Towards a Response Selection System for Spoken Requests in a Physical Domain

Andisheh Partovi, Ingrid Zukerman, Quan Tran
Faculty of Information Technology, Monash University
Clayton, VICTORIA 3800, AUSTRALIA
{andi.partovi,ingrid.zukerman,quan.tran}@monash.edu

Abstract

In this paper, we introduce a corpus comprising requests for objects in physical spaces, and responses given by people to these requests. We generated two datasets based on this corpus: a manually-tagged dataset, and a dataset which includes features that are automatically extracted from the output of a Spoken Language Understanding module. These datasets are used in a classification-based approach for generating responses to spoken requests. Our results show that, surprisingly, classifiers trained on the second dataset outperform those trained on the first, and produce acceptable levels of performance.

1 Introduction

In recent times, there have been significant improvements in Automatic Speech Recognition (ASR) [Chorowski *et al.*, 2015; Bahdanau *et al.*, 2016]. For example, a research prototype of a spoken slot-filling dialogue system reported a Word Error Rate (WER) of 13.8% when using "a generic dictation ASR system" [Mesnil *et al.*, 2015], and Google reported an 8% WER for its ASR API.[1] However, this API had a WER of 54.6% when applied to the Let's Go corpus [Lange and Suendermann-Oeft, 2014].

ASR errors not only produce wrongly recognized entities or actions, but may also yield ungrammatical utterances that cannot be processed by a Spoken Language Understanding (SLU) system (e.g., "the plate inside the microwave" being mis-heard as "*of plating sight* the microwave"), or yield incorrect results when processed by an SLU system (e.g., due to fillers such as "hmm" being mis-heard as "and" or "on").

The problems caused by ASR errors are exacerbated by the fact that people often express themselves ambiguously or inaccurately [Trafton *et al.*, 2005; Moratz and Tenbrink, 2006; Funakoshi *et al.*, 2012; Zukerman *et al.*, 2015]. An ambiguous reference to an object matches several objects well, while an inaccurate reference matches one or more objects partially. For instance, in a household domain, a reference to a "big blue mug" is ambiguous if there is more than one big blue mug in the room, and inaccurate if there are two mugs in the room, one big and red, and one small and blue. Further, ambiguous or inaccurate references may occur as a result of differences in parse trees (e.g., due to variants in prepositional attachments).

In addition to improving ASR and SLU modules, Spoken Dialogue Systems (SDSs) must be able to cope with these problems by generating appropriate responses to users' spoken utterances. Recently, deep-learning algorithms have been used for response generation [Serban *et al.*, 2016; Yang *et al.*, 2016]. However, these algorithms rely solely on requests and responses, without taking into account the (extra linguistic) context, and typically require large amounts of data, which may not be available in some applications. In this paper, we offer a supervised-learning approach to response-generation that is suitable for smaller datasets. Our approach harnesses the properties of utterances, dialogue history and context to choose *response types* for users' requests.

To obtain an upper bound for classifier performance, we trained a classifier using human-observable features of spoken requests and response types selected by participants for these requests. We then trained a second classifier using features that were automatically extracted from the output produced by our SLU system (Section 5). Surprisingly, the second classifier produced significantly better results than the first one.

The rest of this paper is organized as follows. In the next section, we discuss related work. Our corpus is described in Section 3. In Section 4, we detail the human-observable features and the response-classification results obtained with them. We then offer a brief account of our SLU system, followed by a description of the features that are automatically extracted from its output and the resultant classification performance. Concluding remarks appear in Section 7.

2 Related Work

Decision-theoretic approaches have been the accepted standard for response generation in dialogue systems for some time [Carlson, 1983]. These approaches were initially implemented in SDSs in the form of *Influence Diagrams* that make myopic (one-shot) decisions regarding dialogue acts [Paek and Horvitz, 2000], procedures that optimize responses [Inouye and Biermann, 2005; Sugiura *et al.*, 2009], and *Dynamic Decision Networks* that make decisions about dialogue acts over time [Horvitz *et al.*, 2003; Liao *et al.*, 2006].

[1] venturebeat.com/2015/05/28/google-says-its-speech-recognition-technology-now-has-only-an-8-word-error-rate.

Later on, *reinforcement learning* was employed to learn optimal policies over time [Lemon, 2010], with particular attention being paid to *Partially Observable Markov Decision Processes* [Williams and Young, 2007; Young *et al.*, 2013; Gašić and Young, 2014], and their extension *Hidden Information State* [Young *et al.*, 2007; Young *et al.*, 2013]. Owing to the complexity of these formalisms, they have been used mainly in slot-filling applications, e.g., making airline and restaurant reservations [Young *et al.*, 2013].

