LOCALIZED ERROR DETECTION FOR TARGETED CLARIFICATION IN A VIRTUAL ASSISTANT

Svetlana Stoyanchev, Michael Johnston

Interactions Corporation*

ABSTRACT

We propose a novel approach for addressing automatic speech recognition (ASR) and natural language understanding (NLU) errors in an interactive spoken dialog system using targeted clarification (TC). TC applies when a spoken utterance is partially recognized by focusing a clarification question on the misrecognized part of the utterance. A key component of TC is accurate detection of localized ASR and NLU errors in an utterance. In this work, we develop statistical models of *presence* and *correctness* for domain concepts within an ASR/NLU result and use these to drive a targeted clarification (TC) strategy. We evaluate the accuracy of the models and their effect on the dialog strategy in an interactive multimodal assistant.

Index Terms— Clarification, dialog, error detection

1. INTRODUCTION

The ability to clarify information provided by a user is essential for automatic spoken and multimodal dialogue systems such as virtual assistants, automated tutors, or information access systems [1, 2, 3, 4]. To determine whether to accept, reject, or confirm user's input, dialog systems generally rely on confidence scores returned by a speech recognizer or estimated using a combination of speech recognizer result and dialog context [5, 6]. This is achieved using manually selected or automatically learned thresholds on the recognizer's confidence value. When the recognition confidence is above the accept threshold, the system accepts user's input. When the recognition confidence is between the reject and accept thresholds, the system may confirm the input either explicitly *"You said Leaving from San Francisco, is that correct?"* or implicitly *"Leaving from San Francisco. Where are you going to?"*. When recognition confidence is below the reject threshold, systems generally rejects the user's utterance with a prompt, such as *"Please repeat/rephrase","I'm sorry I didn't quite get that.","Can you please try again?"*.

A generic confirmation and rejection approach works well for slot-filling systems where user's utterances are concise and contain a single concept. However, in an intelligent dialog system that allows users to speak naturally and specify multiple concepts in a single utterance, generic rejection and confirmation has some significant disadvantages. The first disadvantage is a lack of naturalness. Research has shown that human speakers use targeted clarifications to recover from errors. Analysis of human dialogs shows that in the majority of cases speakers employ reprise questions, a type of targeted clarification, in which they ask a question that targets a segment of what they heard [7, 8]. The second disadvantage of generic confirmation and rejection is that it is not efficient for long utterances containing multiple concepts. For example, if a user said *What comedies are playing at the Angelica film center tomorrow?*, asking the user to repeat the whole sentence would significantly delay the user's access to information reflecting negatively on user experience.

In this work, we propose to **use targeted clarifications when a user's utterance is partially recognized**. A system may ask *"What type of movies do you want to see?"* or *"When do you want to see a comedy at Angelika Film Center?"* depending on which part of the utterance is misrecognized by the system. Our targeted clarification strategy uses localized error detection (LED). LED identifies when a targeted clarification is appropriate by detecting potential errors in the ASR or NLU. [9] propose the use of targeted clarifications for an open-domain speech-to-speech translation system. The authors identify error segments by predicting which words are likely to have been misrecognized by an ASR and generate a reprise targeted clarification question using rule-based natural language generation [10, 11].

In this work, we evaluate the use of targeted clarifications in a multimodal virtual agent system (MVA) providing access to information about movies, restaurants, and musical events [1]. In contrast with open-domain spoken systems, such as general purpose machine translation, the MVA application covers a specific set of domains with a fixed set of concepts and uses an NLU component to mark concepts in automatically recognized speech. Instead of identifying an error segment, the LED in MVA identifies which of the concepts are likely to be *present* and *correct* using domain knowledge, ASR, and NLU tags and scores. If at least one of the concepts is identified to be present but not correct, the targeted clarification (TC) component uses this information to generate a targeted clarification question.

The paper is organized as follows. In Section 2 we describe the system and the data set used in the experiments. We present our method and results for detecting localized er-

*The experiments described here were conducted while the authors were at AT&T Labs Research.

