LEVERAGING CALL CONTEXT INFORMATION TO IMPROVE CONFIDENCE CLASSIFICATION

Michael Levit

Speech at Microsoft, Microsoft Corporation

ABSTRACT

This paper describes how speech recognition confidence estimation in a typical Directory Assistance scenario can be improved by taking dialog context into account and re-calibrating the original recognition confidences using a statistical classifier that employs classification features extracted from this context. We look at several types of classification features and investigate their utility with respect to semantic and sentence error rates with a view to an improved application behavior, but also with a long term goal of a more efficient semi-supervised selection of model training material. The method leads to significantly better tradeoffs between correct and false recognitions with respect to both error metrics.

Index Terms— Confidence Classification, Dialog Context, Directory Assistance, Feedback Loop

1. INTRODUCTION

Recognition confidences have been used for many years to cope with errors in Automatic Speech Recognition. A confidence of a recognition hypothesis is an estimate of its reliability, of how much we can trust this recognition result. Recognition confidence is often related to the conditional posterior probability of the recognition hypothesis being correct given the acoustic signal [1]. One practical interpretation of this relation is that if we take a large number of recognition hypotheses from an ASR system that all share the same confidence α, on average $100 * \alpha$ percent of them will be correct. It should be noticed however that the identity relation between the two quantities is not strictly required; in practice, having a monotonic dependency is often sufficient. Another view of recognition confidences is embodied in the black box paradigm that converts a multitude of parameters, properties and results of a particular recognition run into a real number, a relation often implemented as a statistical classifier. In the latter case — and this is the one that we adopted for our experiments — the process of extracting confidence is termed *confidence classification*. For a comprehensive review on confidence measures in speech recognition, see [1].

Several factors can lead to recognition mistakes, such as inaccurate models and difficult/mismatched acoustic conditions. As a result, many authors suggest using accuracy indicators from different abstraction levels, such as acoustic or phone level, word level and utterance level [2, 3, 4]. These indicators are treated as features for a binary statistical classifier whose output score (possibly normalized) serves as the final confidence. Separate confidences can be computed for n-best hypotheses as well [5]. To train a successful confidence classifier, one needs a large amount of labeled in-domain training material. Ideally, we would like to train a separate confidence classifier for each language domain and speech application. Alternatively, a generic classifier can be trained and then re-calibrated in a separate step on domain specific data possibly using domain specific features [6, 7].

The approach presented in this paper is primarily intended for interactive human-machine communication scenarios. It follows the re-calibration scheme but also adds a new dimension of classification features to the classifier: *dialog context*. Our method suggests taking into account information about previous dialog turns within each given human-machine dialog. Unlike other methods that utilize dialog-related features such as rejections or elapsed time to ascertain quality of the dialog as a whole (see, for instance, [8, 9]), we stay focused on utterance level confidences. As a special extension of our method, we also consider the oracle case where for each utterance not only previous turns but also future turns are allowed to provide classification features. This extension is useful if our goal is to collect large amounts of in-domain training material for future model (re-)training (*feedback loop*) while avoiding tedious manual transcription effort.

The remainder of this paper is structured as follows: in Section 2 we describe the Directory Assistance application used as a test bed for our experiments. Section 3 explains how classification features are generated. Section 4 presents confidence re-calibration experiments for *locality* and *listing* states of the application with respect to semantic and sentence error rates. We conclude the paper with a short summary and propose directions for future investigations.

email: mlevit@microsoft.com

2. DIRECTORY ASSISTANCE APPLICATIONS

The Directory Assistance application we selected for this experiment is situated within the *Premium DA* scenario [10]. First the system asks for city and state, and then for a listing name. If the subsequent business search is successful the corresponding phone number is released to the caller. Both steps allow for confirmation sub-dialogs. In addition, since callers are paying for this premium service, they always have an option to get connected to a human operator. This can happen either following an explicit request by the caller or if the automated dialog encounters difficulties. A high level diagram of the application call flow is shown in Figure 1.

