

Spoken Dialogue for Information Navigation

Alexandros Papangelis¹, Panagiotis Papadakos², Yannis Stylianou^{1,3}, and Yannis Tzitzikas^{2,3}

¹ Speech Technology Group - Toshiba Research Europe

² Institute of Computer Science - FORTH-ICS, Greece

³ Computer Science Department - University of Crete, Greece

{alex.papangelis, yannis.stylianou}@crl.toshiba.co.uk

{papadako, tzitzik}@ics.forth.gr

Abstract

Aiming to expand the current research paradigm for training conversational AI agents that can address real-world challenges, we take a step away from traditional slot-filling goal-oriented spoken dialogue systems (SDS) and model the dialogue in a way that allows users to be more expressive in describing their needs. The goal is to help users make informed decisions rather than being fed matching items. To this end, we describe the Linked-Data SDS (LD-SDS), a system that exploits semantic knowledge bases that connect to linked data, and supports complex constraints and preferences. We describe the required changes in language understanding and state tracking, and the need for mined features, and we report the promising results (in terms of semantic errors, effort, etc) of a preliminary evaluation after training two statistical dialogue managers in various conditions.

1 Introduction

There has been an increasing amount of research being conducted on many aspects of Spoken Dialogue Systems (SDS) with applications ranging from well-defined goal-oriented tasks to open-ended dialogue, e.g., (Amazon, 2017). Deep learning and joint optimisations of SDS components are becoming the standard approach e.g., (Chen et al., 2017; Li et al., 2016; Williams et al., 2017; Liu et al., 2017; Wen et al., 2017; Cuayáhuitl et al., 2017; Yang et al., 2017), showing many benefits but also limitations and disadvantages. Due to the complexity of the problem, most of these approaches focus on limited applications e.g., information retrieval on small domains or shallow-understanding chat-bots.

Moving towards conversational AI, we shift the paradigm to information navigation and present in this work a more realistic goal-oriented setup. The proposed paradigm is designed towards complex interactions using semantic knowledge bases and linked data (Heath and Bizer, 2011), and allows users to be more expressive in describing their constraints and prefer-

ences. We aim to enable users to make informed decisions by understanding their needs and priorities through conversation with an intelligent agent.

In this work we extend the Linked Data Spoken Dialogue System (LD-SDS) system proposed in (Papangelis et al., 2017) in the following directions: a) we propose features mined over the set and the order of objects in the current user focus, b) we modify the language understanding and belief state tracking modules to support the proposed complex interactions over rich information spaces, c) we apply an agenda-based user simulator to train two statistical dialogue manager models, and d) we conduct a preliminary evaluation with promising results.

2 Challenges and Background

2.1 Challenges and Requirements

As our paradigm moves towards information navigation, we assume that the users have a vague idea of what they are looking for and through interaction with the system they can understand their own needs better. The user's intents, therefore, do not always express hard restrictions (constraints) but often express *preferences*¹ that users may or may not be willing to relax as the dialogue progresses. Such preferences may refer to the *importance of attributes* over other attributes (e.g., *location* is much more important than *has-free-wifi* when searching for accommodation), or may refer to *preferred values* of a given attribute (e.g., prefer *central* over *northern* locations but *northern* may still be okay under certain circumstances), etc. Moreover, it is worth *highlighting* aspects of items that may have not been mentioned but have high *discriminative power* within their cluster (e.g., 5 hotels match the user's preferences but there's one with vegan menu).

Towards this objective, we propose the interaction of SDS with exploratory systems that offer the aforementioned functionality over semantic knowledge bases. This requires extensions in language understanding and state tracking, and the need for mined features.

