*INTERSPEECH 2016*
September 8–12, 2016, San Francisco, USA

# Making Personal Digital Assistants Aware of What They Do Not Know

*Omar Zia Khan, Ruhi Sarikaya*

Microsoft Corporation, Redmond WA 98052, USA
{OmarZia.Khan, Ruhi.Sarikaya@microsoft.com}

## Abstract

Personal digital assistants (PDAs) are spoken (and typed) dialog systems that are expected to assist users without being constrained to a particular domain. Typically, it is possible to construct deep multi-domain dialog systems focused on a narrow set of head domains. For the long tail (or when the speech recognition is not correct) the PDA does not know what to do. Two common fallback approaches are to either acknowledge its limitation or show web search results. Either approach can severely undermine the user's trust in the PDA's intelligence if invoked at the wrong time. In this paper, we propose features that are helpful in predicting the right fallback response. We then use these features to construct dialog policies such that the PDA is able to correctly decide between invoking web search or acknowledging its limitation. We evaluate these dialog policies on real user logs gathered from a PDA, deployed to millions of users, using both offline (judged) and online (user-click) metrics. We demonstrate that our hybrid dialog policy significantly increases the accuracy of choosing the correct response, measured by analyzing click-rate in logs, and also enhances user satisfaction, measured by human evaluations of the replayed experience.

**Index Terms**: personal digital assistants, dialog policies, dialog fallback

## 1. Introduction

PDAs, also known as intelligent personal assistants or virtual assistants, are software systems that can understand user requests and queries in natural language (spoken and/or typed) by leveraging personalization and contextual information to provide appropriate responses [1]. The type of tasks handled by a PDA can range from command and control (calling, music playback, thermostat control, reminder creation) to infotainment (weather, local business lookup, factoid based question answering, stock price), some hybrid of both (such navigation instructions to a restaurant) or even some random conversation in the form of chit-chat. Recently, various mobile and desktop operating systems are integrating PDAs in their core [2, 3, 4] and various stand-alone applications are also offering the functionality of a PDA [5, 6, 7].

PDAs are similar to open-domain dialog systems that comprise automatic speech recognition (ASR), natural language understanding (NLU), dialog management, language generation and text-to-speech components. The complexity of handling natural language queries effectively can often result in no valid interpretation available to a PDA. This can happen due to possible ASR errors, issue with NLU models not analyzing the query correctly [8, 9], or a mistake in the dialog belief state tracking process. In such situations, a fallback strategy is needed.

Even without mistakes, the inherent complexity of implementing open-domain dialog systems can require the use of a fallback strategy. Users are not expected to constrain themselves to specific keywords and not limit their conversation to a particular topic or domain. PDAs are typically developed to handle head domains by providing deep and rich dialog experiences, and provide some shallow answers to the less frequent topics. However, for tail queries the PDA may not have any valid response and will need a fallback strategy.

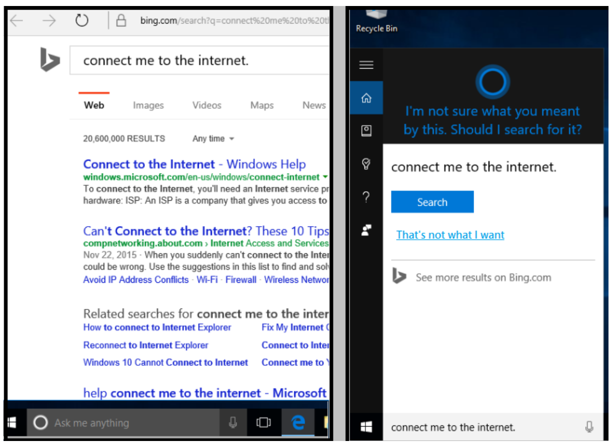
Two common fallback strategies are shown in Fig 1. The first involves the PDA explicitly acknowledging its inability to respond to the user's request whenever it doesn't have a valid response (right screen in Fig 1). The second strategy is web fallback, i.e., always present web search results with the anticipation that the user's intent would be fulfilled through those results (left screen in Fig 1). Each approach has its merits. The first is preferable when an error was made (such as an ASR error) and web search results won't be useful. Furthermore, showing web results typically results in a context switch and makes reformulating the query or correcting the ASR more cumbersome. The first approach is also better if it is clear that the user is conversing with the PDA rather than expecting some knowledge-based lookup. The second strategy is useful when user is asking for some information or instructions which the PDA may not be able to provide but the user would be able to locate on the web.