The application is implemented using *VoiceXML* [11] and utilizes the available logging functionality to write logtags about events happening in each call. For instance, one logtag entry type contains the original recognition confidence for the locality state and another indicates that the call went into a confirmation sub-dialog after the listing input state, and so on. There are about 1200 distinct types of logtag entries in the entire application.

The application is built around the Microsoft speech recognition engine [12]. The recognizer has a pre-trained application-independent confidence classifier implemented as a Multilayer Perceptron that takes a variety of features such as posterior probability, acoustic stability, language model fan-out, and others, generated from recognition lattices.

3. CLASSIFICATION FEATURES

In order to make the results of our approach generalizable to other application scenarios, we decided to adopt a "non-invasive" strategy and extract classification features for confidence re-calibration from the information in the already existing logtag entries. We did not alter the application in any way to introduce logtags specifically intended to convey information we believe would improve confidence re-calibration process.

Thus, the first subtask of the feature extraction task became to automatically reverse-engineer definition domains for each of the 1200 logtag types thrown by the application. This has been done on a random set of 40K calls. Based on this analysis about 200 logtags have been dismissed as they did not occur enough in the sample. Of the remaining one thousand, 90% ended up being binary (presence) features and the rest was equally split among numerical, categorical and string feature types.

From the perspective of each utterance subjected to confidence re-calibration, all features can be grouped into four categories according to how much call context[1] and deep knowledge of the application is needed to generate them. These categories (levels) are described below.

1. **First level features** neither require any knowledge of the dialog future, nor pre-suppose any additional application insights. They include all logtags that have been thrown prior or during recognition of the utterance in question, up to the moment when new input was solicited or obtained from the caller. While these logtags do extend past the point of actual decision making for an utterance in the application, all information in them can be used to change recognizer behavior and ultimately improve caller's experience.

2. **Second level features** still do not look into the future. However, they assume intimate understanding of the application as they have been compiled by a human expert who analyzed information in logtags and suggested how to extract helpful bits from it. One example is a binary feature that is set to 1.0 if and only if the utterance constitutes user's second attempt to provide locality name after the first one failed (something one can derive from other logtags). Another example is a feature that indicates that the recognized locality had a name of a U.S. state in it. This kind of information is certainly already contained in the first-level features implicitly, but is hard to derive from them using traditional data mining methods and without understanding the mechanisms behind the application.

3. **Third level features** add the possibility of looking into the future, but only future human-machine interactions. Namely, if the caller was later redirected to a human operator, no information in the logtags describing the subsequent human-human interaction is accessible for confidence re-calibration. This is a reasonable restriction to make, considering how few speech applications involve a human back-up.

4. **Forth level features** additionally extract information from the caller's interactions with a human operator in cases where an operator did get involved. The logtags from these interactions are available to us because, once the operator has decided to release a phone number, the control is passed back to the system in order to relay it to the caller. This allows us to define classification features such as whether the recognized locality was the same as the locality of the phone number released by the operator.

In addition to the features based on logtag values, we also compute time intervals between the utterance in question and those logtags (and also other input states). These features are then grouped into the four levels as well.

[1] From now on, we will be using the term *call context* instead of *dialog context* as it reflects the DA domain more adequately.

