STOCHASTIC LANGUAGE MODELS
FOR SPEECH RECOGNITION AND UNDERSTANDING

G. Riccardi and A. L. Gorin

AT&T Labs-Research
180 Park Avenue
Florham Park, NJ 07932-0971, USA
{dsp3,algor}@research.att.com

1. ABSTRACT

Stochastic language models for speech recognition have traditionally been designed and evaluated in order to optimize word accuracy. In this work, we present a novel framework for training stochastic language models by optimizing two different criteria appropriate for speech recognition and language understanding. First, the language entropy and *salience* measure are used for learning the *relevant* spoken language features (phrases). Secondly, a novel algorithm for training stochastic finite state machines is presented which incorporates the acquired phrase structure into a single stochastic language model. Thirdly, we show the benefit of our novel framework with an end-to-end evaluation of a large vocabulary spoken language system for call routing.

2. INTRODUCTION

Traditionally, the design of stochastic language models for data-driven speech understanding systems is partitioned into two sub-problems. In other words, two language models are independently trained as optimize the speech recognition and understanding part of the system. Within this paradigm, the language model for speech recognition is meant to constrain the search space of all possible word sequences W and to assign a high (low) probability to those sequences W (not) allowed by a given information source. In a similar way, the language model for language understanding is trained for mapping text into a semantic representation of the system task. In both cases, the training algorithms do **not** account for the inter-dependencies between the speech recognition and understanding processes. The rationale behind a training procedure that couples the syntactic and semantic features is an accurate modeling of the word sequences needed to be *recognized for understanding*. In this work we will describe and evaluate a novel framework for training language models accounting for the constraints assigned by the syntactic and semantic models in a large vocabulary spoken language task. In the next two sections we describe the baseline stochastic language models for speech recognition and understanding. In the third section we propose an iterative algorithm for combining language features (phrases) pertaining to the two different models, into a single stochastic language model. For this purpose, we provide a training algorithm for stochastic finite state machines so that the constraints delivered by the language features are combined together. We tested our algorithms for language modeling within the *How May I Help You* call-routing task [3]. In the last section, we report on the end-to-end evaluation of these training algorithms for the *How May I Help You* large vocabulary spoken language system.

3. LANGUAGE MODELING FOR SPEECH RECOGNITION

The classic approach to training language models for speech recognition is the word n-gram paradigm, wherein a word sequence $W = w_1, \ldots, w_M$ probability is computed by means of conditional probabilities whose context length is n:

$$P(W) = \prod_i P(w_i|w_{i-n+1}, \ldots, w_{i-1}) \qquad (1)$$

One of the major disadvantages of this approach is the insufficient statistics for estimating models with large n ($n \geq 5$). In [1] we have shown that language models can be trained in order to capture long spanning dependencies between words by acquiring lexical features (phrases) from training word sequences. The selection of phrases from a corpus is designed so that the computation of high order n-grams lets us reduce the entropy of the training and test corpus. Moreover, by selecting the set of phrases, the number of parameters will not grow exponentially as in the case of the word n-gram. As a result the probability of a word sequence W will be computed as:

$$P(W) = \prod_i P(ephr_i | ephr_{i-n+1}, \ldots, ephr_{i-1}) \qquad (2)$$

where $ephr_i$ is the generic phrase acquired by the process of entropy minimization over the training set and its length ranges from 1 to N_e (N_e is a parameter of the learning algorithm, [1]). Moreover, the algorithm for acquiring phrases automatically provides the *best* word bracketing instance for computing the word sequence probability. For example, in the sentence (x ..x denotes a digit sequence)

yes I want I like to make a call to to tucson arizona the new area code is x x x the and the number is x x x x x x x

the probability $P(W)$ will be decomposed according to the following bracketing:

[yes I want] I [like to] [make a call to] to [tucson arizona] the new [area code] [is x x x] the and [the number is] [x x x x x x x]

For each context length n, phrase n-gram language models have a number of parameters similar to the word n-gram. In particular, in [1] we have shown that the phrase bigram outperforms the word bigram and trigram while its model size is comparable to the word bigram.

