ROBUST SEMANTIC CONFIDENCE SCORING

Didier Guillevic, Simona Gandrabur, Yves Normandin

InfoSpace Speech Solutions,
460 St-Catherine W., Suite 801, Montreal (QC) H3B 1A7 Canada
{dguillev,sgandra}@infospace.com, ynormandin@sympatico.ca

ABSTRACT

This paper describes an approach for defining robust, application-independent confidence measures for dialogue systems. A concept-level confidence score is computed using a Multi-Layer Perceptron (MLP) classifier trained to discriminate between correct and incorrect concepts. Three types of concept-level confidence features are considered: features based on the confidence score of the underlying words, parsing specific features, and novel semantic features (weighted semantic purity and time consistency) that are indicators of the coherence among various semantic recognition hypotheses. Confidence scores at the semantic hypothesis and utterance levels are derived from the confidence scores of the corresponding concepts. We report our results on a database of $40,000$ utterances from various application contexts. By using features based only on word scores for concept classication we obtained a 46% correct rejection (CR) rate at a 95% correct acceptance (CA) rate. Adding semantic measures to the classifier boosted the CR rate to 71%, which corresponds to a 46.3% relative improvement.

1. INTRODUCTION

Confidence measures can be applied at various levels of ASR results. Most commonly, confidence is measured at the word level [2, 3, 5, 6, 7] and at the utterance level [1, 2, 3, 6]. Typically, the confidence is measured one utterance at a time but more recently attempts have been made to derive confidence measures over multiple dialogue turns [1]. This paper presents an approach to define robust, application independent confidence measures for automatic speech recognition (ASR) within the InfoSpace SoftDialogue™ SDK. SoftDialogue™ is a speech-application development environment that integrates grammar-based ASR, natural language processing and confidence measures at various levels.

In speech understanding systems, recognition confidence must be reflected at the semantic level. This can be achieved by computing word and utterance confidence scores and feeding them to the natural language processing unit in order to influence its parsing strategy [2]. Alternatively, a confidence score can be computed directly at the concept level [6]. We adopted the latter approach for two reasons. First, semantic confidence measures allow the application to apply various confirmation strategies directly at the concept level. Second, word scores alone do not suffice to derive concept-level confidence scores: parsing specific and semantic coherence features (weighted semantic purity and time consistency) proved to be very efficient for accurate semantic confidence scoring.

In our current experiment, we used data coming from four different kinds of applications; namely corporate directory, yes/no, money, and date. Sample utterances from these applications are shown below:

```
i'd like to speak to nicole decarie please
Jacqueline Chang
yes please
six hundred sixteen dollars and fifty-one cents
august nineteenth nineteen ninety three
```

We split our database of $40,000$ utterances into three sets: $49/64$ of the data for training, $7/64$ for validation and the remaining $8/64$ makes up the testing set.

2. POST-PROCESSING OF ASR RESULTS

The ASR result is a set of N-best recognition hypotheses. Each recognition hypothesis is a sequence of words with associated acoustic scores. A *sentence score* is computed for each recognition hypothesis. The sentence score results from the combination of the acoustic scores and the language model probabilities of the word sequence.

The N-best recognition hypotheses are post-processed in order to extract their meaning and to measure the confidence in the recognition accuracy. First, each recognition hypothesis is semantically interpreted according to a parse grammar. The result of the semantic interpretation is a semantic hypothesis which consists of a sequence of *concepts* or *semantic slots*. Then, a confidence score is computed for each hypothesized word by a MLP trained on acoustic decoder-based features. The word confidence scores along with parsing specific information and semantic coherence features are fed to a MLP which computes a confidence

score for each semantic slot of each semantic hypothesis. Confidence scores for each semantic hypothesis and for the utterance are derived from the concept confidence scores. The resulting set of semantic hypotheses is re-ordered according to the hypothesis confidence scores. Finally, we perform a semantic collapsing of the set of semantic hypotheses in order to eliminate redundancies. These post-processing steps are illustrated in the following example:

```
ASR Hypotheses:
  0; -5347.312; McPhearson
  1; -5624.917; Paul McPhearson
  2; -5772.213; Mike Larson
Semantic hypotheses before sorting and collapsing:
  ASR hypIndex  slot = (userId:alias)  conf. score
      0         11572:"McPhearson"         62
      1         11572:"Paul McPhearson"     2
      1         11572:"McPhearson"         25
      2         11062:"Mike Larson"        12
      2         10876:"Larson"              6
Semantic hypotheses after sorting and collapsing:
  ASR hypIndex  slot = (userId:alias)  conf. score
      0         11572:"McPhearson"         62
      2         11062:"Mike Larson"        12
```

