PERSONAL KNOWLEDGE GRAPH POPULATION FROM USER UTTERANCES IN CONVERSATIONAL UNDERSTANDING

Xiang Li[1,2] *Gokhan Tur*[1] *Dilek Hakkani-Tür*[1] *Qi Li*[1,3]

[1] Microsoft Research
[2] New York University
[3] Rensselaer Polytechnic Institute

xiangli@cs.nyu.edu, gokhan.tur@ieee.org, dilek@ieee.org, liq7@rpi.edu

ABSTRACT

Knowledge graphs provide a powerful representation of entities and the relationships between them, but automatically constructing such graphs from spoken language utterances presents the novelty and numerous challenges. In this paper, we introduce a statistical language understanding approach to automatically construct personal (user-centric) knowledge graphs in conversational dialogs. Such information has the potential to better understand the users' requests, fulfilling them, and enabling other technologies such as developing better inferences or proactive interactions. Knowledge encoded in semantic graphs such as Freebase has been shown to benefit semantic parsing and interpretation of natural language utterances. Hence, as a first step, we exploit the personal factual relation triples from Freebase to mine natural language snippets with a search engine, and the resulting snippets containing pairs of related entities to create the training data. This data is then used to build three key language understanding components: (1) *Personal Assertion Classification* identifies the user utterances that are relevant with personal facts, e.g., *"my mother's name is Rosa"*; (2) *Relation Detection* classifies the personal assertion utterance into one of the predefined relation classes, e.g., *"*`parents`*"*; and (3) *Slot Filling* labels the attributes or arguments of relations, e.g., *"*`name(parents)`*:Rosa"*. Our experiments using the Microsoft conversational understanding system demonstrate the performance of this proposed approach on the population of personal knowledge graphs.

Index Terms— spoken language understanding, knowledge graph, personal assertion, relation detection, slot filling

1. INTRODUCTION

With the rapid proliferation of smart phones aligned with advances in automatic speech recognition (ASR) and machine learning technologies, virtual personal assistant (VPA) systems, such as Apple Siri and Microsoft Cortana, have started to emerge. These systems are typically more complex than applications like voice search or voice messaging, and require advanced spoken language understanding (SLU) capabilities, which are robust to variability in natural language, ASR noise, and spontaneous ungrammatical spoken input.

In VPA systems, at each turn, a user's speech, S_i, is recognized, and then the SLU component semantically parses that into a task-specific semantic representation of the user's intention, U_i, (e.g., *play music* or *check weather*) with associated arguments (e.g., *name of the artist* or *location*) [1]. Since SLU is not a single stand-alone technology like speech recognition or synthesis, there is no established definition of a semantic parse and depends on the task, domain, or application. The dialog manager then interprets U_i and decides on the most appropriate system action, A_i, exploiting semantic context, user specific meta-information, such as geo-location and personal preferences, and other contextual information. For example, if the user clicks on a map on the screen and says "*How much is the cheapest gas around here?*", the system should be able to interpret the domain, intent, and the associated arguments [2], like:

> *Domain*: Local Business; *Intent*: Get_Price
> *Slots*: *good*: gas; *cost_relative*: cheapest; *location*: (lat,long)

Typically, spoken dialog queries to a dialog system may be classified as *informational*, *transactional*, and *navigational* in a similar way to the taxonomy for web search [3]. Informational queries seek an answer to a question, such as "*find the movies of a certain genre and director*", transactional queries aim to perform an operation, such as "*play a movie*", or "*reserve a table at a restaurant*", and navigational queries aim to navigate in the dialog, such as "*go back to the previous results*". However, in the VPA systems, in addition to these three main categories, more and more *personal assertion* utterances are conveyed from the users, where users are talking about themselves (e.g., "*I am vegetarian*" or "*My daughter is getting married*". In such utterances, instead of instructing the VPA to perform some unambiguous specific intents in users' minds, users interact with the VPA in a more intimate way. This is an uncharted area of research in the SLU literature, since the users have no intention.

More formally, an assertion is defined as a declarative sentence (instead of imperative, interrogative, or any other types). The personal assertion sentences are more focused on describing the personal facts, where the subject of the sentence is either the user (i.e., "*i*") or somebody/something related to the user (i.e., "*my wife*", "*my birthday*", etc.). While such personal information may vary greatly, as a first step towards processing such personal assertions, we exploit the semantic knowledge graphs of the semantic web [4, 5] and semantic search [6]. A knowledge graph is a collection of triples, which consist of two entities linked by some relation, similar to the well-known predicate/argument structure. An example would be `directed_by` (*Avatar*, *James Cameron*). A commonly used ontology is provided in schema.org, with consensus from academia and major search companies like Microsoft, Google, and Yahoo. In this ontology, the personal relation types, such as education or family are also defined for individuals. Triple stores covering various domains have already emerged, such as freebase.org.

