CLSP Research Note No. 52

Acoustic and Language Modeling for Czech ASR in MALACH

Izhak Shafran
The Center for Language and Speech Processing (CLSP)
The Johns Hopkins University (JHU)
3400 N. Charles Street, Baltimore, MD 21218

Aug 25, 2006

1 Introduction

Automatic transcription of Czech testimonials in MALACH posses a unique problem and a new opportunity as documented extensively elsewhere [2]. Briefly, the problem of automatic transcription is made difficult because the speech is largely from older speakers, often speaking with heavy accent on an emotionally charged topic. The corpus, however, provides a challenge which reflects the complexities of transcribing real-world spoken documents in large number of languages.

The work performed on the MALACH project is described in three sections. The first section describes the development of an automatic speech recognition (ASR) system for transcribing the testimonies and the two releases of transcripts. The second and third sections delve into new models developed to correct systematic errors in ASR on a task using a generative and a discriminative framework respectively. These models are general enough that they can be applied to a number of tasks beyond MALACH. The third section points out the educational benefits from this NSF project.

2 Automatic Transcription

Automatic transcription of the speech was undertaken to provide access to the large collection of interviews in the MALACH corpus. The task consists of generating a word-level transcript of speech from both the interviewer and the interviewee. Additionally, time marks corresponding approximately to the beginning and the end of words are needed to locate the segment of interest to a user.

A state-of-the-art system was developed to transcribe the Czech testimonies automatically, which is described in 2.1. Initially, the system was designed to generate the spoken words verbatim. Czech suffers from diglossia in that written words could have a number of informal variants, which can confound a user who needs to formulate a query for an spoken document retrieval system. This concern was addressed by modifying the ASR system to produce the transcripts in the formal variant of the language. In other words, the ASR system was taxed with the burden of capturing all the variants of a written form that may be encountered in a spoken utterance. This unusual demand on the ASR system created a hitherto unforeseen side-effect on the discriminative training of acoustic models, which is described further in Section 2.2 along with various other design trade-offs.

2.1 System Description

The baseline ASR system uses perceptual linear prediction (PLP) features which was computed on 44KHz input speech at the rate of 10 frames per second, and was normalized to have zero mean and unit variance per speaker. The acoustic models were made of 3-state HMM triphones, whose observation distributions were clustered into about 4500 allophonic (triphone) states. Each state was modeled by a 16 component Gaussian mixture with diagonal covariances. The parameters of the acoustic models were initially estimated by maximum likelihood and then refined by five iterations of maximum mutual information estimation (MMI).

For the first pass decoding, a language model was created by interpolating the in-domain model (weight=0.75), estimated from 600k words of transcripts with an out-of-domain model, estimated from 15M words of Czech National Corpus [12]. Both models were parametrized by a trigram

language model with Katz back-off. The decoding graph was built by composing the language model, the lexical transducer and the context-dependent transducer (phones to triphones) into a single compact finite state machine.

The baseline ASR system decodes test utterance in two passes. A first pass decoding was performed with CML/MMIE acoustic models, whose output transcripts were bootstrapped to estimate two maximum likelihood linear regression transforms for each speaker using five iterations. A second pass decoding wass then performed with the new speaker adapted acoustic models.

The system was developed mostly using public domain software. The acoustic models were trained mostly using Hidden Markov Model Toolkit (HTK) from Cambridge University. The toolkit was augmented with a module to train acoustic model discriminatively using conditional maximum likelihood (CML), popularly also known as maximum mutual information (MMI), estimation. The CML/MMI training requires several lattice operations and our implementation exploits the modularity and efficiency afforded by the finite state machine (FSM) toolkit from AT&T [10]. The decision tree-based phone-to-allophone mapping, generated by the HTK software, were transformed into an equivalent context-dependency transducer. Decoding was performed using a compact static finite state transducer, generated from the context-dependent transducer, dictionary transducer and the language model acceptor as outlined in [11]

2.2 Trade-Offs in System Design

During the system development, a number of algorithmic choices were investigated to design an efficient and accurate system. A sample of such design choices are enumerated below.

1. *Does the bias in language model, due to the inclusion of transcripts of acoustic training data, hurt the performance of an acoustic model trained using CML/MMI criterion?*

   Several previous studies (e.g. [16, 14]) have observed that the decoding of the training data is significantly less biased when the reference transcripts are excluded from the language model, particularly, in tasks where the in-domain corpus is small, such as in MALACH. To test the impact of holding-out the transcripts from the language model, a 20-fold jack-knife procedure was evaluated in the CML/MMI training. As illustrated in Figure 1, a small performance improvement was observed, however, this was largely confined to the early iterations of MMI. Thus, the standard but counter-intuitive practice of including training utterances in LM does not appear to degrade the performance significantly.

2. *What is the optimal order of n-gram in CML/MMI training?*

   The optimal order of n-gram for generating the lattices and for estimating the model parameters needs to empirically determined for each task [18]. For MALACH, bigram and unigram language models were tested for CML/MMI estimation. Unigram language model provides an improvement over bigram models of about 0.1 to 0.2 % absolute gain. However, this comes with a 10% increase in computation cost, most of which is incurred in generating the denominator lattice.

