Online SLU model adaptation with a partial Oracle

Pierre Gotab¹, Geraldine Damnati², Frederic Bechet³, Lionel Delphin-Poulal²

¹Universite d’Avignon ²France Telecom, Orange Labs ³Aix Marseille Universite

pierre.gotab@gmail.com, geraldine.damnati@orange-ftgroup.com frederic.bechet@lif.univ-mrs.fr, lionel.delphinpoulal@orange-ftgroup.com

Abstract

Deployed Spoken Dialog Systems (SDS) evolve quickly while new services are added or dropped, and while users’ behaviour change. This dynamic aspect of SDS justifies the need for a process allowing the system to keep up to date the Automatic Speech Recognition (ASR) and the Spoken Language Understanding (SLU) models. This process usually consists in collecting new data from the deployed system, transcribing and annotating them, then adding these new examples to the ASR and SLU training corpora in order to retrain the models. This strategy, even when used with an active learning scheme, is costly as the transcription and annotation processes of the new collected samples has to be done manually. This paper proposes a supervised approach for updating the SLU models of a deployed SDS which doesn’t need any additional manual transcription or annotation processes. The limited supervision needed for this alternative approach is given by the users calling the SDS: each user can be seen as a partial Oracle who could confirm if a system prediction is right or wrong.

Index Terms: Spoken Language Understanding

1. Introduction

Since the deployment of the AT&T "How May I Help You?" [3] service in 2001, it has been shown that Spoken Dialog Systems (SDS) can be deployed with success on a very large scale, for a very large range of customers, if the complexity of the task is kept relatively low (call routing, simple transactions). Despite this obvious success, the main limitation of the roll out of this type of technology is firstly the cost of building the Automatic Speech Recognition and Spoken Language Understanding models and secondly the cost of maintaining them: SDS evolve quickly while new services are added or dropped, and while users’ behaviour change. This dynamic aspect of SDS justifies the need for a process keeping up to date the ASR and SLU models in order to take into account this variability.

For SDS based on a statistical approach, this process usually consists in collecting new data from the deployed system, transcribing and annotating them, then adding these new examples to the ASR and SLU training corpora in order to retrain the models. This strategy, even when used with an active learning scheme, is costly as the transcription and annotation processes of the new collected samples has to be done manually. Because of this cost the models can’t be adapted on a daily bases and the SDS remain unchanged between two revisions.

In the framework of statistical SDS, this paper proposes an alternative approach to this updating strategy that does not need any manual transcription or semantic annotation process. The main idea is to use instead the answers of the callers of a deployed SDS: each user can be seen as a "partial Oracle" that could confirm if a system prediction is right or wrong. This partial Oracle is used into a novel SLU and ASR model adaptation strategy. This paper illustrates on a real deployed SDS the efficiency of the method proposed for the online adaptation of the models and compare it to the standard updating approach based on a "full Oracle".

2. Related work

As mentioned in the introduction, reducing the need for manual transcription and annotation data is the key for the rapid deployment of statistical SDS on a large scale. From a machine learning point of view, this can be seen as studying weakly or unsupervised methods for training and updating ASR and SLU models. Most of the weakly supervised methods applied to SLU are based on the active learning paradigm. Active learning has been used for training different kinds of models involved in a SDS, from acoustic models [6] and language models [7] to SLU models [9]. In all these previous works two issues are addressed: reducing the manual data annotation effort without impacting the performance of the models; finding the optimal set of data to annotate from a large set of raw data. Deployed SDS can be seen as unlimited sources of dialogue corpora, and therefore active learning methods are well suited for adapting ASR and SLU models to this context. In this situation the baseline models are those available at the launch of the deployed service and the unlabelled data is made of all the traces collected while the service is running. Carefully selecting the new examples to process manually in order to maximize the improvement in the models performance, while keeping the annotating cost as low as possible is the key of an effective maintenance strategy based on active learning [8, 4]. However, regardless of the method used, the active learning paradigm always needs an Oracle to process the portion of the raw data selected.