Recently, deep learning has been applied to various aspects of SDSs [Wen *et al.*, 2015; Li *et al.*, 2016; Mrkšic *et al.*, 2016; Prakash *et al.*, 2016; Serban *et al.*, 2016; Yang *et al.*, 2016]. Wen *et al.* [2015] focused on the generation of linguistically varied responses, and Mrkšic *et al.* [2016] proposed a dialogue-state tracking framework. The generation of dialogue contributions was studied in [Li *et al.*, 2016; Prakash *et al.*, 2016] for chatbots; in [Serban *et al.*, 2016] for help-desk responses and Twitter follow-up statements; and in [Yang *et al.*, 2016] for slot tagging, and user-intent and system-action prediction in slot-filling applications. A combination of deep learning and reinforcement learning has been used in end-to-end dialogue systems that query a knowledge-base, where user utterances are mapped to a clarification question or a knowledge-base query [Williams and Zweig, 2016; Zhao and Eskenazi, 2016; Dhingra *et al.*, 2017]. All these systems learn to generate complete responses from large corpora comprising request-response pairs.

Our work follows this supervised-learning trend in a setting where the appropriateness of a response depends both on the request and on the physical context. Further, our dataset is significantly smaller than those used by neural mechanisms.

3 The Corpus

Our corpus, which was gathered in two experiments, comprises requests to fetch or move household objects, and responses to these requests.

Experiment 1 – This experiment replicates the experiment described in [Zukerman *et al.*, 2015] using the Google ASR API, instead of Microsoft Speech SDK 6.1 — the WER of the Google API was 13% for our corpus. 35 participants were asked to describe 12 designated objects (labelled A to L) in the scenarios depicted in Figure 1. Each scenario contains between 8 and 16 household objects varying in size, colour and position. The participants were allowed to repeat a description up to two times. In total, they recorded 478 descriptions such as the following: "the computer under the table", "the picture on the wall", "the green plate next to the screwdriver at the top of the table", "the plate in the corner of the table", and "the large pink ball in the middle of the room".

Experiment 2 – This experiment took the form of an online survey where participants had to indicate how they would respond to a (potentially mis-heard) request. Each participant was shown the top four ASR outputs for the request versions of 12 descriptions generated by one participant in the first experiment, along with the images in Figure 1. For instance, "the pot plant on the table", uttered in the context of Figure 1(a), was converted to "get the pot plant on the table"; and "the green bookcase", uttered in the context of Figure 1(d), was presented as "move the green bookcase".

Each participant was then asked to choose a response for each request from the following four response types (participants were given a description of each response type):

- DO: suitable when the addressee is sure about which object the request refers to.
- CONFIRM: suitable when the addressee feels the need to confirm the requested object before taking action.
- CHOOSE: suitable when the addressee hesitates between several objects.
- REPHRASE: suitable when part or all of a request is so unintelligible that the addressee cannot understand it.

These choices were made under two cost settings: *low-cost* – where participants were told that the requested object must be delivered to someone in the same room; and *high-cost* – where they were told that the object must be delivered to a far-away location. These settings were designed to discriminate between situations where mistakes are fairly inconsequential and situations where mistakes are costly.

40 people took part in this experiment (six of them also participated in the first experiment); half of the participants were native English speakers, and half were male. Thirteen people participated in an initial version of the experiment where they first chose response types for all the requests under the low-cost setting, and then chose response types for the same requests under the high-cost setting. We modified the experiment on the basis of the participants' feedback, so that the remaining 27 participants considered each request under the low-cost setting, and were immediately asked how their response would differ under the high-cost setting. This experimental variation had no effect on response-classification performance (Sections 4.1 and 6.1).

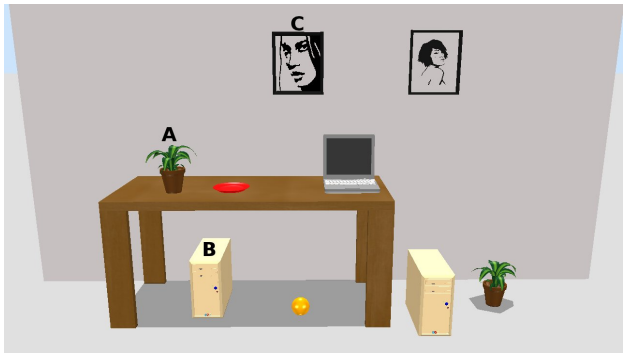
To determine the effect of personal variations on classification performance, one of the authors, who is familiar with the system, selected response types for all the requests.

3.1 Analysis and Post Processing

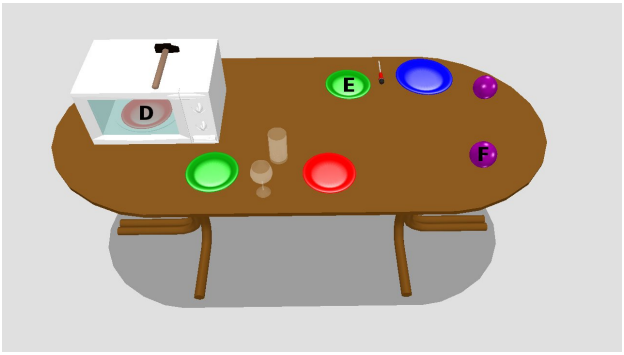
In total, we collected 960 request-response pairs ($=$ 12 requests $\times$ 2 cost factors $\times$ 40 participants). 24.2% of these requests had an unintelligible semantic role in at least one ASR output, with the vast majority occurring in the OBJECT of the descriptions; 17.9% were ambiguous (i.e., they had more than one reasonable referent); and only 3.8% were inaccurate (i.e., they did not match perfectly any referent).