3. *Is a presegmentation step necessary in transcribing an entire channel of speech from an interview tape?*

   In tasks such as broadcast news, the speech is first segmented into utterances before transcribing the utterances. This is necessary for discarding segments consisting entirely of

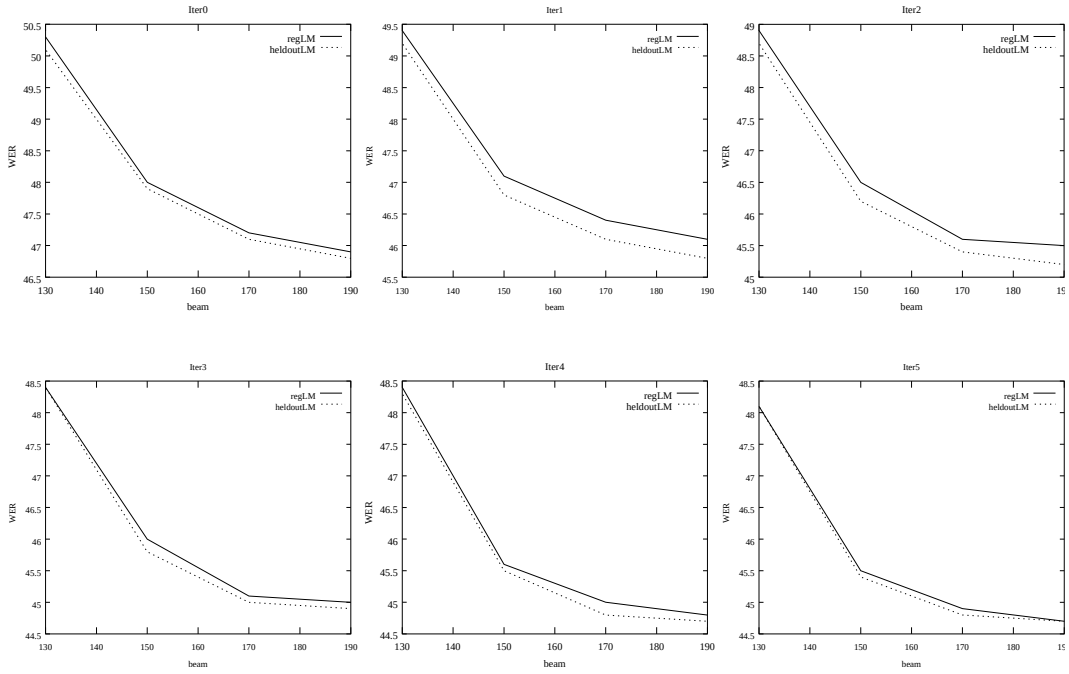

Figure 1: Comparison of ASR performance (WER) using MMI-trained acoustic models trained with standard LM and held-out LM.

intervening filler music or ads, as well as for grouping speakers appropriately while applying speaker adaptive transforms on the acoustic models. Such concerns are unlikely to pose as much problem in MALACH corpus and the speech is largely from one speaker in most tapes. Consequently, the transcription may not need presegmentation. This was investigated in a set of controlled experiments where the decoding was performed with manual segmentation (ManSeg) and without any segmentations (Std). Additionally, different settings were investigated for decoding without segmentation. In one instance, the language model was configured to allow the tape to be decoded as multiple utterances (ClosureLM), unlike the standard configuration where all of input speech was regarded as a single utterance. In another instance, the efficiency of the decoder was evaluated when partial hypotheses was continually generated during Viterbi at immortal nodes, nodes where only a lone word hypothesis survives the pruning (ImmortalVit). The Table 1 shows the average and worst case word error rate measured across speakers along with the real-time factor for decoding.

The word error rate increases only by about 2% when the manual segmentations are not used. Further, the different configurations of language model and search does not impact the word error rate significantly. However, when the input is decoded using a language model with closure, the number of hypotheses evaluated in the search increases significantly and consequently a higher computation cost is incurred. However, the resulting improvement in average word error rate is marginal at best and does not justify the additional computation cost.

Configuration	Average WER	Worst WER	Time
ManSeg	44.6	53.8	14.4
Std	46.8	57.7	14.7
ClosureLM	46.7	57.6	15.8
ImmoralVit	46.8	57.4	14.4

Table 1: Effect of decoding an entire interview tape with/without manual segmentations, and for different configurations of language model and search.

Most automatic segmentation system require carefully tuning of operating threshold, and wrong settings could cost significant increase in deletion or insertion, potentially much more than the 2% loss seen in these experiments. Based on these experiments, the automatic transcription of the interview tapes were performed without presegmentation using standard language model and search configurations.

4. *Is cepstral mean and variance normalization beneficial for decoding interviews?*

   Cepstral mean and variance normalization is often applied as a standard feature normalization procedure, without giving much thought to it. This normalization provides about 0.5% absolute gain on manually segmented test set in MALACH. However, cepstral variance normalization causes significant problem when applied on an entire interview, as experiments on MALACH revealed. The reason for this discrepancy is the mismatch in the training and testing conditions. For a successful flat start, the acoustic training data needs segments with low amounts of silence or non-speech events. However, while decoding entire interviews, a large number of frames contribute very little to the variance, and as a result the estimated variance is much smaller than without silences. Since this normalization effects every frame of the input speech, the ASR performance takes a big hit. The cepstral mean normalization does not suffer from this problem and can be used safely to obtain consistent gains.

5. *What are the trade-offs between multiple pronunciations and discriminative training?*

   As mentioned earlier, for languages that suffer diglossia such as Czech it may be necessary to produce ASR output in the formal form, thus the acoustics models need to represent the informal variants implicitly. This mapping can be most conveniently represented in the lexicon. However, depending on the extent of the diglossia, the resulting dictionary could potentially contain large number of pronunciation variants. For example, the Czech portion of MALACH contains a relatively high percentage of colloquial words – about 9% of the vocabulary and 7% of the tokens. When these informal forms are subsumed in the lexicon, common words contain several pronunciation variants, and a few have as many as 14 variants.

   CML/MMI-based training attempts to maximize the conditional probability of the reference transcripts given the lattice of possible hypotheses. When common words contain large number of pronunciation variants, there is a danger that the numerator and the denominator lattices start resembling each other at the phone and model level. This renders the CML/MMI estimation less effective as revealed in experiments on MALACH. An initial ASR system was developed to produce transcripts verbatim. This system contained very few multiple pronunciations and the CML/MMI estimation provided an absolute gain of 3.4%. The

estimation procedure converged in about three iterations. Subsequently, the ASR system was redesigned to generate formal variants. When the informal forms were included in the dictionary to allow most possible variants, the gains from CML/MMI diminished to 1.7% absolute and the estimation took at least six iterations to converge. Subsequently, a large number of pronunciations variants, about 7% in all, were pruned automatically based on their occurrence in training data. The resulting lexicon improved the gain from CML/MMI training to 2.6% absolute, but still required about six iterations to converge.

