REMEMBERING WHAT YOU SAID: SEMANTIC PERSONALIZED MEMORY FOR PERSONAL DIGITAL ASSISTANTS

Vipul Agarwal, Omar Zia Khan

Ruhi Sarikaya *

Microsoft Corporation

Amazon

ABSTRACT

Personal digital assistants are designed to assist users in easy information retrieval or execute the tasks they are interested in. The conversational medium implies an additional level of intelligence but typically these systems do not support any reference to the user's past interactions. We propose a domain-agnostic approach that enables the system to address queries referring to the past by using an information retrieval approach to rank various entities for a given query. We also add semantic enrichment to the recall process by augmenting the entities with information from a knowledge graph and leverage that in the retrieval process. We mined user interactions for the Cortana digital assistant to extract queries with location and business entities and show that our technique can achieve an accuracy of 89.8% for such recall queries.

Index Terms— Personal digital assistants, referring expressions, information retrieval, spoken language understanding, dialog management

1. INTRODUCTION

Personal digital assistants (PDAs) are getting more acceptance as they get embedded within various operating systems, such as Cortana, Siri, and Google Now as well as released as stand-alone apps such as Alexa, Hound and Dragon. PDAs are general purpose assistants as they can answer queries spanning various domains and perform a multitude of tasks [1]. The user can converse using natural language queries with the system either to provide some information to complete a task or query about some information [2]. However, in both cases, these systems are typically not designed to reason over their own previous interactions with the same user. If the user asks about some information they previously provided or that system presented to the user, the system lacks the ability to understand the query and recall a past referred entity for the user.

Queries that deal with recall over information previously presented by the system is a segment of queries that gets overlooked as it is not a part of the immediate task execution. This can be considered a meta-query over a previous task. For instance, a system may have the ability to order pizza or reserve

*work performed while at Microsoft Corporation

a restaurant on behalf of the user, but the task is considered complete once the pizza has been ordered or the restaurant reservation has been confirmed. The user, however, may have questions about this task later, such as what pizza size they ordered last time or the name of the restaurant they reserved last weekend. With the proliferation of digital devices and experiences and how embedded they are in our daily lives, the need for semantic memory recall is even greater.

Semantic recall can also enable extension to automatically store user preferences and history which PDAs can leverage to speed up the interaction on future similar tasks. This is similar to the concept of auto-fill for web-forms or auto-complete for web URLs where the system assists the user by using the user's previous history but still allows the user to change any aspect of the interaction, if needed.

For the rest of this paper, we formulate the problem in Section 2, describe our approach in Section 3, detail our data collection and labeling methodology in Section 4, and report on experimental results in Section 5. We discuss the relationship of this work with reference to related work in Section 6. We conclude with the take-away message and directions for future work in Section 7.

2. PROBLEM FORMULATION

We formulate the problem of recalling the right entity from a user's past in relation to a new utterance as an information retrieval problem. Given a user u_i we have a sequence of past t queries $Q(u_i, t) = \langle q_1 \dots q_t \rangle$ from her, ordered by query time. For each query q_j , we extract zero, one or more entities relevant for that query and store them as a sequence $E(Q(u_i, t)) = \langle e_1 \dots e_k \rangle$ ordered by the time the entity was shown to a user. Note, that k can be less than, equal to or greater than t. For a new query, q_{t+1} , the task then is to learn a ranking function $\Phi(E(Q(u_i, t)), q_{t+1})$ and a threshold θ . The ranking function, Φ orders the entities in $E(Q(u_i, t))$ by their relevance to the query q_{t+1} , such that q_{t+1} is referring to a past entity, and also outputs ranking scores for each entity. The threshold θ , determines the cut-off on the ranking scores to decide which entities are recalled.

3. BUILDING A PERSONALIZED SEMANTIC MEMORY

To build a personalized semantic memory, we do the following steps. First, we extract entities from the user's interaction with the PDA. Then we determine which entities need to be stored for future recall. For these relevant entities, we store their meta-data and attributes that are used as part of the feature engineering for more robust recall. Finally, we use a ranker that returns a past entity relevant to a new user query. Each component is described in more detail next.