4. LANGUAGE MODELING FOR LANGUAGE UNDERSTANDING

Without loss in generality in this work, we will view language understanding for unconstrained language input as the mapping from input text W to a finite number of machine actions $c_i \in C$ [5] [1]. In our previous work, we have introduced the notion of *salience* for evaluating this input-output ($W \Longrightarrow c_i$) association in a quantitative manner. Given the set of machine actions c_k and the phrase $sphr_i$, its *salience* is computed as the Kulbach-Leibler distance between the $P(sphr_i|c_k)$ distribution and the prior distribution $P(c_k)$. The *salience* measure lets us acquire the *meaningful* features (phrases) from a training corpus using the automatic algorithm described in [3]. Hence, we can exploit the set of *salient* fragments to extract the most likely association between W and all possible machine actions c_i. In the case of the *How May I Help You* call-routing task [1], [3] we have 15 call-types (e.g. CALLING CARD, COLLECT, etc.) and a set of $3K$ salient fragments. To illustrate how the input-output association works we consider the sentence:

yes I'd like to make an international call and put it on my credit card my phone credit card please

By using a peak of fragment classifier we get the following interpretation in terms of *salient* fragments:

yes I'd like to [make an international call] [and put] [it on my credit] [card] my phone [credit card please]
CALLING CARD [it on my credit] 1.0

where the detected *salient* fragments $sphr_i$ are bracketed and the second line gives the most likely call-type and its associated *salient* fragment ($argmax_{c_j, sphr_i} P(c_j|sphr_i)$) along with its posteriori probability ($max_{c_j, sphr_i} P(c_j|sphr_i)$) [3]. We tested this understanding model on unconstrained text input for a relatively small number of finite machine actions and proved its effectiveness on speech recognizer outputs as well [1], [3].

5. LANGUAGE MODELING FOR SPEECH RECOGNITION AND UNDERSTANDING

The training of a stochastic language model for speech recognition **and** understanding is directly related to the combination of the set of features ($ephr_i$ and $sphr_i$) acquired through the algorithms described above. More-

[1] Here we will consider only text input, however the underlying model applies to a generic input

over, such a language model should be delivered as a single stochastic finite state machine so that the probability $P(W)$ will computed in a straightforward way for use in the large vocabulary speech recognizer [2]. There are three main steps in training a stochastic finite state machine combining the classes of language features $ephr_i \in \mathbf{ephr}$ and $sphr_i \in \mathbf{sphr}$ (see fig. 1):

- Given the sets $\mathbf{ephr_i}$, $\mathbf{sphr_i}$ and the training set $\mathcal{T}$ (with lexicon V), build two bracketed training sets $\mathcal{T}_e$ and $\mathcal{T}_s$.
- Compute the probability $P(W)$ according to the different bracketing derived by $\mathcal{T}_e$ and $\mathcal{T}_s$ for the word sequence W.
- Train the stochastic finite state machine that recognizes all possible word sequences $W = w_i, \ldots, w_M$ ($W \in V^*$) and delivers the probability $P(W)$.

Both algorithms described in the two previous sections give for each sentence W in the training set $\mathcal{T}$ the bracketing instance corresponding to the feature sets $\mathbf{ephr}$ and $\mathbf{sphr}$. Thus, for each $W = w_i, \ldots, w_M$ in $\mathcal{T}$ we have:

$\mathcal{T}$ ξ_1: $w_1, w_2, w_3, w_4, \ldots$

$\mathcal{T}_e$ ξ_2: $w_1, [w_2, w_3], w_4, \ldots$

$\mathcal{T}_s$ ξ_3: $w_1, [w_2, w_3, w_4], \ldots$

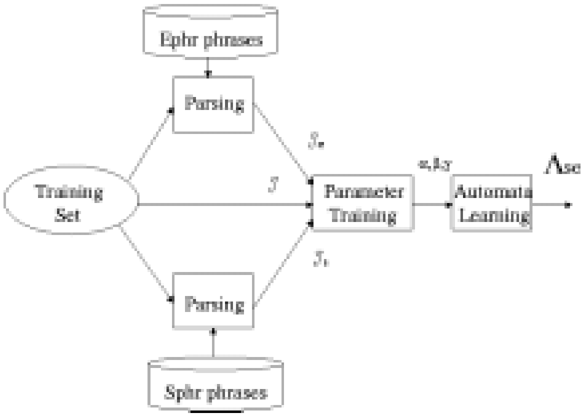

Figure 1: Block Diagram for the phrase-based stochastic finite state machine

Then, the computation of the probability $P(W)$ can be decomposed according to the three parses ξ_1, ξ_2 and ξ_3. In other words, we consider ξ_2 and ξ_3 as *hidden* instances of W being generated by the stochastic models corresponding to $\mathcal{T}_e$ and $\mathcal{T}_s$. For example, in the simple case of $W = w_1, w_2, w_3, w_4$ and being ξ_i ($i = 1, 2, 3$) the only parses allowed, $P(W)$ is calculated as:

$$\begin{aligned} &P(W) \\ &\quad = P(w_1)(\alpha P(w_2|w_1)P(w_3|w_2)P(w_4|w_3) + \\ &\qquad \beta P(w_2 w_3|w_1)P(w_4|w_2 w_3) + \gamma P(w_4 w_3 w_2|w_1)) \end{aligned} \tag{3}$$

where α, β and γ are estimated via the smoothed Maximum Likelihood estimates proposed in [2].
The third step in training a stochastic finite state machine Λ_{se} (see fig 1) is accomplished by using the Variable Ngram Stochastic Automaton (VNSA) learning algorithm [2]. By feeding the three sets $\mathcal{T}$,$\mathcal{T}_e$ and $\mathcal{T}_s$ into the self-organizing automata algorithm we finally get a stochastic finite state machine that estimates $P(W)$ ($W \in V^*$) with the phrase n-gram model as in equation 2. It is worth noting that the non-deterministic automata learning algorithm in [2] let us take advantage of the phrase-based probability computation while the first term in the sum in equation 3 guarantees a non-zero probability estimate for each $W \in V^*$.

6. APPLICATION TO A LARGE VOCABULARY SPOKEN LANGUAGE SYSTEM

We have applied these algorithms for language modeling to the *How May I Help You* call-routing task [3]. In this telecommunications application, we consider people's responses to the open-ended prompt of *How May I help You?* for the purpose of mapping user's utterances into 15 call-types (e.g. CALLING CARD, COLLECT, etc.). Thus, we are aiming at extracting a relatively small number of semantic actions from the responses of a very large number of users who are not trained to the system's capabilities and limitations.
The speech understanding system is composed of a large vocabulary speech recognizer (V=3.6K words) and a language understanding module [3]. We acquired entropy-based and salience-based phrases on 8K training sentence set and tested our language models on 1K held-out set. The sets **ephr** and **sphr** contain respectively $\simeq$ 1K and 3K phrases and 30% of the phrases in **ephr** are shared with **sphr**. Examples of phrases $\in$**ephr**, $\in$**sphr** and $\in$ **ephr**$\bigcap$**sphr** are respectively, *I was wondering if you could, collect call to my*

