THE JOHNS HOPKINS UNIVERSITY

human language technology
center of excellence

Sub-Lexical and Contextual Modeling of Out-of-Vocabulary Words in Speech Recognition

Carolina Parada, Mark Dredze, Abhinav Sethy, Ariya Rastrow

TECHNICAL REPORT 10

APRIL 27, 2013

Acknowledgements This work is supported, in part, by the Human Language Technology Center of Excellence. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

HLTCOE
810 Wyman Park Drive
Baltimore, Maryland 21211
http://hltcoe.jhu.edu

Sub-Lexical and Contextual Modeling of Out-of-Vocabulary Words in Speech Recognition

Carolina Parada[1], **Mark Dredze**[1], **Abhinav Sethy**[2], **and Ariya Rastrow**[1]
[1]Human Language Technology Center of Excellence, Johns Hopkins University
3400 N Charles Street, Baltimore, MD, USA
carolinap@jhu.edu, mdredze@cs.jhu.edu, ariya@jhu.edu
[2]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA
asethy@us.ibm.com

Abstract

Large vocabulary speech recognition systems fail to recognize words beyond their vocabulary, many of which are information rich terms, like named entities or foreign words. Hybrid word/sub-word systems solve this problem by adding sub-word units to large vocabulary word based systems; new words can then be represented by combinations of sub-word units. We present a novel probabilistic model to *learn* the sub-word lexicon optimized for a given task. We consider the task of Out Of vocabulary (OOV) word detection, which relies on output from a hybrid system. We combine the proposed hybrid system with confidence based metrics to improve OOV detection performance. Previous work address OOV detection as a binary classification task, where each region is independently classified using local information. We propose to treat OOV detection as a sequence labeling problem, and we show that 1) jointly predicting out-of-vocabulary regions, 2) including contextual information from each region, and 3) learning sub-lexical units optimized for this task, leads to substantial improvements with respect to state-of-the-art on an English Broadcast News and MIT Lectures task.

1 Introduction

Most automatic speech recognition systems operate with a large but limited vocabulary, finding the most likely words in the vocabulary for the given acoustic signal. While large vocabulary continuous speech recognition (LVCSR) systems produce high quality transcripts, they fail to recognize out of vocabulary (OOV) words. Unfortunately, OOVs are often information rich nouns, such as named entities and foreign words, and mis-recognizing them can have a disproportionate impact on transcript coherence.

Hybrid word/sub-word recognizers can produce a sequence of *sub-word units* in place of OOV words. Ideally, the recognizer outputs a complete word for in-vocabulary (IV) utterances, and sub-word units for OOVs. Consider the word "Slobodan", the given name of the former president of Serbia. As an uncommon English word, it is unlikely to be in the vocabulary of an English recognizer. While a LVCSR system would output the closest known words (e.x. "slow it dawn"), a hybrid system could output a sequence of multi-phoneme units: `s_l_ow`, `b_ax`, `d_ae_n`. The latter is more useful for automatically recovering the word's orthographic form, identifying that an OOV was spoken, or improving performance of a spoken term detection system with OOV queries. In fact, hybrid systems have improved OOV spoken term detection (Mamou et al., 2007; Parada et al., 2009), achieved better phone error rates, especially in OOV regions (Rastrow et al., 2009b), and obtained state-of-the-art performance for OOV detection (Rastrow et al., 2009a).

In this work, we consider how to optimally create sub-word units for a hybrid system. These units are variable-length phoneme sequences, although in principle our work can be used for other unit types. Previous methods have relied on simple statistics computed from the phonetic representation of text to obtain the most frequent units in the dictionary(Rastrow et al., 2009a; Bazzi and Glass, 2001; Bisani and Ney, 2005). However, it isn't clear why these units would produce the best hybrid output. Instead, we introduce a probabilistic model for *learning* the optimal units for a given task. Our model

learns a segmentation of a text corpus given some side information: a mapping between the vocabulary and a label set; learned units are predictive of class labels.

We learn sub-word units optimized for OOV detection. OOV detection aims to identify regions in the LVCSR output where OOVs were uttered. Towards this goal, we are interested in selecting units such that the recognizer outputs them only for OOV regions while prefering to output a complete word for in-vocabulary regions. We combine the learned sub-words in the output of a hybrid recognizer with confidence based metrics to improve OOV detection performance on an English Broadcast News and MIT Lectures task.

The main contributions of the present work are:

- A novel statistical approach to learn sub-lexical units optimized for a given labeling task. Our model is a log-linear model which combines overlapping features from the phonetic segmentation of a corpus. We apply this model to the OOV detection task.

- Several parameters and variations of the proposed sub-lexical model are studied that have not been addressed comprehensively in previous publications. In particular, we examine the effect of the model hyper-parameters on the size of the sub-word lexicon and average sub-word length, as well its impact in OOV detection performance.

- We also discuss several implementation aspects of the proposed model for learning sub-word units, which will be available online at time of publication.

Our final system combines sub-lexical modeling with a state of the art OOV detector for substantial additive gains on a Broadcast News and MIT-Lectures data-set. We present results in terms of OOV detection, and previously unpublished results in terms of hits and false alarms and phone-error-rate.

We begin by presenting our log-linear model for learning sub-word units with a simple but effective inference procedure in Section 2. This model was first described in Parada et al. (2011). Thereby, we detail several implementation aspects which have not been addressed in previous publications. In Section 3 we detail how the learned units are integrated into a hybrid speech recognition system. Section 4 describes the experimental setup and the OOV detection framework (Parada et al., 2010). We present results in Section 5 combining sub-lexical and contextual models, and analyze the effect of several parameters and variations of the model previously unexamined in Parada et al. (2011). We conclude with a review of related work in Section 6.

2 Learning Sub-Word Units

Given raw text, our objective is to produce a lexicon of sub-word units that can be used by a hybrid system for open vocabulary speech recognition. Rather than relying on the text alone, we also utilize side information: a mapping of words to classes so we can optimize learning for a specific task.

The provided mapping assigns labels Y to the corpus. We maximize the probability of the observed labeling sequence Y given the text W: $P(Y|W)$. We assume there is a latent segmentation S of this corpus which impacts Y. The complete data likelihood becomes: $P(Y|W) = \sum_S P(Y, S|W)$ during training. Since we are maximizing the observed Y, segmentation S must discriminate between different possible labels.

We learn variable-length multi-phone units by segmenting the phonetic representation of each word in the corpus. Resulting segments form the sub-word lexicon.[1] Learning input includes a list of words to segment taken from raw text, a mapping between words and classes (side information indicating whether token is IV or OOV), a pronunciation dictionary D, and a letter to sound model (L2S), such as the one described in Stanley F. Chen (2003). The corpus W is the list of types (unique words) in the raw text input. This forces each word to have a unique segmentation, shared by all common tokens. Words are converted into phonetic representations according to their most likely dictionary pronunciation; non-dictionary words use the L2S model.[2]

[1] Since sub-word units can expand full-words, we refer to both words and sub-words simply as units.

[2] The model can also take multiple pronunciations (Section 2.2.1).

2.1 Model

Inspired by the morphological segmentation model of Poon et al. (2009), $P(Y, S|W)$ is a log-linear model parameterized by Λ:

$$P_\Lambda(Y, S|W) = \frac{1}{Z(W)} u_\Lambda(Y, S, W) \qquad (1)$$

where $u_\Lambda(Y, S, W)$ defines the score of the proposed segmentation S for words W and labels Y according to model parameters Λ. Sub-word units σ compose S, where each σ is a phone sequence, including the full pronunciation for vocabulary words; the collection of σs form the lexicon. Each unit σ is present in a segmentation with some context $c = (\phi_l, \phi_r)$ of the form $\phi_l \sigma \phi_r$. Features based on the context and the unit itself parameterize u_Λ.