In order to prevent this need for a manual supervision (the Oracle), the two most popular unsupervised methods that have been proposed are the self-training and co-training algorithms. In the self-training algorithm, a first bootstrap classification model is trained and applied to a set of unlabelled examples. All the hypotheses produced by the classifier with a high confidence score are then added to a new set of automatically labelled examples. When enough data is processed, this new set is used to retrain the original model.

For the co-training algorithm [1], two classifiers are con-
This is done by using the customers of a deployed service as a training corpus from the last data set, we have 4 different corpora: three data annotation campaigns have been achieved for the exact transcription of a spoken message nor its semantic label; have not proved to be useful for updating an already well trained model.

However, in the context of improving an already deployed SDS, the ASR and SLU models to improve are already reasonably good, otherwise it would be impossible to have customers using it. That is why in this paper we use weakly supervised methods, and not unsupervised methods, but the originality of this work is to suppress the need for the manual supervision. This is done by using the customers of a deployed service as a partial Oracle, compared to the full Oracle of the standard manual supervision approach. This partial Oracle does not provide the exact transcription of a spoken message nor its semantic label but rather answers yes or no depending if the semantic label output by the SLU is correct or not. For example, in a call-routing SDS, this would correspond to ask the caller for confirmation after producing a call-type on the caller’s query.

We present in this paper a novel online adaptation strategy, both for ASR and SLU models, based on such a Partial Oracle. The experiments are presented on a call-routing SDS deployed by France Telecom.

3. Partial Oracle Algorithm

For describing the partial Oracle algorithm proposed in this study we assume that we have a corpus of utterances collected through an existing SDS. The SLU module of this SDS consists in predicting a label for each utterance thanks to a set of n binary classifiers \( \Theta_i \) with \( 1 \leq i \leq N \). Each classifier \( \Theta_i \) processing an utterance \( e \) outputs a score \( \Theta_i(e) = s_i \) representing its confidence in the prediction of \( l \) to \( e \). The final predicted class \( \Theta(e) = p \) is the one which obtained the best score among all the \( n \) classifiers: \( s_p = \max \{ \Theta_i(e) \} \).

- Let \( B \) be a bootstrap corpus of fully annotated and transcribed utterances. An Automatic Speech Recognition language model \( ASR_B \) is trained on this corpus.
- Let \( C = C_1, C_2, \ldots, C_4 \) be a set of \( k \) corpora containing each \( \gamma \) speech files. The \( k \times \gamma \) speech files are automatically transcribed using \( ASR_B \).
- Let \( \Phi(e) = c \) be a full Oracle returning the correct semantic label to an utterance \( e \).
- Let \( \Psi(e, l) = 0 \) with \( b \in \{ \text{right}, \text{wrong} \} \) be a partial Oracle returning a boolean answer \( b \) defined as:

\[
\Psi(e, l) = \begin{cases} \text{right} & \text{if } l = \Phi(e) \\ \text{wrong} & \text{otherwise} \end{cases}
\]

At the initialization step of this algorithm, the \( n \) classifiers \( \Theta_i \) are trained on the bootstrap corpus \( B \) as follows:

- each classifier \( \Theta_i \) is trained on a set of positive examples \( S_p^i \) and a set of negative examples \( S_n^i \);
- each example \( x = (e, t, l) \) is an \( n \)-uplet containing the utterance \( e \), the transcription of \( e \) (manual \( t_m \) or automatic \( t_a \)) and a semantic label \( l \) associated to \( e \);
- for an utterance \( e \) with \( \Phi(e) = l \), the example \( x = (e, t, l) \) is added to \( S_p^i \) and the examples \( x = (e, t, i) \) with \( i \neq l \) are added to \( S_n^i \); for the sake of simplicity we will note \( l \) all the labels different from \( l \);
- the features used by the classifiers are the words of the transcriptions \( l \) of all the examples, positive or negative.

