WORD LEVEL CONFIDENCE MEASUREMENT USING SEMANTIC FEATURES

Ruhi Sarikaya, Yuqing Gao and Michael Picheny
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{sarikaya,yuqing,picheny}@us.ibm.com

ABSTRACT

This paper proposes two principled methods to incorporate semantic information into word level confidence measurement. The first technique uses tag and arc probabilities obtained from a statistical classer and parser tree. The second technique uses a maximum entropy based semantic structured language model to use semantic structure of a sentence to assign semantic probabilities to each word. Semantic features provide significant improvements over a posterior probability based confidence measure when used together in an air travel reservation task.

1. INTRODUCTION

Automatic speech recognition systems are far from perfect. There are a number of factors including environment, telephone line quality, and speaker variability that can impair speech recognition performance. Moreover, in some cases a speech understanding unit can generate an incorrect parse result and sends the dialog on a completely wrong path. This may lead to a failed dialog. In order circumvent these problems it is vital to employ a reliable confidence metric that can identify speech recognition errors. This information can be used to generate repair dialogs.

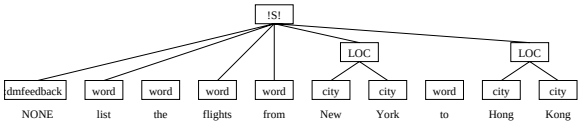
The majority of the approaches to confidence annotation methods use two basic steps: (1) generate as many features as possible based on speech recognition and/or natural language understanding process, (2) use a classifier to combine these features in a reasonable way. Therefore the two main issues for confidence measures are (1) what features are useful for confidence annotation and (2) how to combine these feature in a sensible way. There are a number of studies attempting to answer these questions. Typically, confidence measures depend on the type of the task. For domain independent large vocabulary speech recognition systems, posterior probability based on a word graph is shown to be the single most useful confidence feature [4]. For limited domains features from a speech understanding unit are also helpful. There are a number of cues for poor a speech recognition hypothesis. These cues can be observed from acoustic score, language model score, word counts in an N–best list, lattice density, phone perplexity, language model back-off behavior, and posterior probability [2, 8, 7, 11]. However, many of these features overlap considerably and they have been included in the recognition process directly or indirectly. As a result combination of a number of features from the same source may result in a marginal improvement over the best single feature.

In many of the previous studies the way the semantic information is incorporated into the decision process is rather *ad hoc* with exception of [9]. For example in [11], the semantic weights assigned to words are based on heuristics. Similarly, in [3] such semantic features as "uncovered word percentage", "gap number", "slot number", etc. are generated experimentally in an effort to incorporate semantic information into the confidence metric.

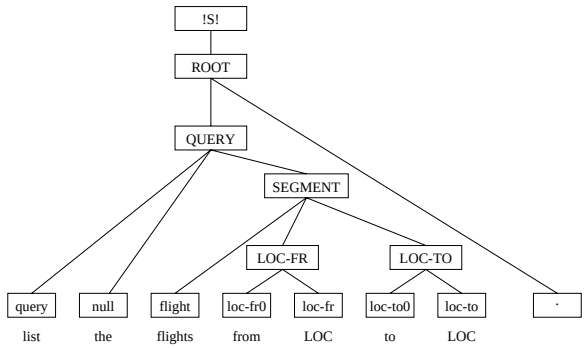
Confidence measurement can be applied either at the word level, phrase/concept level, utterance level or their combinations. In this study, we use the posterior probability as the one single feature obtained from the speech recognition unit and combine with the proposed semantic features in a probabilistic framework for each word. The rest of the paper is organized as follows. In Section 2, we briefly describe the semantic analysis employed in our work. We describe the maximum entropy based semantic structured language models in Section 3. Section 4 defines the semantic confidence features followed by the experimental results. Finally, Section 6 summarizes the findings and possible future research directions.