In this study, more specifically, we follow the Freebase semantic knowledge graph schema[1], including 18 types of relations about the *people.person* entity, such as `nationality` (the country (or

[1] `http://www.freebase.com/schema`

Number	Utterance	Relation	Slot
1	*my mother's name is Rosa*	`parents`	`parents` : *Rosa*
2	*my wife her name is Amy*	`spouse_s`	`spouse_s` : *Amy*
3	*my children are Alex and Eileen*	`children`	`children` : *Alex* ; `children` : *Eileen*
4	*I was born on November 17 1991 in New York City*	`date_of_birth` `place_of_birth`	`date_of_birth` : *November 17 1991* `place_of_birth` : *New York City*
5	*I work for Microsoft as a software engineer*	`profession` `employment_history`	`profession` : *software engineer* `employment_history` : *Microsoft*

Table 1. Example Utterances with Semantic Space

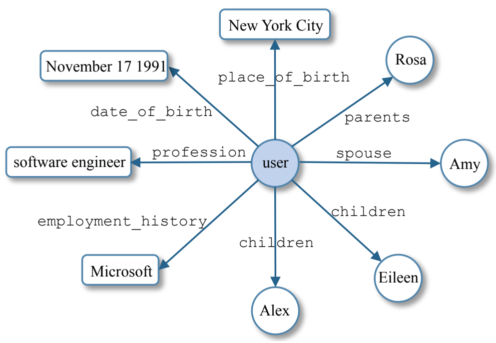

Fig. 1. Example Personal Knowledge Graph

countries) that the person is a citizen of), `profession` (the name of the person's primary occupation(s), during their working life), `parents` (the biological parents and adoptive parents), and so on. A list of the personal factual relations that are encountered in the spoken utterance evaluation dataset is shown in Section 5. For illustration, example utterances with defined semantic space are shown in Table 1, and a sample user-centered knowledge graph is shown in Figure 1 based on these utterances.

For each relation, we leverage the complete set of entities that are connected to each other in the Freebase knowledge graph with the specific relation, and search these entity pairs on the web using Microsoft Bing search engine (`www.bing.com`). We use the snippets that the search engine returns to create natural language examples that can be used as the training data for each relation, based on our earlier work [7]. We further refine and augment the annotations of these examples, which is similar to [8, 9].

This paradigm of constructing personal knowledge graphs in SLU can advance the user experiences, since the SLU component knows more about the user's relationships and behaviors. In addition to customizing knowledge about users, it can also help enhance the performance of SLU systems from many aspects. For example, the SLU component may not appropriately respond to an utterance like "*show me the direction to my daughter's school*" previously. But once the SLU has built a user-centered knowledge graph, where "*my daughter's school*" has been associated with the address of the user's daughter's school, the SLU is able to interpret more utterances and act accordingly by taking the advantages of possessing more knowledge about the user. Moreover, once the VPA constructs a user-centric knowledge graph for each user, then a global knowledge network may be populated by aggregating and integrating personal knowledge graphs through entity linking. For these reasons, we are highly motivated to research on this task.

2. RELATED WORK

Conventional SLU approaches typically focus on user intent determination and slot filling tasks. Intent determination systems have roots in call routing systems used in call centers (e.g., *Billing* vs. *Sales*), such as the AT&T How May I Help You system [10]. They are usually modeled as an utterance classification task aiming at classifying a given speech utterance S_i into one of M semantic classes, $\hat{C}_r \in \mathcal{C} = \{C_1, ..., C_M\}$ (where r is the utterance index). To this end, researchers have tried various classification methods such as Boosting [11, 12, 13], support vector machines (SVMs) [14], and more recently deep learning [15, 16].

On the other hand, slot filling systems have flourished after DARPA sponsored Airline Travel Information System (ATIS) [17] project. These systems attempted to convert the user utterance into an SQL query. The approaches ranged from generative models such as hidden Markov models [18, 19], discriminative classification methods [20, 21, 22], knowledge-based methods, probabilistic context free grammars [23, 24], and more recently deep learning methods [25, 26, 27]. Recently, the state of the art approach for slot filling is framing the task as a sequence classification problem, similar to part of speech tagging or named entity extraction, in order to find both the boundaries and labels of phrases which are used to fill the semantic template. The non-slot filler words are assigned to a special null state.