In order to train both classifiers on the same corpus, we removed requests that don't fit the requirements of the automatic feature-extraction process (Section 6). Specifically, we excluded 62 descriptions (13%) that had more than one prepositional phrase, and 43 descriptions (9%) that could not be processed by our SLU module [Zukerman *et al.*, 2015] (Section 5). As a result, our corpus contains 375 descriptions, which yield a total of 750 requests for both cost settings. The responses to these requests were distributed as follows: 51.9% DO (majority class), 21.6% CHOOSE, 14.1% REPHRASE, and 12.4% CONFIRM.

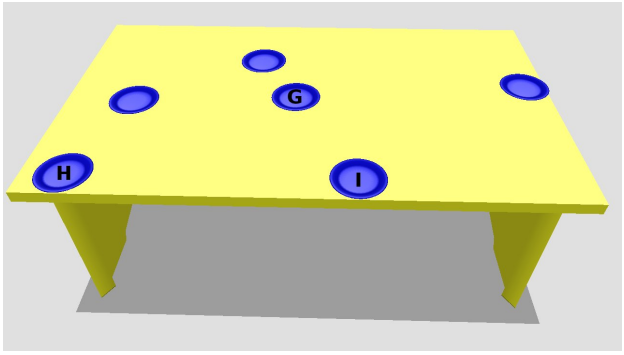
It is worth noting that the response types chosen for the excluded requests were included in the dataset as features in



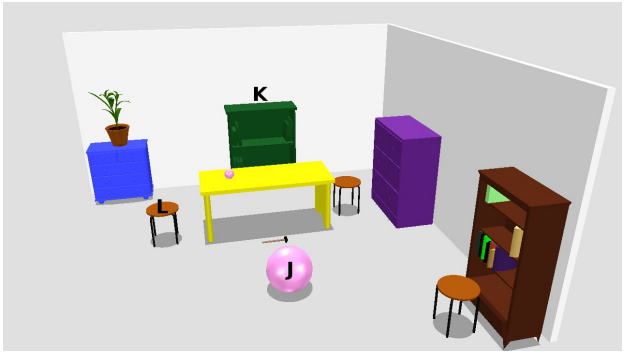
(a) Positional relations in a room



(b) Colour, size and positional relations on a table



(c) Projective and positional relations on a table



(d) Colour, size and positional relations in a room

Figure 1: Household scenes used in our experiments

order to enable us to determine the effect of dialogue history on performance (Sections 4 and 6). Clearly, removing requests disrupts the actual sequence of events, which has an adverse effect on the performance of sequence classifiers (Section 4.2). In the near future, we will address this problem by including a feature set for all the requests in a sequence.

4 Classification with Manually-Tagged Features

Two team members annotated each description obtained from the first experiment with the following features, which are indicative of inaccuracy and ambiguity, and were deemed relevant to a person's decision regarding how to respond to a request (the first annotator labelled the features, and the second annotator verified the annotations; disagreements were resolved by consensus).

1. *Unintelligible role* – This is the semantic role of a garbled portion of a description, where the possible values are {NONE, ALL, OBJECT, LANDMARK, OTHER}. For example, "the *hottest* under the table" has an unintelligible OBJECT, and "the green plate on the left of the *Blues play*" has an unintelligible LANDMARK.
2. *# of reasonable interpretations* – How many objects are reasonable referents for a description? For instance, the first of the above requests has two reasonable referents in the context of Figure 1(a), as there are only two objects under the table. Similarly, the two green plates on the table in Figure 1(b) are reasonable interpretations for the second request.

   People often compensate for mis-heard utterances by postulating reasonable words that sound similar to what was heard. We take this behaviour into account by splitting this feature into two sub-features: (2a) *With phonetic similarity* and (2b) *Without phonetic similarity*. For example, when considering phonetic similarity in the context of Figure 1(b), "blue plate" is a sensible replacement for "*Blues play*", yielding one reasonable interpretation for the second request (the green plate labeled E).
3. *Do the reasonable interpretations include fewer than all the objects in the context?* (YES, NO) – This feature indicates how much information can be extracted from a description, e.g., the value of this feature is NO for "the blue plate on the table" in the context of Figure 1(c), since all the objects on the table are reasonable referents for this description. As above, this feature is split into two subfeatures: (3a) *With phonetic similarity* and (3b) *Without phonetic similarity*.
4. *# of perfect interpretations* – How many objects match perfectly a description? For example, the two balls in Figure 1(d) match perfectly the description "the ball". Note that the difference between *# of reasonable interpretations* and *# of perfect interpretations* indicates the accuracy of a description.
5. *Do the perfect interpretations include fewer than all the objects in the context?* (YES, NO) – This feature is similar

to Feature #3. However, since we are considering only interpretations that match a request perfectly, there is no need to take into account phonetic similarity.