In this paper, we empirically analyze the use of these two fallback strategies using query logs from Cortana, Microsoft's PDA deployed to millions of users. We demonstrate that each approach is useful in different situations but a hybrid approach that can pick and choose between employing one or the other fallback strategy would be ideal. We present two different methods to construct a dialog policy that can determine which fallback strategy to employ for each query. The first method uses the ASR confidence and the second method is a machine learning classifier that analyzes various characteristics of a query. We evaluate these different dialog policies and report results using Cortana logs through click-based offline results as well as human-judged online evaluations.

## 2. Problem Statement

We formulate the problem of choosing the right fallback as a binary classification task. The first alternative is to show a fallback screen in which the PDA acknowledges it does not understand the user but offers to search the web (right screen in Fig 1). The second alternative is to show web search results (like in web browser) and hope the user can locate the information in it. In this paper, we do not assign any utility values to different types of errors and assume that the mistake of showing a web search is no worse or better than the mistake of showing a fallback screen. However, for other scenarios one error may cause greater user dissatisfaction than the other. Our approach can easily be adapted to incorporate such information.

Modern search engines offer search results beyond the ten



connect me to the internet.

20,600,000 RESULTS

Connect to the Internet - Windows Help
windows.microsoft.com/en-us/windows/connect-internet
To connect to the Internet, you'll need an Internet service pr
hardware: ISP: An ISP is a company that gives you access to

Can't Connect to the Internet? These 10 Tips
compnetworking.about.com › Internet Access and Services
Nov 22, 2015 · When you suddenly can't connect to the Inter
could be wrong. Use the suggestions in this list to find and sol
Avoid IP Address Conflicts · Wi-Fi · Firewall · Wireless Networ

Related searches for connect me to the inter
How to connect to Internet Explorer | Fix My Internet C
Reconnect to Internet Explorer | Connect to Inter
Windows 10 Cannot Connect to Internet | Connect me to Y

help connect me to the internet - Microsoft

I'm not sure what you meant by this. Should I search for it?

connect me to the internet.

That's not what I want

See more results on Bing.com

connect me to the internet.

Figure 1: *Alternate fallback strategies. Left screen shows web results and right screen shows fallback screen for same query*

blue links [10] in the form of entities (person, movie, local business) or enriched answers (currency conversion, weather, stock price). If a query would result in a web search with such an enriched answer it would be better to offer web results rather than the fallback screen. For the purpose of this paper, we consider the presence of an enriched web result as an extension of the PDA and exclude such queries from our analysis of different fallback strategies as we defer to a deterministic choice (showing web results) in this case. We also limit our analysis to the first query of a session. In the middle of a session, the user is trying to complete a task with the system likely making a mistake in understanding the user's intent, so allowing the user to correct the system through a fallback screen would be better than looking at web search results. In fact, showing web results in the middle of the task will result in abrupt termination of that task with loss of context, and the user will have to restart the whole process resulting in greater user dissatisfaction. For both these cases, we exclude such queries from our analysis to avoid undue benefit when a deterministic policy would be better.

It is worth noting that nature of queries for the PDA is likely different than those in web results [11], primarily because search engines typically receive keyword-based queries as opposed to natural language queries. For this reason, if no enriched answer is available, the user is unlikely to receive their desired result on a web search without a click. Thus, clicks are a strong indicator of user satisfaction when employing web search as a fallback strategy for PDAs.

## 3. Dialog Policies for Fallback

In this section, we describe the various approaches to developing dialog policies. The first two approaches presented are agnostic to the actual user query and are treated as possible baseline policies. The subsequent two approaches choose a particular fallback strategy by examining the characteristics of the user query.