Similar to the slot filling task defined in SLU, another Slot Filling task is constructed in the Knowledge Base Population (KBP) track, organized by U.S. NIST's Text Analysis Conference (TAC) [28]. The KBP Slot Filling (SF) task aims at collecting from a large-scale multi-source corpus the values ("slot fillers") for certain attributes ("slot types") of a query entity, which is a person or some type of organization. KBP2013 has defined 25 slot types for persons (per) (e.g., age, spouse, employing organization) and 16 slot types for organizations (org) (e.g., founder, headquarters-location, and subsidiaries). Some slot types take only a single slot filler (e.g., per:birth_place), whereas others take multiple slot fillers (e.g., org:top_employees). More information can be found in the task definition [29]. Various approaches have been proposed to perform the task, including information extraction [30], question answering [30, 31], hand-coded heuristic rules [32, 33], pattern matching [34], distant supervision [35, 36, 37, 38, 34], hybrid [30, 34], knowledge graph based [39], etc.

As we know, knowledge graphs have been demonstrated useful and powerful in many conversational understanding research tasks. [7, 40] compute entity type weights to enrich semantic knowledge graph entities with probabilistic weights for the SLU relation detection task. [41] proposes a technique to enable SLU systems to handle user queries beyond their original semantic schemas defined by intents and slots. [42] presents a full pipeline to leverage semantic web search and browse sessions for a semantic parsing problem in

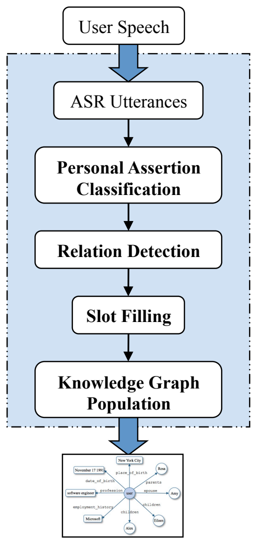

Fig. 2. Framework of Personal Knowledge Graph Construction

multi-turn spoken dialog systems. [43, 44] present studies towards bringing together the semantic web experience and unsupervised statistical natural language semantic parsing modeling. [9] proposes an unsupervised training approach for SLU systems on the intent detection task, which exploits the structure of semantic knowledge graphs from the web.

3. FRAMEWORK

In this work, we align our SLU semantic space with the back-end semantic knowledge repositories such as Freebase and aim to identify knowledge graph relations invoked in users utterances. To achieve this goal, we propose the statistical language understanding framework, as shown in Figure 2, with three key language understanding components: *Personal Assertion Detection*, *Relation Detection*, and *Slot Filling*. Each of these components will be introduced in detail in the following respective subsection.

3.1. Personal Assertion Classification

This component aims to classify the spoken utterances into binary classes according to the fact that the utterance depicts personal facts. For example, one positive case could be like "*i was born in 1999*", and, on the other hand, an instance in the negative class could be similar to "*how is the weather today?*". We formulate this problem as a binary classification task and apply Support Vector Machines (SVM) [45, 46] framework to perform the classification.

SVMs, in their most basic formulation, are a binary classification method based on the intuition of maximizing the margin around the classification boundary. Given a training set of instance-label pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, l$ where $\mathbf{x}_i \in R^n$ and $\mathbf{y} \in \{1, -1\}^l$, the Support Vector Machines require the solution of the following optimization problem:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_i$$

$$\text{subject to} \quad y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq 1 - \xi_i,$$

$$\xi_i \geq 0.$$

where training vectors $\mathbf{x}_i$ are mapped into a higher (maybe infinite) dimensional space by the function ϕ. SVM finds a linear separating hyperplane with the maximal margin in this higher space. $C > 0$ is the penalty parameter of the error term.

We use linear kernels, $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T\mathbf{x}_j$, as provided in the SVM$^{\text{light}}$ [47] package, since they are extremely efficient. The outputs of this stage provide us with coarse-grained information on whether we could further extract fine-grained personal factual relations from next two levels.