Domain	Tag
movies	MOVIE_EVENT_CAT, MOVIE_EVENT_VENUE, MOVIE_EVENT_TITLE
music	MUSIC_GENRE, MUSIC_EVENT_PERFORMER, MUSIC_EVENT_VENUE
restaurants	CUISINE, RESTAURANT_NAME, CHAIN_RESTAURANT
general	LOC, NBHOOD, TIME

Table 1. MVA Concepts.

rors and analysis of system actions in Sections 3 and 4. We conclude in Section 5 and discuss future directions.

2. SYSTEM AND DATA

The Multimodal Virtual Assistant (MVA) is a mobile application that allows users to plan a day or evening out with friends using spoken natural language and gesture input. Users can search and browse over multiple interconnected domains, including music events, movie showings, and places to eat.

Audio input is processed using the AT&T Watson^{SM} speech recognition engine [12]. Recognition utilizes a generic statistical N-gram language model trained on data from a broad range of different domains. Natural language understanding is performed in two steps. First a discriminative stochastic sequence tagger assigns domain specific concept tags to phrases within the input. An intent classifier then uses a combination of lexical features with phrase tags to assign the input to one of a set of intents.

For the experiments described in this paper we use an initial set of 2499 spoken user inputs that were collected in an initial trial version of the application. The utterances were manually transcribed and annotated with semantic tags and intents. Although not a controlled data collection, the recorded commands are nevertheless representative of the typical usage of the system and serve as a starting point for our evaluation.

The MVA domain has twelve concepts distributed across movies, music, restaurants, and three general concepts that apply across all of the domains (Table 1). A user's command specifies search criteria or refines search criteria using one or more of the concept types. For example, the NLU component will apply concept tags to a user command *"Jazz concerts in San Francisco next Saturday"* as: *[Jazz]/MUSIC_GENRE concerts around [San Francisco]/LOC [next Saturday]/TIME*. 41% of the utterances contain multiple concepts where targeted clarifications may apply.

3. METHOD

The NLU component of the system tags an automatically recognized input string. The baseline MVA system either accepts or rejects an utterance based on the NLU confidence score. On *accept*, the system executes the user's command. On *reject*, the system asks a generic clarification, e. g. *please repeat*. In addition to *accept* and *reject* actions, the proposed system may also ask a targeted clarification (*TC action*).

Feature	Description
LEX	words from the ASR output
NLU-conf	overall NLU confidence score
for each NLU-tagged concept:	
NLU-concept-score	average ASR confidence of tagged words
NLU-concept-var	variance of the ASR confidence of tagged words
NLU-concept-num	number of tagged words

Table 2. Features used for presence and correctness models.

The localized error detection (LED) component consists of the presence and correctness prediction modules (PRES, CORR) and the dialogue manager (DM). The LED component identifies mistakes in the ASR and the NLU of the system and the DM component uses this input to determine an optimal system action.

3.1. Presence and Correctness

We train maximum entropy models for binary classification of presence and correctness for each concept used in the MVA system [13]. The models are trained on the features generated by the ASR and NLU system components (Table 2). Lexical features (LEX) are the words, bigrams, and trigrams from the 1-best ASR output. The NLU features include overall NLU confidence score and NLU-per-concept features. For each concept identified by the NLU module, we compute the average ASR score of the words tagged with the concept, ASR score variance, and number of words in this concept[1].

The presence model returns for each of the twelve MVA concepts the probability that it is present in the input. The correctness model returns the probability for each concept of whether it is recognized correctly by the ASR. Using thresholds empirically derived on development data, we identify a set of *present* and *correct* concepts.