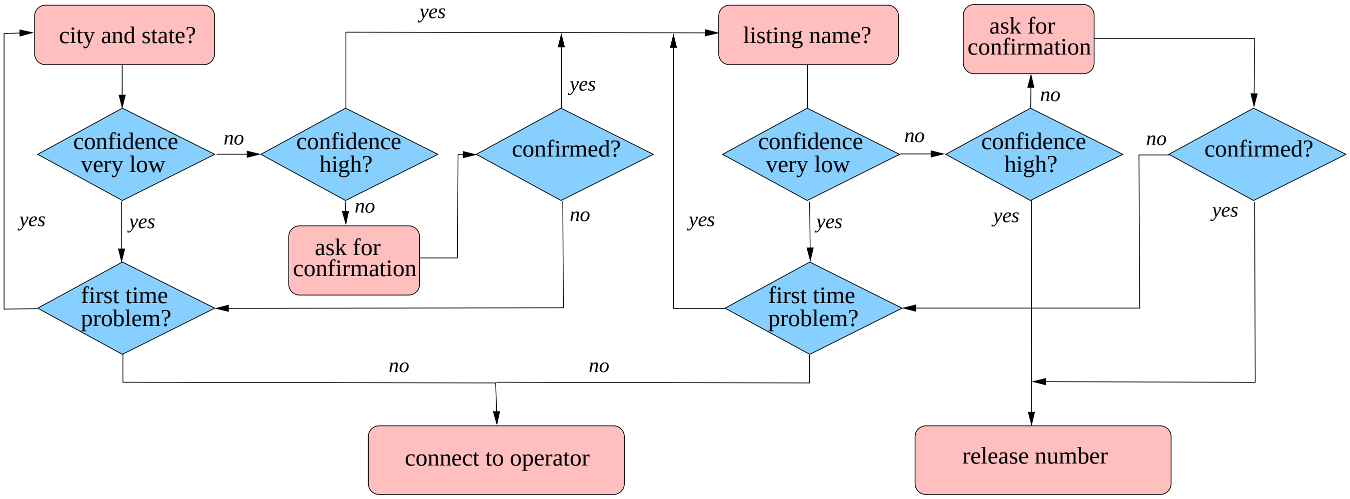

Fig. 1. Call flow of Directory Assistance application.

4. CONFIDENCE RE-CALIBRATION EXPERIMENTS

4.1. Data Sets and Evaluation Metrics

We collected and transcribed two corpora for *locality* and *listing* states of the application. The locality corpus contains 3882 utterances with all cases recognized as special keywords (such as requests for an operator that most of the time were reliably recognized) filtered out. Three quarters of these utterances are first requests and the rest are repeats. The listing corpus contains 4375 utterances selected in the same manner.

Our production recognizer is evaluated in terms of a *semantic error rate*: each recognition hypothesis is associated with corresponding semantics that are used by the dialog manager to continue the dialog; for instance, in the request: *"Hi, it's in eh... Mountain View I think"* the semantics is a codeword *mountain_view@ca*, and in the non-informational *"well... wait a second."*, semantics are empty. *Correct Accept rate (CA)* is the proportion of recognition hypotheses with non-empty correctly guessed semantics in the entire corpus, and *False Accept rate (FA)* is the proportion of recognition hypotheses with non-empty but incorrectly guessed semantics. We ignore the hypotheses with empty semantics, since confidence re-calibration would add little practical value there. Each hypothesis is also associated with a confidence score (the one we are about to re-calibrate). By changing the confidence threshold that this score has to exceed in order for the hypothesis to be accepted in the first place, we get a curve of different tradeoffs between these two metrics. The goal of re-calibration is to make this curve more concave (extending towards the upper-left corner; see below).

While correctly recognizing semantics is crucial for successful dialog continuation, it is the word and sentence error rates that are of more importance when the goal is to select "good" utterances for model re-training and adaptation. The differences between semantic and sentence error rates are large for both locality and listing utterances. In the locality state, semantics are automatically associated with each recognition hypothesis by the decoder. However, not all of the recognized words participate in semantics determination. For listing recognition, the difference is even more striking, as a specialized search back-end is used to convert the recognized word string into business ID. Therefore, we also need to extend our effort to predict recognition problems on the sentence error rate metric.