3.2. Relation Detection

Relation detection aims to determine with relations in the part of knowledge graph related to the utterance has been invoked in the user utterances. For example, Table 1 shows example utterances that invoke various relations in the knowledge graph, and one utterance can also invoke more than one relations. Hence, the detection of the relation as being invoked in the utterance is necessary for formulating the query to the back-end. We frame this subtask as a multi-class classification problem, and we also apply the SVM$^{\text{light}}$ package to classify each utterance into one or more relation classes. But instead of directly using the extended algorithm, SVM$^{\text{multiclass}}$, for multi-class scenarios, we still apply the binary, linear kernels in the SVM$^{\text{light}}$ package through a *one-vs-rest* approach. We construct k SVM models where k is the number of relation classes. The ith SVM is trained with all the examples in the ith class with positive labels, and all other examples with negative labels. Then apply all k SVM models on each utterance to determine which relations are invoked in it. Depending on whether in-domain annotated data is available or not, the models trained using training data for each relation can be used in two ways:

- Case 1: (Supervised Baseline) Use only the in-domain annotated data for training and testing;
- Case 2: (Unsupervised) In cases where there is absolutely no in-domain annotated data, the distantly mined data can be used to build relation detection SVM models;

The formulation of the complete query to the back-end requires detection of the invoked entities in the users utterance, in addition to detecting the graph relations that are invoked. Hence, we will extract the specific entities or arguments of detected relations with the following Slot Filling component.

3.3. Slot Filling

The semantic structure of an application domain is defined in terms of the semantic frames. The semantic frame contains several typed components called "slots". The task of slot filling is then to instantiate the semantic frames. Check Table 1 for slot filling in the example utterances. In this case, the semantic frame is represented as a flat list of attribute-value pairs, similar to [48].

Following the state-of-the-art approaches for slot filling [49, 50, among others], we use discriminative statistical models, namely Conditional Random Fields (CRFs) [51], for modeling. More specifically and formally, slot filling is framed as a sequence classification problem to obtain the most probable slot sequence:

$$\hat{Y} = \underset{Y}{\text{argmax}}\, p(Y|X)$$

where $X = x_1, ..., x_T$ is the word sequence and $Y = y_1, \ldots, y_T$, $y_i \in C$ is the sequence of associated class labels C.

CRFs are shown to outperform other classification methods for sequence classification [1], since the training can be done discriminatively over a sequence with sentence level optimization. The baseline model relies on a word n-gram based linear chain CRF, imposing the first order Markov constraint on the model topology. Similar to maximum entropy models, in this model, the conditional probability, $p(Y|X)$ is defined as [51]:

$$p(Y|X) = \frac{1}{Z(X)} \exp(\sum_k \lambda_k f_k(y_{t-1}, y_t, x_t))$$

with the difference that both X and Y are sequences instead of individual local decision points given a set of features f_k (such as n-gram lexical features, state transition features, or others) with associated weights λ_k. $Z(X)$ is the normalization term. After the transition and emission probabilities are optimized, the most probable state sequence, $\hat{Y}$, can be determined using the well-known Viterbi algorithm [52]. In this study, we follow the popular IOB (in-out-begin) format in representing the data and use CRF++[2], an open source implementation of CRFs.

3.4. Knowledge Graph Population

Once the relations and the associated entities or arguments are identified from the utterances, the user-centered personal knowledge graph would be populated with the newly extracted information and get updated. Then if the user intends to talk more about himself/herself, the system will repeat the above procedures to integrate more personal facts into the current knowledge graphs. Otherwise, the system can either output the knowledge graph for demonstration with better visibility or store the knowledge graph into a knowledge repository such as a knowledge base or database.

4. DATA COLLECTION

In this study, we utilize the semantic space that is defined in a knowledge base, or a triple store, such as *people.person* related facts in Freebase, for the SLU model to be built. A triple typically consists of two entities linked by some relation, similar to the well-known predicate/argument structure. An example would be `place_of_birth` (*Bill Gates*, *Seattle*).

These semantic ontologies are not only used by search engines, which try to semantically parse them, but also by the authors of the in-domain web pages (such as `imdb.com`) for better visibility. While the details of the semantic web literature is beyond the scope of this paper, it is clear that these kinds of semantic ontologies are very close to the semantic ontologies used in goal-oriented natural dialog systems and there is a very tight connection between the predicate/argument relations and intents, as explained below.

To create a training data set for our framework, we mine training examples by searching entity pairs that are related to each other in the knowledge graph on the web. As in our earlier work [9, 7], we extract a set of entity pairs in a given domain that are connected with a specific relation from the knowledge base[3]. Our approach for mining examples guided by relations in the knowledge base is similar to [53], but we directly detect relations invoked in user utterances, instead of parsing utterances with a combinatory categorical grammar [54]. Furthermore, we enhance our data with web search queries which are inquiring similar information as dialog system users.