In addition to scoring a segmentation based on features, we include two priors inspired by the Minimum Description Length (MDL) principle suggested by Poon et al. (2009). The **lexicon prior** favors smaller lexicons by placing an exponential prior with negative weight on the length of the lexicon $\sum_\sigma |\sigma|$, where $|\sigma|$ is the length of the unit σ in number of phones. Minimizing the lexicon prior favors a trivial lexicon of only the phones. The **corpus prior** counters this effect, an exponential prior with negative weight on the number of units in each word's segmentation, where $|s_i|$ is the segmentation length and $|w_i|$ is the length of the word in phones. Learning strikes a balance between the two priors. Using these definitions, the segmentation score $u_\Lambda(Y, S, W)$ is given as:

$$\begin{aligned} u_\Lambda(Y, S, W) \;=\; \exp\Bigg(& \sum_{\sigma,y} \lambda_{\sigma,y} f_{\sigma,y}(S, Y) \\ & + \sum_{c,y} \lambda_{c,y} f_{c,y}(S, Y) \\ & + \alpha \cdot \sum_{\sigma \in S} |\sigma| \\ & + \beta \cdot \sum_{i \in W} |s_i|/|w_i| \Bigg) \end{aligned} \qquad (2)$$

$f_{\sigma,y}(S, Y)$ are the co-occurrence counts of the pair (σ, y) where σ is a unit under segmentation S and y is the label. $f_{c,y}(S, Y)$ are the co-occurrence counts for the context c and label y under S. The model parameters are $\Lambda = \{\lambda_{\sigma,y}, \lambda_{c,y} : \forall \sigma, c, y\}$. The negative weights for the lexicon (α) and corpus priors (β) are tuned on development data. The normalizer Z sums over all possible segmentations and labels:

$$Z(W) = \sum_{S'} \sum_{Y'} u_\Lambda(Y', S', W) \qquad (3)$$

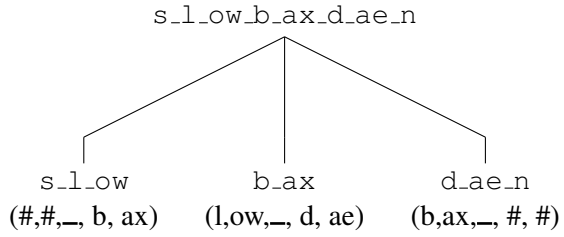


Figure 1: Units and bigram phone context (in parenthesis) for an example segmentation of the word "slobodan".

Consider the example segmentation for the word "slobodan" with pronunciation s,l,ow,b,ax,d,ae,n (Figure 1). The bigram phone context as a four-tuple appears below each unit; the first two entries correspond to the left context, and last two the right context. The example corpus (Figure 2) demonstrates how unit features $f_{\sigma,y}$ and context features $f_{c,y}$ are computed.

2.2 Model Training

Learning maximizes the log likelihood of the observed labels Y^* given the words W:

$$\ell(Y^*|W) = \log \sum_S \frac{1}{Z(W)} u_\Lambda(Y^*, S, W) \qquad (4)$$

We use the Expectation-Maximization algorithm, where the *expectation step* predicts segmentations S given the model's current parameters Λ (Section 2.2.1), and the *maximization step* updates these parameters using gradient ascent. The partial derivatives of the objective (4) with respect to each parameter λ_i are:

$$\frac{\partial \ell(Y^*|W)}{\partial \lambda_i} = E_{S|Y^*,W}[f_i] - E_{S,Y|W}[f_i] \qquad (5)$$

The gradient takes the usual form, where we encourage the expected segmentation from the current model given the correct labels to equal the expected segmentation and expected labels. We now discuss how to compute these expectations.

Labeled corpus: president/$y = 0$ milosevic/$y = 1$
Segmented corpus:
p_r_eh_z_ih_d_ih_n_t/0
m_ih/1 l_aa/1 s_ax/1 v_ih_ch/1
Unit-feature:Value
p_r_eh_z_ih_d_ih_n_t/0:1
m_ih/1:1 l_aa/1:1 s_ax/1:1 v_ih_ch/1:1
Context-feature:Value
(#/0,#/0,_,l/1,aa/1):1,
(m/1,ih/1,_,s/1,ax/1):1,
(l/1,aa/1,_,v/1,ih/1):1,
(s/1,ax/1,_,#/0,#/0):1,
(#/0,#/0,_,#/0,#/0):1

Figure 2: A small example corpus with segmentations and corresponding features. The notation m_ih/1:1 represents unit/label:feature-value. Overlapping context features capture rich segmentation regularities associated with each class.

2.2.1 Inference

Inference is challenging since the lexicon prior renders all word segmentations interdependent. Consider a simple two word corpus: cesar (s,iy,z,er), and cesium (s,iy,z,iy,ax,m). Numerous segmentations are possible; each word has 2^{N-1} possible segmentations, where N is the number of phones in its pronunciation (i.e., $2^3 \times 2^5 = 256$). However, if we decide to segment the first word as: {s_iy,z_er}, then the segmentation for "cesium":{s_iy,z_iy_ax_m} will incur a lexicon prior penalty for including the new segment z_iy_ax_m. If instead we segment "cesar" as {s_iy_z,er}, the segmentation {s_iy,z_iy_ax_m} incurs double penalty for the lexicon prior (since we are including two new units in the lexicon: s_iy and z_iy_ax_m). This dependency requires joint segmentation of the entire corpus, which is intractable. Hence, we resort to approximations of the expectations in Eq. (5).

One approach is to use Gibbs Sampling: iterating through each word, sampling a new segmentation conditioned on the segmentation of all other words. The sampling distribution requires enumerating all possible segmentations for each word (2^{N-1}) and computing the conditional probabilities for each segmentation: $P(S|Y^*, W) = P(Y^*, S|W)/P(Y^*|W)$ (the features are extracted from the remaining words in the corpus). Using M sampled segmentations $S_1, S_2, \ldots S_m$ we compute $E_{S|Y^*,W}[f_i]$ as follows:

$$E_{S|Y^*,W}[f_i] \approx \frac{1}{M} \sum_j f_i[S_j]$$

Similarly, to compute $E_{S,Y|W}$ we sample a segmentation and a label for each word. We compute the joint probability of $P(Y, S|W)$ for each segmentation-label pair using Eq. (1). A sampled segmentation can introduce new units, which may have higher probability than existing ones.

Using these approximations in Eq. (5), we update the parameters using gradient ascent:

$$\bar{\lambda}_{new} = \bar{\lambda}_{old} + \gamma \nabla \ell_{\bar{\lambda}}(Y^*|W)$$

where $\gamma > 0$ is the learning rate.

To obtain the best segmentation, we use deterministic annealing. Sampling operates as usual, except that the parameters are divided by a value, which starts large and gradually drops to zero. To make burn in faster for sampling, the sampler is initialized with the most likely segmentation from the previous iteration. To initialize the sampler the first time, we set all the parameters to zero (only the priors have non-zero values) and run deterministic annealing to obtain the first segmentation of the corpus.

2.2.2 Efficient Sampling

Sampling a segmentation for the corpus requires computing the normalization constant (3), which contains a summation over all possible corpus segmentations. Instead, we approximate this constant by sampling words independently, keeping fixed all other segmentations. Still, even sampling a single word's segmentation requires enumerating probabilities for all possible segmentations.

We sample a segmentation efficiently using dynamic programming. We can represent all possible segmentations for a word as a finite state machine (FSM) (Figure 3), where arcs weights arise from scoring the segmentation's features. This weight is the negative log probability of the resulting model after adding the corresponding features and priors.