4.1 Response Classification

We experimented with several classification algorithms, including Naïve Bayes, Support Vector Machines, Decision Trees (DT) and Random Forests (RF), to learn response types from the data collected in our experiments. Here we report on the results obtained with DT and RF, which had the best performance.[2] We used the above features to determine baseline performance, and experimented with four additional features: *Gender*; *English nativeness* – whether the participant is a native English speaker; *3-Back responses* (vector of length 4) – the counts of the response types provided by an Experiment 2 participant for the three preceding requests;[3] and *Cost* – high or low. This feature worsened performance in all cases, and was removed.

We performed 10-fold cross-validation to evaluate classifier performance; statistical significance was computed using the Wilcoxon signed-ranked test. Rows 2-4 in Table 1 display the best results obtained by our classifiers for each feature configuration.

RF yielded the best results for the manually-tagged features alone, and for these features plus *Gender* and *English nativeness*; while DT produced the best results overall when *3-Back responses* were added (statistically significant with *p-value*=0.05). The most influential features in the decision tree were *# of perfect interpretations*, *# of reasonable interpretations with phonetic similarity*, and *# of rephrases in 3-Back responses*. The per-class performance of DT appears in the second and third columns of Table 5. Note the poor precision and recall obtained for CONFIRM, which was often confused with DO. DT's deficient performance for REPHRASE may be attributed to the fact that requests that had the same features, in particular those with partially or completely unintelligible ASR outputs, elicited the different responses from the participants.

As mentioned in Section 3, we also trained and tested the classifiers using response types selected by only one person – the first author. The best performance was achieved with an RF classifier that includes *3-Back responses*, denoted RF_{1P}. This performance was much better than of the classifiers trained with the response types of 40 participants, which indicates that personal attributes affect people's responses.[4]

4.2 Sequence Classification

In order to investigate the influence of a sequence of request-response pairs on future responses, we trained and tested a

[2]We used over- and under-sampling to try to deal with the large majority class, but neither affected the classifiers' performance.

[3]We experimented with several sequence lengths, of which *3-Back* yielded the best results. We also investigated a setting where the counts of the response types chosen for all the other 23 requests were used as features. This setting, which is clearly unfeasible, gave the best results, achieving 0.70 precision and 0.68 recall.

[4]We tried to address this issue by clustering users based on the number of times they chose each response type, but didn't get good clusters for $k < 10$.

Table 1: Performance with manually-tagged features

Classifier	Manually-tagged Features	Precision	Recall
RF		0.58	0.65
RF	+ *Gender & English nat.*	0.62	0.67
DT	+ *Gender & English nat.* + *3-Back responses*	**0.63**	**0.68**
RNN	+ entire previous sequence	0.55	0.62
RF_{1P}	+ *3-Back responses*	0.81	0.82

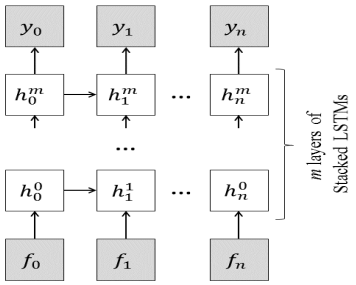


Figure 2: RNN for response-type selection

Recurrent Neural Network (*RNN*) as a sequence classifier.

Our RNN model is based on the *Long-Short-Term-Memory* (*LSTM*) architecture [Hochreiter and Schmidhuber, 1997], which can capture long-range dependencies. If we denote the features of the t-th utterance as f_t, the hidden state of the RNN at time step $t+1$ is calculated as a function of the input at time step $t+1$, f_{t+1}, and the previous hidden state, h_t: $h_{t+1} = \mathit{LSTM}(h_t, f_{t+1})$. With this mechanism, the model maps the sequence of features to a sequence of hidden vectors, which are decoded into a sequence of labels by a linear neural net layer: $y_t \sim \mathit{softmax}(W\mathbf{h}_t + b)$.

A natural extension of this model is to stack the LSTM layers, i.e., the outputs of the first LSTM layer are given as input to the second layer, and so on; our model stacks 15 layers of LSTMs. This model was implemented with Keras [Chollet, 2017] and Theano [Theano Development Team, 2016], and was trained to minimize categorical cross-entropy loss using the Adam SGD learner [Kingma and Ba, 2014].

Owing to time limitations, we performed only 5-fold cross-validation. The results of the RNN appear in the penultimate row of Table 1. The RNN's disappointing performance may be attributed to the relatively small dataset combined with the disruption of several sequences due to the removal of request-response pairs (in order to reduce sequence disruption, we retained the 43 pairs corresponding to descriptions that could not be processed by our SLU system).