¹Preferences can be considered as soft constraints or wishes that might or might not be satisfied

2.2 Background: Preference-Enriched Faceted Search and Hippalus

Faceted search is currently the de facto standard in e-commerce (e.g., eBay, booking.com), and its popularity and adoption is increasing. The enrichment of Faceted Search with *preferences*, hereafter *Preference-enriched Faceted Search* (PFS), was proposed in (Tzitzikas and Papadakis, 2013). It has been proven useful for recall-oriented information needs, because such needs involve decision making that can benefit from the gradual interaction and expression of not only restrictions (hard constraints) but also preferences (soft constraints). It is worth noting that it allows expressing preferences over attributes, whose values can be *hierarchically organized* and/or *multi-valued*, it supports *preference inheritance*, and it offers scope-based rules for *automatic conflict resolution*.

PFS offers various preference actions (e.g., relative, best, worst, around, etc.) that allow the user to order facets (i.e. slots), values, and objects. Furthermore, the user is able to *compose* object related preference actions². Essentially, a user u can express gradually a set of qualitative (i.e. relative) preferences over the values of each facet (slot), denoted by $Pref_u$. These actions define a preference relation (a binary relation) over the values V_{s_i} of each slot s_i , denoted by \succ_i , which are then composed to define a preference relation over the elements of the information space, i.e. over $V = V_{s_1} \times \dots \times V_{s_n}$ (in the case of multi-valued slots $V = \mathcal{P}(V_{s_1}) \times \dots \times \mathcal{P}(V_{s_n})$). Since the descriptions of the objects in the current user focus \mathcal{F}_u are a subset of V , the actions in $Pref_u$ define a preference relation over \mathcal{F}_u denoted as $(\mathcal{F}_u, \succ_{Pref_u})$, from which a bucket order of \mathcal{F}_u , i.e. a linear order of subsets of \mathcal{F}_u ranked based on preference and denoted by $B(\mathcal{F}_u, Pref_u) = \langle b_1, \dots, b_z \rangle$, is derived through topological sorting.

Hippalus (Papadakis and Tzitzikas, 2014) is an exploratory search system (publicly accessible³) that materializes PFS over semantic views gathered from different data sources through SPARQL queries. The information base that feeds Hippalus is represented in RDF/S and objects can be described according to dimensions with hierarchically organized and set-valued attributes. Preference actions are validated using the preference language described in (Tzitzikas and Papadakis, 2013). If valid, the system computes the respective preference bucket⁴ order and returns the corresponding ranked list of objects.

In addition, Hippalus implements the scoring function defined in (Tzitzikas and Dimitrakis, 2016), that expresses the degree up to which an object in \mathcal{F}_u fulfills the preferences in $Pref_u$ and is a real

²There are different composition modes like Pareto, Pareto optimal (i.e. skyline), Priority-based, etc.

³<http://www.ics.forth.gr/isl/Hippalus/>

⁴A preference bucket holds incomparable objects with respect to the given soft-constraints

number (in our case its range is the interval $[1, 100]$). The specific scoring function, exploits all available composition modes available in Hippalus enriching the bucket orders with scores respecting the *consistency of the qualitative-based bucket order* that is defined as: A scoring function $score$ is *consistent with the qualitative-based bucket order*, if for any two objects o, o' and any set of user actions $Pref_u$, it holds: if $pos(o) < pos(o')$ then $score(o, Pref_u) > score(o', Pref_u)$ where $pos(o)$ is the position of o in $B(\mathcal{F}_u, Pref_u)$.

3 Features

3.1 Motivation

In order to reduce the complexity of the dialogue system while at the same time improving its efficiency and effectiveness, we enriched the response of the Hippalus system with a number of features, which provide cues about interesting slots/values (as mentioned in §2.1) that can be exploited by the Belief Tracker, Dialogue Manager, Natural Language Generator, and other statistical components of the SDS. These features are extracted from: a) the set of objects of the current user focus (*selectivity* and *entropy*); and b) from the imposed ordering of the objects according to the expressed user preferences (*avg, min and max preference score per bucket* and *pair-wise wins of objects per slot per bucket*).