However, the lexicon prior poses a problem for this construction since the penalty incurred by a new unit in the segmentation depends on whether that

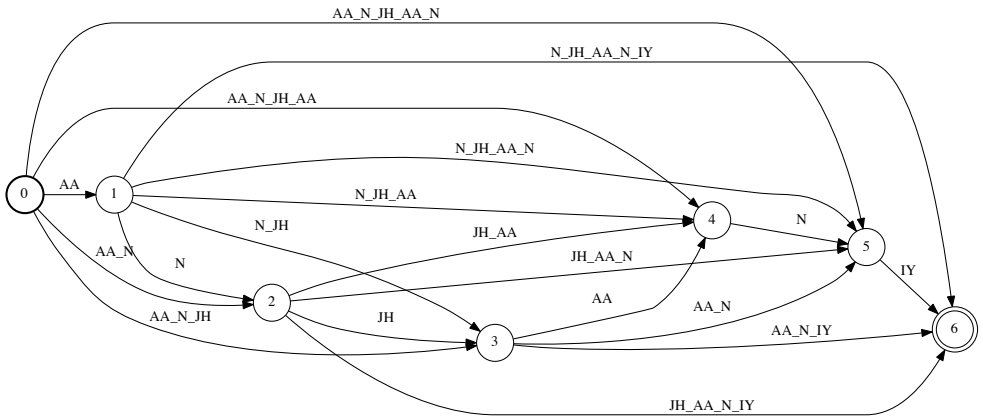


Figure 3: FSM representing all segmentations for the word ANJANI with pronunciation: AA,N,JH,AA,N,IY

unit is present elsewhere in that segmentation. For example, consider the segmentation for the word ANJANI: AA_N, JH, AA_N, IY. If none of these units are in the lexicon, this segmentation yields the lowest prior penalty since it repeats the unit AA_N. [3] This global dependency means paths must encode the full unit history, making computing forward-backward probabilities inefficient.

Our solution is to use the *Metropolis-Hastings* algorithm, which samples from the true distribution $P(Y, S|W)$ by first sampling a new label and segmentation (y', s') from a simpler proposal distribution $Q(Y, S|W)$. The new assignment (y', s') is accepted with probability:

$$\alpha(Y', S'|Y, S, W) = \min\left(1, \frac{P(Y', S'|W)Q(Y, S|Y', S', W)}{P(Y, S|W)Q(Y', S'|Y, S, W)}\right)$$

We choose the proposal distribution $Q(Y, S|W)$ as Eq. (1) omitting the lexicon prior, removing the challenge for efficient computation. The probability of accepting a sample becomes:

$$\alpha(Y', S'|Y, S, W) = \min\left(1, \frac{\sum_{\sigma \in S'} |\sigma|}{\sum_{\sigma \in S} |\sigma|}\right) \quad (6)$$

Alg. 1 shows our full sub-word learning procedure, where sampleSL (Alg. 2) samples a segmentation and label sequence for the entire corpus from $P(Y, S|W)$, and sampleS samples a segmentation from $P(S|Y^*, W)$.

[3] Splitting at phone boundaries yields the same lexicon prior but a higher corpus prior.

Algorithm 1 Training

Input: Lexicon L from training text W, Dictionary D, Mapping M, L2S pronunciations, Annealing temp T.
Initialization:
Assign label $y^*_m = M[w_m]$. $\bar{\lambda}_0 = \bar{0}$
S_0 = random segmentation for each word in L.

for $i = 1$ **to** K **do**
/* **E-Step** */
S_i = bestSegmentation(T, λ_{i-1}, S_{i-1}).
for $k = 1$ **to** NumSamples **do**
(S'_k, Y'_k) = sampleSL($P(Y, S_i|W)$,$Q(Y, S_i|W)$)
$\tilde{S}_k$ = sampleS($P(S_i|Y^*, W)$,$Q(S_i|Y^*, W)$)
end for

/* **M-Step** */
$E_{S,Y|W}[f_i] = \frac{1}{NumSamples} \sum_k f_{\sigma,l}[S'_k, Y'_k]$
$E_{S|Y^*,W}[f_{\sigma,l}] = \frac{1}{NumSamples} \sum_k f_{\sigma,l}[\tilde{S}_k, Y^*]$
$\bar{\lambda}_i = \bar{\lambda}_{i-1} + \gamma \nabla L_{\bar{\lambda}}(Y^*|W)$
end for

S = bestSegmentation(T, λ_K, S_0)
Output: Lexicon L_o from S

2.3 Implementation Aspects

This section highlights some important aspects of our implementation of the training algorithm 1). The key operation for training is sampling a segmentation for a word given the current model parameters (needed for $bestSegmentation$, $sampleSL$, and $sampleS$ methods). We sample a segmentation for a word by sampling a path in a finite state machine representing all possible segmentations of that word. We use the OpenFst toolkit [4] and available

[4] http://www.openfst.org

Algorithm 2 sampleSL($P(S, Y|W), Q(S, Y|W)$)

for $m = 1$ **to** M (NumWords) **do**
  (s'_m, y'_m) = Sample segmentation/label pair for word w_m according to $Q(S, Y|W)$
  $Y' = \{y_1 \dots y_{m-1} y'_m y_{m+1} \dots y_M\}$
  $S' = \{s_1 \dots s_{m-1} s'_m s_{m+1} \dots s_M\}$
  $\alpha = \min\left(1, \frac{\sum_{\sigma \in S'} |\sigma|}{\sum_{\sigma \in S} |\sigma|}\right)$
  with prob $\alpha : y_{m,k} = y'_m, s_{m,k} = s'_m$
  with prob $(1 - \alpha) : y_{m,k} = y_m, s_{m,k} = s_m$
end for
return $(S'_k, Y'_k) = [(s_{1,k}, y_{1,k}) \dots (s_{M,k}, y_{M,k})]$

operations as follows.

Given a word and a label (0-IV or 1-OOV) we build a "segmentation" finite state machine, where arc labels indicate the features for that segmentation and label. For example, in Figure 3, the label for the arc between nodes 2 and 3 would be: {JH/1,AA/1_N/1_AA/1_N/1}, where the portion of the label before the comma indicates the sub-word JH, and its label 1, and the rest indicates the bi-gram phone context of that sub-word. Note that each phone/sub-word has its associated label.

Each time we need to sample a segmentation for a word, we load the transducer associated with that word and label and assign a score to each arc using a $Mapper$.[5] In our implementation, the mapper assigns weights to each arc according to the features encoded in the label and the current model parameters, plus the corpus prior: $\beta/|w_i|$.

To sample a path we run the forward-backward algorithm, where the backward computations are carried out explicitly by computing the $ShortestDistance$ to the final state in the $LogSemiring$. The forward pass is done through sampling, i.e. we traverse the machine only computing forward probabilities for arcs leaving the sampled state. Specifically, we use the $RandGen$ operation with a customed arc selector $BetaArcSelector$. This operation randomly selects an outgoing transition at a given state with respect to the weights of each arc $Times$ the probability of leaving the destination state (computed in the backward pass). All weights are treated as negative log probabilities after normalizing for the total weight leaving each state. Once we sample a path, we accept it according to Eq. (6) or keep the previous segmentation if rejected.

To sample a segmentation for a word given its true label ($sampleS$), we sample a path through an acceptor including all segmentations for that word with its true-label. To sample a segmentation and label ($sampleSL$) we sample a path through the union of two acceptors for the same word one with label 0 and a second one with label 1. Finally, to obtain a segmentation of the entire corpus, we sample one word at a time assuming the segmentations of all other words fixed. When using annealing all weights are divided by a given temperature.

3 OOV detection using hybrid models

To evaluate our model for learning sub-word units, we consider the task of out-of-vocabulary (OOV) word detection. Since detectors process hybrid recognizer output, we can evaluate different sub-word unit lexicons for the hybrid recognizer and measure the change in OOV detection accuracy.

Our model (Section 2.1) can be applied to this task by using a dictionary D to label words as IV ($y_i = 0$ if $w_i \in D$) and OOV ($y_i = 1$ if $w_i \notin D$). This results in a labeled corpus, where the labeling sequence Y indicates the presence of out-of-vocabulary words (OOVs). Given a sub-word lexicon, the word and sub-words are combined to form a hybrid language model (LM) to be used by the LVCSR system. This hybrid LM captures dependencies between word and sub-words. In the LM training data, all OOVs are represented by the smallest number of sub-words in a left to right greedy fashion which corresponds to their pronunciation. Pronunciations for all OOVs are obtained using grapheme to phone models (Stanley F. Chen, 2003).

The output can be the one-best transcripts, lattices or confusion networks. While lattices contain more information, they are harder to process; confusion networks offer a trade-off between richness and compactness (Mangu et al., 1999). Figure 4 depicts a confusion network decoded by the hybrid system for a section of an utterance in our test-set. Below the network we present the reference transcription. In this example, two OOVs were uttered: "slobo-

[5] "Mappers are function objects used by the Map operation to transform arcs and/or final states" (http://www.openfst.org/twiki/bin/view/FST/FstAdvancedUsage).