2. SEMANTIC ANALYSIS

Semantic analysis involves finding the semantic units that span words or word groups and the relationship among these units in a sentence. The semantic units are assigned certain tags and labels. Moreover, higher level relationships among semantic unit groups can also be determined. Our semantic analysis is based on statistical classing and parsing and is currently used in limited domain dialog systems. Domain independent statistical semantic classers and parsers are not feasible to develop due to the possibly unlimited number of concepts that may occur in a domain independent task. Like any other statistical system, our statistical parser and classer requires annotated training data. Basically, during annotation we impose the semantic relationships among the words and word groups in a hierarchical manner. The decision tree based statistical classer/parser uses the training data to assign probabilities to each node and arc in a parser tree. Once the decision tree is built, during testing our parser works in a left–to–right and bottom–up fashion. Each parser action is assigned a probability given the current context. A parser action can be in many different forms. For example, assigning a certain tag to a word or extending a tag to a parent label, or assigning a certain label to a set of tags etc. is considered a parser action. Classing can be considered as a shallow parsing. An example of the classer tree is shown in Fig. 1. As seen in the figure each word is assigned a tag and certain tags are grouped under a label to form a constituent. The classer output is used as input to parser. Therefore, parsing is a two step process. The function of the classer is to group together the words that are part of a concept. The parser takes the classer output and builds a hierarchical full semantic parse tree. The corresponding parse tree for the classer tree is given



(A) An example of a semantic classer output.



(B) The parser output for the same example.

Figure 1: Classer and Parser outputs for an example sentence.

in the same figure. Here, semantically related concepts are grouped at a higher level.

3. MAXIMUM ENTROPY BASED SEMANTIC STRUCTURED LANGUAGE MODELING

The Maximum entropy (ME) method presents a framework to combine multiple overlapping information sources in an effective way. ME has been widely used in statistical language modeling [12]. Maximum entropy modeling matches the feature expectations exactly while making as few assumptions as possible in the model. The multiple information sources are combined in the following way:

$$P(o|h) = \frac{e^{\sum_i \lambda_i f_i(o,h)}}{\sum_{o'} e^{\sum_i \lambda_i f_i(o',h)}}, \qquad (1)$$

where o is the current word, f_i are the feature indicators that are activated for a certain history, and h represents the history which may include previous words as well as tags and labels that can be used in predicting the current word. In [1], we used ME to model sentence based syntactic and higher level semantic information. Semantic information is obtained from the semantic classer and parse trees. We computed the joint probability of a word sequence and a parse tree: $P(W,C)$ [1]. The first step in building the maximum entropy model is to represent a classer/parse tree as a sequence of words, tags, and labels. The labels are divided as begin-label and end-label.

Basically, this representation (an example is given in Section 4.1 along with the token probabilities) is equivalent to enriching the original text which is composed of word sequences with the tags and labels. This representation allows us to define the boundaries for the semantic constituents and take the long range semantic information into account. Since the tags are already included in the classes used in language modeling, we ignored them in our analysis. In [1], we proposed a set of maximum entropy based structured language modeling (MELM) techniques. MELM2 is one the language models proposed there and employed 7 types of questions about the current token in a sentence (MELM1 corresponds to a regular n–gram). In addition to regular n–gram questions for trigram, four more questions are used regarding the semantic structure of the sentence. These questions are (1) current active parent (L_i), (2) L_i and number of words to the left since starting the current constituent (N_i), (3) L_i, N_i and previous word token, (4) the previous completed constituent (O_i) and number of words to the left since completing O_i. The history given in Eq. 1 above consists of answers to these questions.

Interpolating MELM2 with the class based trigram provided significant improvement over a sophisticated class-based language model [1]. This improvement is due to the inclusion of new semantic information that was not part of the original speech recognition system.

4. SEMANTIC CONFIDENCE FEATURES

Our semantic analysis is based on a statistical classer and parser. The issue that we want to address is what features we can obtain from the semantic analysis. We answer these questions in two ways resulting in two methods along with two feature sets to incorporate semantic information into confidence measurement.

4.1. Semantic Tags and Labels

The classer/parser performs a left–to–right bottom-up search to find the best parse tree for a given sentence. During search, each node and arc in the parse tree is assigned a probability. Node probability represent the probability of having that node there in the parse tree given previous words, tags and labels. Similarly, an arc probability represents the probability of placing that arc between the current node and its parent. The example below shows a classer tree in text format along with the node and arc probabilities. Note that each token is assigned a pair of probabilities. The first probability is for the node and the second probability is for the arc.

```
:NONE   list the flights from New York to Hong Kong
{0.4273 {!S!_1_1 :NONE_dmfeedback_1_0.9979
list_word_0.9996_0.9957 the_word_0.9882_0.9957
flights_word_0.9996_0.9957 from_word_0.9848_0.9957
{LOC_0.9999_0.9998 new_city_0.4901_0.9813
york_city_0.9998_0.9989 LOC_0.9999_0.9998}
to_word_0.9986_0.9957 {LOC_0.9999_0.9981
hong_city_0.9979_0.9881 kong_city_0.9601_0.9957
LOC_0.9999_0.9981} !S!_1_1} }
```

Any of the probabilities in this tree can potentially be used as a semantic feature. We considered classer tag (cTag), classer tag-arc (cTagArc), parser tag (pTag), and parser tag-arc (pTagArc) to combine with the posterior probability. In the example above, "0.4901" is a cTag, and "0.9813" is a cTagArc probability for the word "new". Similarly, the corresponding pTag and pTagArc probabilities are extracted from the parse tree.