5 The SLU System *Scusi?*

Scusi? [Zukerman *et al.*, 2015] is a system that implements an anytime, numerical mechanism for the interpretation of spoken descriptions, focusing on a household context. It has four processing stages, where (intermediate) interpretations in each stage can have multiple parents in the previous stage,

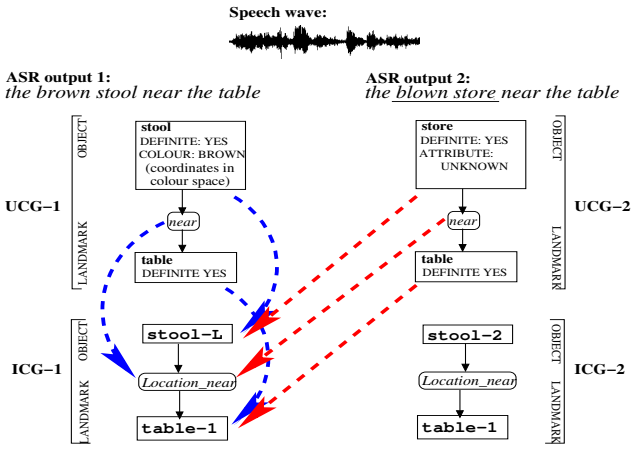


Figure 3: *Scusi?*'s workflow and UCG-to-ICG relations

and can produce multiple children in the next stage; early processing stages may be probabilistically revisited; and only the most promising options in each stage are explored further.

Scusi's workflow – The system takes as input a speech signal, and uses an ASR to produce candidate texts. Each text is assigned a score given the speech wave, and passed to an error-detection module that postulates which words were correctly or wrongly recognized by the ASR [Zukerman and Partovi, 2017] — this component is required, as in real life we don't have access to transcriptions. Next, *Scusi?* applies Charniak's probabilistic parser (`bllip.cs.brown.edu/resources.shtml#software`) to syntactically analyze the texts, yielding at most 50 parse trees per text. The third stage applies mapping rules to the parse trees to generate *Uninstantiated Concept Graphs* (*UCGs*) [Sowa, 1984] that represent the semantics of the descriptions. The final stage instantiates the UCGs with objects and relations from the current context, and returns candidate *Instantiated Concept Graphs* (*ICGs*) ranked in descending order of merit (score).

Figure 3 illustrates this process for the description "the brown stool near the table" in the context of Figure 1(d). All stages produce several outputs, but we show only two outputs for each of three stages (ASR, UCG and ICG). In addition, in this example, both UCGs are parents of the two ICGs, but only the match with ICG-1 is shown in Figure 3. The first ASR output is correct, and the second has "*blown store*" instead of "brown stool". Each of these outputs yields one UCG (via a parse tree), where the object in the second UCG has an *unknown* attribute, as *Scusi?* doesn't recognize the modifier "*blown*" (*unknown* attributes occur when a user employs out-of-vocabulary noun modifiers or the ASR mis-recognizes noun modifiers).

The score of each ICG depends on two factors: (1) how well the concepts and relations in it match the corresponding concepts and relations in its parent UCGs, and (2) how well the relations in the ICG match the context. For example, ICG-1 matches UCG-1 well, as `stool-L` can be called "stool" and it is brown, and `table-1` can be called "table"; but its match-score with UCG-2 is lower, as `stool-L` cannot be called "store" and doesn't match the *unknown* attribute specified in UCG-2. ICG-1 matches the context well, as `stool-L` is near `table-1`. The details of the calculation of the scores are described in [Zukerman *et al.*, 2015]. The aspects that are most relevant to this paper are that scores are represented on a logarithmic scale in order to avoid underflow, and scores of value 0 are smoothed to a low value ϵ in order not to invalidate any interpretation.

Table 2: Features obtained from the word-error detector

Is there an ASR output with all correct words?
% of wrong words in the top ASR output
% of wrong words in all ASR outputs
% of ASR outputs with all correct words

Table 3: Features extracted from the top-10 ICGs

Number of top-ranked ICGs with similar scores	(×1)
Location match score between an ICG and the context	(×10)
Per-node features for an ICG in relation to its parent UCGs	
Best colour-match score for a content node	(×20)
Best size-match score for a content node	(×20)
Maximum # of *unknowns* for a content node	(×20)
For a content node, % of UCG parents with corresponding node	
• with a colour match for this node	(×20)
• with a size match for this node	(×20)
• that have *unknowns*	(×20)
For a node, % of UCG parents with corresponding node	
• that lexically match this node	(×30)

6 Classification with Automatically-Extracted Features

We automatically extracted features from the top-10 ICGs generated by *Scusi?* for each description (the correct interpretation is in the top-10 ICGs in about 90% of the cases) — these features appear in Tables 2 and 3. The features in Table 2, extracted from the output of *Scusi?*'s word-error detector, pertain to the intelligibility of the descriptions. The second and third feature in Table 2 are among the most influential ones.[5] The last feature is noteworthy because, even though only one ASR output is correct, the error-detection component may decide that several ASR outputs are correct, e.g., "the *flower* on the table" and "the *flour* on the table".