4.2. Confidence Re-calibration for Locality Recognition

The first experiments have been conducted to re-calibrate confidences in the locality state of the application. The transcribed and labeled corpus was split into five equal subsets for 5-fold cross-validation while making sure that no two utterances from the same call end up in different folds. Then, a binary statistical classifier (SVM) was trained to predict correctness of recognized utterances from either sentence level or semantic point of view, and the scores were normalized to approximate class posteriors.

For various feature levels, Table 1 compares *"missed errors"*, fractions of those falsely recognized (FA) utterances that have not been identified as such by the classifier, while keeping the fraction of correctly recognized (CA) utterances that have been accepted by the classifier constant at about 80%. As a baseline, we use simple thresholding on recognizer confidence. This baseline also determined the selection of the constant above (to keep recall for both classes approximately equal in this setup). Table 2 shows a similar comparison for locality recognition using sentence error rate instead of semantic error rate. This time, the recall of correct recognition was fixed at about 82%.

feature level	reco	1	1,2	1,2,3	1,2,3,4
missed errors (%)	18.6	22.7	17.2	16.1	13.8

Table 1. Missed semantic errors in the locality input state.

feature level	reco	1	1,2	1,2,3	1,2,3,4
missed errors (%)	17.8	16.0	13.3	16.5	16.6

Table 2. Missed sentence errors in the locality input state.

The tables indicates that for the locality state, only features extracted from the dialog future can help spotting utterances with poorly recognized semantics (those include features like caller's disconfirmations or repeated requests). However, as far as the sentence error rate is concerned, even extracting the features that pertain only to past and present of an utterance in question, helps reduce the fraction of not discovered false recognitions relative to the recognition-only baseline by 7-22% relative.

Note that the the tables represent only a snapshot of the classification results for a particular confidence threshold, a single point on a dependency curve between CA and FA. These curves, however, are not always smooth and therefore an overall "worse" curve can produce a "better" result for a particular operating point. This explains why level 1 and 2 features seem to produce better results than all features in Table 2 and, to a certain extent, the very bad performance of level 1 features in Table 1[2]. Figure 2 offers a more complete view of the achieved improvements from the perspective of the tradeoff between correctly and falsely recognized utterances (CA/FA), subject to changing confidence threshold, where the confidence can either come from the recognizer, or as a multiplicative product of recognition confidences and normalized classifier scores. In our experiments we determined that multiplying the normalized classification score with the original recognition confidence produces the best results with respect to sentence error rate in both locality and listing states, while semantic error rate is best served by the normalized classifier score alone.

The plot (only shown for sentence error rates) demonstrates that the classifier-adjusted versions outperform the recognition-only setup in the range of FA below 10%. The significance of this result from the feedback loop perspective can be exemplified by a scenario in which we wish to use the captured utterances along with their automated transcriptions for model re-training and restrict the set to only 70% of the highest confidence recognitions. Computing proportions of utterances with transcription errors FA/(FA+CA) in this new training set, we would get $0.05/(0.65+0.05) \approx 7.1\%$ for the unaltered confidences and almost half this amount, $0.03/(0.67+0.03) \approx 4.3\%$, for the re-calibrated confidences.

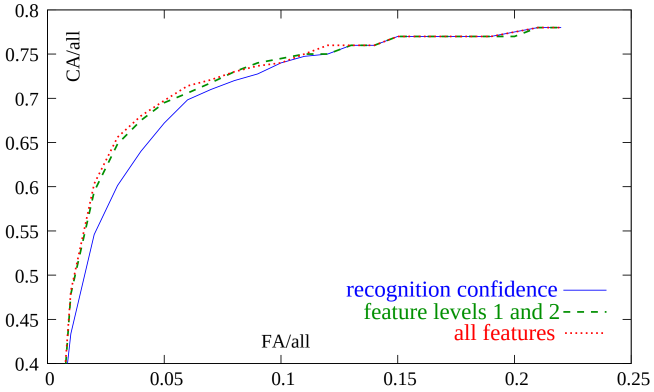

Fig. 2. CA/FA curves for the locality input state (sentence error rate) based on recognizer confidence measure (solid); recognizer confidences combined with classifier scores trained on features from levels 1 and 2 (dashed) and on all features (dotted).

feature level	reco	1	1,2	1,2,3	1,2,3,4
missed errors (%)	34.0	23.2	18.7	16.7	15.6

Table 3. Missed semantic errors in the listing input state.