4.2. MELM2 Features

The language model score for a given word in MELM2 model is conditioned not only on previous words but also tags, labels and relative coverage of these labels over words. MELM2 presents an effective statistical method to combine word sequences with the semantic parse tree. Therefore we can use the MELM2 score

as a feature for confidence measurement. However, MELM2 for a given word only depends on the previous word sequence and the parse tree up to that word. In [5], it is observed that on subset of the Switchboard development test data correctness on w_i has a significant effect on w_{i+1}. For example, w_{i+1} is correct 87% of the time when w_i is correct and only 48% of the time when w_i is incorrect. Even though it is a different data set, this observation suggests we can expect a low score for the current word if the previous word is recognized incorrectly. Besides the MELM2 score for the current word w_i, we considered a window of three words ($[w_{i-1}\ w_i\ w_{i+1}]$), MELM2-ctx3, and five words, MELM2-ctx5, to capture the context information.

5. EXPERIMENTAL RESULTS AND DISCUSSION

We have carried out experimental investigations of confidence measurement with the IBM DARPA communicator system. MELM2 is trained on 137K sentences in air travel domain. An additional 18K sentences are used for smoothing. The MELM2 model is trained using the improved iterative scaling algorithm using fuzzy smoothing [1, 12]. The confidence measurement training data is obtained by pooling eight other DARPA communicator sites' evaluation data. This data was from the calls received by those communicator systems during the June 2000 evaluations. The corresponding evaluation data for the IBM DARPA Communicator system is used as test data. One should note that many of these communicator sites have different dialog strategies. Although the task is the same, the dialog questions and the answers can be quite different. Having no overlap within the training and test data as far as the systems go adds one more degree of difficulty to our experiments. The training data consist of 10640 sentences and 28666 words. The test data consist of 1173 sentences and around 3600 words. Therefore an average sentence contains around three words. The acoustic models are trained using air travel and generic telephony data. A separate class based trigram language model with deleted interpolation is trained on the MELM2 training and held out data and used during speech recognition.

The posterior probabilities are obtained from the sausages [10] which is motivated by minimizing the word error rate rather than sentence error rate. A sausage is a simplified word graph with a specific topology. The word graph is converted into a sequence of confusion sets along time. Each confusion set consists of a group words which are competing hypotheses for a certain time interval. The posterior probabilities for each word is obtained by summing the probabilities of all the paths going over that word.

For each sentence in the confidence training and test data, a sausage is generated and the consensus hypothesis, which is the best path from sausage is hypothesized as the speech recognition output. The best path computed based on the posterior probability resulted an average of 1.4% improvement over the confidence measurement training and test data compared to regular Viterbi based decoding (21.1% versus 19.7%). Each word is labeled as correct ("1") or incorrect ("0") after aligning the hypothesis with the reference transcripts. All the recognition hypotheses are classed using statistical semantic classing. Each sentence is scored with MELM2 to assign semantic probabilities to each word. The corresponding semantic features are extracted for all the words in the sentence. All of the positive (correct recognition) and negative (misrecognition) examples are pooled in two sets. A decision tree is built using the respective features. The decision tree has used the raw scores of each feature. In our decision tree algorithm, the tree is grown by partitioning the data recursively in each node until either the node becomes homogeneous or the node contains too few observation (≤ 200). In order to predict the correctness of a word from the features, one follows the path from the root, to a leaf, according to splits at the interior nodes.