- **Always Fallback Screen (AFS):** This dialog policy is considered a baseline and in this policy the PDA always presents the fallback screen (right side of Fig 1) but offers the user an option to review web search results. The user can respond in affirmative and get web results. Otherwise the user can reject this and either terminate the current interaction or possibly continue with a rephrasing of the query.
- **Always Web Results (AWS):** This dialog policy is considered an alternative baseline and in this policy the PDA always presents web results (left side of Fig 1). If the user is not interested in web results, they do not need to click on them and can return to the PDA to continue the interaction or terminate the session. AWS is the exact opposite of AFS.
- **ASR Confidence Threshold (ACT):** A subset of the mistakes made by the PDA are caused by ASR errors. To target those cases, we use a dialog policy based on ASR confidence. The intuition here is that if the ASR confidence is low, it is more likely that the ASR made a mistake in its recognition and offering web results would be counterproductive. On the other hand, if the ASR confidence is high, it is less likely that the ASR misrecognized the user so offering web results can be helpful. This policy is based on choosing an ASR threshold, $\theta$, that is applied to each query, $q_i$ with the ASR confidence, $\mathbb{C}_i$, such that $\text{ACT}(q_i) = \begin{cases} \text{if } \mathbb{C}_i < \theta & \text{then AFS} \\ \text{otherwise} & \text{AWS} \end{cases}$
  The threshold, $\theta$, is determined by analyzing logs to optimize the F1 score.
- **ML Classier (MLC):** This policy is based on a machine learning classifier that analyzes different aspects of the query. We evaluate classifiers based on SVM, Logistic Regression and Boosted Decision Tree algorithms, with each classifier sharing the same feature space. The feature space comprises around 75 features that can be grouped in different categories:
  - **ASR Features:** that include the ASR confidence, acoustic model and language model scores, ASR nbest list size, and relative difference and ratio of top vs. second best ASR choice.
  - **Word-Level Features:** that include information regarding the raw query such length of the query, existence of different classes such as stop words/numerical characters/carrier phrases (e.g., "can you", "tell me", "how can I").
  - **NLU Features:** that include best domain, intent, confidence score, NLU nbest list size, and the relative difference and ratio of top vs. second best NLU choice, and existence of various tagged entities from NLU (e.g., places, people, dates).

## 4. Evaluation

In this section, we evaluate various dialog policies described in the previous section on queries extracted from Cortana logs from real-world users. We describe our data set, define our evaluation metrics, and then report our results.

### 4.1. Data Set

We randomly sampled Cortana usage logs and filtered for queries which didn't have any Cortana response or enriched answer on Bing. Our filtered sample has 318,100 speech queries. We separate 222,670 queries for training MLC and determining the threshold for ACT. All policies were tested on the remainder 95,430 queries. AWS policy was enabled on Cortana for these logs, so we also have click information for this set. A click

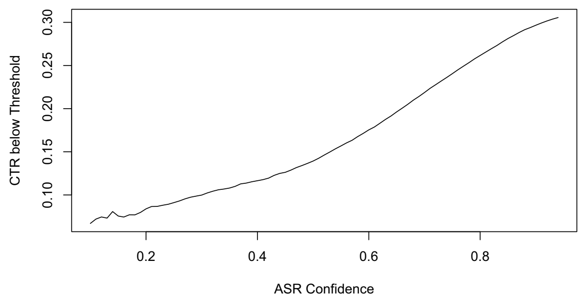


Figure 2: *CTR plotted against the ASR threshold to show that queries with higher confidence have higher CTR*

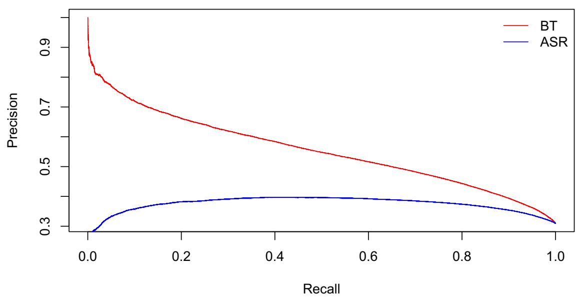


Figure 3: *Precision vs Recall curve for ACT and MLC - BT*

| | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| AFS | 60.1% | - | - | - |
| AWS | 30.9% | 30.9% | 100% | 47.0 |
| ACT | 49.5% | 36.5% | 85.1% | 51.1 |
| MLC - SVM | 56.6% | 39.8% | 78.7% | 52.8 |
| MLC - LR | 60.5% | 42.4% | 77.2% | 54.7 |
| MLC - BT | 65.5% | 46.4% | 74.8% | 57.3 |

Table 1: Offline (Click-based) Evaluation of Dialog Policies

indicates showing web results was the correct choice because user is unlikely to receive the information without clicking in the absence of enriched answers. We treat the presence/absence of a click as our label to indicate if AWS or AFS is the correct choice. Beyond clicks, we also asked human judges to evaluate 2,500 queries by viewing the two dialog policies side-by-side (such as in Fig 1) and manually label which is better. We also evaluate our policies on this annotated set to measure user satisfaction using the human judged labels.