3.2. Dialogue Manager

The Dialogue manager (DM) identifies a set of *present & incorrect* (PI) and *present & correct* (PC) concepts from the result of PRES and CORR models. Using the following algorithm, DM selects the next system action:

- PRESENT ← the set of concepts with presence probability above threshold
- CORRECT ← the set of concepts with correct probability above threshold

[1]Numerical features are binned into four categorical values: Low, Med-Low, Med-High, High

ASR/NLU output				
Sent	Jazz concerts	in	San Francisco	tonight
ASR	Jazz concerts	in	ERROR	tonight
NLU	MUSIC_GENRE			TIME
LED prediction				
PRES	MUSIC_GENRE, TIME, **LOC**			
CORR	MUSIC_GENRE, TIME			
DM Table Lookup Values				
PC	MUSIC_GENRE, TIME			
PI	LOC			
Template	Where do you want to see MUSIC_GENRE concerts TIME?			
→	Where do you want to see Jazz concerts tonight?			

Table 3. LED/TC example.

- PC ← *PRESENT* ∩ *CORRECT*
- PI ← *PRESENT* ∩ *not CORRECT*
- If the set of PI is empty → **accept utterance**
- Else: If the set of PC is empty → **reject utterance**
- Else: Question ← Table lookup for PC and PI
- If Question not NULL → **ask a TC**
- Else: **ask a generic clarification question**

In Table 3 we show an example of processing a partially misrecognized sentence *"Jazz concerts in San Francisco tonight"*. ASR and NLU correctly recognize and identify the MUSIC_GENRE (*"jazz"*) and the TIME (*"tonight"*) concepts but fail to recognize and identify the LOC concept. The set of *present & correct* (PC) in this example includes MUSIC_GENRE and TIME. The set of *present & incorrect* (PI) in this example includes LOC. Using this information the DM looks up a clarification question template querying the LOC, and instantiates it with the correctly recognized values for the concepts MUSIC_GENRE and TIME.

4. RESULTS

4.1. Presence and Correctness Models

To evaluate the PRES and CORR components, we compute precision, recall, F-measure, and sentence concept accuracy for each of the models. We perform all experiments with 10-fold cross validation on the data set of 2499 sentences automatically recognized with a generic language model. True concept presence is identified from the manual annotations on the reference transcript. True concept correctness is identified by matching concept strings in the reference transcripts and automatic ASR/NLU output. A maximum entropy classifier returns the probability of presence/correctness for each concept [13]. In Table 4 we report the results with a threshold optimizing sentence concept accuracy of each model.

Precision of the *presence model* is the proportion of concepts correctly identified by the model as 'present'. Recall of the *presence model* is the proportion of concepts in reference that were successfully identified by the model as 'present'. Sentence concept accuracy is the proportion of all sentences in the data set where the model's prediction matches exactly the actual presence of the concepts in a reference.

The majority baseline assigning the most frequent set of concepts (LOC and MUSIC_GENRE occuring in 15.9% of the sentences) achieves an F-measure of .45. The NLU baseline method uses the output of the NLU system to predict presence by assigning *True* to the 'presence' value for each concept tagged by the NLU model. The NLU baseline method achieves an F-measure of .82 and a sentence accuracy of 67.4%. Using the LEX features, the model achieves an F-measure of .90 and a sentence accuracy of 77.6%. Using the NLU features, the model achieves an F-measure of .82 and a sentence accuracy of 66.4%, which is equivalent to the performance of the NLU system baseline. Not surprisingly, applying the maximum entropy classifier to the NLU features does not improve the performance over the *NLU system* baseline because NLU features are not indicative of concept presence. The performance using a combination of LEX & NLU features is equivalent to the performance using LEX features alone and outperforms the *NLU system* baseline by 10.4% points absolute on the sentence accuracy measure.

Method	P	R	F	Sent Con Acc
PRESENCE Baseline				
Majority	.38	.56	.45	15.9%
NLU System	.88	.76	.82	67.4%
PRESENCE Model				
LEX T=.4	.84	.96	.90	77.6%
NLU T=.4	.77	.88	.82	66.4%
LEX & NLU T=.5	.84	.96	.90	77.8%
CORRECTNESS Baseline				
Present-predicted	.73	.93	.82	66.4%
NLU System	.79	1.0	.88	80.2%
CORRECTNESS Models				
LEX T=.4	.91	.96	.93	88.3%
NLU T=.5	.91	91	.91	83.4%
LEX & NLU T=.5	.92	.96	.94	88.8%

Table 4. Precision, Recall, F-measure (P,R,F), and sentence concept accuracy evaluation of the *presence* and *correctness* models predicting whether a concept is present/correct in a user's utterance and is correctly recognized by the system.

