Investigation on Mandarin Broadcast News Speech Recognition

Mei-Yuh Hwang[1], *Xin Lei*[1], *Wen Wang*[2], *Takahiro Shinozaki*[1]

[1]Univ. of Washington, Dept. of Electrical Engineering, Seattle, WA 98195 USA
[2]SRI International, Menlo Park, 94025 USA

{mhwang,leixin,staka}@ee.washington.edu, wwang@speech.sri.com

Abstract

This paper describes our efforts in building a competitive Mandarin broadcast news speech recognizer. We successfully incorporated the most popular speech technologies into our system. More importantly, we present two novel algorithms in smoothing pitch features and segmenting Chinese characters into word units. Additionally, we propose to borrow the principle of pointwise mutual information for creating a Chinese word lexicon automatically. Our final system achieved 6.0% character error rate (CER) on dev04 and 16.0% on eval04, with simpler acoustic models, less training data, and simpler decoding architecture compared with other state-of-the-art systems, yet was equally competitive.

Index Terms: Mandarin speech recognition, character error rate, pitch smoothing, word segmentation.

1. Introduction

Due to economic and national security reasons, automatic speech recognition for Arabic and Mandarin languages has drawn great attention lately, particularly for broadcast news and broadcast conversational speech. This paper focuses on our efforts to build and improve our Mandarin broadcast news speech recognition system.

The organization of this paper starts from language modeling, where an n-gram based maximum likelihood (ML) word segmentation algorithm is presented. We argue that it generates more meaningful segmentation, which will benefit machine translation, than the blind longest-first match algorithm. Next we explain our acoustic feature representation, in particular on the use of improved smoothing and normalization of pitch features. We then build our system with the above algorithms and incorporate popular speech technologies. Section 4.2 shows the contributions of major acoustic components on benchmark test sets. Finally we outline our future work to further advance our system.

2. Language Modeling

2.1. Training Text and Preprocessing

We used several corpora for training our language models (LMs): the HUB4 1997 Mandarin broadcast news acoustic transcripts (Hub4), the LDC Chinese TDT2, TDT3, TDT4, Multiple-Translation Chinese Corpus (MTC) part 1, 2, and 3, and Mandarin Gigaword corpus. Due to limits of machine memory for LM training, we only used a portion of the Mandarin Gigaword corpus: all of the materials from XIN, ZBN, and CNA, spanning from the years of 2001 to 2004. We sampled a small heldout set, lmdev-06 (about 60K characters), from TDT4 and some broadcast conversations from GALE collection as the LM development set. lmdev-06 and all text from November 2003 and April 2004 were excluded from LM training. This gave us about 420M words of training text.

Before training an LM, we first performed text normalization on the Chinese text data to remove HTML tags, get rid of phrases with bad or corrupted codes, convert numbers into their verbalized forms in Chinese, and delete punctuations. Word fragments and background noise transcriptions were mapped to a special garbage word, and laughters to a laughter word. Both the garbage word and the laughter word were treated as lexical words, and therefore their n-grams would be trained.

2.2. Word Segmentation

Since Chinese characters are written without space, word segmentation needs to be performed after text normalization. Our word segmentation algorithm can be summarized as follows:

1. Create an initial lexicon of words with the following greedy merge algorithm:
   (a) Start from a lexicon where all words are single-character.
   (b) Compute the pointwise mutual information [1] for every pair of words in the current lexicon:

$$\text{PMI}(w_1, w_2) \quad = \quad \log \frac{p(w_1 w_2)}{p(w_1)p(w_2)}$$

where $p(w_1 w_2)$ is the probability that w_1 is followed by w_2.

(c) Choose the pair with the maximum PMI and merge the two words into a new longer word. Add the new longer word into the lexicon.

(d) Go to Step (b) to re-compute PMI, until a certain threshold is met.

Due to time constraint, we adopted an initial lexicon with phonetic pronunciations from BBN Technologies. In the future, we would like to study the effectiveness of the PMI based auto lexicon as it is valuable when we extend our work to other Asian languages.

2. Perform longest-first match for word segmentation on all training text, using the the above word lexicon.

3. Train a close-vocabulary n-gram LM for the most frequent V words. Unselected words are mapped to the garbage word.

4. With the above n-gram, do a second iteration word segmentation by searching for the segmentation with the maximum n-gram probability.

The longest-first match is a blind match, which can result in non-logical segmentation as the following example shows:

民进党/和亲/民党...
(The Green Party made peace with the Min Party via marriage...)

A more informed segmentation is to search for the ML segmentation path if a word-based n-gram LM is available. Particularly in the above example, the segmentation was fixed correctly in our experiment:

民进党/和/亲民党...
(The Green Party and the QinMin Party...)

Although the better segmentation may not necessarily imply significantly better recognition accuracy, it can be crucial for machine translation. Furthermore, to compare the perplexity of different word-based Chinese LMs (with the same lexicon), we should compute the lowest perplexity among all possible word segmentations on any test data, given each LM. Therefore, we advocate the n-gram ML segmenter. To avoid being trapped at local optimum, it is better to train the n-gram LM at Step 3 discriminatively [2] or to use a lower-order n-gram (such as unigram) at the second iteration of segmentation.

