

Task-Specific Minimum Bayes-Risk Decoding using Learned Edit Distance

Izhak Shafran and William Byrne

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

{zakshafran,byrne}@jhu.edu

Abstract

This paper extends the minimum Bayes-risk framework to incorporate a loss function specific to the task and the ASR system. The errors are modeled as a noisy channel and the parameters are learned from the data. The resulting loss function is used in the risk criterion for decoding. Experiments on a large vocabulary conversational speech recognition system 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.

1. Introduction

The performance of an automatic speech recognition is measured using word error rate (WER). However, in most speech recognition systems, the best hypothesis is chosen using the maximum a-posteriori (MAP) estimator. Further, the MAP estimator is computed using empirically estimated component densities whose true forms are not known, and is given by,

$$\hat{W} = \arg \max_W P(W|O) = \arg \max_W P(O|W)P(W).$$

Here, W is a hypothesis, $P(O|W)$ is the acoustic score, and $P(W)$ is the prior likelihood as obtained from a language model. When the true distributions are known, improving the MAP estimate is guaranteed to improve the sentence error rate, but not the word error rate. This mismatch in cost function has been the subject of several studies [10, 9, 5, 1]. In large vocabulary tasks, the word error rate of the 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 of the above mentioned studies. They address the problem of picking a hypothesis from a set of likely hypotheses with a word error rate lower than the MAP estimator.

In [9], a perceptron based algorithm is trained to create an error-corrective language model whose feature space consists of N-grams observed in the training data. An iterative perceptron training algorithm is used to train the weights, and the resulting model can be used to re-score a lattice of hypotheses. The authors demonstrate performance gain on Switchboard corpus. In a different approach, [5] collapses the likely hypotheses into “sausages” by clustering with heuristics rules that utilize word identities and time boundaries. These “sausages” are then used to learn rules that help select the hypothesis with a low word error rate. They show that these rules improve performance and can reveal deficiencies of the ASR system.

The minimum Bayes-risk (MBR) decoding provides an alternate framework to address the mismatch between MAP and WER. 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 [10, 1]. 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, as-

sume 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. Subsequent section explains the experiments performed on a large vocabulary task and the results obtained from them. Finally, the contributions of the paper are summarized.

2. Bayes Risk With Learned Edit Distance

2.1. 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 $(a, b) \in (\Sigma \cup \{\epsilon\}) \times (\Sigma \cup \{\epsilon\}) \setminus \{(\epsilon, \epsilon)\}$. For clarity, it is also denoted by $a \rightarrow b$. An alignment A of two strings s_1 and s_2 is a sequence $(a_1 \mapsto b_1, \dots, a_h \mapsto b_h)$ of edit operations such that $s_1 = a_1 \dots a_h$ and $s_2 = b_1 \dots b_h$, with possible insertions of ϵ . To each edit operation a weight function δ assigns a real number $\delta(a \rightarrow b)$. The weight $\delta(A)$ of an alignment A is defined as $\delta(A) = \sum_{a \rightarrow b \in A} \delta(a \rightarrow b)$, and the edit distance of two strings is the minimum weight over all alignments of the string, $D_u(s_1, s_2) = \min_A \delta(A)$. 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 $\delta(a \rightarrow a) = 0$, $\delta(a \rightarrow b) = 1, \forall a \neq b$, then δ is the unit weight function. Another popular weight function which is often used in scoring ASR output has the following weights: $\delta(a \rightarrow a) = 0$, $\delta(a \rightarrow \epsilon) = 3$, $\delta(\epsilon \rightarrow b) = 3$, $\delta(a \rightarrow b) = 4, \forall a \neq b$. In the rest of the paper, edit distance derived from such pre-defined costs will be referred to as the untrained edit distance.

2.2. 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 $X = x_1, x_2, \dots, x_M$ 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, $Y = y_1, y_2, \dots, y_N$. 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

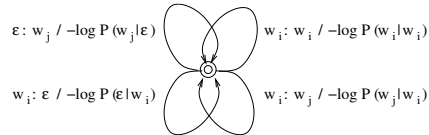


Figure 1: Probabilistic finite state transducer to model noisy channel.

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 [2]. For each word w_i in the vocabulary, four types of transitions are created, as shown in the Figure 1. 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. In this work, the cost of elementary edit operation is assigned the log of these weights, as in [8]. As in edit distance, to enforce zero cost for a distortionless 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 $\delta(a \rightarrow a) = 0$, $\delta(a \rightarrow b) = -\log P_s + \log P_c$, and $\delta(a \rightarrow \epsilon) = -\log P_d + \log P_c$. 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.

2.3. 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. In this work, 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 $W \circ T \circ W'$ and compute the shortest path through the resultant transducer. These operations can be performed using publicly available AT&T's FSM toolkit [6]. The Bayes risk can then be computed over the set of all hypotheses to locate one with the least risk.