It is useful to have a single measure of performance for confidence measurement. The Equal Error Rate (EER) is one such measure. ERR is the operating point on an Receiver Operating Characteristics (ROC) curve where False acceptance is equal to false rejection. However, for spoken dialog systems it is not a useful operating point as one needs to accept as many correct words as possible at a very low False Acceptance (FA) rate. Table 2 summarizes the Correct Acceptance (CA) rates for word level recognition errors at false alarm rates of 5% and 10%. The FA and CA are calculated using the following formula:

$$\text{FA} = \frac{\#\text{ of falsely accepted words}}{\text{Total }\#\text{ of negative examples}} \times 100 \quad (2)$$

$$\text{CA} = \frac{\#\text{ of correctly accepted words}}{\text{Total }\#\text{ of positive examples}} \times 100 \quad (3)$$

In Fig. 2 we present the ROC curve for MELM2 based features. Here, MELM2 refers to the language model score for a given word, and MELM2-ctx3 refers to MELM2 score of context three where previous and the next scores are included as part of the current score. Similarly, MELM2-ctx5 refers to a window of five scores around the current score. Including context around the current word improves the performance. at 5% FA rate MELM2-ctx3 outperforms MELM2 by 16%. Note that on overall MELM2-ctx5 does not perform as well as MELM2-ctx3. We attribute this to very short sentences (average of 3 words each). Combining each MELM2 based features with the posterior probability improves the CA rate significantly. Note that the most interesting part of the ROC curve for dialog systems is between 0–10%, and the feature combination is particularly effective in this range. Although, the individual MELM2 feature performances compared to posterior probability is fairly low, when combined with the posterior it improves the overall result. This is because of the fact that MELM2 based features bring complementary new information for posteriors. We extracted the CA rates at 5% and 10% FA rate from the ROC curve and presented them in Table 2. The best improvement at 5% FA is 14.6% for posterior combined with MELM2-ctx5.

The results for classer/parser based features are shown in Fig. 3. The features considered here are cTag, cTagArc, pTag and pTagArc. Although there are a number of combinations of these features among themselves and with the posterior, not all of them are included in Fig. 3. The performance of the some of the remaining combination are given in Table 1. Even though the relative improvement of these features combined with posterior probability is similar, the best performance is obtained when posterior is combined with pTag and pTagArc: at 5% FA rate they outperformed posterior by 13%.

The improvement in CA for both feature sets at 10% FA rate is moderate (4–5%). Note that the posterior probability has

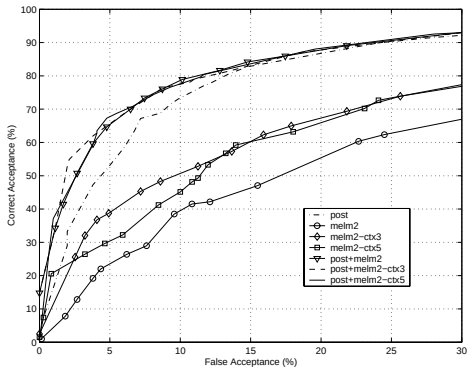


Figure 2: ROC for combination of posterior probability with MELM2 based features.

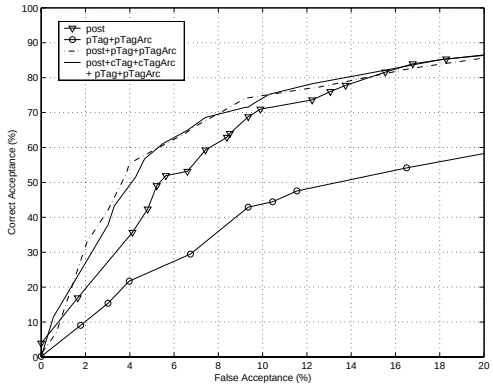


Figure 3: Receiver Operating Characteristics (ROC) for posterior probability, classer and parse probabilities.

different CA rates at the same FA rates in the tables. This is because of the fact that some of the word units used by classer/parser and MELM2 are different. For example, "BUFFALO_NEW_YORK" is a unit for posterior probability and MELM2 but it is three units: "BUFFALO", "NEW", and "YORK" for classer/parser. Therefore the same posterior score is repeated three times when combined with the classer/parser scores. As a result total number of positive and negative examples are different for MELM2 and classer/parser based feature sets which leads to different ROC for the posterior probability.

6. CONCLUSIONS AND FUTURE WORK

We proposed two methods to generate word level semantic features and integrate them with the speech recognition based posterior probability feature in a principled manner. The first set of semantic features consists of tag, tag-arc probabilities for statistical classer and parse trees. The second set of semantic features are derived from the maximum entropy based semantic structured language models (MELM2) with variable context around a given word. Combination of these features with posterior probability provided an improvement of around 13–14% for correct acceptance at 5% false acceptance rate, over posterior probability. Our future research will focus on including dialog state or turn information [6] as well as using semantic features from MELM3 [13]. Moreover, we will attempt to extend word level confidence measurement to concept-level, in which case we expect the semantic features to be more effective.