4.3. Confidence Recalibration for Listing Recognition

For the listing state of the application, we carried out the cross-validation experiments as we did for the locality state, using semantic and sentence error rates to determine utterance dispositions.

We have mentioned in the beginning of this section that these two metrics are decoupled. In addition, it should be noted that the search back-end leads to a large number of rejections. Thus, even if the recognizer produced an output word string for some audio, we are still not guaranteed to have any semantics associated with this utterance, as the search engine can still dismiss the string. As a result, the number of utterances with valid semantics (and those are the ones we are interested in for the purpose of confidence re-calibration) is about 50% of the total number of utterances. However, missing semantics is not necessarily a problem, since callers often do not provide useful information in this input state. On the other hand, the search itself results in additional logtags that we can use as classification features to aid confidence re-calibration. Table 3 shows how missed semantic errors can be reduced for the listing state using a classifier (correct recognition recall fixed at about 67%).

This time a very significant improvement could be achieved even with the level 1 features. We propose two possible explanations for this. First, level 1 features for the listing state include information about what has previously happened in the locality state. If the confidence of the locality recognition was low, this increases the chance of a mistake in the listing state because a wrong grammar could be used there. A related

[2] The other reason being that with the parameters fixed for the entire series of the experiments, the classifier indeed failed to produce a good solution in this case; other parameter constellations resulted in much better results.

feature level	reco	1	1,2	1,2,3	1,2,3,4
missed errors (%)	28.3	31.4	22.1	22.3	22.7

Table 4. Missed sentence errors in the listing input state.

reason is that having problems recognizing locality can also mean a “difficult” caller for whom listing recognition is more likely to encounter problems as well. Second, search-related features are now part of level 1 set (because the logtags have been thrown before we got any new input from the caller); in other words, listing utterances have more prior context. Having not found any business could (though does not have to) mean a recognition problem.

Finally, Table 4 offers a comparison of missed errors for sentence error rate driven confidence re-calibration in the listing state. Overall, a relative reduction of about 20% relative was achieved.

Note that in general the task to detect utterances with recognition (sentence) errors is much more difficult than using semantic errors to guide the search. In fact, for cases where semantics has been guessed correctly in the presence of word recognition errors, the features from the call context can be detrimental. Indeed, whenever there is an explicit confirmation from the caller, call context features would suggest error free recognition. This is possibly the reason why features of up to the second level ended up predicting errors as well as all features together.

The plot in Figure 3 shows how combining normalized classifier scores with recognition confidences improves CA/FA tradeoff.

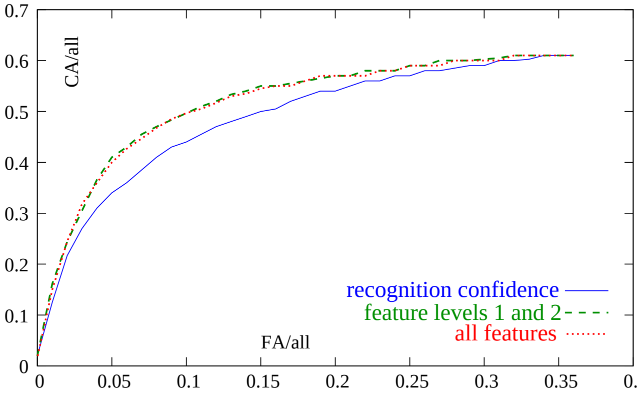

Fig. 3. CA/FA curves for the listing state (sentence error rate) based on recognizer confidence measure (solid); recognizer confidences combined with classifier scores trained on features from levels 1 and 2 (dashed) and on all features (dotted).