### 4.2. Evaluation Metrics

Framing the problem as a binary classification enables us to use standard evaluation measures. We assume that showing the web search result when the user needed a web search is the true positive (TP), showing the web results when the user didn't need them is false positive (FP), showing the fallback screen when the user needed web search is false negative (FN), and finally showing fallback screen when user didn't want web results is true negative (TN). Given these definitions, for each dialog policy we report the Accuracy, Precision, Recall, and F1 score.

### 4.3. Offline Evaluation through Logs

Of the 318,100 queries we sampled from the logs, 98,394 had clicks which provided a 30.9% click rate. For AFS, we have TP = 0 (we never show the web results which is our positive label) and TN = 219,706. For AWS, we have TP = 98,394 and TN = 0 (we always show web results). Table 1 shows results for our baseline dialog policies (AFS and AWS). We can see that while AFS provides us with 60.1% accuracy, its F1 score is 0 due to no positive predictions (never correctly predicting when the user would have benefited from web results). AWS has a lower accuracy, however, it does have a higher F1 score as it is able to predict certain positive cases.

To report results for ACT, we need to determine the ASR confidence threshold to use as cut-off. First, we analyze the relationship between the ASR confidence and the click-through rate (CTR). In Fig 2, we plot the CTR on the vertical axis for queries that are below a certain ASR threshold. The ASR confidence itself is plotted on the horizontal axis. Setting ASR confidence = 1 includes all queries and we see the maximum CTR = 0.309. We can also see that the CTR increases as the ASR confidence increases confirming our intuition that users are more likely to click through on results when the query has been recognized correctly, with the ASR confidence acting as a proxy for correct recognition. The slight instability at the lower end of ASR confidence is because fewer cumulative queries are available as the threshold decreases.

Next, we plot the Precision-Recall curve based on ASR confidence in Fig 3. This curve also shows us that we can set the threshold to 1 to get the behavior of AWS that will provide 100% recall but with lower 30.1% precision. The unusual dips on both ends of the ASR curve indicate the ASR is overconfident in its calibration when outputting extreme confidence values (near 0 and 1). We plot the F1 score against the ASR confidence cut-off in Fig 4 to identify the ASR threshold that optimizes for the F1-score. We report results in Table 1 for ACT, showing improvements in the accuracy and precision at the cost of lowering the recall, but still increasing the F1 score.

We also reports results for MLC across different metrics in Table 1. We use different three different binary classification algorithms, namely SVM (MLC - SVM), logistic regression (MLC - LR), and boosted trees (MLC - BT). We report results for each classifier separately. No parameter tuning is performed on any classifier. We choose the threshold for each classifier in the same manner as for ACT. We see that all MLC variants are better than ACT on all metrics, except recall, as MLC variants have access to a larger number of features and can model additional effects that cannot be captured by ACT which is only using the ASR threshold. We can also see that MLC - LR and MLC - BT have a higher accuracy than AFS but are still able to show web results in a large number of cases where the user would benefit from them. We show the P/R curve and F1/cutoff curve for MLC - BT in Figs 3 and 4.

In this paper we have assumed the cost of a false positive is the same as the cost of a false negative. If we prefer one over the other, it is straightforward to incorporate this preference. We only need to consider a weighted harmonic mean (with different weights assigned to precision and recall reflecting their importance) instead of a balanced F1 and choose a threshold that maximizes this weighted F1. The rest will stay the same.

|  | Accuracy | Precision | Recall | F1 Score |
|---|---|---|---|---|
| AFS | 51.0% | - | - | - |
| AWS | 48.9% | 48.9% | 100% | 65.7 |
| ACT | 59.8% | 56.1% | 81.6% | 66.5 |
| MLC - SVM | 66.4% | 63.0% | 75.9% | 68.9 |
| MLC - LR | 70.5% | 68.1% | 74.6% | 71.2 |
| MLC - BT | 72.0% | 71.8% | 70.3% | 71.0 |

Table 2: Online (Human Judged) Evaluation of Dialog Policies

### 4.4. Online Evaluation through Manual Annotations

We also evaluate the different dialog policies on the data set annotated by human judges. The results for different classifiers are shown in Table 2. We can see that human judges have marked more queries as better for web search (48.9%) where as the click rate was 35% on this set. However, the evaluation results are still similar. ACT is better than AFS and AWS for both accuracy and F1 score. Similarly, all three MLC variants also have higher accuracy and F1 score than ACT. Finally, we also see that MLC - BT has the highest accuracy with an F1 score that is almost as close as MLC - LR. This leaves MLC - BT as the best classifier from both online and offline analysis.