2.3. Compact Bigram and Large Four-gram

After the second word segmentation, we chose the most frequent 49K words as our decoding vocabulary, which included several dozen of English words. We trained 8 separate 4-gram LMs using the SRILM toolkit [3] with Kneser-Ney smoothing, for Hub4, TDT2, TDT3, TDT4, MTC123, Gigaword-XIN, Gigaword-ZBN, and Gigaword-CNA respectively. We optimized the interpolation weights of these LMs to maximize the likelihood of lmdev-06 given the interpolated LM. Our final 4-gram LM included 20M bigrams, 74M trigrams, and 57M 4-grams. Section 4.1 explains our decoding architecture. For the first pass fast decoding, we pooled together about half of the 4-gram LM training data to create a compact bigram LM, with 15M bigram entries. The perplexities on lmdev-06 using the 2-gram vs. 4-gram LMs are 495 vs. 288. The big 4-gram LM achieved almost 42% of perplexity reduction.

3. Acoustic Modeling

3.1. Acoustic Training and Test Data

The acoustic training data included Hub4 and the CCTV and VOA programs of the TDT4 corpus[1]. The TDT4 data comes with closed caption, but no accurate transcription. Therefore we used the flexible alignment algorithm described in [4] to select the segments with high confidence in the closed caption. There were in total about 97 hours of acoustic data after selection.

The Mandarin RT-04 development data, dev04, was used for system development. It consists of half an hour of CCTV broadcast news programs from November 2003. After system parameters were tuned, we then applied them to the RT-04 evaluation set, eval04, which includes one hour of broadcast news from CCTV, RFA, and NTDTV in April 2004. Due to these two test sets, all text from November 2003 and April 2004 were excluded during LM training.

3.2. Acoustic Feature Extraction and Pitch Processing

We used the 12th order mel-scale cepstrum coefficients (MFCC) to do automatic speaker clustering and compute vocal track length normalization warping for all *auto* speakers, both based on Gaussian mixture models.

F_0 was extracted with ESPS's `get_f0` and then passed to a lognormal tied mixture model [5] to alleviate pitch halving and doubling problems. In our previous systems, a smoothing algorithm similar to [6] was applied. Recently we have obtained improved performance by using the following smoothing algorithm: (a) Spline interpolation for the unvoiced regions, (b) Taking $\log$ of F_0, (c) Moving window normalization, and (d) 5-point moving average smoothing.

After we obtained the pitch feature for all time frames, we applied both mean and variance normalization to all dimensions, including c_0, on a per utterance basis. The final 42-dimension feature vector included first and second order

[1]The CNR programs were not used in the experiments here because we hadn't seen any further improvement by adding this subset in training.

Smoothing Algorithm	dev04	eval04
no pitch	14.5	24.1
IBM style	14.0	22.2
SPLINE	12.7	21.4

Table 1: CER comparison of different pitch smoothing algorithms.

differences.

To achieve a fast turnaround on investigating the best pitch smoothing algorithm, we used an acoustic model which was ML trained with Hub4 acoustic data only; decoded with the small bigram and unsupervised maximum likelihood linear regression (MLLR) adaptation [7]. Table 1 demonstrates our superior smoothing algorithm. For more details, please refer to [8].

3.3. Pronunciation Dictionary

With large vocabulary, it is natural to use phonetic models. Our phone set and pronunciation dictionary were derived from BBN, with very minor bug fixes and the addition of a silence phone and a noise phone, for a total of 72 base phones. The phone set follows the main-vowel principle in [9, 10]. Our garbage word was modeled by the noise phone, *rej*, with a pronunciation graph which allowed two or more *rej* phones. There were not many examples of laughter in our acoustic training data. Therefore we set the laughter word to have the same pronunciation as the garbage word. However, these two words were treated as two different lexical words in order to play different language roles. When future training samples contain significant laughter, we can easily create a new phone to model laughter separately without changing the training text transcription. All phonetic HMMs have the same 3-state left-to-right Bakis model topology.

3.4. Acoustic Models

We began by mapping our existing English context-independent (CI) phone models to the Mandarin phone set, followed by training the Mandarin CI model with the Hub4 acoustic data. Once we had a well trained CI model, it was used to train context dependent models, clustered down to 2500 shared Markov states (senones) with decision trees [11]. Each senone was modeled by 32 Gaussians. While building decision trees, we allowed clustering across triphones which represent the same toneless phone, and added different combinations of tone questions. We built both crossword and non-crossword triphone models, with the objective of either ML or minimum phone error rate (MPE) training with phone lattices [12]. We also conducted experiments with and without speaker adaptive training (SAT). All models were gender independent. For SAT experiments, we computed the 1-class constrained MLLR transformation [7] for each training speaker. The transformation was then converted to the feature domain as the SAT feature transform for each speaker: $N(x; A\mu + b, A\Sigma A^t) = |A|^{-1} N(A^{-1}(x - b); \mu, \Sigma)$.