The SLU update algorithm is the following:

- For each \( C_i \):
  - Prediction of the label \( \Theta(e) = p \)
  - If \( \Psi(e, p) = \text{right} \):
    - add \((e, t_a, p)\) to \( S_p^i \) and \((e, t_a, \neg p)\) to \( S_n^i \)
  - else if \( \Psi(e, p) = \text{wrong} \):
    - add \((e, t_a, p)\) to \( S_p^i \)
    - Retrain each classifier \( \Theta_i \) on \( S_p^i \) and \( S_n^i \)

The automatic transcriptions \( t_a \) are obtained with the language model \( ASR_B \), therefore the only supervision needed in this algorithm is the partial Oracle \( \Psi(e, p) \), no other manual process is required. There are two methods of obtaining this partial Oracle from a deployed SDS: 1. by explicitly asking the users to confirm each hypothesis produced by the system; 2. by analyzing the logs of the dialogues between the users and the system in order to estimate if the dialog was a success or a failure.

In the experiments reported in this paper we have simulated these methods by considering that we had the confirmation answers of the users for all the utterances of a corpus collected through a deployed call-routing SDS for which we had the true labels manually annotated.

4. Experiments

4.1. Experimental framework

1013 is the short number for France Telecom residential lines aftersales service. Initially deployed as a DTMF call-routing service, it has been progressively improved into an advanced natural language Interactive Voice Response (IVR) service since November 2007.

Among other functionalities, the system automatically detects calls that are outside the scope of the service and redirect customers to the correct service. It also detects when customers report a problem on their telephone line and for some particular motives, it performs automatic line diagnosis before routing to a customer representative.

The SLU model is composed of 66 target predicate/argument interpretations. A garbage class is also modelled for utterances that are not associated to any valid interpretation. Distribution of interpretations is not uniform. Interpretations related to the “telephone line” predicate represent 63% of the training corpus.

The deployed application relies on the Disserto suite and SLU is performed with a rule-based semantic analyser following a procedure described in [2]. In this paper however, we use instead statistical SLU models and the hand-crafted rules are not exploited.

Three data annotation campaigns have been achieved for three different versions of the system. After extracting a test corpus from the last data set, we have 4 different corpora:
On the overall, utterances are rather long as they are composed of 8.7 words on average. This is rather characteristic of call-routing applications and constitutes a significant difference between this study and previous studies on French dialogue applications [4].

As data are collected in real conditions, we test our methods on the whole test corpus including ill-formed utterances. 17% of the test utterances correspond to the garbage class, whether because of false voice activity detection (6.2%), out-of-domain speech (3.7%) or uncovered utterances (7.1%).

The speech recognition engine for this study is Nuance Recognizer 9. The bigram Statistical Language Model is trained on the bootstrap corpus for a lexicon of 4,1k words. The acoustic model is the default French acoustic model and all compilation and runtime parameters are set to default values. Finally, only the lexical and language models are specifically tuned for the application. The Word Error Rate on the test corpus is 47.7%.

### Table 1: Corpora collection

<table>
<thead>
<tr>
<th>corpus</th>
<th># utt.</th>
<th># words</th>
<th>period</th>
</tr>
</thead>
<tbody>
<tr>
<td>bootstrap</td>
<td>3200</td>
<td>25.3k</td>
<td>March → July 2008</td>
</tr>
<tr>
<td>corpus1</td>
<td>2700</td>
<td>24k</td>
<td>October 2008 → April 2009</td>
</tr>
<tr>
<td>corpus2</td>
<td>13500</td>
<td>130.3k</td>
<td>May 2009</td>
</tr>
<tr>
<td>test</td>
<td>4860</td>
<td>42.2k</td>
<td>May 2009</td>
</tr>
</tbody>
</table>

#### 4.2. SLU adaptation with the Partial Oracle algorithm

In the following experiments, the bootstrap $B$ contains 3154 examples. The pool of candidate utterances for training is the concatenation of corpus1 and corpus2. This pool is separated into 12 successive partitions $C_i$, of around 1300 examples each, simulating the successive data collections mentioned in section 3. We have 66 semantic labels $l \in \{1, 2, ..., 66\}$, and thus 66 binary classifiers $\Theta_l$ specialized on each label.