Acknowledgment

The authors thank to Hakan Erdogan, Lidia Mangu, Raimo Bakis, Ruben San-Segunda, Mark Epstein and Kishore Papineni for fruitful discussions.

Performance of the MELM2 Based Features.(%)		
	5% FA	10% FA
Posterior	53.1	73.4
MELM2	23.5	39.5
MELM2-ctx3	39.0	50.7
MELM2-ctx5	30.4	45.1
Posterior + MELM2	65.4	78.6
Posterior + MELM2-ctx3	65.6	77.5
Posterior + MELM2-ctx5	67.7	77.6

Table 1: Correct Acceptance (CA) rates at 5% and 10% False Acceptance (FA) rates for MELM2 based features.

Performance of the Classer/Parser Features.(%)		
	5% FA	10% FA
Posterior (Post)	45.7	71.0
cTag	17.4	35.2
cTag + cTagArc	20.1	37.9
pTag	16.8	34.2
pTag + pTagArc	24.6	43.8
Post + cTag	54.5	70.9
Post + cTag + cTagArc	55.3	71.3
Post + pTag	52.9	73.2
Post + pTag + pTagArc	58.9	74.9
Post + cTag + pTag + pTagArc	54.9	71.9
Post + cTag + cTagArc + pTag + pTagArc	58.5	74.1

Table 2: Correct Acceptance (CA) rates at 5% and 10% False Acceptance (FA) rates for Classer/Parser based features.

References

[1] H. Erdogan, R. Sarikaya, Y. Gao and M. Picheny, "Semantic Structured Language Models", *ICSLP–2002*, Denver, CO, Sept. 2002.

[2] R. San-Segundo, B. Pellom, K. Hacioglu, and W. Ward, "Confidence Measures for Spoken Dialog Systems", *ICASSP-2001*, pp. 393–396, Salt Lake City, UT, May 2001.

[3] P. Carpenter, C. Jin, D. Wilson, R. Zhang, D. Bohus and A. Rudnicky, "Is This Conversation on Track", *Eurospeech-2001*, pp. 2121–2124, Aalborg, Denmark, Sept. 2001.

[4] F. Wessel, K. Macherey and H. Ney,"A Comparison of Word Graph and N-best list based Confidence Measures", pp. 1587–1590, *ICASSP-2000*, Istanbul, Turkey, June 2000.

[5] C. Neti, S. Roukos and E. Eide, "Word–Based Confidence Measures as a Guide for Stack Search in Speech Recognition", pp. 883–886, *ICASSP–97*, Munich Germany, April 1997.

[6] R. Sarikaya, H. Erdogan, Y. Gao and M. Picheny, "Turn Based Language Modeling for Spoken Dialog Systems", *ICASSP–2002*, Orlando, FL, May 2002.

[7] R. Zhang and A. Rudnicky, "Word Level Confidence Annotation Using Combination of Features", *Eurospeech–2001*, Aalborg, Denmark, Sept., 2002.

[8] S. Pradhan, and W. Ward, "Estimating Semantic Confidence for Spoken Dialog systems", *ICASSP–2002*, Orlando, FL, May 2002.

[9] K. Hacioglu, and W. Ward, "A Concept Graph Based Confidence Measure", *ICASSP–2002*, Orlando, FL, May 2002.

[10] L. Mangu, E. Brill and A. Stolcke, "Finding Consensus Among Words: Lattice Based Word Error Minimization", *Eurospeech–1999*, pp. 495–498, Budapest, Hungary, Sept. 1999.

[11] C. Pao, P. Schmid and J. Glass, "Confidence Scoring for Speech Understanding Systems", *ICSLP–98*, Sydney, Australia, Dec. 1998.

[12] S.F. Chen and R. Rosenfeld, "A Survey of Smoothing Techniques for Maximum Entropy Models", *IEEE Trans. on Speech and Audio Proces.*, vol. 19, no. 3, pp. 37–50, 2000.

[13] H. Erdogan, R. Sarikaya, Y. Gao and M. Picheny, "Semantic Structured Language Models for Spoken Dialog Systems", *Computer Speech and Language*, submitted for publication.