Precision of the *correctness* model is the proportion of concepts identified by the model as 'correct' that are correctly recognized by the system. Recall of the *correctness model* is the proportion of correctly identified concepts that were successfully identified by the model as 'correct'.

The *Present-predicted* baseline assigns 'correct' tag using *presence* model assignment with *LEX & NLU, T=.5* parameters and achieves 66.4% overall sentence accuracy. The *NLU*

system baseline assigns 'correct' tag to all concepts tagged and recognized by the system correctly and achieves 80.2% sentence accuracy. It has a recall of 1.0 as the set of correct hypothesis tags is a subset of all correctly recognized tags.

With LEX features alone, the model achieves F-measure of .93 (.05 points above the *NLU system baseline*) and sentence accuracy 88.3%. The increase in performance using LEX features alone over the baseline is not surprising since the *correctness* models combined presence and correctness: a concept can be correct only when it is present. Hence, the *correctness* model benefits from lexical features for the same reasons as the *presence* model. With NLU features alone, the model achieves F-measure .91 (.03 points above the *NLU system* baseline) and sentence accuracy 83.4%. Combining LEX & NLU features, the model achieves F-measure of .94 (.06 points above the *NLU system baseline*) and sentence accuracy of 88.8% outperforming each individual feature sets. While LEX features are the most salient in the prediction of correctness NLU features are also useful.

4.2. System Action

To evaluate the impact of the proposed TC component on the system, we analyze the effect of the model on the system actions. Table 5 shows frequencies of clarification questions asked using Oracle prediction of *presence* and *correctness* and using the proposed models. With an Oracle prediction, the system accepts 63% of the utterances, rejects 22% and asks a targeted clarification for 14%. Hence, in 38% (=14/(22+14)) of the cases where a generic error recovery (without TC) rejects an utterance, the proposed system with Oracle presence and correctness would ask a targeted clarification question.

System action depends on the threshold on the presence and correctness models. We first evaluate system actions using thresholds that achieved the highest sentence accuracy on the data set (T-pres=T-corr=.5). This model accepts 68% of utterances with precision and recall .83 /.90, asks a targeted clarification for 15% of the utterances with precision and recall of .66, and rejects 18% of the utterances with precision and recall .85 /.65. In comparison, Model 2 with a lower presence threshold (T-pres=.1) accepts 57% of the utterances with precision and recall of .92 /.83. It asks a targeted clarification with precision and recall of .59 /.75, and rejects with precision and recall .79/.86.

In Table 6, we further analyze error types for the two models. False accept errors occur when the system accepts an utterance that should have been rejected or clarified. False reject/TC errors occur when the system rejects or clarifies an utterance that should have been accepted. Asking a TC instead of a rejection will result in an inappropriate clarification question [14]. With lower presence threshold, more concepts are predicted to be present by Model 2, resulting in higher proportion of false rejects and TCs and lower false accepts. Threshold values can be adjusted for a desirable system behavior.

Gold/Model	ACC	TC	REJ
Clarification Questions Asked			
Oracle	63% (1577)	14% (362)	22% (560)
Model 1	68% (1707)	15% (364)	18% (428)
Model 2	57% (1420)	18% (461)	25% (618)
Precision / Recall			
Model 1	.83 / .90	.66 / .66	.85 / .65
Model 2	.92 / .83	.59 / .75	.78 / .86

Table 5. System actions using Oracle and experimental models where Model 1 uses thresholds T_pres=T_corr=.5. Model 2 uses thresholds T_pres=.1, T_corr=.5.