4.3.1. *Selecting Classification Features*

Different features contribute different amounts to classification success. One way to assess this contribution is to look at how much a feature’s presence affects the prior distribution of the classes. Mutual information is a common criterion to measure this change. For instance, for the listing confidence re-calibration task, consider the feature representing the logtag that registers recognition confidence of a preceding locality recognition. Figure 4 shows how different value ranges of this feature change priors of classes CA and FA. For example, for the confidence value range between 0.33 and 0.44, the point-wise mutual information with misrecognitions (FA) is strongly positive, and correspondingly, for the class of correct recognitions (CA) it is strongly negative. This means that this range of confidences induces a conditional posterior class distribution that differs from the priors in that a recognition error becomes much more likely. In general, we see that there is a correlation between correct recognitions in the listing state and recognition confidences in the preceding locality input state.

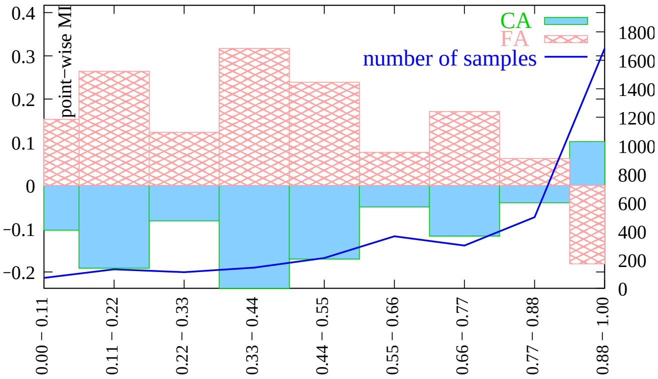

Fig. 4. Point-wise mutual information for listing recognition success/error on one side and confidence ranges of previous locality recognition on the other.

To make the re-calibration algorithm practical and robust, we want to extract a small, informative subset of the large original set of classification features. One way of doing this is to employ a greedy algorithm that sorts all features according to their mutual information with respect to binary dichotomy CA versus FA, and then keeps only n highest ranked candidates. Assuming that c is class and v represents the feature value/interval, mutual information is then computed as:

$$I = \sum_{c,v} P(c,v) \log_2 \frac{P(c,v)}{P(c)P(v)}.$$

In Figure 5 we plotted a dependency of the false recognition rate (“FA” in the CA/FA curve with CA fixed at 50%) on the number of classification features for confidence re-calibration. The analysis has been conducted for the listing input state and sentence error rate as the guiding metric for re-calibration.

While we do not claim our feature ranking and selection to be optimal, we did observe that with about 100 of the most salient features, the false recognition rate was reduced by 6% relative to the setup that uses all classification features (which amounts to over 36% in total relative to the recognition-only baseline). Since we did not use a separate validation set to select features, our only conclusion from this experiment is that the number of features can be dramatically reduced without negatively affecting performance. A similar effect has also been observed for the locality state, where the saturation could be achieved with 75 most salient features.

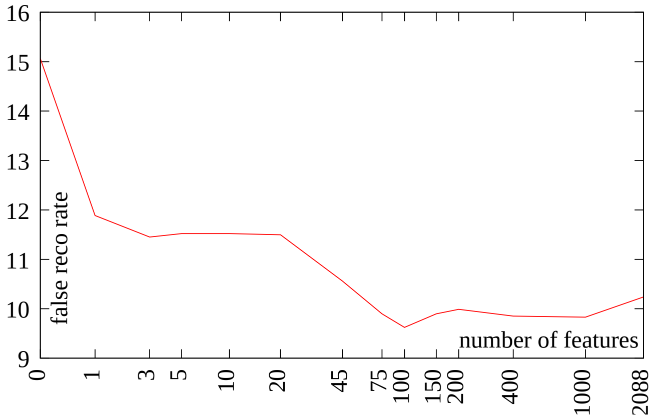

Fig. 5. Effect of feature selection on proportion of falsely recognized listing requests.