The gains observed from various system design trade-offs are summarized in Table 2. The results are measured on a 2-hour test sets with no speakers overlapping with the training data.

	System	1-best
(a)	ML + Bigram LM	42.8
(b)	(a) + CMN/CVN	42.5
(c)	ML + CMN/CVN + Trigram LM	40.3
(d)	ML + New Silence Phones	39.2
(e)	(d) + CML/MMI Estimation	36.7
(f)	(e) + MLLR-2Xfm	35.8

Table 2: The performance of the baseline ASR system is reported, showing the improvements from various stages of system development.

The initial system consisted of acoustic models trained with maximum likelihood estimation and bigram language model. Cepstral mean and variance normalization provided a gain of 0.5% gain as expected. Replacing the bigram language model with the interpolated trigram language model improved the performance by about 2.2% absolute. The standard HTK recipe suggests separating silences into two categories: a context-transparent short silence and an end-of-utterance long silence. Simplifying the representation of silence with three state HMM containing large number of mixture components improved performance by about 1% absolute. Subsequent CML/MMI training with a pruned dictionary reduced word error rate by about 2.5% absolute, and speaker adaptation using two transforms, estimated with six iterations, gave a further gain of about 1% absolute. The word error rate of the complete system is still significantly higher than that of other popular tasks such as Switchboard. The gap in performance largely reflects the difficulty of transcribing the peculiarities of speech from the elderly in MALACH corpus. The baseline CML/MMI estimated system was used for generating the final transcripts. The transcription accuracy was measured on a set of 10 complete interview tapes and was found to be about 60%.

3 Corrective Models using Minimum Bayes Risk

Often the output of ASR system contains systematic errors specific to a task and an ASR system. These errors are best modeled in a post-processing step, where the parameters of the models are entirely focused on correcting the errors rather than capturing the interaction with the acoustic and language modeling components. Two such corrective modeling strategies were investigated on MALACH task. The first approach models the word level confusions using a generative framework. The second approach uses a simple log-linear model which is trained discriminatively using a maximum entropy criterion and can incorporate a large variety of features including morphology

and acoustic cues. The following sections describe the crux of the two approaches and the details can be found in [16] and [15] respectively.

Briefly, the minimum Bayes-risk approach can be extended to model systematic errors specific to the task and the ASR system. The standard formulation uses an uninformative edit distance as a loss function. Instead, the errors are modeled as a noisy channel and the parameters of the new edit distance are learned from the data. The resulting loss function is used in the risk criterion for decoding. Experiments on MALACH demonstrate significant gains of about 1% absolute over MAP hypothesis and about 0.6% absolute over untrained loss function. The approach is general enough to be applicable to other sequence recognition problems such as in Optical Character Recognition (OCR) and in analysis of biological sequences.

In large vocabulary tasks, the word error rate of the maximum aposteriori (MAP) hypothesis ranges anywhere from 20-50% depending on the complexity of the task. Even at high word error rates, the list of most likely hypotheses often contains the spoken sentence, and the hypothesis closest to the spoken sentence (oracle) has word error rate as low as 5%. The large potential gain in WER has motivated several studies to address the problem of picking a hypothesis from a set of likely hypotheses with a word error rate lower than the MAP estimator.

The minimum Bayes-risk (MBR) decoding provides a framework to pick a hypothesis with lower WER than the MAP solution. The Bayes risk criterion is formulated as follows. For a given task, a cost $C(W, W')$ is assigned for picking a particular hypothesis W when the reference is known to be W'. During test, the total risk of picking a hypothesis W is the expected cost over all W', and is given by,

$$R(W) = \sum_{W}^{'} C(W, W')P(W'|O),$$

where $P(W'|O)$ is the posterior probability of W' given the observation sequence O. The optimal recognizer picks a hypothesis W for which the risk $R(W)$ is minimum. When all incorrect sentences are penalized equally, irrespective of the number of words in error, the optimal Bayes estimator reduces to the MAP estimator.

In ASR applications, the Bayes risk is computed over a set of most likely recognized word strings represented as an N-best list. So far, all studies have used string edit (Levenshtein) distance to compute the cost between two hypotheses strings [17, 5]. The Levenshtein distance, as described in the next section, assigns a fixed cost for substitution, deletion and insertion, irrespective of the identities of the words involved. Such a cost function may not be a good choice in several situations.

For example, consider the three hypotheses generated by an ASR system: (1) *Look who's here*, (2) *Book is here*, and (3) *Yeah right here*. For the sake of illustration, assume that they have equal posterior probabilities. This may happen in a noisy environment, where the first sentence may be confusable with the second, and the third may have a high bi-gram (*prior*) probability. The Levenshtein distance between any two pairs of hypotheses is the same, and therefore, the three strings are equally good candidates for minimum Bayes-risk decoding. If it is known a-priori (say, by examining the training data) that "Look" often gets misrecognized as "Book", and "who's" gets shortened as "is", then the first hypothesis is a "better" candidate. This notion can be incorporated into the risk minimization criteria through a learned edit distance, as described in the following section.

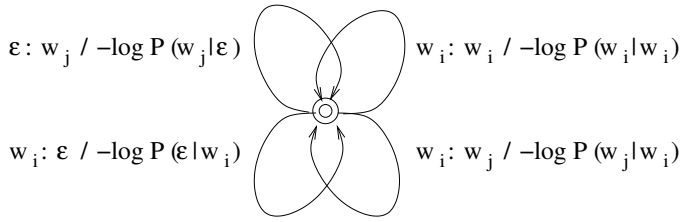

Figure 2: Probabilistic finite state transducer to model noisy channel.

