natural Language Processing*, pages 383–406. MIT Press, Cambridge, MA, 1996.
12. D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall, 2000.
13. G. Riccardi, R. Pieraccini, and E. Bocchieri. Stochastic automata for language modeling. *Computer Speech and Language*, 10:265–293, 1996.
14. P. Ruch, R. H. Baud, A.-M. Rassinoux, P. Bouillon, and G. Robert. Medical document anonymization with a semantic lexicon. In *Proceedings of the American Medical Informatics Association Symposium*, pages 729–733, 2000.
15. R. E. Schapire. The boosting approach to machine learning: An overview. In *Proceedings of the MSRI Workshop on Nonlinear Estimation and Classification*, Berkeley, CA, March 2001.
16. B. Thuraisingham. Data mining, national security, privacy and civil liberties. *SIGKDD Explorations*, 4(2):1–5, 2002.