3. WORD SCORE

3.1. Features

Traditionally, our ASR engine has been returning a *frame weighted* acoustic score for each word. This score is computed by averaging the acoustic score of each frame aligned with the word, normalized by an on-line garbage model. Here we investigate the use of other measures extracted from the acoustic alignment to produce hopefully a better word score, in terms of discrimination power. In addition to the *frame weighted score*, the ASR engine now returns three more measures for each word. In the *state weighted score*, each Hidden Markov Model state contributes equally as opposed to each frame contributing equally in the traditional score. The *frame weighted raw score* is the average frame acoustic score non-normalized by the on-line garbage model. Finally, we consider the *length* in frames of each word.

3.2. Tag

A boolean tag is associated with each hypothesized word. It is true if the hypothesized word can be found in roughly the same position in the utterance's reference string. This step is performed by aligning the hypothesis and the reference word sequences using a dynamic programming algorithm with equal cost for insertions, deletions, and substitutions. As a result, each hypothesized word is tagged as either: *insertion*, *substitution*, or *ok*, with the *ok* words tagged as *correct*, while the others are tagged as *incorrect*. Note that the comparison between two words is based on the pronunciation, not the orthography. If two words share a common pronunciation, they will be considered equivalent.

3.3. Classifier

Given the four features mentioned above, plus the correct tag, we trained a MLP network with two nodes in the hidden layer. Prior to being fed to the network, the features are scaled appropriately. Currently, this step consists of removing the mean and projecting the feature vectors on the eigenvectors computed on the pooled-within-class covariance matrix. The pooled matrix is the one used in Fisher's Linear Discriminant Analysis.

3.4. Results

Acoustic Word Score Classifier		
Feature	NCE	CA $= 95\%$
Original word score	0.25	CR $= 34\%$
New word score	0.32	CR $= 47\%$

At a correct acceptance (CA) rate of 95%, we were able to boost our correct rejection (CR) rate from 34% to 47%. This represents a 38% relative increase in our ability to reject incorrect words. We also measured the relative performance of our new word score using the Normalized Cross Entropy (NCE) introduced by NIST. We were able to boost the entropy from 0.25 to 0.32. Note that by simply using the class priors, we would obtain a NCE of 0.0. On the other hand, a "perfect" classifier would result in a NCE of 1.0. A perfect classifier is one system that would have a confidence measure output of 1.0 when the hypothesized word is correct, and 0.0 when the word is incorrect.

4. SEMANTIC SLOT CONFIDENCE SCORING

A semantic slot is defined as the smallest semantic value in our system. In a corporate directory application, a given hypothesis is made of a single slot *userId* corresponding to a unique user identification in a phone directory. When recognizing dates, a semantic hypothesis can have up to 5 slots named *dayName, dayNumber, monthName, monthNumber* and *year*. When recognizing dollar amounts, we can have up to 3 slots; namely *dollar, currency* and *cents*.

4.1. ASR Hypothesis A Posteriori Probability

The sentence score of a recognition hypothesis taken in isolation does not have much significance since it depends on the number of recognized frames. It is more appropriate for further processing to convert sentence scores relative to each other into *a posteriori* probabilities. Mangu [4] uses a similar scheme to compute word posteriors. Here we will use hypothesis posteriors in the computation of some of the semantic slot features. Consider the following example of three recognition hypotheses with their associated sentence scores S_i:

```
0; -5610.114; September twenty-fifth
1; -5647.917; September twenty-sixth
2; -5650.505; September twenty-six
```

The computation of the posteriors is illustrated in the table below, where the constant C has been set experimentally to 0.075 in our applications, $\Delta_i = S_i - S_0$,

$$p_i = \exp\left[C\Delta_i\right], \quad \text{and} \quad \hat{p_i} = p_i / \sum_{j=0}^{N} p_j.$$

S_i	Δ_i	$C\Delta_i$	p_i	$\hat{p_i}$
-5610.114	0.0	0.0	1.0	0.9033
-5647.917	-37.803	-2.835	0.0587	0.0530
-5650.505	-40.391	-3.029	0.0483	0.0436

4.2. Features

For each semantic slot, we consider six measures, namely: word score average and standard deviation, semantic purity, time consistency, as well as the fraction of unparsed words and frames.