3.1 Untrained Edit Distance

Let Σ be a finite set, a vocabulary, whose elements are words, and ϵ denote a null symbol. An elementary edit operation is a pair $(a, b) \in (\Sigma \cup \{\epsilon\}) \times (\Sigma \cup \{\epsilon\}) \backslash \{(\epsilon, \epsilon)\}$. For clarity, it is also denoted by $a \mapsto b$. An alignment A of two strings $s1$ and $s2$ is a sequence $(a_1 \mapsto b_1, \ldots, a_h \mapsto b_h)$ of edit operations such that $s1 = a_1, \ldots, a_h$ and $s2 = b_1, \ldots, b_h$, with possible insertions of ϵ. To each edit operation a weight function δ assigns a real number $\delta(a \mapsto b)$. The weight $\delta(A)$ of an alignment A is defined as $\delta(A) = \sum_{a \mapsto b \in A} \delta(a \mapsto b)$, and the edit distance of two strings is the minimum weight over all alignments of the string, $D_u(s1, s2) = \min_A \delta(A)$. Note that the alignment A associated with this minimum is invariant to scaling of δ by a constant, a property that will be used later in this work.

If $\delta(a \mapsto a) = 0,\ \delta(a \mapsto b) = 1, \forall a \neq b$, then δ is the unit weight function. Another popular weight function which is often used in scoring ASR output has the following weights: $\delta(a \mapsto a) = 0,\ \delta(a \mapsto \epsilon) = 3,\ \delta(\epsilon \mapsto b) = 3,\ \delta(a \mapsto b) = 4,\ \forall a \neq b$. In the rest of the paper, edit distance derived from such pre-defined costs will be referred to as the untrained edit distance.

3.2 Learned Edit Distance: Noisy Channel Model

Instead of assigning a pre-defined cost to edit operations, a stochastic model can be learned from the data.

To motivate this approach, consider the ASR system as a noisy channel. Let $X = x_1, x_2, \ldots, x_M$ be the spoken word sequence corresponding to the input waveform. This input word sequence enters the ASR channel and exits in the form of corrupted word sequence, $Y = y_1, y_2, \ldots, y_N$. The ASR errors could have multiple sources such as incorrect form of acoustic/language model, bad estimation of its parameters and the mismatch in cost functions. Now, the problem is to learn a model that maps the corrupted sequence to the actual spoken word sequence, $Y \mapsto X$.

The noisy channel can be modeled using a simple probabilistic finite state transducers or a Markov chain with no input memory [1]. For each word w_i in the vocabulary, four types of transitions are created, as shown in the Figure 2. They correspond to the transitions $(w_i \mapsto w_i),\ (w_i \mapsto w_j),\ (\epsilon \mapsto w_j)$ and $(w_i \mapsto \epsilon)$ with weights $P(w_i|w_i),\ P(w_j|w_i),\ P(\epsilon|w_j)$ and $P(w_i|\epsilon)$, respectively. The cost of elementary edit operation is assigned the log of these weights, as in [13]. As in edit distance, to enforce zero cost for a distortion-less channel, a normalization is applied to the model, $\delta(w_i \mapsto w_j) = -\log P(w_j|w_i) + \log P(w_i|w_i), \delta(w_i \mapsto w_i) = 0, \delta(w_i \mapsto \epsilon) = -\log P(\epsilon|w_i) + \log P(w_i|w_i)$. Since this is a simple Markov chain, the parameters can be estimated using Expectation Maximization or the corresponding Viterbi approximation.

In a practical ASR task, the vocabulary is large and the training data is limited, making

it difficult to estimate all the parameters robustly. In such a scenario, robust estimates can be obtained only for the most frequent words. For the rest, parameters need to be shared to obtain good estimates. An extreme scenario consists of grouping the transitions into four sets: $\{(a,a)\}$, $\{(a,b)\ \forall a{\neq}b\}$, $\{(\epsilon,b)\}$ and $\{(a,\epsilon)\}$. If the probabilities are denoted by P_c, P_s, P_i and P_d, respectively, then, the stochastic model corresponds to the edit distance with fixed costs, where $\delta(a{\mapsto}a) = 0$, $\delta(a{\mapsto}b) = -\log P_s + \log P_c$, and $\delta(a{\mapsto}\epsilon) = -\log P_d + \log P_c$. The alignment with minimum edit distance corresponds to the maximum likelihood alignment of the stochastic model. Thus, the Levenshtein distance can be seen as an instance of this noisy channel model. The complexity of the noisy channel model can be increased with the order of Markov chain.

3.3 Decoding with Learned Edit Distance

The noisy channel model captures task- and decoder-specific confusions, and this knowledge can be used to pick better candidate hypotheses. To do so, the Bayes risk criterion is computed using the learned edit distance, $D_l(W, W')$ and the hypothesis with least risk is picked.

Using the noisy channel model, the minimum Bayes-risk decoding can be seen as a variant of probabilistic decoding with unconstrained costs. In a probabilistic framework, the best hypothesis can be computed as, $\arg\max_W \sum_{W'} P(W|W')P(W'|O)$. Here, $P(W|W')$ replaces $C(W, W')$ of MBR and imposes additional probabilistic constraints on the costs, that the conditional probabilities for a given word W_i sum to one. The weights are normalized negative log probabilities and so are not treated as probabilities.

For any two hypotheses, the learned edit distance can be computed using the Viterbi algorithm. The edit distance model can be implemented as a finite state transducer T. To obtain the edit distance between two hypotheses $D_l(W, W')$ and to generate the best alignment, compose $WoToW'$ and compute the shortest path through the resultant transducer. These operations can be performed using publicly available AT&T's FSM toolkit [10]. The Bayes risk can then be computed over the set of all hypotheses to locate one with the least risk.

3.4 Parameter Estimation

To learn the parameters of the stochastic model of edit distance, the Viterbi approximation was used [13]. The edit distance model was initialized using uniform weights. The training procedure consists of decoding the training data, collecting the counts, and estimating the empirical edit costs.