In the first set of experiments we evaluate the partial Oracle algorithm on the Test corpus. For this first set of experiments the ASR model remains unchanged and is the ASR$_B$ model trained on the bootstrap corpus; only the classifiers are iteratively adapted with new training examples. When an example $(e, t, l)$ is added to the training corpus of a classifier, the transcription $t$ is the one obtained automatically by applying ASR$_B$.

Three strategies are compared:

- The full Oracle experiment consists in replacing in our algorithm the predicted label $\Theta_l(e)$ by the correct label $\Phi(e)$ thanks to a full Oracle. This ‘cheating’ experiment represent the upper bound that can be reached by our algorithm.

- The partial Oracle only YES experiment consists in adding new examples to the training corpus of the $\Theta_l$ classifiers only if the user answers yes to the confirmation prompt (therefore, $\Psi(e, \Theta_l(e)) = \text{right}$). If the partial Oracle returns wrong, the example is simply dropped. This experiment consists in simulating a self-training experiment: only the new examples correctly classified are added to the training corpus.

- The partial Oracle YES+NO experiment implements fully the algorithm presented in the previous section. The originality of this method is to use as well the wrongly interrelated utterances as negative examples in the training process.

The results are given in figure 1 with the Interpretation Error Rate (IER) measure. The IER is defined as the summation of insertions (giving an interpretation instead of Reject), deletions (the opposite) and substitutions over the total amount of valid interpretations. IER is higher than the classical Sentence Error Rate which is relative to the total amount of utterances regardless of their semantic validity, but is considered to reflect more accurately the performances as perceived by users.

The YES + NO method clearly outperforms the only YES approach. Only incorporating in the training data utterances for which the bootstrap model was already good (only YES curve) doesn’t yield much improvement, confirming the small gain that can be achieved by self-training algorithms. When negative examples are used, the IER is significantly reduced. Nearly half of the maximum gain that could be achieved with a full Oracle is reached with our algorithm, with no additional human annotation process.

#### 4.3. ASR adaptation with the Partial Oracle algorithm

The second set of experiments aims at showing the impact of introducing new examples also at the Language Model level. Here again no manual data transcriptions are taken into account, only the word sequences output by ASR$_B$ model are used for the data to be incorporated into the training corpora. The SLU models remain unchanged with respect to the previous figure. On the other hand, the test corpus is decoded with different ASR models.

- no ASR adaptation : the bootstrap ASR$_B$ model is used on Test data (Word Error Rate = 47.7%).
- with ASR adaptation (all messages) : all the word sequences automatically obtained on the pool data are added to the train the Language Model (Word Error Rate = 47.7%).
- with ASR adaptation (only messages validated by partial Oracle) : only the hypotheses for which the predicted class has been validated by the partial Oracle are incorporated into the training data of the Language model (Word Error Rate = 47.3%).

The word error rate is rather unchanged by this addition of potentially noisy transcriptions to the language model training corpus. However, the impact of these different transcriptions on the SLU performance is bigger as we can see in figures 2 and 3. Three curves on the figure 2 are plotted for the only YES, and on the figure 3 for the YES+NO partial Oracle experiments. As we can see, by adding only the automatic transcriptions of the utterances considered as correct by the partial Oracle, we obtain a significant decrease of IER, especially for the YES+NO strategy.

#### 5. Conclusion

This paper proposes a lightly supervised approach for updating the SLU models of a deployed SDS which doesn’t need any additional manual transcription or annotation processes. The limited supervision needed for this alternative approach is given by the users calling the SDS: each user can be seen as a partial Oracle who could confirm if a system prediction is right or wrong. We have demonstrated the efficiency of this simple method on a real deployed SDS for the online adaptation of the SLU and ASR models.
6. Acknowledgment

This work is partially supported by the Conseil Régional Provence-Alpes-Côte-d’Azur and by the 6th Framework Research Program of the European Union (EU), Project LUNA, IST contract no 33549. (http://www.ist-luna.eu)

7. References