3.1. Query Entity Extraction

Users are either completing a task or querying for certain information. In either case, there can be one or more entities that are relevant to the user's query. Each relevant entity has to be resolved (grounded) by the system, to enable a user to fetch information or perform an action involving it. For instance, the user may ask for directions to a restaurant, requiring the system to resolve the tagged restaurant name text to a physical entity, which can then be looked up in a knowledge graph (KG). This return result from the knowledge graph lookup provides us the entities and all of the meta data associated with each entity (e.g. address, telephone number, hours etc.), for a given query that we can store for future recall. The benefit of intercepting the KG lookup result is that we are not constrained by which task or domain required the lookup. Another advantage is that even if the query did not explicitly mention the entity name, if a task made a correlation and had a KG lookup, we can still leverage it.

3.2. Entity Relevance

All entities returned to a user in response to a query may not be relevant. This is especially true if the user is being asked to disambiguate between a list of entities returned by the KG. An entity is defined as relevant if a task was completed using it (ordered pizza from a store, asked for driving directions) or otherwise a user clicked on an entity when only a single entity was displayed to her. For clicks on multiple entities, it is not clear which one (or more) entity was of interest to the user and so we do not store those for recall in this paper.

3.3. Entity Attribute and Meta-Data Enrichment

For each entity, we also augment it using the additional knowledge available for it from the KG. For a local business entity, this additional information can include full business name, alternate names, street and web address, phone number or even some meta-information such as categorical information about that entity. This enrichment allows for semantic recall on attributes of the entity beyond the name or what was included in the original user utterance which led to the entity in the first place.

3.4. Feature Extraction

The user query q_{t+1} is treated as a referring entity expression *(RE)* utterance and is represented as a sequence of words. The entities $E(Q(u_i, t))$ in the KB for a user u_i are serialized into text documents $\langle e_1^i \dots e_k^i \rangle$ with multiple sections. Each section groups entity attributes into coherent units like name, address, entity type, incident utterances, other attributes etc. For an input RE we compute a set of bag of words based similarity features over each e_i^i in the user's KB $E(\cdot)$.

For each section in the entity document, we compute ngram matches over the section and the RE. We use character n-gram to capture morphological inflections in the user's RE. Aggregates like count, sum, max over n-gram matches across RE and e_j^i are defined as numerical features. We also leverage BM25F [3], a ranking function that takes document structure into account and represents state-of-the-art Term Frequency-Inverse Document Frequency like retrieval function in information retrieval. We compute BM25F features for the RE over the entire document (entity) as well as each section individually.

3.5. Ranking Mechanism

We approach the semantic recall problem as an information retrieval problem [4]. The collection of relevant entities for a user are treated as a set of documents. The information from KG is treated as additional text that is associated with each document (entity) and is used for feature extraction. The task is then to return the ranked set of documents (entities) relevant to the incoming query. With this formulation, we can apply approaches similar to web-based document ranking. We evaluate the use of various learning algorithms (SVM, Logistic Regression and Gradient Boosted Decision Trees).

4. DATA COLLECTION AND EVALUATION MECHANISM

4.1. Relevant Entity Collection

We sampled logs from anonymized Cortana users and filtered on queries that contained a physical location. We further pruned for queries that only contained a relevant entity (as defined in Section 3.2). For each query, we also collected all other queries for the user associated with that query. This now provides us with a set of users with associated entities, who may be interested in querying over their past information.

4.2. Referring Entity Expression (RE) Generation

Currently, the ability to query about your past information is not available in Cortana so recall queries do not exist in the logs. To generate a corpus of referring expressions for past entities, we created a Mechanical Turk like task in which the

Entity Name	24 Hour Laundry Express - Tonawanda Or Amherst		
Entity Type	Professionals & Services, Home & Family		
Address	2336 Niagara Falls Boulevard Tonawanda, Ny - {EntityZip}		
Website	http://www.24hrlaundry.com/		
Phone Number	(716) 692-5326		
	24 hr laundromat near me		
Query	24 hr laundromat near me		
Query Date & Time	24 hr hundroman near me Wed, July 27, 2016		
Query Date & Time List three ways to	P A landomat over me Wed, July 27, 2016 uvery for the entity above? Summt		
Query Date & Time List three ways to 1.	Ph & Insudemant near me West, July 27, 2016 uery for the entity above? Summ C		