First, the training data is decoded using conditions similar to the test setup. Following the observations in [14] about the need to exclude the contribution of the utterance being decoded from the language model, the training data was divided up into 20 sets. A set of language models were build holding one set out at a time. In all cases, the selectively sampled Czech National Corpus was also used appropriately. Each set was then decoded with a language model that excluded transcripts from that set. The same MMIE acoustic model was used for decoding all the training data. Thus, the MAP hypothesis was generated for each utterance in the training set.

For each utterance, the edit distance between the MAP hypothesis and the reference transcript is computed and the corresponding alignment is saved. The counts for all elementary edit operations were collected for the entire training data. Counts for words that occur fewer than 8 times were ignored, as is often done in language modeling. Subsequent experiments showed that the performance is not sensitive to this threshold. The cost of an elementary edit operation was

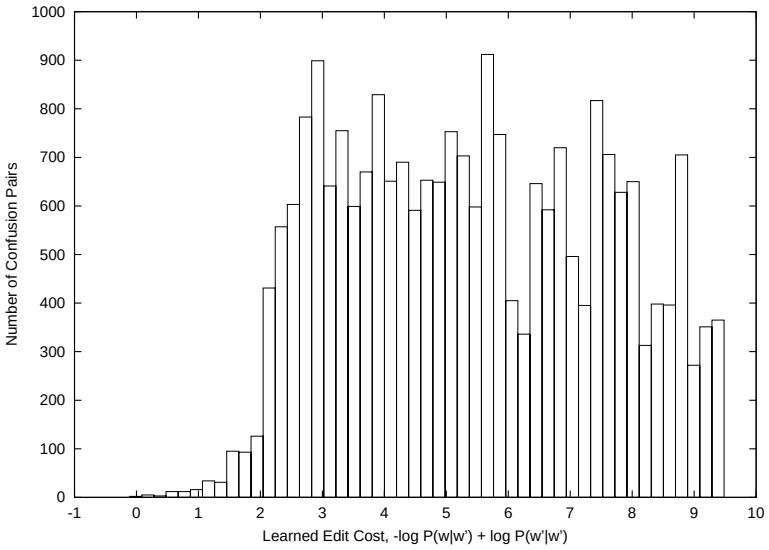

Figure 3: Histogram of costs of edit operations learned from the corpus.

estimated as in $\delta(w \mapsto w') = -\log P(w|w') + \log P(w'|w')$. All in all, about 19k edit costs were estimated from the training data. Most edit operations have a cost that occupy a range from zero to 9.5, as shown in Figure 3. In a few cases, the words preferred substitution or deletions more than identity, thus have a negative cost associated with them.

For edit operations that were not represented adequately in the training data, the standard fixed cost was assigned as a back-off. Since the best alignments from edit distance are scale invariant, the default costs were scaled so that they were slightly above the range of learned costs. Specifically, the back-off cost of substitution, deletion, and insertion were set to 9, 9 and 12, respectively.

3.5 Evaluation

The MBR decoding was applied on a N-best list of 50 and 200 hypotheses on the test set, which were generated using the MMIE models and without speaker adaptation. The likelihoods were squashed by dividing it with the grammar scale factor (14.0) and the posteriors were computed over the N-best. Words such as silence which do not get scored in the WER were removed. The results of MBR decoding were compared with the baseline MAP estimator, and are shown in Table 3.

On an N-best list of 200 hypotheses, the standard MBR with untrained edit distance improved performance by 0.4% over MAP. This decoder chose hypotheses other than the MAP hypotheses about 33% of the time. The MBR decoder with learned edit distance preferred such hypotheses

Decoder	Word Error Rate	
1. MAP	45.4	
	N=50	N=200
2. MBR with untrained edit distance	45.2	45.0
3. MBR with learned edit distance	44.5	44.4

Table 3: Comparison of MBR decoding using edit distance with fixed weights and weights learned from the task.

about twice the number of times (approx. 61%). A significant gain of 1% absolute was observed over MAP. The computational cost of MBR rises exponentially with number of hypotheses, increasing by about 16-fold from 50 to 200-hypotheses set. Using learned edit distance, much of the improvement can be obtained from the smaller 50-hypotheses set. This makes it possible to improve performance with MBR decoding even when the computational resources are significantly limited.

The minimum Bayes-risk decoding with learned edit distance can be applied to tasks other than speech recognition such as machine translation where Bayes-risk framework has been found to be useful [8]. One limitation of this approach, however, is that the model needs to parameterize the space of all possibilities and resorts to smoothing techniques to account for observations not seen in the training data. This poses a significant hurdle when the number of discrete input symbols increases by a magnitude, such as when modeling morphological features. To tackle such situations, an alternative discriminative model was investigated.

4 Discriminative Log-Linear Models

Briefly, a discriminative log-linear model picks a hypothesis with maximum score, where the score is computed over a range of features from the hypothesis. The features can be categorical or continuous and this provides an opportunity to model a variety of cues for correcting errors, including morphology and acoustics. As a first step, a corrective model consisting of morphological and word n-gram features was investigated. Experiments on the Czech portion of the MALACH corpus demonstrate performance gain of about 1.1–1.5% absolute in word error rate, wherein morphological features contribute about a third of the improvement. A simple feature selection mechanism based on χ^2 statistics is shown to be effective in reducing the number of features by about 70% without any loss in performance, making it feasible to explore yet larger feature spaces.

High inflection in a language is generally correlated with some level of word-order flexibility. Morphological features either directly identify or help disambiguate the syntactic participants of a sentence. Inflectional morphology works as a proxy for structured syntax in a language. Modeling morphological features in these languages not only provides an additional source of information but can also alleviate data sparsity problems.