3.2 Features extracted from object focus

Assume a dataset \mathcal{D} that contains $|\mathcal{O}_{\mathcal{D}}|$ objects, where $\mathcal{F}_u \subseteq \mathcal{O}_{\mathcal{D}}$ is the current focus of the user u (i.e. the objects that satisfy the expressed hard-constraints). Let $S_{|\mathcal{F}_u} = \{s_1, \dots, s_n\}$ denote the set of available slots in \mathcal{D} under focus \mathcal{F}_u and $V_{s_i|\mathcal{F}_u} = \{v_{s_{i1}}, \dots, v_{s_{im}}\}$ denote the set of values for slot $s_i \in S_{|\mathcal{F}_u}$ respectively⁵. We define the following metrics:

Definition 3.1. The *selectivity* of a slot s_i under focus \mathcal{F}_u is defined as:

$$Selectivity(s_i|\mathcal{F}_u) = \frac{|V_{s_i|\mathcal{F}_u}|}{|\mathcal{F}_u|} \quad (1)$$

Definition 3.2. The *entropy* of a slot s_i under focus \mathcal{F}_u is defined as:

$$Entropy(s_i|\mathcal{F}_u) = - \sum_{j=1}^{|V_{s_i|\mathcal{F}_u}|} (P(v_{s_{ij}|\mathcal{F}_u}) * \log_2(\frac{1}{P(v_{s_{ij}|\mathcal{F}_u})})) \quad (2)$$

where $P(v_{s_{ij}|\mathcal{F}_u})$ is the probability of value $v_{s_{ij}}$ in slot s_i under focus \mathcal{F}_u .

Both selectivity and entropy metrics provide insights about the discreteness and the amount of information contained in the values of a specific slot for the objects

⁵The set of values can be hierarchically organized through a subsumption binary relation (V_{s_i}, \leq_i)

under focus \mathcal{F}_u . Selectivity is an inexpensive but rough metric that takes values in $[0, 1]$. If the value of each object for a specific slot is unique, then selectivity is 1 (high selectivity), while it is near 0 for the opposite (low selectivity). On the other hand entropy is a refined but more expensive metric, with bigger values when the probabilities of values in $V_{s|\mathcal{F}_u}$ are equal. Hippalus returns the values of both metrics for each slot of the current user focus \mathcal{F}_u on the fly, along with the pre-computed values for the whole dataset.

3.2.1 Features extracted from object order

Other interesting features can be extracted from the imposed ordering of objects based on the user preferences, including *min*, *max*, and *average preference score* of objects in each bucket, and for each object of a bucket *the sum of pair-wise wins* per each slot over which the user has expressed a preference. The last feature can be used as an indication about the number of wins of each object over all different preference criteria (slots), pinpointing criteria that affect only a small number of objects.

Definition 3.3. The **pair-wise wins PWW** metric under focus \mathcal{F}_u of objects contained in a bucket $b \in B_{\mathcal{F}_u, Pref_u}$ derived by preference actions $Pref_u$ of user u for slot s , is defined as:

$$PWW(b, Pref_{u|s}) = \sum_{o \in b} \sum_{o' \in b} \frac{2 * win_s(o, o')}{|b|(|b| - 1)} \quad (3)$$

where $Pref_{u|s}$ denotes the preference actions of a user u over the slot s and $win_s(o, o') = 1$ if $pos_s(o) < pos_s(o')$, where $pos_s(o)$ is the position of o in $B_{\mathcal{F}_u, Pref_{u|s}}$, else $win_s(o, o') = 0$.

Notice that big PWW values mean that we have a small number of objects, even a single object, that win the rest objects of the bucket for the preference actions of a specific slot. As an example consider a bucket that contains the cheapest hotel. This hotel wins the rest objects of the bucket for the slot *price* and could be used by the dialogue system to ask if *price* is considered more important than the rest slots (i.e. expression of priority). On the other hand lower values mean that we have a number of ties for the objects of a bucket, and that the dialogue system is not able to pin-point specific slots that could further restrict the top-ranked objects.

