A Real-Time Japanese Broadcast News Closed-Captioning System

Olivier Siohan[†] *Akio Ando*[‡] *Mohamed Afify*[†] *Hui Jiang*[†]
Chin-Hui Lee[†] *Qi Li*[†] *Feng Liu*[†] *Kazuo Onoe*[‡] *Frank K. Soong*[†] *Qiru Zhou*[†]

[†]Bell Labs – Lucent Technologies
600 Mountain Ave, Murray Hill, NJ 07974, USA

[‡]NHK, Science and Technical Research Laboratories
1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, JAPAN

siohan@research.bell-labs.com ando@strl.nhk.or.jp

Abstract

This paper describes Bell Labs and NHK (Japan Broadcasting Corp.) collaboration to develop a real time large vocabulary speech recognition system for live closed-captioning of NHK news programs. Bell Labs broadcast news recognition engine consists of a two pass decoder using bigram language models (LM) and right biphone models during the first pass, and trigram LM with within-word triphone models in the second pass. Various pruning strategies are used to achieve real time decoding, together with a noise compensation procedure aimed at improving recognition on noisy segments of the program. The system operates in real-time mode and delivers less than 2% of word error rate (WER) on studio news conditions and about 5% of WER on noisy news and reporter speech when evaluated on a real broadcast news program.

1. Introduction

Developing a broadcast news (BN) transcription system for live subtitling is a challenging problem since it requires high recognition accuracy with real-time performance for a large variety of acoustic environments, ranging from high quality anchor speech in studio condition to reporters speech in the field. Since March 2000, NHK (Japan Broadcasting Corp.) has been providing live closed-captioning of its broadcast news programs using a real-time speech recognizer followed by manual correction of recognition errors [1]. While current automatic speech recognition (ASR) technologies can provide high recognition accuracy on anchor speakers in studio condition, the inherent variability of broadcast news programs leads to poor ASR performance on some portions of the broadcast. In an attempt to improve core BN technology, NHK Labs and Bell Labs have teamed up and this paper presents some of our recent results using Bell Labs' Japanese recognition engine.

In section 2, we present Bell Labs' core 2-pass decoder and outline a new noise compensation procedure that significantly improves recognition results on noisy speech [2]. Experimental results on an NHK's broadcast news database are described in section 3, followed by some conclusions.

2. Bell Labs' Japanese Broadcast News Transcription System

2.1. System Overview

To satisfy both real-time decoding and high recognition accuracy requirements, we have developed a 2-pass decoder for broadcast news applications. Broadly speaking a 2-pass search consists of a very fast first pass which limits the potential search space to a set of most likely candidates which are rescored in the second pass by more detailed models. The success of the first pass lies in its ability to retain the most likely word sequence, hence not resulting in any search error. The basic idea is to use a simple search strategy and/or simple acoustic and language models. The first pass of our 2-pass decoder is based on the following:

- A single static tree search structure. This is in contrast to most tree search in speech recognition where multiple tree copies depending on the language model history are used. This single tree structure is also used by BBN [3] and NHK [1] systems and results in a very efficient first pass.
- Left biphone acoustic models trained using a decision tree state tying algorithm [4]. These models take advantage of context while still maintaining efficiency. This can be compared to averaging the triphones in the BBN system[1], and using full triphones in the NHK recognizer.
- Bigram language models. Bigram probabilities are distributed among nodes in the tree to benefit as

[1]Essentially leading to models very similar to monophones

early as possible from the language model. This is the traditional language model lookahead used in many systems, which is efficiently implemented using caching.

Based on the above ingredients the first pass uses a classical Viterbi beam search, where at each time instant the top K candidates are propagated back into the root of the tree. The list of the best word candidates at each time instant are kept for later use in the backward pass. To further improve the efficiency of the first pass, three additional types of pruning are used:

- Word end pruning, where a tighter beam is used at the word ends.
- Histogram pruning, where only a specified maximum number of arcs are retained at each time. Interestingly, this type of pruning was found particularly useful in noisy speech decoding.
- Phoneme lookahead based pruning. This is based on phoneme pruning using a normalized local acoustic score as discussed in [5].

After using the above techniques it was observed that the computational effort is mostly spent in the likelihood evaluation. This also was observed by others and fast likelihood computation techniques are usually called for to speed up the system. We use a vector quantization (VQ) Gaussian selection scheme to reduce the effort spent in Gaussian calculation [6]. The combination of all of the above techniques result in a very fast and efficient first pass which runs at less than $0.5\times$RT for clean speech, and contains the correct word candidates about 99% of the time.