Czech speech recognition needs to deal with two sources of errors which are absent in English, namely, the inflectional morphology and the differences in the formal (written) and colloquial (spoken) forms. Table 4 presents an example output of our speech recognizer on an utterance from a Holocaust survivor, who is recounting General Romel's desert campaign during the Second World War. In this example, the feminine past-tense form of the Czech verb for *to be* is chosen mistakenly, which is followed by a sequence of incorrect words chosen primarily to maintain agreement with

REF	no	Ježíš	to	už	byl	Romel	hnedle	před	Alexandrií
gloss	well	Jesus	by that time	already	was	Romel	just	in front of	Alexandria
translation	oh Jesus, Romel was already just in front of Alexandria by that time								
HYP	no	Ježíš	to	už	byla	sama	hned	lepší	Alexandrie
gloss	well	Jesus	by that time	already	(she) was	herself	just	better	Alexandria
translation	oh Jesus, she was herself just better Alexandria by that time								

Table 4: An example of the *grouping* effect. The incorrect form of the verb *to be* begins a group of incorrect words in the hypothesis, but these words agree in their morphological inflection.

the feminine form of the verb. When the acoustic model prefers a word with an incorrect inflection, the language model effectively propagates the error to later words. A language model based on word-forms prefers sequences observed in the limited training data, which will implicitly force an agreement with the inflections of preceding words, making it difficult to stop propagating errors. Morphological cues can correct for such errors and improve recognition of inflected languages in general.

The approach based on log-linear model differs from previous work such as [19] and [4]. By choosing a discriminative framework and maximum entropy based estimation, the model allows arbitrary features or constraints and their combinations without the need for explicit elaboration of the space and its backoff architecture. Thus, morphological features can be incorporated in the absence of knowledge about their inter-dependencies.

4.1 Inflectional Morphology

Inflectional abundance in a language generally corresponds to some flexibility in word order. In a free word-order language, the order of sentential participants is relatively unconstrained. This does not mean a speaker of the language can arbitrarily choose an order. Word-order choice may change the semantic and/or pragmatic interpretation of an utterance. Czech is known as a free word-order language allowing for subject, object, and verbal components to come in any order. Morphological inflection in these languages must include a syntactic *case* marker to allow the determination of which participants are subjects (nominative case), objects (accusative or dative) and other such entities. Additionally, morphological inflection encodes features such as gender and number. The agreement of these features between sentential components (adjectives with nouns, subjects with verbs, etc.) may further disambiguate the target of a modifier (e.g., identifying the noun that is modified by a particular adjective).

The increased flexibility in word order aggravates the data sparsity of standard n-gram language model for two reasons: first, the number of valid configurations of a group of words increases with the free order; and second, lexical items are decorated with the inflectional morphemes, multiplying the number of word-forms that appear.

In addition to modeling sequences of word-forms, the model incorporates sequences of morphologically reduced *lemmas*, sequence of morphological *tags* and sequences of various factored representations of the morphological tags, *factored tags*. Factoring a word into the semantics-bearing lemma and syntax-bearing morphological tag alleviates the data sparsity problem to some extent. However, the number of possible factorizations of n-grams is large. The approach adopted in this work is to provide a rich class of features and defer the modeling of their interaction to the learning procedure.

4.2 Extracting Morphological Features

The extraction of reliable morphological features critically effects further morphological modeling. The first step in the process is to select the most likely morphological analysis for each word using a morphological tagger. This is performed with the Czech feature-based tagger distributed with the Prague Dependency Treebank [6]. The tagger is based on a morphological analyzer which uses a lexicon and a rule-based tag guesser for words not found in the lexicon. Trained by the maximum entropy procedure, the tagger uses left and right contextual features from the input string. Currently, this is the best available Czech-language tagger. See [7] for further details on the tagger.

Label	Description	# Values
lemma	Reduced lexeme	$< \|vocab\|$
POS	Coarse part-of-speech	12
D-POS	Detailed part-of-speech	65
gen	Grammatical Gender	10
num	Grammatical Number	5
case	Grammatical Case	8

Table 5: Czech morphological features used in the current work. The # Values field indicates the size of the closed set of possible values. Not all values are used in the annotated data.

From the morphological features assigned by the tagger, only a subset is retained and the less reliable features, which are semantic in nature, are discarded. The basic morphological features used are detailed in Table 5. In the tag-based model, a string of 5 characters representing the 5 morphological fields is used as a unique identifier. The derived features include n-grams of POS, D-POS, gender (gen), number (num), and case features as well as their combinations.

POS, D-POS Captures the sub-categorization of the part-of-speech tags.

gen, num Captures complex gender-number agreement features.

num, case Captures number agreement between specific case markers.

POS, case Captures associated POS/Case features (e.g., adjectives associated with nominative elements).

The paired features allow for complex inflectional interactions and are less sparse than the composite 5-component morphological tags. Additionally, the morphologically reduced lemma and n-grams of lemmas are used as features in the models.

Word-form	to	období	bylo	poměrné	krátké
gloss	that	period	was	relatively	short
lemma	ten	období	být	poměrně	krátký
tag	PDNS1	NNNS1	VpNS-	Dg—	AAFS2

Table 6: A morphological analysis of Czech. This analyses was generated by the Hajič tagger.

Table 6 presents a morphological analysis of the Czech sentence *To období bylo poměrně krátké.* The encoded tags represent the first 5 fields of the Prague Dependency Treebank morphological encoding and correspond to the last 5 rows of Table 5. Features for this sentence include the word-form, lemma, and composite tag features as well as the components of each tag and the above mentioned concatenation of tag fields. Additionally, n-grams of each of these features are included. Bigram features extracted from an example sentence are illustrated in Table 7.

form	to	období	bylo	poměrné	krátké
	to_období	období_bylo	bylo_poměrné	poměrné_krátké	
lemma	ten	období	být	poměrně	krátký
	ten_období	období_být	být_poměrně	poměrně_krátký	
tag	PDNS1	NNNS1	VpNS-	Dg—	AAFS2
	PDNS1_NNNS1	NNNS1_VpNS-	VpNS-_Dg—	Dg—_AAFS2	
POS	P	N	V	D	A
	P_N	N_V	V_D	D_A	
...			...		
case	1	1	-	-	2
	1_1	1_-	-_0	-_2	
num/case	S1	S1	S-	–	S2
	S1_S1	S1_S-	S-_–	–_S2	
...			...		