4 The LD-SDS

Figure 1 shows the architecture of our system. Hippalus is responsible for feeding information regarding the current knowledge view to the SLU and DST components. In addition, it provides the previously mentioned features and the current ranked list of results to the multi-domain policy, and Natural Language Generation (NLG) and Text to Speech (TTS) components respectively. Spoken Language Understanding (SLU) and dialogue state / belief tracking

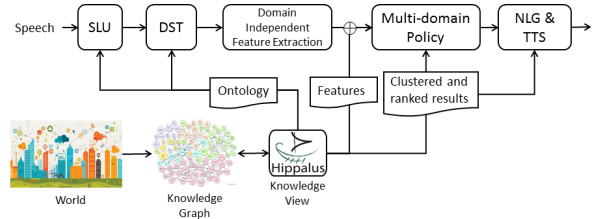


Figure 1: The architecture of our prototype.

(DST / BT), have been extended with operations that correspond to the actions supported by Hippalus. Since Hippalus supports hierarchical and multi-valued attributes, the notion of slot has been extended to allow the definition of relations between slot values.

4.1 Dialogue Management

The objective is to conduct dialogues with as few semantic errors as possible that result in successfully completed tasks and satisfied users. As baselines for dialogue management, we created a hand-crafted Dialogue Manager (DM) and trained two statistical DMs in simulation. To this end, we developed an agenda-based user simulator (Schatzmann et al., 2007) that was designed to handle the complexities and demands of our SDS, e.g., real values for slots, intervals, hierarchies, all of our operators, hard constraints and preferences, etc., as well as to be able to handle multiple items being suggested by the system (in the sense of an overview of current results) and tell if these items satisfy the user’s constraints. In order to handle a wide range of domains, we use the method proposed in (Wang et al., 2015), which extracts features describing each slot and action plus some general features pertaining to the dialogue so far and the current state of the knowledge base. Thus, even if new slots are added to the knowledge base, our dialogue manager does not need to be re-trained. Specifically, we use some of the features proposed in (Wang et al., 2015; Papangelis and Stylianou, 2016) and the features described in the previous section, which are necessary to handle the increased complexity of the interaction.

4.2 Understanding and State Tracking

Translating the identified user intentions from SLU into a belief state is not trivial, even for slot filling models with one or two operators (e.g., =, ≠). Moreover, as we aim to connect our system to live knowledge bases, it is important for SLU to be able to adapt over time, as well as handle out-of-domain input gracefully. As an initial approach to belief tracking, we follow some simple principles (Papangelis et al., 2017) in conjunction with an existing belief tracker (Ultes et al., 2017). While this is straightforward for regular slots, we need a different kind of belief update for hierarchically valued or multi-valued slots. Specifically, for hierarchical slots we need to recursively perform the belief update, while still following the basic principles. As the constraints become more complex, traversing the hierar-

Variables	E1	E2	E3	E4
Semantic Error	15%	30%	45%	45%
SLU N-Best Size	3	5	7	7
Sim. User Patience	5	3	3	2
Max User Constraints	3	5	7	10
Acceptable Num. Items	7	5	3	2

Table 1: Four environments (parameter settings) under which our DMs were evaluated.

chy of values becomes non-trivial. In our prototype, we traverse the hierarchy once for each constraint (relevant to a specific hierarchical slot) and then combine the updates into a single belief update as the average for each value. When updating multi-valued slots, we assign the probability mass to each value that was mentioned (and not negated); this can be seen as generating (or removing) a single-valued “sub-slot” for each value on the fly.