The forward pass results in a set of likely words ending at each time frame. This pass is followed by a backward pass whose search space is constrained by the results of the first pass. The backward pass uses the following models:

- Within word triphone models. In an earlier version of our system, right biphone models were used in the backward pass but significant improvement have been obtained when using within word triphone models. One the other hand, we have found that using cross-word triphones leads to significant increase in the search space and computation time.
- A backward trigram language model.

The backward search is again a Viterbi beam search employing word-end (in this case word-begin) pruning, histogram pruning, and fast likelihood computation techniques. To make the pruning and hence the search even more efficient the following techniques are used:

- Forward-backward pruning. Here, the score from the forward pass is kept and combined with the score from the second pass. This allows the beam width to be drastically reduced.
- Grammar spreading. To avoid the abrupt change in score when applying the language model at word ends (again word-begin in the backward case), the language model score is distributed uniformly over the word states and hence incrementally applied.

The combination of the above techniques leads to a very efficient backward pass, and hence overall search. For example, the 2-pass decoder runs at $0.7\times$RT for clean speech and results in almost the same accuracy as our 1-pass decoder [7].

2.2. Noise Compensation

To improve recognition accuracy on noisy data, we have developed a new noise compensation approach that operates in the feature space in a completely unsupervised manner [2]. The noise compensation takes as input an utterance in the cepstral domain as well as a Gaussian mixture model representing the distribution of clean speech in the log-spectral domain, and generates as output the cepstrum sequence of the compensated utterance where the noise contribution has been attenuated.

The compensation approach is designed to compensate the effect of an additive noise. The basic idea is to derive an estimate of the noisy speech log-spectrum probability density function (pdf) given a test utterance. Then, given the pdf of the clean speech (which can be derived on the training data), a MMSE estimate of the compensated (clean) speech is derived in the log-spectral domain and is mapped back into the cepstrum domain. This compensated cepstrum is then used for recognition.

Since the noise is supposed to be additive in the spectral domain, the relation between noise, clean speech and noisy speech is non-linear in the log spectral domain and can be represented by the function $z = f(x,n)$ where z, x and n are the noisy speech, clean speech and noise log-spectrum, respectively. Let us assume that the pdf $p(x)$ of the clean speech has been derived from the clean training data. We will assume that $p(x)$ can be represented using a Gaussian mixture model. The compensation process can then be described as follows:

1. Represent $f(x,n)$ by its m-th order Vector Taylor Series (VTS), $f^m(x,n)$.
2. Approximate the VTS by a linear function, $g^m(x,n) = A^m x + B^m n + C^m$ and derive the MMSE estimates for A^m, B^m and C^m. These 2 steps are required to get a linear relation between z, x and n, so that it becomes possible to express the pdf of z as a function of x and n pdfs'.

3. Initialize the noise pdf $p(n)$ using the first N frames of the test utterance. Typically we use $N = 10$ and assume that $p(n)$ can be represented using a Gaussian distribution or a mixture of Gaussian distribution.

4. Derive $p(z)$ given $p(x)$, $p(y)$, and the linear approximation function $g^m(x,n)$.

5. For each test utterance, refine the estimate of μ_n using EM and derive $p(z)$ (using batch or sequential estimation).

6. Derive an MMSE estimate of $\hat{x}$ given z, $p(x)$ and $p(z)$: $\hat{x} = E\{x|z\}$.

7. Map $\hat{x}$ into the cepstrum domain.

A nice property of this algorithm is that the noise pdf estimation (step 5) can be performed either in batch or online mode, and is therefore suitable for real-time applications. In [2], we have shown that the noise estimation can be done sequentially, allowing the noise compensation to track slowly varying noises thanks to an optimal forgetting factor. For more details, the reader is referred to [2].

3. Experimental Results

3.1. Database description

The training data consists of about 80 hours of broadcast news recorded by NHK between April and July 2000. About 1/5th of the data come from anchor speakers, the remaining data coming from reporters. Both left-biphone and triphone models are trained using a decision tree state tying algorithm, leading to a total of about 20K and 90K Gaussian probability density functions for the biphone and triphone models, respectively.

The test data consists of NHK's typical news programs recorded from June 1st, to June 7th, for a total of about 3 hours of speech material. The test data is excluded from the training material. To study the performance of the system based on the input speech characteristics, the test data has been subdivided into 8 different categories to reflect speaker and acoustic environment, as indicated in Table 1. In this evaluation, the same language model is used for every category, but a different LM is used each day, built according to the approach described in [8] to emphasize recent news stories. As a result, the perplexity of the test data varies widely across category, ranging from less than 10 on studio_news to about 60 on sports.noisy, weather and spontaneous. The decoder uses a lexicon of about 20K Japanese words (morphemes), leading to an out-of-vocabulary rate ranging from about 0.5% to 5% based on the test category.