Table 7: Examples of the n-grams extracted from the Czech sentence *To období bylo poměrně krátké.* A subset of the feature classes is presented here. The morphological feature values are those assigned by the Hajič tagger.

The following section describes how the features extracted above are modeled in a discriminative framework to reduce word error rate.

4.3 Corrective Model and Estimation

For incorporating morphological features, a log-linear model is adopted which is similar in form to the re-ranking framework of Charniak and Johnson [3]. The model scores each test hypothesis y using a linear function, $v_\theta(y)$, of features extracted from the hypothesis $f_j(y)$ and model parameters θ_j, i.e., $v_\theta(y) = \sum_j \theta_j f_j(y)$. The hypothesis with the highest score is then chosen as the output.

The model parameters, θ, are learned from a training set by maximum entropy estimation of the following conditional model:

$$\prod_s \sum_{y_i \in Y_s : g(y_i) = max_j\, g(y_j)} P_\theta(y_i|Y_s)$$

Here, $Y_s = \{y_j\}$ is the set of hypotheses for each training utterance s and the function g returns an extrinsic evaluation score, which in our case is the WER of the hypothesis. $P_\theta(y_i|Y_s)$ is modeled by a maximum entropy distribution of the form, $P_\theta(y_i|Y_s) = \exp v_\theta(y_i) / \sum_j \exp v_\theta(y_j)$. This choice simplifies the numerical estimation procedure since the gradient of the log-likelihood

with respect to a parameter, say θ_j, reduces to difference in expected counts of the associated feature, $E_\theta[f_j|Y_s] - E_\theta[f_j|y_i \in Y_s : g(y_i) = max_j g(y_j)]$. To allow good generalization properties, a Gaussian regularization term is also included in the cost function.

A set of hypotheses Y_s is generated for each training utterance using a baseline ASR system and a jack-knife procedure, as in section 3.

The model allows the exploration of a large feature space, including n-grams of words, morphological tags, and factored tags. In a large vocabulary system, this could be an enormous space. However, in a discriminative maximum entropy framework, only the observed features are considered. Among the observed features, those associated with words that are correct in all hypotheses do not provide any additional discrimination capability. Mathematically, the gradient of the log-likelihood with respect to the parameters of these features tends to zero and they may be discarded. Additionally, the parameters associated with features that are rarely observed in the training set are difficult to learn reliably and may be discarded.

To avoid redundant features, attention was focused on words which are frequently incorrect; this is the *error region* to be modeled. In the training utterance, the error regions of a hypothesis are identified using the alignment corresponding to the minimum edit distance from the reference, akin to computing word error rate. To mark all the error regions in an ASR lattice, the minimum edit distance alignment is obtained using equivalent finite state machine operations [9]. From amongst all the error regions in the training lattices, the most frequent 12k words in error are shortlisted. Features are computed in the corrective model only if they involve words for the shortlist. The parameters, θ, are estimated by numerical optimization as in [3].

4.4 Empirical Evaluation

The model was empirically evaluated on MALACH corpus. A portion of the training data containing speech from 44 speakers, about 21k words in all is treated as development set (dev). The test set (eval) consists of about 2 hours of speech from 10 new speakers and contains about 15k words.

A set of contrastive experiments was carried out to gauge the performance of the corrective models and the contribution of morphological features. For training the corrective models, 50 best hypotheses are generated for each utterance using the jack-knife procedure mentioned earlier. For each hypothesis, bigram and unigram features are computed which consist of word-forms, lemmas, morphologoical tags, factored morphological tags, and the likelihood from the baseline ASR system. For testing, the baseline ASR system is used to generate 1000 best hypotheses for each utterance. These are then evaluated using the corrective models and the best scored hypothesis is chosen as the output.

Table 8 summarizes the results on two test sets – the dev and the eval set. A corrective model with word bigram features improve the word error rate by about an absolute 1% over the baseline. Morphological features provide a further gain on both the test sets consistently.

The gains on the dev set are significant at the level of $p < 0.001$ for three standard NIST tests, namely, matched pair sentence segment, signed pair comparison, and Wilcoxon signed rank tests. For the smaller eval set the significant levels were lower for morphological features. The relative gains observed are consistent over a variety of conditions that we have tested including the ones reported below.

Features	Dev	Eval
Baseline	29.9	35.9
Word bigram	29.0	34.8
+ Morph bigram	28.7	34.4

Table 8: The word error rate of the corrective model is compared with that of the baseline ASR system, illustrating the improvement in performance with morphological features.

4.5 Analysis of Features

Given the improvements, two questions arise naturally. Can the number of features be reduced significantly and still retain most of the gains? Which features contribute most to the performance gain? Experiments with simple feature reduction using χ^2 statistics were performed to answer the first question and the results showed that at least 70% of the features can be removed without hurting the performance significantly.

An exact answer to the second question requires a large number of leave-one-out experiments. Instead, an approximate contribution was gauaged using the weights learned from the maximum entropy model when all the features compete with each other. The impact of feature classes can then be analyzed by zeroing the weights corresponding to all features from a particular class and evaluating the performance of the resulting model without re-estimation. Figure 4 illustrates the effectiveness of different features class. The y-axis shows the gain in F-score, which is monotonic with the word error rate, on the entire development dataset. In this analysis, the likelihood score from the baseline ASR system was omitted since the interest is in understanding the effectiveness of categorical features such as words, lemmas and tags.

The most independently influential feature class is the factored tag features. This corresponds with our belief that modeling morphological features requires detailed models of the morphology; in this model the composite morphological tag n-gram features (TNG) offer little contribution in the presence of the factored features.

The ranking of χ^2 statistics reveals a similar story. When features are ranked according to their χ^2 statistics, about 57% of the factored tag n-grams occur in the top 10% while only 7% of the word n-grams make it. The lemma and composite tag n-grams give about 6.2% and 19.2% respectively. Once again, the factored tag is the most influential feature class.

The large feature reduction allows exploration of yet larger feature space. In addition to the n-gram counts, the feature could potentially incorporate the component acoustic model scores. Thus, allow corrective model to be acoustic-sensitive and this extension is being investigated currently.