Word Score average and standard deviation. The word score average and standard deviation are computed over all words that make up a given slot. E.g.:

```
five thousand eight hundred and thirteen cents
```

The slot *dollar* is made of the four words `five thousand eight hundred` and *cents* is made of the single word `thirteen`. We use the word scores described in Section 3.

Semantic Purity. This is a measure of how well a given semantic slot is represented in the N-best ASR hypotheses. It takes into account all the hypotheses weighted by their *a posteriori* probabilities. The measure does not take into account the position of the slot within the hypothesis. E.g. consider the three hypotheses shown below with respective *a posteriori* probabilities $(0.72, 0.18, 0.10)$ where A, B, C and D represents semantic slot values:

```
0; 0.72; A B C
1; 0.18; B D
2; 0.10; D C
```

The purity of A is 0.72, B's is 0.90 $(0.72+0.18)$, C's is 0.82 and finally D's purity is 0.28.

Time Consistency. The ASR gives the start and end frames for each hypothesized word. We constructed a scheme that translate these frame indexes into a normalized time scale. The start and end times of a semantic slot are then respectively defined as the start time of the first word and the end time of the last word that make up the slot.

The measure of *time consistency* is similar to the semantic purity described above. For a given slot, we add the *a posteriori* probabilities of all hypotheses where the slot is present and where the start / end times match. We then normalize that value by the sum of the *a posteriori* probabilities of all hypotheses where the slot is present (irrespective of the time indexes). If a slot appears in one single hypothesis, it is guaranteed to have a time consistency of 1.0.

Fraction of unparsed words and frames. This measure indicates how many of the hypothesized words are left unused when generating a given slot value. The classifier will learn that a full parse is often correlated with a correct interpretation, and a partial parse with a false interpretation. Consider the example below:

```
----- ASR Hypothesis -----
0; -1023.178; Sheri Mitchell Buckley
----- Semantic Hypotheses -----
0; userId = 10542:"Sheri Mitchell"
2; userId = 10082:"Buckley"
```

In order to generate the slot *userId* $= 10082$, two words out of three are left unparsed; namely "Sheri" and "Mitchell". When parsing for *userId* $= 10542$, only one word out of three is left unparsed.

The measure of unparsed frames is similar to the measure of unparsed words, with the difference that the words are now weighted by their length in frames.

4.3. Tag - Classifier

We tag each semantic slot as either "correct" or "incorrect". A semantic slot is "correct" if it is also present in the list of slots associated with the reference.

We feed the six measures mentioned above into a Multi Layer Perceptron (MLP) network with five hidden neurons. Once again, the features are pre-processed before being fed to the network. We remove the mean and project the feature vector on the eigenvectors of the pooled-within-class covariance matrix.

4.4. Results

We measure the performance of the resulting semantic slot classifier by analyzing the rejection ability at various correct acceptance rates. We report in the table below the performance resulting from the various combinations of our features. The features are indexed as follows: (0) average of the original word acoustic score among the hypothesized words from a given slot, (1) word score average and (2) standard deviation, (3) semantic purity, (4) time consistency and finally (5)(6) fraction of unparsed words and frames.

Semantic Slot Classifier					
Features	NCE	CA $\rightarrow$	90%	95%	99%
(0)	0.32	CR $\rightarrow$	59%	36%	6%
(1,2)	0.35	CR $\rightarrow$	64%	46%	15%
(3,4)	0.48	CR $\rightarrow$	81%	66%	21%
(1,2,3,4)	0.55	CR $\rightarrow$	86%	70%	21%
(1,2,3,4,5,6)	0.56	CR $\rightarrow$	87%	71%	27%

We see that by feeding our six measures into a MLP we are able to significantly enhance the performance of a base system that uses only the average of the original word acoustic scores. The entropy jumps from 0.32 to 0.56 and the CR rate goes from 36% to 71% for a given CA rate of 95%.