Assume AS is the set of all snippets returned for the pair of entities a and b via web search[4]. We choose a subset of AS, SAS, that include snippets with both entities: $SAS = \{s : s \in AS \wedge includes(s, a) \wedge includes(s, b)\}$, where $includes(x, y)$ is a binary function that has a value of 1 if string x contains y

[2] `http://crfpp.googlecode.com`

[3] `http://www.freebase.com`

[4] In this work, we use Bing search engine and download the top 10 results for each entity pair.

Category	Data	Number
Positive	web mined snippets	$72,820$
	pattern mined utterances	$12,989$
Negative	Cortana domain data	$150,915$

Table 2. Training Data for Personal Assertion Classification

as a substring. One approach is using the complete strings of the snippets for each relation as training examples. However, the snippets can contain more than one correct relation tuples. In order to capture more relations in the mined snippet sentences, we apply an additional procedure to post-process these sentences to augment the relation tags from Freebase, since many crawled instances actually contain more than one relations. (Even though we cannot guarantee that the augmented relations are "complete", because the Freebase is not complete as well as our collected data.) For example, we extract two relations regarding "Jacques Berthier", which are `date_of_birth`(*February 10, 1916*) and `place_of_birth`(*Paris, France*). This newly added step would generate two following instances with all corresponding tags rather than two instances with incomplete tags: *Jacques Berthier was born on* <`date_of_birth`>*February 10, 1916*</`date_of_birth`> *in* <`place_of_birth`>*Paris, France*</`place_of_birth`>.

5. EXPERIMENTS

5.1. Evaluation Dataset

We first create a set of test examples to evaluate each key component of the proposed framework. To extract a set of testing instances, we have collected a total of 10 million utterances from Microsoft conversational understanding, Cortana, query logs. In order to mine real cases that are personal assertions and contain personal factual relations, we use 7 simple yet general patterns to extract a candidate pool, where the patterns are "*i am a **", "*i am from **", "*i have a **", "*i live **", "*i was born **", "*i work **", and "*my **". Then we randomly sample a subset of the pooled candidate utterances, and manually annotated each utterance with three levels of annotations, corresponding to the three main components of our proposed framework: (1) investigate whether the utterance is a personal assertion; (2) identify the relations invoked in the utterance; and (3) tag the entities or argument of the invoked relations in the utterance. The final set of annotated data consists of $12,989$ examples about personal assertions, among which $1,811$ utterances contain at least one of the predefined relations, while the remaining $11,178$ instances do not. We then experimentally investigate the performance of each key component based on this evaluation data set.

5.2. Personal Assertion Classification

To evaluate the performance of the Personal Assertion Classification component, a 10-fold cross-validation approach is applied on a combined data set, which contains the automatically mined snippets from the web, the annotated utterances from Cortana query logs, and a subset of Cortana related in-domain data. The Cortana related in-domain data consists utterances in 7 distinct domains such as "weather" or "calendar". We use this data as negative assertion examples, while we label both snippets and annotated utterances as positive training data. Table 2 shows the number of the examples from each data source. Then the data set is randomly split into 10 equal size subsamples. Of the 10 subsamples, a single subsample is retained as the validation data for testing the model, and the remaining 9 subsamples are used as training data. The cross-validation process is then repeated 10 times (the folds), with each of the 10

Relation Type	Count	Relation Detection		Slot Filling		
		unsupervised	supervised	supervised		
		Precision@Count (%)	Precision@Count (%)	Precision (%)	Recall (%)	F-Measure (%)
place_of_birth	8	0.00	0.00	0.00	0.00	0.00
religion	8	0.00	50.00	0.00	0.00	0.00
ethnicity	17	0.00	70.59	100.0	17.65	30.00
employment_history	40	7.50	52.50	50.00	12.50	20.00
nationality	47	0.00	63.83	75.00	82.98	78.79
profession	61	0.00	54.10	50.00	1.64	3.72
gender	63	6.35	82.54	90.91	47.62	62.50
date_of_birth	73	46.58	75.34	56.25	36.99	44.63
places_lived	121	2.48	68.59	69.91	65.29	67.52
sibling_s	248	86.29	90.32	85.92	71.08	77.80
children	260	23.08	87.31	80.92	47.31	59.71
parents	401	19.95	86.78	83.97	65.17	73.39
spouse_s	464	82.11	94.39	86.81	68.10	76.33
Total	1811	42.85	84.32	82.01	58.58	68.34

Table 3. Performance of Relation Detection and Slot Filling

subsamples used exactly once as the validation data. The 10 results from the folds are then combined to produce a single estimation. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. Among total $236,724$ data samples, $234,650$ instances are correctly classified while only $2,074$ are classified with wrong class labels, which achieves 99.12% accuracy. This demonstrates the reliable performance of this SVM-based Personal Assertion classifier.