5 Conclusions

This report describes acoustic and language modeling for Czech ASR in MALACH. The techniques developed here are equally applicable to other spoken archives, in particularly, those with rich morphology. The report enumerates many of the technical challenges in building a large vocabulary ASR system for a real-world spoken archive. Further, two new techniques were investigated to correct systematic errors in ASR output and have wider applications. Each of them provided an improvement in ASR performance by about 1-1.5% absolute gain.

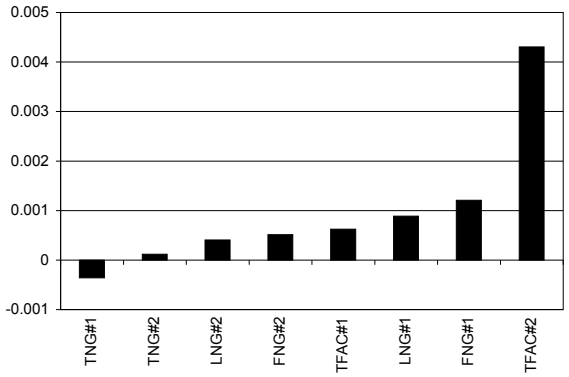

Figure 4: Analysis of features classes for a bigram form, lemma, tag, and factored tag model. Y-axis is the contribution of this feature if added to an otherwise complete model. Feature classes are labeled: TNG – tag n-gram, LNG – lemma n-gram, FNG – form n-gram and TFAC – factored tag n-grams. The number following the # represents the order of the n-gram.

6 Acknowledgments

The work reported here was performed in collaboration with several colleagues. In particular, section 4 represents joint work with Keith Hall. I thank William Byrne for sharing his technical insight all along this project. Peter Nemec and Vaclav provided valuable suggestions from a native speaker's perspective on correcting common errors in ASR output, which inspired the work in section 4. Vlasios Doumpiotis and Shankar Kumar provided baseline legacy systems for comparison. Thanks also to Joseph Psutka and colleagues for providing the transcripts and the Czech National Corpus, and Mehryar Mohri and colleagues for the AT&T FSM and GRM toolkits. This work was partly supported by NSF (USA) under the Information Technology Research (ITR) program by NSF IIS Award No. 0122466.

References

[1] L. R. Bahl and F. Jelinek. Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition. In *IEEE Trans. Info. Theory*, volume 21(4), pages 404–411, 1975.

[2] W. Byrne, D. Doermann, M. Franz, S. Gustman, J. Hajic, D. Oard, M. Picheny, J. Psutka, B. Ramabhadran, D. Soergel, T. Ward, and W.-J. Zhu. Automatic recognition of spontaneous speech for access to multilingual oral history archives. *IEEE Transactions on Speech and Audio Processing (Special Issue on Spontaneous Speech Processing)*, July 2004.

Center for Language and Speech Processing
Johns Hopkins University, Baltimore MD 21218

[3] Eugene Charniak and Mark Johnson. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, 2005.

[4] Ghinwa Choueiter, Daniel Povey, Stanley Chen, and Geoffrey Zweig. Morpheme-based language modeling for Arabic LVCSR. In *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, 2006.

[5] V. Goel and W. Byrne. Minimum Bayes-risk automatic speech recognition. In W. Chou and B.-H. Juang, editors, *Pattern Recognition in Speech and Language Processing*. CRC Press, 2003.

[6] Jan Hajič, Eva Hajičová, Petr Pajas, Jarmila Panevová, Petr Sgall, and Barbora Vidová Hladká. The prague dependency treebank 2.0, 2005. http://ufal.mff.cuni.cz/pdt2.0.

[7] Jan Hajič and Barbora Vidová-Hladká. Tagging inflective languages: Prediction of morphological categories for a rich, structured tagset. In *Proceedings of the COLING-ACL Conference*, pages 483–490, Montreal, Canada, 1998.

[8] S. Kumar and W. Byrne. Minimum Bayes-risk decoding for statistical machine translation. In *Proc. HLT-NAACL*, 2004.

[9] Mehryar Mohri. Edit-distance of weighted automata: General definitions and algorithms. *International Journal of Foundations of Computer Science*, 14(6):957–982, 2003.

[10] Mehryar Mohri, Fernando Pereira, and Michael Riley. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231(1):17–32, 2000.

[11] Mehryar Mohri, Fernando C. N. Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech and Language*, 16(1):69–88, 2002.

[12] Josef Psutka, Pavel Ircing, Josef V. Psutka, Vlasta Radovic, William Byrne, Jan Hajič, Jiri Mirovsky, and Samuel Gustman. Large vocabulary ASR for spontaneous Czech in the MALACH project. In *Proceedings of the 8th European Conference on Speech Communication and Technology*, Geneva, Switzerland, 2003.

[13] E. S. Ristad and P. N. Yianilos. Learning string-edit distance. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 20(5), pages 522–532, 1998.

[14] Brian Roark, Murat Saraclar, and Michael Collins. Corrective language modeling for large vocabulary ASR with the perceptron algorithm. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, 2004.

[15] I. Shafran and K. Hall. Corrective models for speech recognition of inflected languages. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia, July 2006.

[16] Izhak Shafran and William Byrne. Task-specific minimum Bayes-risk decoding using learned edit distance. In *Proceedings of Interspeech - International Conference on Speech and Language Processing*, pages 1945–48, Jeju Islands, Korea, 2004.

[17] A. Stolcke, Y. Konig, and M. Weintraub. Explicit word error minimization in n-best list rescoring. In *Proc. EUROSPEECH*, volume 1, pages 163–166, 1997.

[18] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young. MMIE training of large vocabulary speech recognition systems. *Speech Communication*, 22:303–314, 1997.

[19] Dimitra Vergyri, Katrin Kirchhoff, Kevin Duh, and Andreas Stolcke. Morphology-based language modeling for arabic speech recognition. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP/Interspeech 2004)*, 2004.