Fig. 1. RE Expression Generation Task Screen-shot

Please look at the user query. Query : show me the place that serves seafood I searched for And the predicted entity:				
Address	1148 River Rd Sw Calabash, Nc - 28467			
Website	http://www.ellasofcalabash.com/			
Entity Type	Food & Drink, Restaurants, Food & Dining, Seafood Restaurants			
Is the intent of the query answered by the predicted entity? O Yes O No C ant Sav (Please provide instification if you choose this)				
Comments:				
	Ç			
Submit				

Fig. 2. Evaluation of Recalled Entity

turks were displayed the user query, the relevant entity associated with that query and various attributes of that entity extracted from the KG. The task then was to provide three different ways how this entity could be referred to in the future. A screen-shot for the view available to the judge is shown in Figure 1. We collected data for 287 users, with 2,172 unique entities, and got 30,957 RE generated for these entities, with multiple judges asked to generate RE for the same entity.

4.3. Model Evaluation

To evaluate the ranking results for a RE expression, the top results were judged manually. We used a human judgment task to label the relevance of ranked top entity for a RE. A screen shot for the judgment task is shown in Figure 2.

We also report results on a negative test set of 2,935 randomly sampled Cortana queries so that we can ensure that our model does not over-trigger for queries that are not relevant. For this purpose, we again randomly sample conversations for Cortana users for queries with any location-based entities (local queries will trigger the local answer that will over-ride the semantic recall) and use our model to determine if any entity would be displayed using our approach. The results are again showed to human judges to determine if any scored relevant entities are useful. The judgment task shown in Figure 2 is

	Accuracy	False Positive
Utterance - FR	60.11	4.2%
Entity Name - FR	66.61	2.3%
SR - SVM	82.49	4.9%
SR - LR	82.97	11.1%
SR - FR	89.80	1.2%

Table 1. Model Accuracy and False Positive Rates

reused for this purpose.

5. EXPERIMENTAL RESULTS

Of the 30,957 RE collected, we used 23,692 (76%) for training, 4,636 (15%) for testing as blind set and 2,629 (9%) was held out for parameter tuning. The data was randomly split in these groups. Note that the data collected was for location entities (of all types, such as business, restaurant, hotel, school, hospital, cinema, and so on). The queries also had various intents (asking for directions, phone number, address, open/close hours etc).

We present results for this data in Table 1. The utterance match refers to matching between the original user utterance that led to the RE as the new query. The entity name match refers to matching only between the entity name and RE, without the enrichment from the KG and we use utterance match and entity match as our baselines. SR refers to our semantic recall approach with different rankers such as FastRank (FR), Logistic Regression (LR) and SVM with RE, entity and KG attributes used. We observe that enriching the entity by adding additional attributes from the KG delivers significantly better results. The precision/recall curves for all three semantic recall approaches based on different rankers are also plotted in Figure 3. We observe that FR outperforms LR and SVM. In Table 1, we also present false triggering rates for the different techniques on non-location related Cortana data and see that false triggering are extremely low, particularly for SR - FR (1.2%).

In Figure 4 we plot the P/R curve for different amounts of training data, where for Precision k, k is the fraction of data used for training with recall on x-axis and precision on y-axis. We can see that the use of further data to train the model may help a little in increasing the accuracy but the amount of existing data used is already providing reasonable results.

5.1. Discussion

For queries with an overlap between the entity name and the query (Query:"Do you remember the Gamestop I wanted?", Entity:"Gamestop (Fair Plain, Mi)"), a ranking technique using entity name would work. For queries that refer to the original utterance (Query: "Find the store whose closing hours I asked for", Original Utterance: "Closing hours for Costco in Lihue", Entity: "Costco (Lihue, Hi)"), a technique based on



Fig. 3. Precision/Recall Curves for Semantic Recall



Fig. 4. Effect of Additional Training Data

matching on previous utterances would work. However, if the query is referring to an attribute of the entity (Query: "Do you remember the name of the shooting range I searched for", Entity: "Strip Gun Club", Entity Category: "Sports & Recreation, Archery & Shooting"), the use of a KG is needed. Similarly, if an attribute of when the entity was surfaced (Query: "Which movie theater did I visit in Tulsa", Entity: "Cinemark Movies 8"), additional meta-data about the context of the entity's original display is needed. Our semantic recall technique uses all these features (original utterance, entity name, context of query, as well as information from KG) which helps it outperform other approaches.