The feature vector consists of 39 components, including 12 MFCC coefficients plus energy with their first and second derivative. The first cepstral coefficient C_0 is used as energy coefficient since it is required by our noise compensation algorithm.

Condition	Description
noisy_news	Noisy news programs
field_report	Reporters in field environment
sports.noisy	Sports news in noisy environment
sports.clean	Sports news in clean environment
weather	Weather reports
studio_report	Reporters in studio environment
spontaneous	Spontaneous speech
studio_news	Anchor speakers in studio environment

Table 1: List of test data categories.

3.2. Recognition Results

The performance of the system is reported in terms of word accuracy and real time factor. System evaluations have been carried out on a high end PC (quad 700Mhz Xeon processors, 2MB L2 cache, 3GB SD100 RAM) running Linux with only one recognition process running on a single CPU.

Our noise compensation algorithm has been initially validated using our 1-pass recognition engine, which uses the same triphone acoustic models and trigram LM as our 2-pass decoder. Table 2 represents the average word accuracy for each broadcast category with and without noise compensation. It can be seen that the noise compensation can lead to significant improvement on the noisy speech, especially on the sports.noisy category where the error rate has been reduced by more than 7% absolute. Reduction of word error rates are observed on all noisy categories, while some minor degradation in performance is sometimes observed for clean speech recognition. Considering that the benefits of the compensation clearly overcome its drawbacks, we now use the noise compensation as a standard feature of our system. We should also point out that the feature compensation algorithm has been applied only during testing, the acoustic models being trained on the original uncompensated cepstrum.

When using our 2-pass decoder, real-time performance is obtained while still preserving a high recognition accuracy, as illustrated in Table 3. In our experiments, we also observed that one additional advantage of using the noise compensation algorithm is that it also leads to some speed-up in decoding, especially on the noisy data. Overall, every category with a word error rate of about 5% is suitable for live closed-captioning.

Environment	Noise Comp.	
	On	Off
field_report	94.5	93.8
noisy_news	96.1	95.7
spontaneous	80.0	80.8
sports.clean	91.2	91.2
sports.noisy	81.3	74.0
studio_news	98.8	98.9
studio_report	89.8	90.1
weather	89.0	89.3

Table 2: Word Accuracy (%) using Bell Labs' 1-pass decoder with and without noise compensation.

Environment	Acc.	RT
field_report	93.9	1.1
noisy_news	96.3	1.1
spontaneous	78.3	1.2
sports.clean	88.7	0.8
sports.noisy	78.0	1.6
studio_news	98.4	0.6
studio_report	90.8	0.9
weather	82.8	1.2

Table 3: Word Accuracy (%) and real time factor (RT) for Bell Labs' 2-pass decoder with noise compensation.

4. Conclusion

In this paper, we present our 2-pass decoder for broadcast news applications and show that we are able to obtain real-time decoding with moderate degradation in performance compared to our high accuracy 1-pass system.

Experimental results indicate that our noise compensation algorithm is able to improve the word accuracy on average and leads to significant reduction of the real time factor. On field_report, noisy_news and studio_news conditions, the performance is high enough to allow live close-captioning of NHK's broadcast news with minimal manual correction of recognition errors. Additional work is required on the other categories before the system can be used for live subtitling.

5. References

[1] T. Imai, A. Kobayashi, S. Sato, H. Tanaka, and A. Ando, "Progressive 2-pass decoder for real-time broadcast news captioning," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2000.

[2] M. Afify and O. Siohan, "Sequential noise estimation with optimal forgetting for robust speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, Utah, USA, May 2001, ICASSP'01.

[3] L. Nguyen and R. Schwartz, "Single-tree method for grammar-directed search," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, USA, 1999.

[4] W. Reichl and W. Chou, "Robust decision tree state tying for continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, sep 2000.

[5] F. Liu, M. Afify, H. Jiang, and O. Siohan, "Incremental fast match with normalized scores for large vocabulary speech recognition," To be submitted to Eurospeech'01.

[6] S. Herman and R. Sukkar, "Variable threshold vector quantization for reduced continuous density likelihood computation in speech recognition," in *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 1997, pp. 331–338.

[7] Q. Zhou and W. Chou, "An approach to continuous speech recognition based on self-adjusting decoding graph," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Munich, Germany, 1997, pp. 1779–1782.

[8] A. Kobayashi, K. Onoe, T. Imai, and A. Ando, "Time dependent language model for broadcast news transcription and its post-correction," in *Proc. Int. Conf. on Spoken Language Processing*, Sydney, Australia, 1998, pp. 2435–2438.

[9] P. Q. Li, F. K. Soong, and O. Siohan, "A high-performance auditory feature for automatic speech recognition," in *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, 2000.