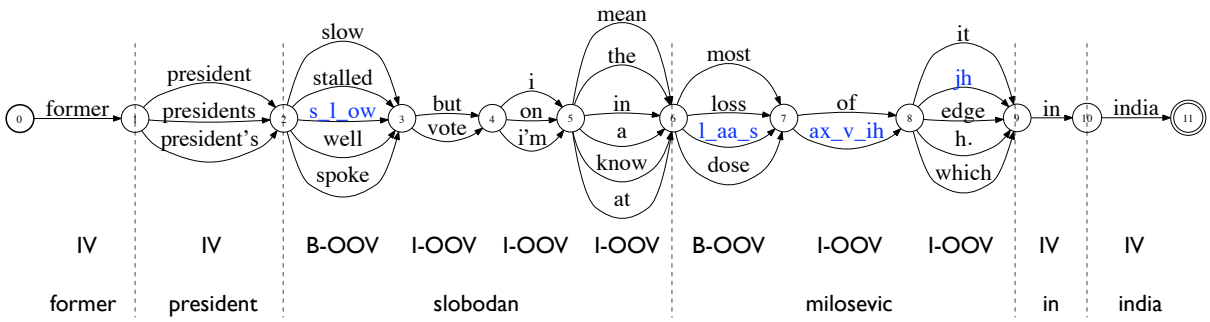


Figure 4: Example confusion network from the hybrid system with OOV regions and BIO encoding. It represents a compact representations of the recognizer's hypotheses, where the top path corresponds to the most likely output from the recognizer. A confusion network is composed of a sequence of *confused regions*, indicating the set of most likely word/sub-word hypotheses uttered and their posterior probabilities.

dan" and "milosevic" and decoded as four and three in-vocabulary words, respectively. A *confused region* (also called "bin") corresponds to a set of competing hypothesis between two nodes. The goal is to correctly label each of the "bins" as OOV or IV. Note the presence of both fragments (e.g. `s_l_ow`, `l_aa_s`) and words in some of the hypothesis bins.

Since sub-words represent OOVs while building the hybrid LM, the existence of sub-words in ASR output indicates an OOV region. Better sub-word output yields better recognition.

Baseline Unit Selection: We used Rastrow et al. (2009a) as our baseline unit selection method, a data driven approach where the language model training text is converted into phones using the dictionary (or a letter-to-sound model for OOVs), and a N-gram phone LM is estimated on this data and pruned using a relative entropy based method. The hybrid lexicon includes resulting sub-words – ranging from unigrams to 5-gram phones, and the 83K word lexicon.

3.1 OOV Detectors

We consider two models: a Maximum Entropy OOV detection system with features from filler and confidence estimation models proposed by Rastrow et al. (2009a), and a CRF sequence based model using contextual information proposed by Parada et al. (2010). While the sequence model does substantially better than the Maximum Entropy classification model, we evaluate both to consider how our learned sub-word units compliments different types of information. We briefly review both models here.

Maximum Entropy: Rastrow et al. (2009a) present a MaxEnt classification based OOV detector, in which each region is classified as either IV (in-vocabulary) or OOV. Their two features are the existence of a) sub-words and b) high entropy in a network region.

CRF: While Maximum Entropy assigns a label to each region in the confusion network independently, OOV words tend to co-occur since they are often recognized as two or more IV words. In our example (Figure 4), the OOV word "slobodan" was recognized as four IV words: "slow but i mean". Therefore, Parada et al. (2010) used a sequence model, specifically a Conditional Random Fields (CRF) (Lafferty et al., 2001), to capture this contextual information. In addition to using this new model, they introduced several sets of contextual features:

- **Current Word:** Words from the LVCSR decoding of the sentence are used in the CRF OOV detector. For each bin in the confusion network, we select the word with the highest probability (best hypothesis). We then add the best hypothesis word as a feature of the form: `current_word=X`.

- **Context-Bigrams** Unigrams and bigrams from the best hypothesis in a window of 5 words around current bin. This feature ignores the best hypothesis in the current bin, i.e., `word[-2],word[-1]` is included, but `word[-1],word[0]` is not.

- **Current-Trigrams** Unigrams, bigrams, and trigrams in a window of 5 words around and including current bin.

- **All-Words** All of the above features.

In addition to local context features, they also included global context features based probabilities from a language model: a standard 4-gram language model with interpolated modified KN discounting and a syntactic language model (Filimonov and Harper, 2009). Both language models were trained on 130 million words from Hub4 CSR 1996 (Garofolo et al., 1996) and were restricted to LVCSR system's vocabulary. See Parada et al. (2010) for a full description of these features.

4 Experimental Setup

4.1 Data sets

We used the data set constructed by Can et al. (2009) (OOVCORP) for the evaluation of Spoken Term Detection of OOVs since it focuses on the OOV problem. The corpus contains 100 hours of transcribed Broadcast News English speech. There are 1290 unique OOVs in the corpus, which were selected with a minimum of 5 acoustic instances per word and short OOVs inappropriate for STD (less than 4 phones) were explicitly excluded. Example OOVs include: NATALIE, PUTIN, QAEDA, HOLLOWAY, COROLLARIES, HYPERLINKED, etc. This resulted in roughly 24K (2%) OOV tokens.

For LVCSR, we used the IBM Speech Recognition Toolkit (Soltau et al., 2010)[6] to obtain a transcript of the audio. Acoustic models were trained on 300 hours of HUB4 data (Fiscus et al., 1998) and utterances containing OOV words as marked in OOVCORP were excluded. The language model was trained on 400M words from various text sources with a 83K word vocabulary. The LVCSR system's WER on the standard RT04 BN test set was 19.4%. Excluded utterances amount to 100hrs. These were divided into 5 hours of training for the OOV detector and 95 hours of test. Note that the OOV detector training set is different from the LVCSR training set.

We also use a hybrid LVCSR system, combining word and sub-word units obtained from either our approach (Section 2) or a state-of-the-art baseline approach (Rastrow et al., 2009a) (Section 3). Our hybrid system's lexicon has 83K words and 5K or 10K sub-words. Note that the word vocabulary is common to both systems and only the sub-words are selected using either approach. The word vocabulary used is close to most modern LVCSR system vocabularies for English Broadcast News; the resulting OOVs are more challenging but more realistic (i.e. mostly named entities and technical terms). The 1290 words are OOVs to both the word and hybrid systems.

In addition we report OOV detection results on a MIT lectures data set (Glass et al., 2010) consisting of 3 Hrs from two speakers with a 1.5% OOV rate. These were divided into 1 Hr for training the OOV detector and 2 Hrs for testing. Note that the LVCSR system is trained on Broadcast News data. This out-of-domain test-set help us evaluate the cross-domain performance of the proposed and baseline hybrid systems. OOVs in this data set correspond mainly to technical terms in computer science and math. e.g. ALGORITHM, DEBUG, COMPILER, LISP.

4.2 Learning parameters

For learning the sub-words we randomly selected from training 5,000 words which belong to the 83K vocabulary and 5,000 OOVs[7]. For development we selected an additional 1,000 IV and 1,000 OOVs. This was used to tune our model hyper parameters (set to $\alpha = -1$, $\beta = -20$). There is no overlap of OOVs in training, development and test sets. All feature weights were initialized to zero and had a Gaussian prior with variance $\sigma = 100$. Each of the words in training and development was converted to their most-likely pronunciation using the dictionary for IV words or the L2S model for OOVs.[8]

The learning rate was $\gamma_k = \frac{\gamma}{(k+1+A)^\tau}$, where k is the iteration, A is the stability constant (set to $0.1K$), $\gamma = 0.4$, and $\tau = 0.6$. We used $K = 40$ itera-

[6]The IBM system used speaker adaptive training based on maximum likelihood with no discriminative training.

[7]This was used to obtain the 5K hybrid system. To learn sub-words for the 10K hybrid system we used 10K in-vocabulary words and 10K OOVs. All words were randomly selected from the LM training text.