5. CONCLUSIONS AND FUTURE WORK

We have shown how information in the call context can be used to adjust recognition confidences of individual utterances. Using classification features ranging from application-agnostic to describing parts of the calls served by human operators, we were able to improve recognition confidences with respect to semantic and sentence error rates for locality and listing input states of a typical Directory Assistance speech application. The semantic error rate of listing utterances saw the largest reduction at 54% relative, but significant gains have also been observed for other combinations of error rate metrics and locality/listing utterances. A greedy feature selection based on mutual information of individual feature candidates yielded additional improvements. Throughout the experiments presented in this paper, we have assumed a fixed set of logtags to extract classification features from. In the future, we plan to address the complementary task of designing a set of logtags to optimize confidence re-calibration performance for this and similar applications, with and without involvement of human operators. Finally, while confidences re-calibrated using classification features from dialog's past will be used in the production version of the application to improve immediate user experience, we also plan to capitalize on the improved CA/FA tradeoffs while assembling a semi-supervised set of utterances for model re-training.

6. ACKNOWLEDGMENTS

The author would like to thank Shawn Chang, Yan Huang, Sara Krehbiel, other members of Speech at Microsoft as well as anonymous reviewers for valuable feedback on this paper.

7. REFERENCES

[1] Hui Jiang, "Abstract confidence measures for speech recognition: A survey," *Speech Communication*, vol. 45, no. 4, pp. 455–470, April 2005.

[2] Timothy J. Hazen, Theresa Burianek, Joseph Polifroni, and Stephanie Seneff, "Recognition confidence scoring for use in speech understanding systems," *Computer Speech and Language*, pp. 49–67, 2000.

[3] Lin Chase, "Word and Acoustic Confidence Annotation for Large Vocabulary Speech Recognition," in *Eurospeech*, Rhodes, Greece, 1997, pp. 815–818.

[4] Thomas Schaaf and Thomas Kemp, "Confidence Measures for Spontaneous Speech Recognition," in *ICASSP*, 1997, pp. 875–878.

[5] Jason D. Williams and Suhrid Balakrishnan, "Estimating Probability of Correctness for ASR N-Best Lists.," in *SIGDIAL*, London, UK, 2009.

[6] Dong Yu, Shizhen Wang, Jinyu Li, and Li Deng, "Word Confidence Calibration Using a Maximum Entropy Model with Constraints on Confidence and Word Distributions," in *ICASSP*, Dallas, TX, 2010, pp. 4446–4449.

[7] Dong Yu and Li Deng, "Semantic Confidence Calibration for Spoken Dialog Applications," in *ICASSP*, 2010, pp. 4450–4453.

[8] Diane J. Litman, Marilyn A. Walker, and Michael S. Kearns, "Automatic Detection of Poor Speech Recognition at the Dialogue Level," in *37th Annual Meeting ACL*, 1999, pp. 309–316.

[9] Diane J. Litman and Shimei Pan, "Predicting and Adapting to Poor Speech Recognition in a Spoken Dialogue System," in *17th National Conference on Artificial Intelligence (AAAI2000)*, 2000, pp. 722–728.

[10] Shuangyu Chang, Susan Boyce, Katia Hayati, Issac Alphonso, and Bruce Buntschuh, "Modalities and Demographics in Voice Search: Learnings from Three Case Studies," in *ICASSP*, Las Vegas, NV, 2008, pp. 5252–5255.

[11] W3C, "Voice Extensible Markup Language (VoiceXML) 2.1," June 2007, url: http://www.w3.org/TR/voicexml21/.

[12] Julian Odell and Kunal Mukerjee, "Architecture, User Interface, and Enabling Technology in Windows Vistas Speech Systems," vol. 56, no. 9, 2007.