and *make a collect call*. The phrase length for **ephr** (**sphr**) varies in the range $2-16$ $(1-4)$. We have evaluated the combined stochastic language model Λ_{se} to test its effectiveness for improving the understanding rate of our system. As expected the perplexity of Λ_{se} is similar to the language model using entropy-based phrases [1]. In fact, the phrases **sphr** were selected as part of the language model training procedure with the goal of improving the understanding rate. In table 1 we show the word accuracy results for the baseline system (using a word bigram language model), the entropy-based-only language model and the combined model Λ_{se}. In interpreting the user's responses we used a peak-of-fragment classifier that would search for all *salient* fragments $sphr_i$ in the decoded utterance $\hat{W}$. The figures of merit of this evaluation are the probability of false rejection, where a call is falsely rejected and the probability of correct classification where the correct call-types are associated to $\hat{W}$. In fig 2 the entropy-based and salience-based language model shows the best understanding performance with 25% error rate reduction with respect to the baseline system and 15% compared to the entropy-based-only language model for speech recognition (for a false rejection rate of 40%).

unit type	VNSA order	
	2	3
word	49.5	52.7
ephr	50.2	52.7
ephr & sphr	50.5	53.2

Table 1: Word accuracy versus variable VNSA order using words, e-phr and e-phr & s-phr combined in the model Λ_{se}.

7. CONCLUSION

In this work we have proposed a language model training algorithm targeted at speech recognition **and** understanding. In doing so, we have selected two sets of language features (phrases) that account for the constraints suitable for the speech recognition and understanding part of our system. The combined model has been learned from raw and bracketed data and the scheme for the word sequence probability computation has been provided. We have applied these language models to a large vocabulary spoken language system and demonstrated the effectiveness of our language model training algorithm by reducing the understanding error rate by 25% compared to the baseline system.

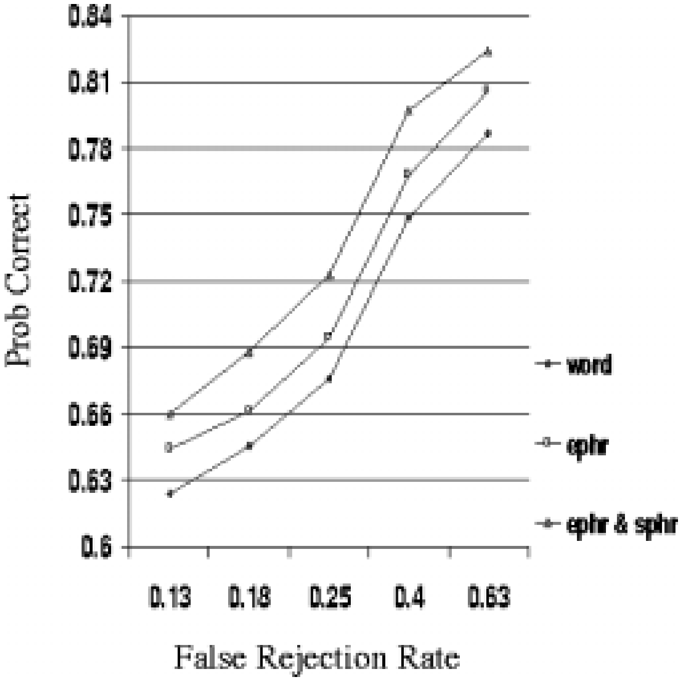

Figure 2: Understanding performances for the word-based, ephr-based and Λ_{se} language model

8. REFERENCES

[1] G. Riccardi, A. L. Gorin, A. Ljolje and M. Riley, "A Spoken Language System for Call Routing," *Proc. ICASSP 97*, pp. 1140-1143, Munich, 1997.

[2] G. Riccardi, R. Pieraccini and E. Bocchieri, "Stochastic automata for language modeling ," *Computer Speech and Language*, **10**, pp. 265-293, 1996.

[3] A. L. Gorin, G. Riccardi and J. W. Wright, "How May I Help You" to appear on *Speech Communication*, 1997.

[4] A. L. Gorin, "Processing of semantic information in fluently spoken language" *Proc. ICSLP*, pp 1001-1004, Philadelphia, 1996.

[5] A. L. Gorin, "On Automated Language Acquisition" *J. Acoust. Soc. Am.*, 97, pp 3341-33461, 1995.