Semantic recall is not accurate for a few cases, such as when there are multiple valid matches due to the query not constraining the set enough (Query: "The restaurant we ordered asian food from", Recalled Entity: "Skewers By Morimoto", Correct Entity: "Kobe Sushi Bar Restaurant", both are Asian food restaurants) and when the recall mistakes some attribute (Query: "The restaurant I called in Gainesville", Recalled Entity: "Poor Richard's Gainesville", Correct Entity: "Fuji Hibachi Express"). The latter is owing to model's reliance on lexical features. Features capturing semantics like word embeddings can help fix such cases. The former essentially requires presenting multiple options to the user for disambiguation rather than choosing a single entity.

6. RELATED WORK

Referring entity expressions (descriptive, anaphoric as well as deitic) have been extensively studied in dialog systems [5, 6, 7, 8, 9]. The resolution of REs for disambiguation for onscreen items has also been studied [10]. In both these cases, the focus is to identify entities from the recent session/past whereas our work is focused on longer term recall and reference resolution.

KGs have been used to improve accuracy of entity disambiguation [11], spoken language understanding [12] and query lookup [13, 14]. We leverage KGs for recalling and resolving past entities of interest. Li *et al.*, [15] build a personal KG based on user assertions where the user explicitly provides information. We do not rely on explicit assertions and store all implicit entities for future reference resolution. Sansonnet [16] identified automatically building memory for digital agents as a key challenge. This includes the ability to query the agent about its current state (possibly influenced by a previous user interaction) [17] using natural language [18] over possibly a structured information store [19, 20] . We have presented a way to address this challenge for PDAs.

7. CONCLUSION AND FUTURE WORK

In this paper, we presented an approach to add semantic memory to PDAs to recall entities from the user's past. We improve the semantic recall by enriching the entities using a knowledge graph lookup. We adopt an information retrieval based approach to rank the entities and then determine which of the ranked entities should be displayed to the user. We sample location entities from Cortana logs and then ask human judges to generate expressions of how a user would like to recall these entities. Using this data, we construct domain agnostic models for semantic recall and demonstrate it is possible to achieve 89.8% accuracy. This includes cases where the recall expressions only have some attribute of the entity mentioned rather than the full entity name. The model is able to answer such queries by enriching the entities from additional information available through a KG.

This effectively creates a personalized knowledge repository that can be extended for various other purposes such as storing user preferences (automatically add entities for non-recall queries such as drive me to the hospital) or even perform complex inference on these entities to accomplish more complicated tasks (e.g., give me directions to the Italian restaurant I asked about last time). This work can also be extended to address temporal queries (the pizza place I asked about last month) by tagging temporal expressions and then constraining the set of entities to be ranked using these timebased constraints. Word embeddings improve accuracy of disambiguation [21, 22] and KGs have been used to construct word embeddings [23, 24, 25, 26]. We plan to investigate their effect on the accuracy of semantic recall.