As mentioned before, the ASR hypothesis *a posteriori* probabilities (Section 4.1) are used as weights in the computation of the *semantic purity* and *time consistency* measures. In order to measure the impact of these weights, we built classifiers based on the two above features, computed with and without the weights (i.e., assuming equal probabilities for all hypotheses).

Influence of the hypothesis a posterior			
Features	NCE	CA →	90%
(3,4) without hypAPostProb	0.39	CR →	68%
(3,4) with hypAPostProb	0.48	CR →	81%

As the results clearly demonstrate, the impact of the weights is quite significant. We get a 25% relative increase in the entropy and the correct rejection jumps from 68% to 81% when correctly accepting 90% of the utterances.

5. SEMANTIC HYPOTHESIS SCORING

Deciding whether a semantic hypothesis is "correct" or "incorrect" is a non-trivial task. We could decide that a hypothesis is "correct" if and only if all its semantic slots are tagged "correct". In the example below, the hypothesis would be tagged as "incorrect":

```
Reference : {DOLLAR=8500,CENT=30}
Hypothesis: {DOLLAR=8500,CENT=13}
```

Moreover, the hypothesis in the following example, too, would be tagged "correct", eventhough it is missing the CURRENCY slot:

```
Reference : {DOLLAR=48000,CURRENCY="CND",CENT=18}
Hypothesis: {DOLLAR=48000,CENT=18}
```

Alternatively, we could tag a hypothesis "correct" if and only if it is an exact match with one of the semantic interpretations associated with the reference. With this tagging scheme both of the above examples would be tagged as "incorrect". We adopted this latter approach in our experiments. With both these tagging schemes a hypothesis is "incorrect" as soon as any of its slots is "incorrect". Therefore a hypothesis can be no better than its weakest slot. Based on this reasoning, we have decided, somewhat arbitrarily, to set the score of a semantic hypothesis as the smallest score among its slots. Eventually, we should probably consider a scheme where each application could specify the relative importance of the semantic slots encountered.

In this setting, there is no training to be performed. The rejection ability versus the correct acceptance is given below at various operating points.

Semantic Hypothesis Scoring				
NCE	CA →	90%	95%	99%
0.56	CR →	87%	72%	32%

6. CONCLUSION

We defined robust confidence measures for spoken dialogue systems with an emphasis on semantic-level scores. In our experiments the semantic coherence features proved to be significantly more discriminative for concept classification than the other features. By adding these features we increased the CR rate from 46% to 71% at a 95% CA rate, a relative improvement of 46.3%. An important factor in the discrimination capacity of these features are the *a posteriori* probability weights that are applied to the raw coherence features. At a 90% CA rate, the relative improvement obtained by using weighted instead of un-weighted coherence features was 19%.

Future work will include experimentations with various robust decoder-based features for word score computations and exploration of more sophisticated semantic hypothesis and utterance-level confidence scores.

7. REFERENCES

[1] P. Carpenter, C. Jin, D. Wilson, R. Zhang, D. Bohus, and A. Rudnicky. Is this conversation on track? In *Eurospeech*, pages 2121–2124, 2001.

[2] T. Hazen, T. Burianek, J. Polifroni, and S. Seneff. Recognition confidence scoring for use in speech understanding systems. In *ASR*, pages 213–220, Paris, France, 2000.

[3] B. Maison and R. Gopinath. Robust confidence annotation and rejection for continuous speech recognition. In *ICASSP*, 2001.

[4] L. Mangu. *Finding consensus in speech recognition*. PhD thesis, Johns Hopkins University, April 2000.

[5] P. Moreno, B. Logan, and B. Raj. A boosting approach for confidence scoring. In *Eurospeech*, pages 2109–2112, 2001.

[6] R. San-Segundo, B. Pellom, K. Hacioglu, and W. Ward. Confidence measures for spoken dialogue systems. In *ICASSP*, 2001.

[7] R. Zhang and A. Rudnicky. Word level confidence annotation using combinations of features. In *Eurospeech*, pages 2105–2108, 2001.