5 Preliminary Evaluation

To assess how well current statistical DMs perform in this setting, we compare a hand-crafted dialogue policy (HDC) against a DM trained with GP-SARSA (GPS) (Gašić et al., 2010) and one trained with Deep Q Networks with eligibility traces (DQN- λ) - an adapted version of (Harb and Precup, 2017). HDC, GPS, and DQN (without eligibility traces) have been the top performing algorithms in a recent benchmark evaluation (Casanueva et al., 2017). We test the DMs under various conditions, presented in Table 1. *Semantic Error* refers to simulated errors, where we change either the type of dialogue act, slot, value, or operator that the simulated user issues, based on some probability. This can happen multiple times, to generate multiple SLU hypotheses. *SLU N-Best Size* is the maximum size of the N-best list of SLU hypotheses, after the simulated error stage. *Sim. User Patience* is the maximum number of times the simulated user tolerates the same action being issued by the DM. *Max User Constraints* is the maximum number of constraints in the simulated user’s goal (e.g., $price \leq 70$). One important observation is that task success is very hard to define, as we consider a cluster of ranked items to be a valid system response. Some users may want to get exactly one option while for some it may be acceptable to get no more than four. Therefore, we add a feature to our user simulator to indicate the number of items a user will accept as a final result (provided that all of them meet the user’s constraints). We sample this uniformly from the set $\{1, \dots, acceptable\}$, as defined in Table 1 (*Acceptable Num. Items*). While this is a rough approximation of real world conditions, we expect that it introduces one more layer of complexity that the statistical DMs need to model.

The dataset used for the evaluation consists of four domains (Hotels, Restaurants, Museums, and Shops) with databases populated with content scrapped from the internet, containing a total of 84 slots and 714 ob-

ENV	E1	E2	E3	E4
Single Domain Per Dialogue				
HDC	83.8 \pm 5	65.8 \pm 6	38.8 \pm 9	35.7 \pm 8
DQN λ	74.4 \pm 8	60.7 \pm 9	52.1 \pm 10	49.6 \pm 9
GPS	88.1 \pm 4	79.5 \pm 3	66.8 \pm 6	60.3 \pm 9
Multiple Domains Per Dialogue				
HDC	82.3 \pm 7	71.6 \pm 7	40.7 \pm 10	30.5 \pm 9
DQN λ	88.3 \pm 2	87.5 \pm 2	85.8 \pm 3	82.9 \pm 4
GPS	44.6 \pm 8	26.3 \pm 2	22.7 \pm 6	13.3 \pm 7

Table 2: Dialogue success rates for the DMs under various semantic error rates \pm std dev.

jects. We evaluated the statistical DMs on a single domain and on a multi-domain setting (as described in section 4.1). Table 2 summarizes the results of our evaluation in simulation in the four environments we have defined, where each entry is the average of 5 runs of 1,000 training and 100 evaluation dialogues. DQN- λ performs better with the rich (dense) domain-independent feature set in the multi-domain scenario, likely because it is exposed to more variability in the data and therefore needs less iterations to learn well-performing policies. In fact, it is able to cope very well in deteriorating conditions, by learning to adapt e.g., by asking for more confirmations. GPS shows the opposite trend, preferring the sparse belief state features of the single-domain scenario, needing many more dialogues (than the 1,000 allowed here) to reach good performance in the multi-domain case.

6 Conclusion

We have presented LD-SDS, a prototype information navigation SDS that connects to semantic knowledge bases to guide users towards making informed decisions. This direction is more challenging compared to other simpler kinds of interaction. To evaluate the quality of the approach that we propose, we developed an agenda-based user simulator and applied it to train two statistical DMs. While we have proven the feasibility of our approach, our system still needs to be trained and evaluated with human users as in some cases statistical DMs may overfit simulators (or take advantage of certain aspects of them). We are therefore in the process of designing studies to collect text-based human-human data that will be used to train LD-SDS either end-to-end or by jointly optimising some of the components. In addition, we plan to evaluate our approach with live semantic knowledge bases and extend our approach to also exploit available unstructured information (out of domain). In the appendix we show an example dialogue with our system that highlights the extensions to the typical slot-filling approach.

References

Amazon. 2017. Amazon alexa prize 2017 proceedings. <https://developer.amazon.com/>