8. REFERENCES

- [1] Ruhi Sarikaya, "The technology behind personal digital assistants," in *IEEE Signal Processing Magazine*, 2017.
- [2] Ruhi Sarikaya et al., "Overview of end-to-end natural language understanding and dialog management for personal digital assistant," in *IEEE SLT*, 2016.
- [3] Stephen Robertson and Hugo Zaragoza, "The probabilistic relevance framework: BM25 and beyond," *Found. Trends Inf. Retr.*, vol. 3, no. 4, pp. 333–389, 2009.
- [4] Tie-Yan Liu, "Learning to rank for information retrieval," *Found. Trends Inf. Retr.*, vol. 3, no. 3, pp. 225– 331, Mar. 2009.
- [5] Norbert Pfleger and Jan Alexandersson, "Towards resolving referring expressions by implicitly activated referents in practical dialogue systems," in *SEMDIAL*, 2006.
- [6] Srinivasan Janarthanam and Oliver Lemon, "Learning to adapt to unknown users: Referring expression generation in spoken dialogue systems," in ACL, 2010, pp. 69–78.
- [7] Konstantina Garoufi and Alexander Koller, "Combining symbolic and corpus-based approaches for the generation of successful referring expressions," in *Workshop* on Natural Language Generation, 2011, pp. 121–131.
- [8] Kotaro Funakoshi Mikio Nakano and Takenobu Tokunaga Ryu Iida, "Modeling referring expressions with Bayesian networks," in *SEMDIAL*, 2012.
- [9] Ingrid Zukerman, Su Nam Kim, Thomas Kleinbauer, and Masud Moshtaghi, "Employing distance-based semantics to interpret spoken referring expressions," *Comput. Speech Lang.*, vol. 34, no. 1, pp. 154–185, 2015.
- [10] Asli Celikyilmaz, Zhaleh Feizollahi, Dilek Z Hakkani-Tür, and Ruhi Sarikaya, "Resolving referring expressions in conversational dialogs for natural user interfaces.," in *EMNLP*, 2014, pp. 2094–2104.
- [11] Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer, "Robust and collective entity disambiguation through semantic embeddings," in *SIGIR*, 2016, pp. 425–434.
- [12] Laurent Mazuel and Nicolas Sabouret, "Generic natural language command interpretation in ontology-based dialogue systems," in *WI-IAT*, 2006, pp. 347–350.
- [13] Yi Ma, Paul A Crook, Ruhi Sarikaya, and Eric Fosler-Lussier, "Knowledge graph inference for spoken dialog systems," in *ICASSP*, 2015, pp. 5346–5350.

- [14] Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky, "Jointly modeling inter-slot relations by random walk on knowledge graphs for unsupervised spoken language understanding," in *NAACL*, 2015, pp. 619–629.
- [15] Xiang Li, Gokhan Tur, Dilek Hakkani-Tür, and Qi Li, "Personal knowledge graph population from user utterances in conversational understanding," in *Spoken Language Technology Workshop (SLT)*, 2014, pp. 224–229.
- [16] Jean-Paul Sansonnet, "Building interaction-based memories in a dialogical world," in *Workshop on distributed collective practices (DCP02)*, 2002.
- [17] Nicolas Sabouret and Jean-Paul Sansonnet, "Automated answers to questions about a running process," in *Proc. CommonSense*, 2001, pp. 217–227.
- [18] Philipp Cimiano, Peter Haase, Jörg Heizmann, Matthias Mantel, and Rudi Studer, "Towards portable natural language interfaces to knowledge bases-the case of the orakel system," *Data & Knowledge Engineering*, vol. 65, no. 2, pp. 325–354, 2008.
- [19] Nicolas Sabouret and Jean-Paul Sansonnet, "Querying knowledge about actions in the semantic web," in *Proc. Semantic Web workshop at KR*, 2002.
- [20] Valentin Tablan, Danica Damljanovic, and Kalina Bontcheva, "A natural language query interface to structured information," in *European Semantic Web Conference*. Springer, 2008, pp. 361–375.
- [21] Ikuya Yamada, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji, "Joint learning of the embedding of words and entities for named entity disambiguation," in *CoNLL*, 2016.
- [22] Hongzhao Huang, Larry Heck, and Heng Ji, "Leveraging deep neural networks and knowledge graphs for entity disambiguation," *preprint arXiv:1504.07678*, 2015.
- [23] Antoine Bordes, Jason Weston, Ronan Collobert, and Yoshua Bengio, "Learning structured embeddings of knowledge bases," in *Conference on Artificial Intelli*gence, 2011, number EPFL-CONF-192344.
- [24] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen, "Knowledge graph and text jointly embedding.," in *EMNLP*. Citeseer, 2014, pp. 1591–1601.
- [25] Jiang Bian, Bin Gao, and Tie-Yan Liu, "Knowledgepowered deep learning for word embedding," in *ECML*, 2014, pp. 132–148.
- [26] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *ICLR*, 2015.