[8]In this work we ignore pronunciation variability and simply consider the most likely pronunciation for each word. It is straightforward to extend to multiple pronunciations by first sampling a pronunciation for each word and then sampling a segmentation for that pronunciation.

tions for learning and 200 samples to compute the expectations in Eq. 5. The sampler was initialized by sampling for 500 iterations with deterministic annealing for a temperature varying from 10 to 0 at 0.1 intervals. Final segmentations were obtained using 10,000 samples and the same temperature schedule. We limit segmentations to those including units of at most 5 phones to speed sampling with no significant degradation in performance. We observed improved performance by dis-allowing whole word units.

4.3 Evaluation

We obtain confusion networks from both the word and hybrid LVCSR systems. We align the LVCSR transcripts with the reference transcripts and tag each confusion region as either IV or OOV. The OOV detector classifies each region in the confusion network as IV/OOV.

We evaluate the performance of the two OOV detection systems (MaxEnt and CRF) with units learned by our model in terms of:

- Hits: sub-word units predicted in OOV regions, and False Alarms: sub-word units predicted for in-vocabulary words
- OOV detection performance. We present results using standard detection error tradeoff (DET) curves (Martin et al., 1997). DET curves measure tradeoffs between misses and false alarms and can be used to determine the optimal operating point of a system.
- Phone Error Rate (PER): PER evaluates whether the sub-word units predicted in OOV regions resemble the true pronunciation of the OOV. [9]

Previous work reported OOV detection accuracy on all test data. However, once an OOV word has been observed in the training data for the OOV detector, even if it never appeared in the LVCSR training data, it is no longer truly OOV. The features used in previous approaches did not necessarily provide an advantage on observed versus unobserved OOVs, but our contextual features do yield an advantage. Therefore, we report accuracy on All OOVs and Unobserved OOVs: OOV words that do not appear in either the OOV detector's or the LVCSR's training data. While the latter penalizes our results, it is a more informative metric of true system performance.

[9] We do not evaluate WER because the units used are phonetic, thus their concatenation does not provide the correct spelling of the OOV uttered.

5 OOV Detection Results

Table 1 shows an example of the predicted sub-words for the baseline and proposed systems. Interestingly, the average sub-word length for the proposed units exceeded that of the baseline units by 0.3 phones (`Baseline 10K` average length was 3.20 with standard deviation 0.82, while that of `Learned Units 10K` was 3.58 with standard deviation 0.87.[10]

Table 2 shows the percent of Hits and False Alarms in `OOVCORP`. We can see that the proposed system increases the Hits by roughly 8% absolute, while increasing the False Alarms by 0.3%.

Table 3 also shows the performance on MIT Lectures. Note that both the sub-word lexicon and the LVCSR models were trained on Broadcast News data, hence this data-set evaluates the robustness of learned sub-words across domains. The OOVs in these domains are quite different: MIT Lectures' OOVs correspond to technical computer science and math terms, while in Broadcast News they are mainly named-entities. However, similar to the results in `OOVCORP`, we found that the learned sub-words provide larger coverage of OOV regions in MIT Lectures domain. These results suggest that the proposed sub-words are not simply modeling the training OOVs (named-entities) better than the baseline sub-words, but also describe better novel unexpected words.

Hybrid System	No. Sub-words	Hits (%)	FAs (%)
Baseline	5k	18.25	1.49
Learned Units	5k	26.78	1.78
Baseline	10k	24.26	1.82
Learned Units	10k	28.96	1.92

Table 2: Coverage of OOV regions by sub-words in OOVCORP.

In order to establish a comparative baseline, we first present results for MaxEnt and CRF using the same feature set (Word-Entropy and Fragment-Posterior from Rastrow et al. (2009a).) For the

[10] This difference is significant at 99% under the T-Test.

Word	Baseline sub-words	Learned units
adrianna	ey_d, r_iy, ae_n, ax	ey_d, r_iy_ae_n_ax
yakusuni	y_ax, k_uw, s_uw, n_iy	y_ax_k_uw, s_uw, n_iy
systembolaget	s_ih, s_t, ax_m, b_ax, l_aa, zh, ey	s_ih, s_t_ax_m, b_ax_l_aa, zh_ey
quicktime	k_w, ih_k, t_ay, m	k_w_ih, k_t, ay_m
natascha	n_ax, t_aa, sh_ax	n_ax, t_aa, sh_ax
lieutanant	l_uw, t_ae, n_ih, n_t	l_uw_t, ae_n, ih_n_t

Table 1: Example representations of OOVs using the Baseline and Learned Subwords.

Hybrid System	No. Sub-words	Hits (%)	FAs (%)
Baseline	5k	17.03	2.33
Learned Units	5k	22.14	2.72
Baseline	10k	21.41	2.55
Learned Units	10k	21.89	2.66

Table 3: Coverage of OOV regions by sub-words in MIT Lectures.

CRF model[11], we used a second order model with BIO encoding and all real-valued features were normalized and quantized using the uniform-occupancy partitioning described in White et al. (2007). [12] Quantization of real valued features is standard for log-linear models as it allows the model to take advantage of non-linear characteristics of feature values and is better handled by the regularization term. For the MaxEnt model quantized and continuous features achieve comparable performance. White et. al. found it improved performance.

Figure 5 depicts DET curves for OOV detection for the MaxEnt and CRF on unobserved OOVs and all OOVs in the test data. Predictions at different FA rates are obtained by varying a probability threshold. For MaxEnt we used the predicted label probability and for CRFs the marginal probability of each bin's label. We present results using the baseline sub-words and the sub-words we proposed in Section 2. The only difference between these systems is the sub-word lexicon used in the hybrid system to decode the test-set.

Comparing performance of proposed sub-words vs baseline sub-words using the MaxEnt detector, we can see that at a 5% FA rate, our system (`Learned Units (5k) MaxEnt`) reduces the miss OOV rate by 6.3% absolute over the baseline (`Baseline (5k) MaxEnt`) when evaluating all OOVs. For unobserved OOVs, it achieves 3.2% absolute improvement. A larger lexicon (`Baseline (10k) MaxEnt` and `Learned Units (10k) MaxEnt`) shows similar relative improvements. Note that the features used so far do not necessarily provide an advantage for unobserved versus observed OOVs, since they ignore the decoded word/sub-word sequence and only include posterior probability information from the decoded networks.

Using a CRF model for detection also shows a clear improvement over using a MaxEnt classifier. This model has up to 4.5% absolute improvement at 5% false alarm rate, despite using the identical features as the MaxEnt baseline. Even a small amount of context as expressed through local labeling decisions improves OOV detection. Since the hybrid system including 10K sub-words in the lexicon consistently outperform the one including only 5K sub-words, we present all future experiments using the 10K hybrid system.

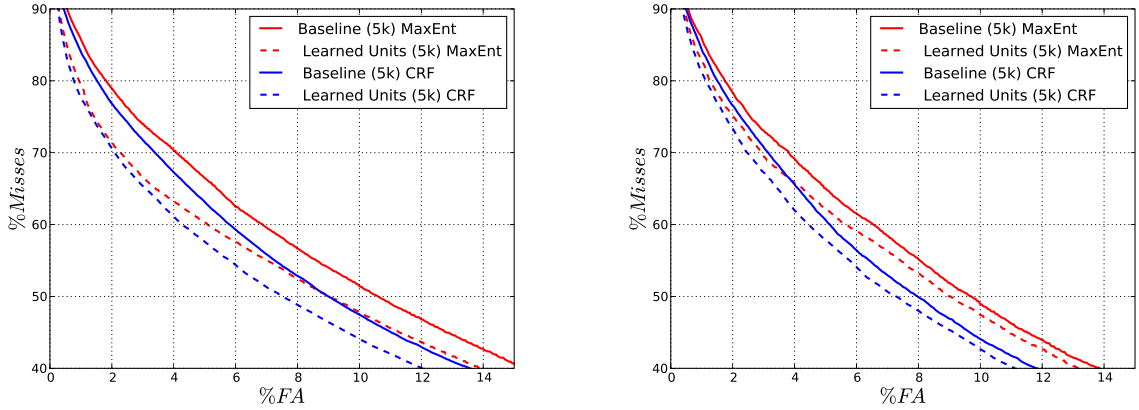
5.1 Size and Number of Sub-words

The size of the sub-word lexicon included in the proposed hybrid system, as well as the average length of the included sub-words is affected by a) the number of unique words in the training corpus, and b) the weight assigned to the corpus and lexicon priors of our model as given by Equation 2 (hyper-parameters). A higher lexicon prior weight α encourages a smaller lexicon (both in number of units and their length), while increasing the corpus prior weight β biases for a shorter description of words (i.e. longer sub-words). Learning strikes a balance between these two priors while maximizing the likelihood of the training labeling sequence.