Error type	False Accept	False Reject/TC	Inapp TC
Model 1	282 / 922 (28%)	152 / 1577 (10%)	19 / 560 (3%)
Model 2	112 / 922 (11%)	269 / 1577 (17%)	32 / 560 (6%)

Table 6. System actions using Oracle and experimental models where Model 1 uses thresholds T_pres=T_corr=.5. Model 2 uses thresholds T_pres=.1, T_corr=.5.

5. CONCLUSIONS

We propose using targeted clarifications for error recovery from ASR and NLU errors in a virtual assistant dialog system. Our approach is motivated by the more natural and efficient targeted clarification strategy in comparison with the generic rejection strategy commonly used for error recovery. In this work, we evaluated the models of *presence* and *correctness* that drive a targeted clarification strategy and their effect on the system's actions. With Oracle presence and correctness detection, 38% of errors can be clarified by the system with a targeted clarification. We have shown that a maximum entropy model trained on a combination of lexical and NLU features achieves a significant improvement over the baseline methods in predicting whether a concept is present and/or correct in a user's utterance. We find that lexical context features are especially useful for the both presence and correctness models. By optimizing presence and correctness thresholds, the system can be tuned to minimize false accept or false reject errors.

In the future work, we will explore other features, such as ASR n-best hypotheses and domain knowledge mined from external sources. We will evaluate targeted clarification strategy in a larger user trial and compare its performance with the baseline generic rejection system. We will also explore automatically optimizing the clarification policy using on-line reinforcement learning with real users using output of the presence and correctness models as features.

6. REFERENCES

[1] M. Johnston et al., "MVA: The Multimodal Virtual Assistant," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 257–259.

[2] D. J. Litman and S. Silliman, "Itspoke: an intelligent tutoring spoken dialogue system," in *Demonstration Papers at HLT-NAACL 2004*, Stroudsburg, PA, USA, 2004, HLT-NAACL–Demonstrations '04, pp. 5–8, Association for Computational Linguistics.

[3] K. Acomb et al., "Technical support dialog systems: issues, problems, and solutions," in *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, 2007, p. 2531.

[4] D. Bohus and E. Horvitz, "Dialog in the open world: platform and applications," in *Proceedings of the 2009 international conference on Multimodal interfaces*, 2009, p. 3138.

[5] D. Bohus and A. I. Rudnicky, "A principled approach for rejection threshold optimization in spoken dialog systems," in *INTERSPEECH*, 2005, pp. 2781–2784.

[6] K. Komatani and T. Kawahara, "Flexible mixed-initiative dialogue management using concept-level confidence measures of speech recognizer output," in *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*. 2000, COLING '00, pp. 467–473, Association for Computational Linguistics.

[7] M. Purver, *The Theory and Use of Clarification Requests in Dialogue*, Ph.D. thesis, King's College, University of London, 2004.

[8] J. Ginzburg and R. Cooper, "Clarification, ellipsism and the nature of contextual updates," *Linguistics and Philosophy*, vol. 27, no. 3, 2004.

[9] S. Stoyanchev, A. Liu, and J. Hirschberg, "Clarification questions with feedback 2012.," in *Interdisciplinary Workshop on Feedback Behaviors in Dialog*, 2012.

[10] S. Stoyanchev, P. Salletmayr, J. Yang, and J. Hirschberg, "Localized detection of speech recognition errors.," in *SLT*. 2012, pp. 25–30, IEEE.

[11] S. Stoyanchev, A. Liu, and J. Hirschberg, "Towards natural clarification questions in dialogue systems," in *Proceedings of AISB2014*, 2014.

[12] V. Goffin et al., "The AT&T WATSON speech recognizer," in *Proceedings of ICASSP*, Philadelphia, PA, USA, 2005, pp. 1033–1036.

[13] P. Haffner, "Llama – General software library for large margin classifiers," http://www.research.att.com/∼haffner/llama.

[14] A. Liu et al., "Detecting inappropriate clarification requests in spoken dialogue systems," in *Proceedings of the 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 2014, pp. 238–242.