3. Experiments

Experiments were performed on a large vocabulary corpus consisting of spontaneously spoken testimonies in Czech language, which is subset of the multilingual MALACH corpus [7].

3.1. Task

For acoustic modeling, the training set is about 84 hours of speech, comprising segments from 336 speakers (145 male and 191 females) and 552k spoken words from a vocabulary of 42k. Unlike other comparable corpora, this corpus contains a relatively high percentage of colloquial words – about 9% of vocabulary and 7% of tokens. The test set is about 2 hrs of speech, consisting of utterances from 5 male and 5 female speakers whose speech was excluded from the training set. It contains about 15k word tokens. Unlike other English corpora where OOV rates are of under 2%, the test set has a high OOV rate of about 6%.

Acoustic models for these experiments were made of 3-state HMM triphones, where each state is modeled by 16 component Gaussian mixture with diagonal covariances. A more complex 176-component HMM was used to model silence [7].

The language model was trained on two sets of transcripts. The first set is from the testimonies in the training set and is a relatively small set. The second set consists of portions from the Czech National Corpus which were selectively sampled to improve the language model. A bi-gram language model with Katz back-off was used for all the experiments. Further details of acoustic and language model can be found in [7]. The acoustic models from [7] were further improved using several iterations of Maximum Mutual Information Estimation (MMIE) to provide the baseline for the experiments reported in this paper.

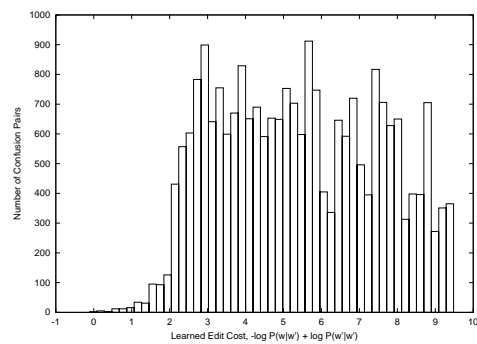


Figure 2: *Histogram of costs of edit operations learned from the corpus.*

3.2. Parameter Estimation

To learn the parameters of the stochastic model of edit distance, the Viterbi approximation was used [8]. 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 [9] 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 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 2. 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.

3.3. Testing

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 1.

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

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 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.

4. Conclusions

This paper extends the minimum Bayes-risk (MBR) framework to pick a hypothesis with word error rate lower than the MAP hypothesis. Systematic errors introduced by an ASR system are modeled as a noisy channel with no input memory. This was then incorporated into a learned edit distance and utilized in the Bayes risk criterion. The experiments on large vocabulary task demonstrates significant gains of about 0.6% and 1% over standard MBR and MAP decoding respectively. Further, experiments show that the most of the gain can be obtained even with set of hypotheses as small as 50. This makes it possible to improve ASR performance even when the computational power is limited and evaluating large sets

of hypotheses is not affordable.

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 [3].

5. Acknowledgments

The authors would like to thank Joseph Psutka and colleagues for providing the acoustic and language models for the task, and Mehryar Mohri and colleagues for the use of AT&T FSM and GRM toolkits.

6. References

- [1] V. Goel, and W. Byrne, "Minimum Bayes-risk automatic speech recognition", Pattern Recognition in Speech and Language Processing, W. Chou, and B.-H. Juang, (editors), CRC Press, 2003.
- [2] L. R. Bahl, and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition", 21(4):404-411, 1975.
- [3] S. Kumar, and W. Byrne, "Minimum Bayes-risk decoding for statistical machine translation", Proc. HLT-NAACL, 2004.
- [4] V. Levenshtein, "Binary codes capable of correcting spurious insertions and deletions of ones", Problems of Information Transmission, 1:8-17, 1965.
- [5] L. Mangu, and M. Padmanabhan, "Error corrective mechanisms for speech recognition", Proc. ICASSP, 2001.
- [6] M. Mohri, F. C. N. Pereira, and M. Riley, "The design principles of a weighted finite-state transducer library", Theoretical Computer Science, 231:17-32, 2000, <http://www.research.att.com/sw/tools/fsfm>.
- [7] J. Psutka, P. Ircing, J. V. Psutka, V. Radovic, W. Byrne, J. Hajic, J. Mirovsky, and S. Gustman, "Large vocabulary ASR for spontaneous Czech in the MALACH project", Proc. EUROSPEECH, 2003.
- [8] E. S. Ristad, and P. N. Yianilos, "Learning string-edit distance", IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(5):522-532, 1998.
- [9] B. Roark, M. Saraclar, and M. Collins, "Corrective language modeling for large vocabulary ASR with the perceptron algorithm". To appear in Proc. ICASSP, 2004.
- [10] A. Stolcke, Y. Konig, and M. Weintraub, "Explicit word error minimization in N-best list rescoring", Proc. EUROSPEECH, 1:163-166, 1997.