Table 4 shows the effect of varying the weight associated with the lexicon and corpus priors during

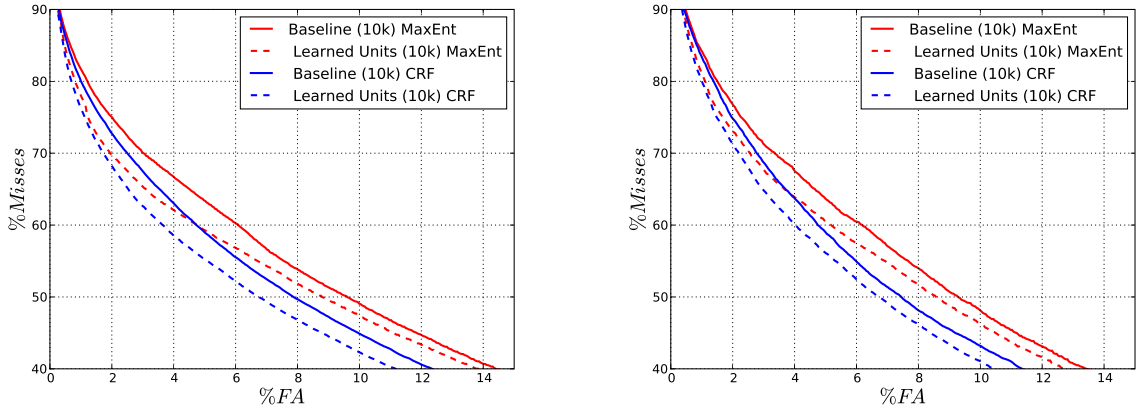
[11] CRF experiments used the CRF++ package http://crfpp.sourceforge.net/

[12] All experiments use 50 partitions with a minimum of 100 training values per partition.



(a) 5K System evaluated on All OOVs.

(b) 5K System evaluated on Un-observed OOVs.



(c) 10K System evaluated on All OOVs.

(d) 10K System evaluate on Un-observed OOVs.

Figure 5: DET curves for OOV detection using a Maximum Entropy (MaxEnt) classifier vs a 2^{nd} order CRF. Solid curves indicate that the hybrid system was built using the Baseline units (Rastrow et al., 2009a) while dashed curves use Learned Units proposed in Section 2. All results are on OOVCORP.

training (5K unique IV words and 5k unique OOV words). We show the size of the sub-word lexicon and the average length of sub-words while varying one of the prior parameters. As expected, a higher absolute value for the lexicon prior α (while keeping the corpus prior β fixed) yields smaller lexicon sizes and smaller average sub-word length. Increasing the effect of the corpus prior β results in larger average sub-word length.

5.2 Convergence behavior

Figure 6 illustrates the behavior of the training algorithm by showing the evolution of the label accuracy on training and held-out set over several iterations, for different values of the lexicon and corpus prior weights. We can see that while the performance in the training set continues to increase even after 60 iterations, it levels off rather quickly on the held-out set which contains 1,000 IV and 1,000 OOV words distinct from those in training. We selected the weights for the lexicon and corpus prior based on the performance of the model on the held-out set. We found that several settings yielded similar performance and selected $\alpha = -1$, and $\beta = -20$.

For completeness of error analysis, we investigated the effect on varying these priors on OOV detection on the test-set. As shown in Figure 7, varying the prior weights did not greatly affect the performance on the test-set. Finally, we varied the temperature schedule and did not find significant changes

α (lexicon)	β (corpus)	No. sub-words	Avg Length	Std-dev
-0.1	-20	6252	3.61	0.96
-1	-20	4490	3.23	0.88
-5	-20	1722	2.62	0.65
-10	-20	1530	2.57	0.64
-20	-20	1560	2.58	0.64
-1	-1	3993	3.04	0.79
-1	-10	4240	3.14	0.85
-1	-20	4490	3.23	0.88
-1	-50	5127	3.51	0.97
-1	-100	5443	3.62	0.99

Table 4: Effect of α (lexicon prior) and β (corpus prior) on the lexicon size and sub-word length.

in convergence performance by modifying the burn in period.

5.3 Contextual Features

Learning sub-words improved both the MaxEnt and CRF model with baseline features. We now investigate how the learned sub-words combined with contextual features (Section 3.1) affects performance. Figure 8 shows the additive improvements of these features when using the baseline hybrid recognizer. For Unobserved OOVs, the current words did not improve performance. However, when the current words are combined with the lexical context (`All-Words`), they give a significant boost in performance: a 4.4% absolute improvement at 5% false alarm rate over the previous CRF system, and 9% over the MaxEnt baseline for un-observed OOVs. Interestingly, only combining context and current word gives a substantial gain. This indicates that OOVs tend to occur with certain distributional characteristics that are independent of the OOV word uttered (since we consider only unobserved OOVs), perhaps because OOVs tend to be named entities, foreign words, or rare nouns.

When evaluating all OOVs, the proposed features achieve 25% absolute improvement at a 5% FA rate, reducing the OOV regions missed from 59% (`CRF`) to 34% (`+All-Words`). With respect to the MaxEnt system the absolute gain is 30%. Since these features include the identity of the decoded words we can see a clear advantage when evaluating all OOVs (including OOVs observed in the OOV detector training set) due to repeated OOVs decoded using the same in-vocabulary word sequences. Adding the *stemmed* versions of the words achieves less than 1% absolute improvement in performance.

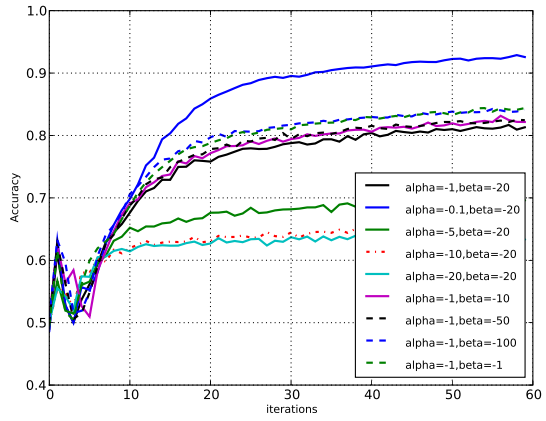
We next add the global contextual features from the language models. At 5% false alarm rate LM features (`+Syntactic LM`, or `4-gram LM`) yield a 5.5% absolute improvement with respect to the previous best result (`All-Words`). The syntactic LM did not provide significant improvements over the 4-gram LM. Higher order language models did not improve.

5.4 Final system

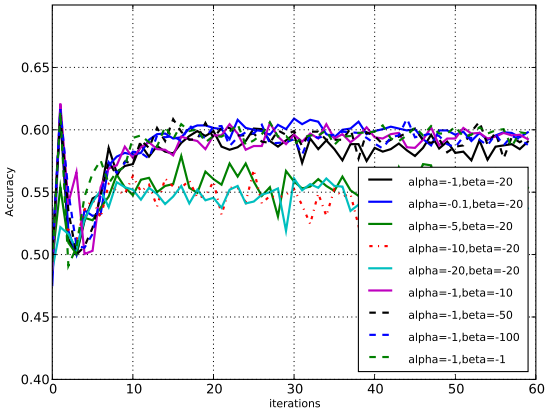
Figure 9 summarizes all lexical and global features in a single second order BIO encoded CRF using baseline and proposed sub-words systems. The Learned sub-words with context features (`Learned Units + Context`) still improves over the baseline (`Baseline 10k + Context`), however the relative gain is reduced.