The first feature in Table 3 represents the ambiguity of a description through the similarity between the scores of successive top-ranked ICGs, which is encoded as the ratio between the (logarithmic) score of the $i+1$-th ICG and the score of the i-th ICG. When this ratio between neighbouring ICGs is below an empirically-derived threshold, they are deemed similar. This feature is among the most influential ones.

The remaining features in Table 3 pertain to the accuracy of a description, which is represented by the goodness of the match between an ICG and its parent UCGs, and between an ICG and the context. The second feature, which represents the accuracy of the location specified in a description, is among the most influential ones (for ICGs ranked 4th, 6th and 9th).

[5] The frequency of features in the top-two levels of 100 trees generated by RF was used as a proxy for their importance.

As seen in Figure 3, content nodes (objects and landmarks) in UCGs may have colour and size descriptors, as well as *unknown* attributes. The first six per-node features in Table 3 represent the goodness of attribute matches between the content nodes (object and landmark) of an ICG and the corresponding nodes in its parent UCGs. Two size-match features, one colour-match feature and one *unknown* feature for objects of ICGs at various ranks are among the most influential features.

The last row in Table 3 represents the goodness of lexical matches between the nodes in an ICG and the corresponding nodes in its parent UCGs. This feature is among the most influential for the objects of most of the top-10 ICGs.

To illustrate these features, let's return to the UCG-ICG matches in Figure 3 for the request "move the brown stool near the table" in the context of Figure 1(d). The score of the top-ranked ICG, viz ICG-1, is significantly higher than that of ICG-2. Hence, the value of the first feature in Table 3 is 1. As mentioned above, `stool-L` is near `table-1`, yielding a high location match score for ICG-1. 50% of the UCG parents have a lexical match with the object in ICG-1, as "*store*" doesn't match any designation of `stool-L`; but 100% of the UCG parents have a lexical match with the landmark in ICG-1 (`table-1`). Due to the *unknown* attribute in the object of UCG-2, the maximum number of *unknowns* for the ICG-1 object is 1, and the percentage of UCG parents that have *unknowns* for the ICG-1 object is 50%; while 0% of UCG parents have *unknowns* for the ICG-1 landmark. Since the colour specified in UCG-1 matches the colour of `stool-L`, the maximum colour match for the object of ICG-1 is 1, but the percentage of UCG parents with a colour match for the ICG-1 object is 50%, as UCG-2 doesn't have a colour attribute.

6.1 Response Classification

We experimented with the classifiers considered in Section 4.1, except the RNN, using the 165 features described in Tables 2 and 3, instead of the manually-obtained ones.[6] The RNN was omitted due to the above-described removal of requests, which disrupts the sequence. As before, we performed 10-fold cross-validation.

Table 4 displays our results. The classifier with the best performance for a particular configuration of manually-tagged features also had the best performance for the corresponding configuration of automatically-extracted features. Surprisingly, overall performance with these features was significantly better (with *p-value*=0.01) than the performance obtained with the manually-tagged features, both for the responses given by 40 participants and for the responses provided by one person. In the former case, *3-Back responses* had an adverse effect on performance, and in the latter case, it had no effect. The best performance for the 40-participant dataset was obtained with RF plus *Gender* and *English nativeness*, but the differences between the classifiers were not statistically significant. The per-class performance of this classifier appears in the fourth and fifth columns of Table 5. As for the manually-tagged features, the worst precision and recall were obtained for CONFIRM, but the performance for REPHRASE was only slightly worse than for the other classes.

[6] Applying Principal Components Analysis to reduce the number of features had no effect on the classifiers' performance.

Table 4: Performance with automatically-extracted Features

Classifier	Automatically-extracted Features	Precision	Recall
RF		0.73	0.74
RF	+ *Gender & English nat.*	**0.74**	**0.74**
DT	+ *Gender & English nat.* + *3-Back responses*	0.72	0.72
RF_{1P}		0.93	0.92

Table 5: Per-class performance of the best classifier for manually- and automatically-extracted features

Class	Manually-tagged Features		Automatically-extracted Features	
	Precision	Recall	Precision	Recall
DO	0.72	0.93	0.82	0.83
CONFIRM	0.28	0.10	0.42	0.38
CHOOSE	0.70	0.64	0.74	0.76
REPHRASE	0.54	0.31	0.70	0.70

7 Conclusion and Future Work

We have offered a corpus comprising requests for objects in physical spaces, and the responses given by people for these requests. We generated two datasets based on this corpus: a manually-tagged dataset, and a dataset which includes features that are automatically extracted from the output of an SLU module. These datasets were used in a classification-based approach for generating responses to spoken requests.