5.3. Relation Detection

In order to measure the quality and effectiveness of Relation Detection component, the models have been trained using the snippets mined from the web and the annotated Cortana utterances in two scenarios, depending on whether in-domain annotated data is available or not:

- Case 1: (Supervised Baseline) Only use the in-domain annotated Cortana utterances for both training and testing, where a 2-fold cross-validation (handout) approach is applied. For each fold, annotated utterances are randomly assigned to two sets d_0 and d_1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). Then the model is trained on d_0 and tested on d_1, following by being trained on d_1 and tested on d_0. This has the advantage that our training and test sets are both large, and each data point is used for both training and validation;
- Case 2: (Unsupervised) To mimic the cases where there is absolutely no in-domain annotated spoken data, the snippets crawled from the web are used to build models, and gauge the model performance on the annotated Cortana utterances;

For evaluation, we used Precision@N (P@N), where N is the number of positive examples for that relation in the test set. Table 3 shows the detailed results in each above case, where only n-gram features are used. The supervised method provides the upper bound of 84.32% P@N, based on manual annotations. Using the proposed unsupervised approach results in a bootstrap model achieving 42.85% P@N overall. However for certain classes such as `sibling` or `spouse`, the model has performed on par with the supervised approach. For relations, requiring a named entity such as location for `place_of_birth` or date for `date_of_birth`, we plan to use a generic named entity tagger to improve the performance. This is left as future research. Another promising direction is adapting this bootstrap model with supervised data, using an online learning mechanism, drawing learning curves for each relation. We suspect that with few manually tagged examples, some relation types may improve significantly, such as `employment_history`.

5.4. Slot Filling

The Slot Filling results in each above case are also shown in Table 3. For slot filling we only used the supervised approach, since the semantic annotation mechanisms of the snippets and the evaluation set are different, as they belong to different genre (e.g., *Jacques Berthier is the son of* <`parents`>*Paul Berthier*<`parents`> vs. *my* <`parents`>*father*<`parents`> *is old*). For evaluation, the slot F-measure is used, following the literature [49] using the CoNLL evaluation script[5]. We can see that the supervised approach can achieve 68.34% F-measure in the overall performance. For most relation types, where the context is obvious, the system achieves reasonable performance levels with minimal annotations. There are few relation types, where the task is nontrivial such as `profession` relation, since `profession` may get invoked with a much larger pool of expressions, such as "computer research scientist", "helicopter trainer", "international standard ballroom dancer", and so on, which cannot easily get trained from a small in-domain data. As part of future research, we plan to extract these patterns from the automatic annotations we mined from the snippets. Similarly, the named entity features would help improving the overall performance.

6. CONCLUSION

In this paper, we have presented a novel SLU framework aiming to construct personal (user-centric) knowledge graphs in spoken utterances. This approach contains three main language understanding components: *Personal Assertion Classification*, *Relation Detection*, and *Slot Filling*. Our experimental results have proven the effectiveness of the proposed scheme on all three levels. While relation detection and slot filling have been studied in many SLU tasks, to the best of our knowledge, this is a pioneering study for systematically building personal knowledge graphs in human/machine conversational systems.

[5] `http://www.cnts.ua.ac.be/conll2000/chunking/output.html`

Since the current slot filling approach cannot handle the utterances that involve two or more links, we plan to integrate an inference scheme into the framework to solve sophisticated relations invoked in the utterances. Given "*my wife was born in China*", for example, directly link `place_of_birth`:*China* to `spouse_s` node. We are also interested in exploring the personal preferences depicted in the utterances, such as "*I am vegetarian*", since we believe this `interested_in`-style relation could enhance the performance of VPA to a great extent, like recommending appropriate restaurants in this case. In addition, we find that it is also very important to identify the negation expression and its scope within the utterances, which is crucial to determine whether a relation should be populated into the knowledge graph. We plan to boost our proposed framework towards these directions in the future work.

7. REFERENCES

[1] Gokhan Tur and Renato DeMori, *Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, John Wiley and Sons, New York, NY, 2011.

[2] G. Tur, Y.-Y. Wang, and D. Hakkani-Tür, *Computing Handbook, Third Edition*, chapter Understanding Spoken Language, Springer, 2014.