For unobserved OOVs our final system achieves a 14.8% absolute improvement at 5% FA rate by adding context. Including the learned sub-words the total gain is 16.5%. The absolute improvement on All OOVs was 30.5% using context and 31.8% including also new sub-words. The result on all OOVs includes *observed* OOVs: words that are OOV for the LVCSR but are encountered in the OOV detector's training data.

Figure 10 shows the OOV detection results in the MIT Lectures data set using proposed sub-words and context features. When evaluating on unobserved OOVs in the MIT Lectures data set, our final system achieves a 1.4% absolute improvement at 5% FA rate by adding context. Including the learned sub-words the total gain is 7.1%. The absolute improvement on All OOVs was 8.3% using context and

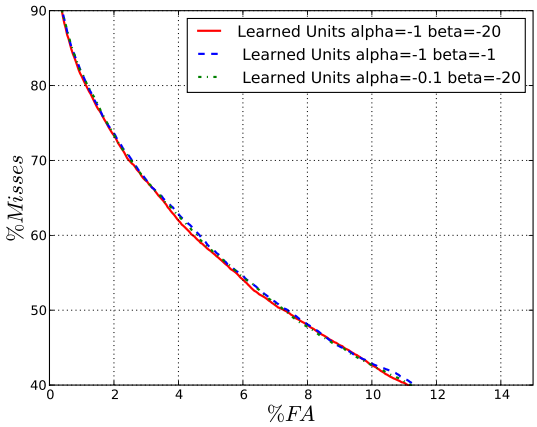


(a)

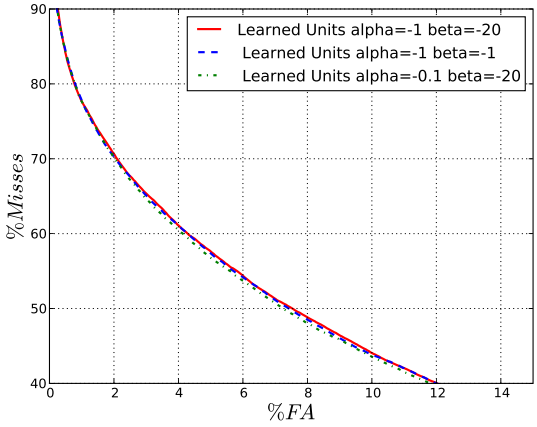


(b)

Figure 6: OOV/IV label accuracy on (a) training set and (b) heldout set over several iterations for different values of the prior parameters.



(a)



(b)

Figure 7: Det curve for OOV detection for (a) unseen OOVs and (b) all OOVs for different values of the prior parameters (evaluated on test-set).

12.3% including also new sub-words.

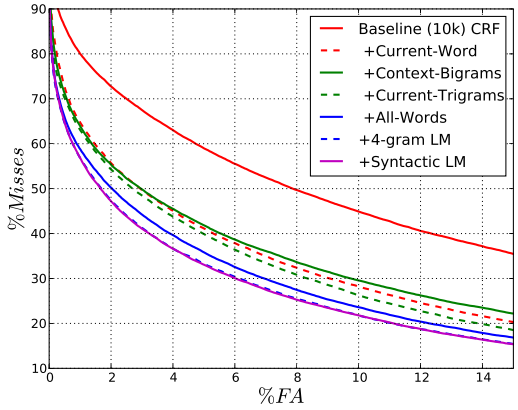
Similar to Broadcast News, we found that including context for OOV detection on the MIT Lectures data-set achieves improvements in performance. However the gains from context are smaller. We conjecture that this is due to the higher WER[13] and the less structured nature of the domain: i.e. ungrammatical sentences, disfluencies, incomplete sentences, making it more difficult to predict OOVs based on context.

The learned sub-words achieve larger gains with respect to the baseline hybrid system in the MIT Lectures data set than in Broadcast News `OOVCORP` data set, specially for un-observed OOVs. These results suggest that the learned sub-words are not simply modeling the training OOVs better than the baseline sub-words, but also describe better novel unexpected words. Recall that these training OOVs were mostly named-entities since it was trained on BN data.

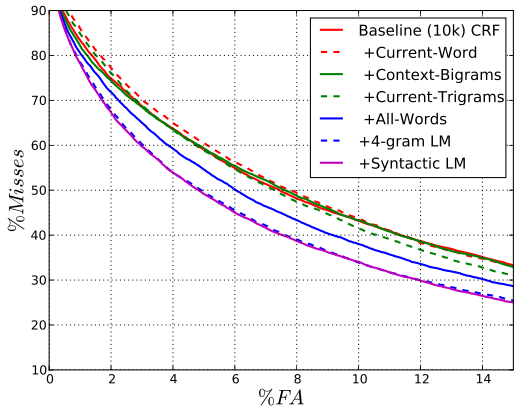
5.5 Improved Phonetic Transcription

We consider the hybrid lexicon's impact on Phone Error Rate (PER) with respect to the reference transcription. The reference phone sequence is obtained by doing *forced alignment* of the audio stream to the

[13] $WER = 32.7\%$ since the LVCSR system was trained on Broadcast News data as described in Section 4.

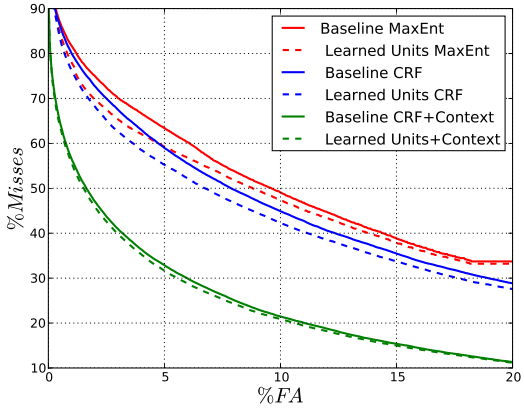


(a)

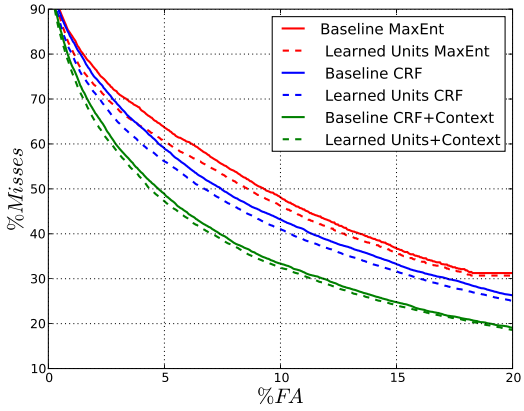


(b)

Figure 8: DET curves for OOV detection with features from local lexical and global context using baseline hybrid system. Evaluation on **All** OOVs (a) and **Unobserved** OOVs (b).



(a)



(b)

Figure 9: Comparing Baseline (Rastrow et al., 2009a) vs Learned Units (10K) with different context features (`OOVCORP`) (`+Syntactic LM`). Evaluation on **All** OOVs (a) and **Unobserved** OOVs (b).

reference transcripts using acoustic models. This provides an alignment of the pronunciation variant of each word in the reference and the recognizer's one-best output. The aligned words are converted to the phonetic representation using the dictionary.

Table 5 and 6 present PERs for the word and different hybrid systems in `OOVCORP` and MIT Lectures corpus respectively. As previously reported (Rastrow et al., 2009b), the hybrid systems achieve better PER, specially in OOV regions since they predict sub-word units for OOVs. Our method achieves modest improvements in PER compared to the hybrid baseline in `OOVCORP`[14]. It is also worth mentioning that the PER results show that the output of hybrid systems are richer and more useful for downstream applications such as Spoken Term Detection (STD).

6 Related Work

The proposed un-supervised segmentation approach we propose in this work was inspired by Poon et al. (2009). Their work presents a log-linear model for un-supervised morphological segmentation. The main differences between their model and the one proposed here are:

1. Their approach learns the joint probability

[14] Statistically significant at p=0.001 using the mapsswe test.

System	No. Subwords	OOV (%)	IV (%)	All (%)
Word	0	1.62	6.42	8.04
Baseline	5k	1.56	6.44	8.01
Baseline	10k	1.51	6.41	7.92
Learned Units	5k	1.52	6.42	7.94
Learned Units	10k	1.45	6.39	7.85

Table 5: Phone Error Rate results for OOVCORP.