### 4.5. Discussion

Analyzing the results of different policies, we can understand when showing web vs. showing fallback screen would be better. If the ASR confidence is low and the resulting query resembles gibberish or not something a human would search, AFS would be better, e.g., {*User: "ATM", ASR: "8am near me"*}. If the ASR has made a mistake but the recognized string contains sufficient information for NLU to indicate it would have been a Cortana query and not a web search, MLC can prevent from showing web results even if the confidence is high, e.g., {*User: "Set my alarm to 6:30 in the morning", ASR: "30 in the morning", ASR conf = 0.65*}. Alternatively, if the ASR confidence is low but the recognized utterance is correct, AWS will be better than AFS, e.g., {*User = ASR = "Can Android use Cortana?", ASR conf = 0.24*}.This is also a case of why MLC variants can out-perform ACT as they are not limited to the use of the ASR confidence, but can include additional features. On the other hand, MLC can also make mistakes that ACT can avoid if the ASR confidence is low but the ASR output resembles a valid query, e.g., {*User: "Who is Pharrel Williams", ASR: "Who is L Williams", ASR conf = 0.14'*}. Finally, if the ASR confidence is low but the ASR mistake is minor, web results can still be better than blocking the search, e.g., {*User: "Gemini and Libra compatibility", ASR: "Gemini and Libra compatibility I know", ASR conf = 0.44'*}. The variety of cases listed indicates that having more features is useful instead of relying only on ASR confidence, and that is why MLC outperforms ACT.

## 5. Related Work

There has been a recent trend of developing open domain dialog systems by dispatching a query to multiple independent components and then analyzing their results [12, 13]. If the response from any component is relevant to the current query, that answer is presented to the user, otherwise some fallback is needed. This paper directly addresses what do when none of the available answers are deemed relevant to a query.

Often multi-domain dialog systems also rank or rerank alternative analyses [12, 14, 15] to determine the correct interpretation as more knowledge is available at subsequent layers of the system. Our approach is complementary with such ranking/reranking as we are catering for the case where the ranker has already determined that no valid analysis is available, and thus a fallback strategy is needed, which is what provide.

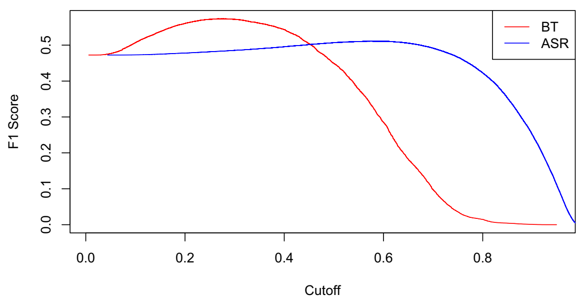


Figure 4: *F1 score plotted vs cutoff to choose optimal cut-off*

Reinforcement learning (RL) [16] has also also been proposed for dialog management [17], especially for confirmation and clarification of actions [18]. Fallback strategies can be considered as a stand-alone next step or otherwise combined inside RL as another available choice. We chose the former and leave the latter extension as future work.

A related problem is to identify when to transfer a caller to an operator from an automated dialog system [19, 20]. This is similar as the operator here is a fallback, but different because while the operator can address any user query, web search can only be useful for specific queries so web search as a fallback can in fact be counterproductive.

Certain dialog systems offer repair acts [21, 22, 23] to rectify errors made earlier in the interaction. Our fallback strategy is slightly different; we are handling the case when we are confident that we don't have a valid response in the current turn but a mistake hasn't been committed yet. The repair act can be triggered in the follow-up turn since we already know there is a higher likelihood user didn't receive the expected response.

## 6. Conclusions and Future Work

In this paper we have presented an approach to enable PDAs to be self-aware and choose the correct action when they have no valid response available. We construct policies that explicitly model the choice between showing a fallback screen or presenting web results. We sample real Cortana user logs and demonstrate improvements in the accuracy and F1 score for the fallback choice using both click log and human judgments. We also discuss that the approach can be easily extended to reduce different types of errors.