[3] Andrei Broder, "A taxonomy of web search," *SIGIR FORUM*, vol. 36, no. 2, pp. 3–10, 2002.

[4] S. A. McIlraith, T. C. Sun, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems*, pp. 46–53, 2001.

[5] N. Shadbolt, W. Hall, and T. Berners-Lee, "The semantic web revisited," *IEEE Intelligent Systems*, pp. 96–101, 2006.

[6] R. Guha, R. McCool, and E. Miller, "Semantic search," in *Proceedings of the WWW*, Budapest, Hungary, 2003.

[7] Dilek Hakkani-Tür, Larry Heck, and Gokhan Tur, "Using a knowledge graph and query click logs for unsupervised learning of relation detection," in *the Proceedings of the ICASSP 2013*. IEEE, 2013.

[8] G. Tur, M. Jeong, Y.-Y. Wang, D. Hakkani-Tür, and L. Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *In Prooceedings of the Interspeech*, Portland, OR, September 2012.

[9] Larry P. Heck and Dilek Hakkani-Tür, "Exploiting the semantic web for unsupervised spoken language understanding," in *Proceedings of the IEEE Workshop on Spoken Language Technology (SLT 2012)*, 2012, pp. 228–233.

[10] A. L. Gorin, G. Riccardi, and J. H. Wright, "How May I Help You?," *Speech Communication*, vol. 23, pp. 113–127, 1997.

[11] R. E. Schapire and Y. Singer, "Boostexter: A boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2/3, pp. 135–168, 2000.

[12] N. Gupta, G. Tur, D. Hakkani-Tür, S. Bangalore, G. Riccardi, and M. Rahim, "The AT&T spoken language understanding system," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 1, pp. 213–222, 2006.

[13] I. Zitouni, H.-K. J. Kuo, and C.-H. Lee, "Boosting and combincation of classifiers for natural language call routing systems," *Speech Communication*, vol. 41, no. 4, pp. 647–661, 2003.

[14] P. Haffner, G. Tur, and J. Wright, "Optimizing SVMs for complex call classification," in *Proceedings of the ICASSP*, Hong Kong, April 2003.

[15] R. Sarikaya, G. E. Hinton, and B. Ramabhadran, "Deep belief nets for natural language call-routing," in *Proceedings of the ICASSP*, 2011.

[16] Y. Dauphin, G. Tur, D. Hakkani-Tür, and L. Heck, "Zero-shot learning and clustering for semantic utterance classification," in *Proceedings of the ICLR*, 2014.

[17] P. J. Price, "Evaluation of spoken language systems: The ATIS domain," in *Proceedings of the DARPA Workshop on Speech and Natural Language*, Hidden Valley, PA, June 1990.

[18] R. Pieraccini, E. Tzoukermann, Z. Gorelov, J.-L. Gauvain, E. Levin, C.-H. Lee, and J. G. Wilpon, "A speech understanding system based on statistical representation of semantics," in *Proceedings of the ICASSP*, 1992.

[19] Y. He and S. Young, "A data-driven spoken language understanding system," in *Proceedings of the IEEE ASRU Workshop*, 2003, pp. 583–588.

[20] R. Kuhn and R. De Mori, "The application of semantic classification trees to natural language understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, pp. 449–460, 1995.

[21] Y.-Y. Wang and A. Acero, "Discriminative models for spoken language understanding," in *Proceedings of the ICSLP*, Pittsburgh, PA, September 2006.

[22] G. Tur, D. Hakkani-Tür, and L. Heck, "What is left to be understood in ATIS?," in *Proceedings of the IEEE SLT Workshop*, Berkeley, CA, 2010.

[23] S. Seneff, "TINA: A natural language system for spoken language applications," *Computational Linguistics*, vol. 18, no. 1, pp. 61–86, 1992.

[24] W. Ward and S.Issar, "Recent improvements in the CMU spoken language understanding system," in *Proceedings of the ARPA HLT Workshop*, 1994.

[25] L. Deng, G. Tur, X. He, and D. Hakkani-Tür, "Use of kernel deep convex networks and end-to-end learning for spoken language understanding," in *In Prooceedings of the IEEE SLT Workshop*, Miami, FL, December 2012.

[26] K. Yao, B. Peng, G. Zweig, D. Yu, X. Li, and F. Gao, "Recurrent conditional random field for language understanding," in *Proceedings of the IEEE ICASSP*, 2014.