System	No. Subwords	OOV (%)	IV (%)	All (%)
Word	0	1.50	18.26	19.76
Baseline	5k	1.47	18.36	19.84
Baseline	10k	1.47	18.35	19.82
Learned Units	5k	1.47	18.31	19.78
Learned Units	10k	1.45	18.31	19.76

Table 6: Phone Error Rate results for MIT Lectures.

$P(S, W)$ for a segmentation S and the text W, while ours takes into consideration the label sequence Y: $P(Y, S|W)$. This motivates the model to find segmentations predictive of class label Y, optimizing the segmentation for a particular labeling task;

2. We don't model dependencies in the text W, note that W is on the right hand side of the conditioning: $P(S, Y|W)$. This makes inference simpler.

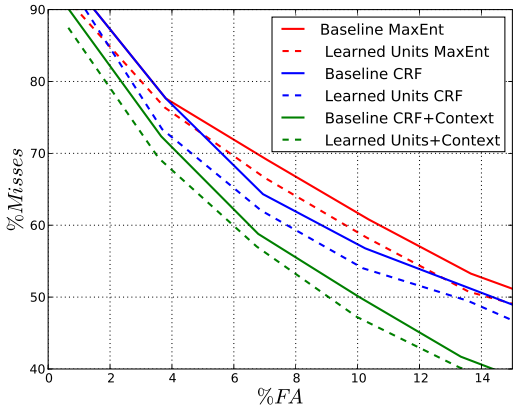
3. We propose an efficient inference procedure using Finite State Methods and the Metropolis Hastings algorithm; speeding sampling up to an order of magnitude;

4. The objective function in Poon et al. (2009) is different from ours, since we maximize the likelihood of the observed labeled sequence Y^*, while they maximize the likelihood of the text W.

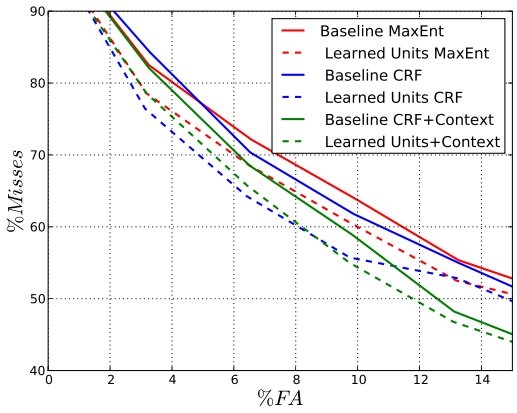
The authors in Creutz and Lagus (2002) also proposed an unsupervised segmentation approach for finding morphological units, and applied this model for speech recognition in morphologically rich languages. Their approach slightly simplified maximizes the posterior probability of the lexicon given the corpus: $P(\text{lexicon}|\text{corpus}) \propto P(\text{lexicon})P(\text{corpus}|\text{lexicon}) = \prod_{\text{letters } \alpha} P(\alpha) \cdot \prod_{\text{morphs } \mu} P(\mu)$, where letter and morph probabilities are maximum likelihood estimates. This approach differs from ours in many respects: 1) it is a generative model while ours is discriminative; 2) it does not take into consideration the context of units when deriving a segmentation; and 3) it does not model a labeling sequence Y, so the segmentation is not optimized for a given task.

7 Conclusion

Our probabilistic model learns sub-word units for hybrid speech recognizers by segmenting a text corpus while exploiting side information. The learned units improve detection of OOV regions on an English Broadcast News task, and an out-of-domain MIT Lectures data-set. Furthermore, we have confirmed previous work that hybrid systems achieve better phone accuracy, and our model makes modest improvements over a baseline with a similarly sized sub-word lexicon. Additionally, we used a simple but effective solution to speed inference using Metropolis-Hastings. This reduces the sampling time by an order of magnitude with no degradation in performance. We integrate the learned subwords with features from local and global contextual information in a CRF to detect OOV regions, improving over baseline sub-word selection methods. In the Broadcast News data set, at a 5% FA rate we reduce the missed OOV rate from 63.4% to 31.6%, a 31.8% absolute error reduction when eval-



(a) Baseline vs Learned Units (10K) evaluated on All OOVs



(b) Baseline vs Learned Units (10K) evaluated on Un-observed OOVs

Figure 10: Comparing Baseline (Rastrow et al., 2009a) vs Learned Units with different context features. Evaluated on MIT Lectures corpus.

uating all OOVs, and by 16.5% absolute when focusing on un-observed OOVs. Most of the gain was achieved by including contextual information in the OOV detector. On the MIT Lectures data set we achieved 7.1% and 12.2% absolute improvement on un-observed and All OOVs respectively. We found that while context helps, most of the gain in this out-of-domain data-set was achieved by integrating the proposed sub-words proposed. Context information might have a smaller effect in this data set given that it is a less-structured domain when compared to Broadcast News.

Acknowledgment

The authors would like to thank Bhuvana Ramabhadran and Denis Filimonov.

References

Issam Bazzi and James Glass. 2001. Learning units for domain-independent out-of-vocabulary word modelling. In *Eurospeech*.

M. Bisani and H. Ney. 2005. Open vocabulary speech recognition with flat hybrid models. In *INTERSPEECH*.

Dogan Can, Erica Cooper, Abhinav Sethy, Chris White, Bhuvana Ramabhadran, and Murat Saraclar. 2009. Effect of pronounciations on OOV queries in spoken term detection. In *ICASSP*.

M. Creutz and K. Lagus. 2002. Unsupervised discovery of morphemes. In *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*, pages 21–30.

Denis Filimonov and Mary Harper. 2009. A joint language model with fine-grain syntactic tags. In *EMNLP*.

Jonathan Fiscus, John Garofolo, Mark Przybocki, William Fisher, and David Pallett, 1998. *1997 English Broadcast News Speech (HUB4)*. Linguistic Data Consortium.

John Garofolo, Jonathan Fiscus, William Fisher, and David Pallett, 1996. *CSR-IV HUB4*. Linguistic Data Consortium.

James Glass, Timothy Hazen, Lee Hetherington, and Chao Wang. 2010. Analysis and processing of lecture audio data: Preliminary investigations. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *International Conference on Machine Learning (ICML)*.

Jonathan Mamou, Bhuvana Ramabhadran, and Olivier Siohan. 2007. Vocabulary independent spoken term detection. In *SIGIR*.

L. Mangu, E. Brill, and A. Stolcke. 1999. Finding consensus among words. In *Eurospeech*.

A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocky. 1997. The det curve in assessment of detection task performance. In *Eurospeech*.

Carolina Parada, Abhinav Sethy, and Bhuvana Ramabhadran. 2009. Query-by-example spoken term detection for OOV terms. In *ASRU*.

Carolina Parada, Mark Dredze, Denis Filimonov, and Frederick Jelinek. 2010. Contextual information improves oov detection in speech. In *NAACL-HLT*.

Carolina Parada, Mark Dredze, Abhinav Sethy, and Ariya Rastrow. 2011. Learning sub-word units for open vocabulary speech recognition. *ACL*.

Hoifung Poon, Colin Cherry, and Kristina Toutanova. 2009. Unsupervised morphological segmentation with log-linear models. In *NAACL '09*, pages 209–217.

Ariya Rastrow, Abhinav Sethy, and Bhuvana Ramabhadran. 2009a. A new method for OOV detection using hybrid word/fragment system. *ICASSP*.

Ariya Rastrow, Abhinav Sethy, Bhuvana Ramabhadran, and Fred Jelinek. 2009b. Towards using hybrid, word, and fragment units for vocabulary independent LVCSR systems. *INTERSPEECH*.

H. Soltau, G. Saon, and B. Kingsbury. 2010. The ibm attila speech recognition toolkit. *IEEE Workshop on Spoken Language Technology*.

Stanley F. Chen. 2003. Conditional and joint models for grapheme-to-phoneme conversion. In *Eurospeech*, pages 2033–2036.

Christopher White, Jasha Droppo, Alex Acero, and Julian Odell. 2007. Maximum entropy confidence estimation for speech recognition. In *ICASSP*.