- [alexaprize/proceedings](#). Accessed: 2018-03-09.
- Iñigo Casanueva, Paweł Budzianowski, Pei-Hao Su, Nikola Mrkšić, Tsung-Hsien Wen, Stefan Ultes, Lina Rojas-Barahona, Steve Young, and Milica Gašić. 2017. A benchmarking environment for reinforcement learning based task oriented dialogue management. *arXiv preprint arXiv:1711.11023*.
- Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *arXiv preprint arXiv:1704.00051*.
- H. Cuayáhuitl, S. Yu, A. Williamson, and J. Carse. 2017. Scaling up deep reinforcement learning for multi-domain dialogue systems. In *2017 International Joint Conference on Neural Networks*, pages 3339–3346.
- Milica Gašić, Filip Jurčićek, Simon Keizer, François Mairesse, Blaise Thomson, Kai Yu, and Steve Young. 2010. Gaussian processes for fast policy optimisation of pomdp-based dialogue managers. In *Proceedings of SIGDial*, pages 201–204. ACL.
- Jean Harb and Doina Precup. 2017. Investigating recurrence and eligibility traces in deep q-networks. *arXiv preprint arXiv:1704.05495*.
- Tom Heath and Christian Bizer. 2011. Linked data: Evolving the web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136.
- Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc’Aurelio Ranzato, and Jason Weston. 2016. Learning through dialogue interactions. *CoRR*, abs/1612.04936.
- Bing Liu, Gokhan Tur, Dilek Hakkani-Tur, Pararth Shah, and Larry Heck. 2017. End-to-end optimization of task-oriented dialogue model with deep reinforcement learning. *arXiv preprint arXiv:1711.10712*.
- Panagiotis Papadakis and Yannis Tzitzikas. 2014. Hippalus: Preference-enriched faceted exploration. In *EDBT/ICDT Workshops*, volume 172.
- A Papangelis and Y Stylianou. 2016. Multi-domain spoken dialogue systems using domain-independent parameterisation. In *Domain Adaptation for Dialogue Agents*.
- Alexandros Papangelis, Panagiotis Papadakis, Margarita Kotti, Yannis Stylianou, Yannis Tzitzikas, and Dimitris Plexousakis. 2017. Ld-sds: Towards an expressive spoken dialogue system based on linked-data. In *SCAI*.
- Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. 2007. Agenda-based user simulation for bootstrapping a pomdp dialogue system. In *HLT 2007, NAACL*, pages 149–152. Association for Computational Linguistics.
- Yannis Tzitzikas and Eleftherios Dimitrakis. 2016. Preference-enriched faceted search for voting aid applications. *IEEE Transactions on Emerging Topics in Computing*.
- Yannis Tzitzikas and Panagiotis Papadakis. 2013. Interactive exploration of multi-dimensional and hierarchical information spaces with real-time preference elicitation. *Fundamenta Informaticae*, 122(4):357–399.
- Stefan Ultes, Lina M. Rojas Barahona, Pei-Hao Su, David Vandyke, Dongho Kim, Iñigo Casanueva, Paweł Budzianowski, Nikola Mrkšić, Tsung-Hsien Wen, Milica Gasic, and Steve Young. 2017. *PyDial: A Multi-domain Statistical Dialogue System Toolkit*. In *Proceedings of ACL 2017, System Demonstrations*, pages 73–78, Vancouver, Canada. ACL.
- Z. Wang, T.H. Wen, P.H. Su, and Y. Stylianou. 2015. Learning domain-independent dialogue policies via ontology parameterisation. In *16th Annual Meeting of the SIGDial*, page 412.
- Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, pages 438–449.
- D. Jason Williams, Kavosh Asadi, and Geoffrey Zweig. 2017. Hybrid code networks: Practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *ACL*, pages 665–677.
- X. Yang, Y. N. Chen, D. Hakkani-Tür, P. Crook, X. Li, J. Gao, and L. Deng. 2017. End-to-end joint learning of natural language understanding and dialogue manager. In *2017 IEEE ICASSP*, pages 5690–5694.

A Supplemental Material

In this section, we provide an example interaction between a human user and the LD-SDS prototype. Figure 2 shows the system in operation. Figures 3 and 4 show examples of slots that can take multiple values or whose values have hierarchical relations, respectively.