Our results show that, surprisingly, classifiers trained on the second dataset outperformed those trained on the first. As mentioned in Section 4, analysis of the data reveals that different users often provide different responses for requests that have identical manually-tagged features. For instance, three participants who were shown the following ASR outputs responded with DO, CONFIRM and REPHRASE (the option chosen by our classifier): (1) "get a blade in the rights of the disabled", (2) "get I played in the rights of the disabled", (3) "get I played in the right of the devil", and (4) "get a blade in the right of the devil". This discrepancy may be partially due to a mixture of individual ability to compensate for misheard utterances combined with risk-taking attitude — traits that may be related to the *English nativeness* and *Gender* features respectively, which improve performance. In light of this, we posit that additional features that reflect personal disposition could yield further improvements. This notion is reinforced by the significantly better classification performance for the responses obtained from a single user (albeit one familiar with the system) compared with the performance for the responses of 40 participants.

A complementary explanation for the worse classification performance obtained for the manually-tagged dataset is that this dataset encodes intelligibility, ambiguity and accuracy of descriptions in a general way, while the specific information encoded in the automatically-extracted dataset (i.e.,

lexical, colour, size and location match for each of the top-10 ICGs) is important for classification. The only aspect where the manual encoding is more informative than the automatic encoding pertains to phonetic similarity, which is one of the most influential features for this dataset. In the future, we will incorporate specific features about lexical, colour, size and location match and out-of-vocabulary words into the manually-generated tags, and phonetic-similarity into the automatically-extracted features.

In terms of dialogue history, our results are inconclusive. Our hypothesis that dialogue history affects users' choices was confirmed (for three preceding requests) for the manually-tagged requests, but not for the automatically-tagged ones.

Finally, as noted in [Inouye and Biermann, 2005; Singh *et al.*, 2002], users may be satisfied with responses that differ from those provided by human consultants. To test this idea, we propose to conduct a follow-up experiment, where participants will be asked to rate the suitability of responses generated by our best classifier.

Acknowledgements

This research was supported in part by grant DP120100103 from the Australian Research Council.

References

[Bahdanau *et al.*, 2016] D. Bahdanau, J. Chorowski, D. Serdyuk, and Y. Bengio. End-to-end attention-based large vocabulary speech recognition. In *ICASSP'2016 – Proceedings of the 2016 IEEE International Conference on Acoustic, Speech and Signal Processing*, pages 4945–4949, Shanghai, China, 2016.

[Carlson, 1983] L. Carlson. *Dialogue Games: An Approach to Discourse Analysis*. D. Reidel Publishing Company, Dordrecht, Holland, Boston, 1983.

[Chollet, 2017] F. Chollet. Keras. `https://github.com/fchollet/keras`, 2017.

[Chorowski *et al.*, 2015] J.K Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio. Attention-based models for speech recognition. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 577–585. Curran Associates, Inc., 2015.

[Dhingra *et al.*, 2017] B. Dhingra, L. Li, X. Li, J. Gao, Y.N. Chen, F. Ahmed, and L. Deng. Towards end-to-end reinforcement learning of dialogue agents for information access. In *ACL'17 – Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, 2017.

[Funakoshi *et al.*, 2012] K. Funakoshi, M. Nakano, T. Tokunaga, and R. Iida. A unified probabilistic approach to referring expressions. In *SIGDIAL'2012 – Proceedings of the 13th SIGdial Meeting on Discourse and Dialogue*, pages 237–246, Seoul, South Korea, 2012.

[Gašić and Young, 2014] M. Gašić and S.J. Young. Gaussian processes for POMDP-based dialogue manager optimization. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 22(1):28–40, 2014.

[Hochreiter and Schmidhuber, 1997] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[Horvitz *et al.*, 2003] E. Horvitz, C. Kadie, T. Paek, and D. Hovel. Models of attention in computing and communication: From principles to applications. *Communications of the ACM*, 46(3):52–57, 2003.

[Inouye and Biermann, 2005] B. Inouye and A. Biermann. An algorithm that continuously seeks minimum length dialogs. In *Proceedings of the 4th IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems*, pages 62–67, Edinburgh, Scotland, 2005.

[Kingma and Ba, 2014] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Lange and Suendermann-Oeft, 2014] P. Lange and D. Suendermann-Oeft. Tuning Sphinx to outperform Google's speech recognition API. In *ESSV2014 – Proceedings of the Conference on Electronic Speech Signal Processing*, Dresden, Germany, 2014.

[Lemon, 2010] O. Lemon. Learning what to say and how to say it: Joint optimisation of spoken dialogue management and natural language generation. *Computer Speech and Language*, 25(2):210–221, 2010.

[Li *et al.*, 2016] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky. Deep reinforcement learning for dialogue generation. In *EMNLP2016 – Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas, 2016.