[27] P. Xu and R. Sarikaya, "Convolutional neural network based triangular crf for joint intent detection and slot filling," in *Proceedings of the IEEE ASRU*, 2013.

[28] Heng Ji and Ralph Grishman, "Knowledge base population: Successful approaches and challenges," in *Proceedings of ACL*, 2011, HLT '11, pp. 1148–1158.

[29] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis, "Overview of the tac2010 knowledge base population track," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[30] Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji, "Cuny-blender tackbp2010 entity linking and slot filling system description," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[31] Lorna Byrne and John Dunnion, "Ucd iirg at tac 2010," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[32] Sanyuan Gao, Yichao Cai, Si Li, Zongyu Zhang, Jingyi Guan, Yan Li, Hao Zhang, Weiran Xu, and Jun Guo, "Pris at tac2010 kbp track," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[33] Jingtao Yu, Omkar Mujgond, and Rob Gaizauskas, "The university of sheffield system at tac kbp 2010.," in *Proceedings of Text Analytics Conference (TAC)*, 2010.

[34] Bonan Min, Xiang Li, Ralph Grishman, and Ang Sun, "New york university 2012 system for kbp slot filling," in *Proceedings of Text Analytics Conference (TAC)*, 2012.

[35] David Nemeskey, Gabor Recski, Attila Zseder, and Andras Kornai, "Budapestacad at tac 2010," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[36] Grzegorz Chrupala, Saeedeh Momtazi, Michael Wiegand, Stefan Kazalski, Fang Xu, Benjamin Roth, Alexandra Balahur, and Dietrick Klakow, "Saarland university spoken language systems at the slot filling task of tac kbp 2010," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[37] Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning, "A simple distant supervision approach for the tac-kbp slot filling task," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[38] Ander Intxaurrondo, Oier Lopez de Lacalle, and Eneko Agirre, "Ubc at slot filling tac-kbp2010," in *Proceedings of Text Analytics Conference (TAC) 2010*, 2010.

[39] Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare Voss, and Malik Magdon-Ismail, "The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding," in *Proceedings of COLING*, 2014.

[40] Dilek Hakkani-Tür, Asli Celikyilmaz, Larry Heck, Gokhan Tur, and Geoff Zweig, "Probabilistic enrichment of knowledge graph entities for relation detection in conversational understanding," IEEE, 2014, ISCA - International Speech Communication Association.

[41] Ali El-Kahky, Derek Liu, Ruhi Sarikaya, Gokhan Tur, Dilek Hakkani-Tur, and Larry Heck, "Extending domain coverage of language understanding systems via intent transfer between domains using knowledge graphs and search query click logs," in *Proceedings of the IEEE ICASSP*, 2014.

[42] Lu Wang, Larry Heck, and Dilek Hakkani-Tur, "Leveraging semantic web search and browse sessions for multi-turn spoken dialog systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) Conference*, 2014.

[43] Gokhan Tur, Minwoo Jeong, Ye-Yi Wang, Dilek Hakkani-Tur, and Larry Heck, "Exploiting the semantic web for unsupervised natural language semantic parsing," in *Proceedings of the ISCA Interspeech Conference*. 2012, International Speech Communication Association.

[44] Larry Heck, Dilek Hakkani-Tur, and Gokhan Tur, "Leveraging knowledge graphs for web-scale unsupervised semantic parsing," in *Proceedings of Interspeech*. 2013, International Speech Communication Association.

[45] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. 1992, pp. 144–152, ACM Press.

[46] Corinna Cortes and Vladimir Vapnik, "Support-vector networks," in *Machine Learning*, 1995, pp. 273–297.

[47] Thorsten Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods*. 1999, pp. 169–184, MIT Press.

[48] Roberto Pieraccini and Esther Levin, "A learning approach to natural language understanding," in *In Speech Recognition and Coding, New Advances and Trends, NATO ASI Series*. 1995, pp. 139–155, Springer Verlag.

[49] Christian Raymond and Giuseppe Riccardi, "Generative and discriminative algorithms for spoken language understanding," 2007.

[50] Ye-yi Wang and Alex Acero, "Discriminative models for spoken language understanding," in *ICSLP*, 2006.

[51] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of ICML '01*, 2001, pp. 282–289.

[52] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[53] Jayant Krishnamurthy and Tom Mitchell, "Weakly supervised training of semantic parsers," in *Proceedings of EMNLP-CoNLL*, 2012, pp. 754–765.

[54] Mark Steedman, *Surface Structure and Interpretation*, The MIT Press, 1996.