We plan to explore the use of n-gram based features for MLC to study possible accuracy improvements. We also plan to examine follow-up turns in the logs for these queries to look for repeats or reformulations as they can provide a strong signal for user dissatisfaction to use in continual refinement. We can also extend this approach for follow-up turns using an RL approach to avoid abrupt task cancellations due to an ASR or NLU or dialog error by introducing a new fallback screen that confirms before terminating or switching the task if there is sufficient uncertainty.

## 7. References

[1] R. Sarikaya, "The technology powering personal digital assistants," Interspeech Keynote Presentation : http://www.superlectures.com/interspeech2015/the-technology-powering-personal-digital-assistants, Dresden, Germany, September 2015.

[2] "Siri," http://www.apple.com/ios/siri/.

[3] "Cortana," http://windows.microsoft.com/en-us/windows-10/getstarted-what-is-cortana.

[4] "Google Now," http://www.google.com/landing/now/.

[5] "Alexa," https://developer.amazon.com/public/solutions/alexa.

[6] "Hound," http://www.soundhound.com/hound.

[7] "Dragon," http://www.nuance.com/dragon/index.htm.

[8] H. Erdogan, R. Sarikaya, Y. Gao, and M. Picheny, "Semantic structured language models," in *Proceedings of 7th International Conference on Spoken Language Processing (ICSLP/INTERSPEECH)*, 2002.

[9] A. Deoras, G. Tur, R. Sarikaya, and D. Hakkani-Tr, "Joint discriminative decoding of words and semantic tags for spoken language understanding," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1612–1621, 2013.

[10] D. Chen, W. Chen, H. Wang, Z. Chen, and Q. Yang, "Beyond ten blue links: Enabling user click modeling in federated web search," in *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining (WSDM)*, 2012.

[11] A. Celikyilmaz, G. Tur, and D. Hakkani-Tur, "IsNL? A Discriminative approach to detect natural language like queries for conversational understanding," in *Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2013.

[12] J.-P. Robichaud, P. A. Crook, P. Xu, O. Z. Khan, and R. Sarikaya, "Hypotheses ranking for robust domain classification and tracking in dialogue systems," in *Proceedings of the 15th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2014.

[13] M. Gasic, N. Mrksic, P. hao Su, D. Vandyke, T.-H. Wen, and S. J. Young, "Policy committee for adaptation in multi-domain spoken dialogue systems," in *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015.

[14] M. Nakano, S. Sato, K. Komatani, K. Matsuyama, K. Funakoshi, and H. G. Okuno, "A two-stage domain selection framework for extensible multi-domain spoken dialogue systems," in *Proceedings of 12th Annual Meeting on Discourse and Dialogue (SIGDIAL)*, 2011.

[15] O. Z. Khan, J.-P. Robichaud, P. Crook, and R. Sarikaya, "Hypotheses ranking and state tracking for a multi-domain dialog system using multiple ASR alternates," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH 2015)*, 2015.

[16] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. MIT Press, 1998.

[17] J. D. Williams and S. Young, "Partially observable Markov decision processes for spoken dialog systems," *Computer Speech and Language*, vol. 21, no. 2, pp. 393–422, April 2007.

[18] Z. Wang, H. Chen, G. Wang, H. Tian, H. Wu, and H. Wang, "Policy learning for domain selection in an extensible multi-domain spoken dialogue system," in *Proceedings of Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[19] E. Horvitz and T. Paek, "Complementary computing: policies for transferring callers from dialog systems to human receptionists," *User Modeling and User-Adapted Interaction*, vol. 17, no. 1, pp. 159–182, 2007.

[20] A. Schmitt, C. Hank, and J. Liscombe, *Proceedings of the 4th IEEE Tutorial and Research Workshop on Perception and Interactive Technologies for Speech-Based Systems: Perception in Multimodal Dialogue Systems*. Springer Berlin Heidelberg, 2008, ch. Detecting Problematic Dialogs with Automated Agents, pp. 72–80.

[21] P. Heeman and J. Allen, "Detecting and correcting speech repairs," in *Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL)*, 1994, pp. 295–302.

[22] M. G. Core, "Dialog parsing: From speech repairs to speech acts," University of Rochester Rochester, NY, USA, Rochester, NY, USA, Tech. Rep., 1999.

[23] M. Frampton and O. Lemon, "Using dialogue acts to learn better repair strategies for spoken dialogue systems," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008.