[Liao *et al.*, 2006] W. Liao, W. Zhang, Z. Zhu, Q. Ji, and W.D. Gray. Toward a decision-theoretic framework for affect recognition and user assistance. *International Journal of Human-Computer Studies*, 64:847–873, 2006.

[Mesnil *et al.*, 2015] G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D.Z. Hakkani-Tür, X. He, L.P. Heck, G. Tur, D. Yu, and G. Zweig. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech & Language Processing*, 23(3):530–539, 2015.

[Moratz and Tenbrink, 2006] R. Moratz and T. Tenbrink. Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. *Spatial Cognition & Computation: An Interdisciplinary Journal*, 6(1):63–107, 2006.

[Mrkšic *et al.*, 2016] N. Mrkšic, Ó.S. Diarmuid, T.H. Wen, B. Thomson, and S.J. Young. Neural belief tracker: Data-driven dialogue state tracking. *arXiv preprint arXiv:1606.03777v1*, 2016.

[Paek and Horvitz, 2000] T. Paek and E. Horvitz. Conversation as action under uncertainty. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 455–464, Stanford, California, 2000.

[Prakash *et al.*, 2016] A. Prakash, C. Brockett, and P. Agrawal. Emulating human conversations using convolutional neural network-based IR. In *Proceedings of the Neu-IR16 SIGIR Workshop on Neural Information Retrieval*, Pisa, Italy, 2016.

[Serban *et al.*, 2016] I.V. Serban, T. Klinger, G. Tesauro, K. Talamadupula, B. Zhou, Y. Bengio, and A. Courville. Multiresolution recurrent neural networks: An application to dialogue response generation. *arXiv preprint arXiv:1606.00776v1*, 2016.

[Singh *et al.*, 2002] S. Singh, D. Litman, M. Kearns, and M. Walker. Optimizing dialogue management with reinforcement learning: Experiments with the NJFun system. *Artificial Intelligence Research*, 16:105–133, 2002.

[Sowa, 1984] J.F. Sowa. *Conceptual Structures: Information Processing in Mind and Machine*. Addison-Wesley, Reading, MA, 1984.

[Sugiura *et al.*, 2009] K. Sugiura, N. Iwahashi, H. Kashioka, and S. Nakamura. Bayesian learning of confidence measure function for generation of utterances and motions in object manipulation dialogue task. In *Proceedings of Interspeech 2009*, pages 2483–2486, Brighton, United Kingdom, 2009.

[Theano Development Team, 2016] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, 2016.

[Trafton *et al.*, 2005] J.G. Trafton, N.L. Cassimatis, M.D. Bugajska, D.P. Brock, F.E. Mintz, and A.C. Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans*, 35(4):460–470, 2005.

[Wen *et al.*, 2015] T.H. Wen, M. Gašić, N. Mrkšic, P. Hao Su, D. Vandyke, and S.J. Young. Semantically conditioned LSTM-based natural language generation for spoken dialogue systems. In *EMNLP2015 – Proceedings ot the Conference on Empirical Methods in Natural Language Processing*, pages 1711–1721, Lisbon, Portugal, 2015.

[Williams and Young, 2007] J.D. Williams and S. Young. Partially observable Markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):393–422, 2007.

[Williams and Zweig, 2016] J.D. Williams and G. Zweig. End-to-end LSTM-based dialog control optimized with supervised and reinforcement learning. *arXiv preprint arXiv:1606.01269*, 2016.

[Yang *et al.*, 2016] X. Yang, Y.N. Chen, D. Hakkani-Tür, P. Gao, and L. Deng. End-to-end joint learning of natural language understanding and dialogue manager. *arXiv preprint arXiv:1612.00913v1*, 2016.

[Young *et al.*, 2007] S. Young, J. Schatzmann, B. Thomson, K. Weilhammer, and H. Ye. The hidden information state dialogue manager: A real-world POMDP-based system. In *NAACL-HLT 2007 – Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics, Demonstration Program*, pages 27–28, Rochester, New York, 2007.

[Young *et al.*, 2013] S.J. Young, M. Gašić, B. Thomson, and J. Williams. POMDP-based statistical spoken dialogue systems: a review. *Proceedings of IEEE*, 101(5):1160–1179, 2013.

[Zhao and Eskenazi, 2016] T. Zhao and M. Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In *SIGDIAL'2016 – Proceedings of the 17th SIGdial Meeting on Discourse and Dialogue*, pages 1–10, Los Angeles, California, 2016.

[Zukerman and Partovi, 2017] I. Zukerman and A. Partovi. Improving the understanding of spoken referring expressions through syntactic-semantic and contextual-phonetic error correction. *Computer Speech and Language*, 2017.

[Zukerman *et al.*, 2015] I. Zukerman, S.N. Kim, Th. Kleinbauer, and M. Moshtaghi. Employing distance-based semantics to interpret spoken referring expressions. *Computer Speech and Language*, pages 154–185, 2015.