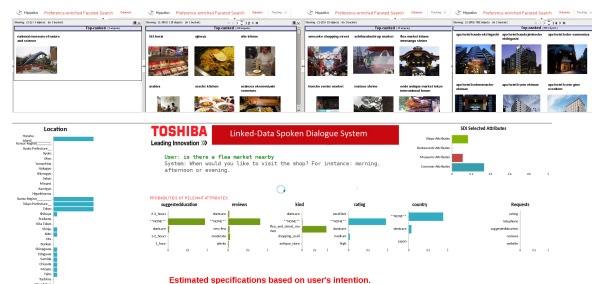


Figure 2: The prototype SDS, acting as a hotel concierge, with live connections to Hippalus.



Figure 3: An item in Hippialus, where the *language* slot can take multiple values.

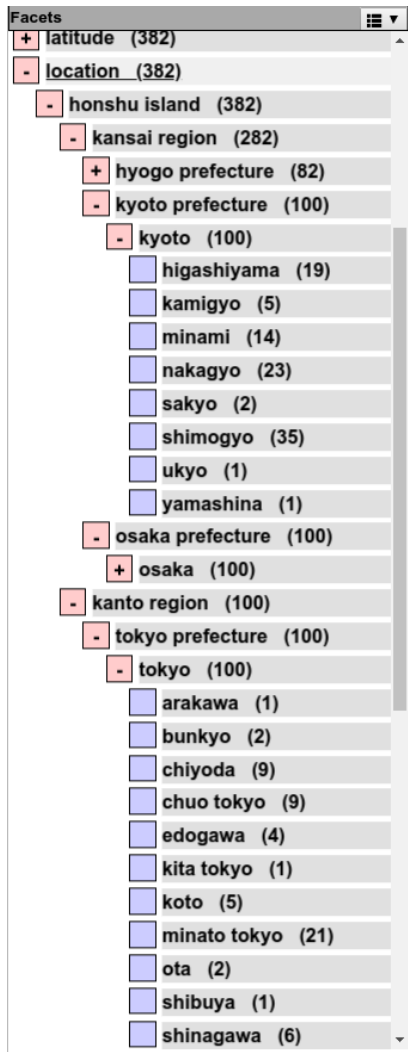


Figure 4: An example of a slot in Hippialus, where the values have hierarchical relations.

System: Hello, welcome to the Japanese Hotels spoken dialogue system. How may I help you?
User: I'm looking for a hotel in Kyoto but not in Minami where they offer free Wi-Fi and have non smoking rooms. (ASR confidence: 0.75154209) <i>Minami is a region in Kyoto.</i>
Constraints: [type=hotel, location=kyoto, location≠minami, amenities={free-wifi, non-smoking-rooms}]
Preferences: []
Hippalus: [267 items in 2 buckets; {60, 207} items in buckets]
S: What price range are you looking for?
U: I want something around 70 pounds and with more than two stars. (0.7065863)
C: [pricerange ~ 70, stars > 2]
P: []
H: [188 items in 45 buckets; {25, 103, 13, ...} items in buckets]
S: What kind of user rating would you prefer the hotel to have?
U: I'd like excellent ratings. (0.92533112)
C: []
P: [ratings=excellent : best]
H: [188 items in 27 buckets; {9, 11, 8, ...} items in buckets; identified slots that can affect ranking and reduce items in first bucket: {stars, location, price}]
S: Which of the following criteria are important for you? stars, price, or location?
U: I prefer location and price. (0.95948964)
C: []
P: [location >> all, price >> all]
H: [188 items in 152 buckets; 3 items in first bucket]
S: 3 hotels match your preferences. Two are located in Shimogyo and one in Nakagyo. Ryokan Kyoraku in Shimogyo is cheaper with 59 pounds per night and 3 stars, and Daiwa Roynet is more expensive at 81 pounds per night but with 4 stars. Royal Park in Nakagyo is expensive as well at 79 pounds per night and 4 stars. Both expensive hotels offer more services than the cheaper one.
U: Thank you, goodbye. (0.97125274)

Table 3: Example interaction between a human user and our LD-SDS prototype. In the interest of space, the notes under each dialogue turn briefly show items that correspond to new information. The belief state is updated accordingly. ASR: Automatic Speech Recognition.