Section 4.2 will present the progress of each of our acoustic models.

4. Experiments

4.1. Decoding Architecture

We used a simple two-stage decoding structure as follows:

1. Automatically identify the speech segments in the input audio program with a finite-state grammar which defines that an audio recording consists of any number of silence and/or speech segments. Each silence segment is modeled by one or more *silence* HMMs; each speech segment by at least 17 *speech* HMMs. Each HMM state is modeled by 300 Gaussians with 39-dimension MFCC and its differences.

   The identified speech portion is then segmented into short utterances of at most 10 secs long.

2. Compute the 42-dimension acoustic feature per frame, as described in Section 3.2, for each utterance.

3. *First stage search:* Run the first pass fast decoding with the non-crossword, non-SAT ML trained acoustic model and the small bigram LM. Output both the best hypothesis and a word lattice for each utterance.

4. Expand the bigram word lattices in Step 3 with the big 4-gram LM.

5. For each auto speaker, compute 3-class (silence, vowel, consonant) MLLR adaptation using the top 1 hypothesis from Step 3. The model being adapted can be the same AM used in Step 3 or another more complicated AM. If an SAT AM is adapted at this step, the speaker dependent SAT feature transform is computed before MLLR adaptation.

6. *Second stage search:* Search through the 4-gram word lattices with the speaker adapted AMs for the maximum-likelihood word sequence.

4.2. Experimental Results

In all experiments reported here, decoding steps 1-4 stayed constant. We varied the acoustic models in Step 5, and applied these speaker adapted acoustic models to Step 6 final decoding, constrained by the same 4-gram word lattices,

acoustic model	dev04	eval04
nonCW, nonSAT, ML	7.4%	-
nonCW, nonSAT, MPE	6.9%	-
nonCW, SAT, ML	6.8%	-
CW, SAT, ML	6.4%	-
CW, SAT, MPE	6.0%	16.0%

Table 2: CERs using different acoustic models at Step 4 of the decoding architecture. CW stands for crossword triphones.

for dev04. As Table 2 shows, MPE training, SAT normalization, and crossword triphone modeling all contributed significant error rate reduction. Finally, we applied the same decoding architecture using the best acoustic model to eval04 and achieved 16.0% CER, without any tuning.

5. Conclusion and Future Work

In this paper, we presented a new pitch smoothing algorithm and a new Chinese word segmentation algorithm. The SPLINE based pitch smoothing algorithm provided improvement over the popular IBM-style smoothing. The n-gram based ML segmentation often offered a better word segmentation. We have successfully incorporated both of these, along with various dominating speech technologies to build a competitive Mandarin broadcast news speech recognition system, with 6.0% CER on dev04 data set and 16.0% on eval04. We used less training data and simpler decoding architecture to achieve essentially identical error rates compared with other state-of-the-art systems.

One issue that we have not investigated is the handling of pure English speech segments. Sometimes test data contains segments of interviews with non-Chinese, who speak in English. These segments are spoken by native English speakers, not occasional English words embedded in Chinese sentences uttered by Chinese speakers. Ideally the system should first identify which language is spoken with 100% accuracy, and then switch to that language AM and LM for decoding. We will be investigating different methods of language ID and code switching.

Formerly we demonstrated that adding MLP posterior features into our conversational telephone speech recognizer improved recognition accuracy [13]. We will be integrating the more advanced ICSI feature [14] into our broadcast news recognizer for improved feature representation.

Improving the pronunciation phone set is another area of investigation. Furthermore, we would like to investigate the effectiveness of using PMI to create our initial word list automatically. Finally, adding more training data (acoustics, text, and web data) is always in the plan, particularly adding broadcast conversation type of text as currently there is not a plentiful supply.

6. Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA.

7. References

[1] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.

[2] A.L. Berger, S.D. Pietra, and V.J.D. Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, vol. 22, no. 1, pp. 39–71, 1996.

[3] A. Stolcke, "SRILM – an extensible language modeling toolkit," in *Proc. ICSLP*, 2002, vol. 2, pp. 901–904.

[4] A. Venkataraman, A. Stolcke, W. Wang, D. Vergyri, V. Gadde, and J. Zheng, "An efficient repair procedure for quick transcriptions," in *Proc. ICSLP*, 2004.

[5] M.K. Sonmez et al., "A lognormal model of pitch for prosody-based speaker recognition," in *Proc. Eur. Conf. Speech Communication Technology*, 1997, vol. 3, pp. 1391–1394.

[6] C.J. Chen et al., "New methods in continuous Mandarin speech recognition," in *Proc. Eur. Conf. Speech Communication Technology*, 1997, vol. 3, pp. 1543–1546.

[7] M.J.F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[8] X. Lei, M. Siu, M. Ostendorf, and T. Lee, "Improved tone modeling for Mandarin broadcast news speech recognition," in *interspeech*, 2006.

[9] C. J. Chen, H. Li, L. Shen, and G. K. Fu, "Recognize tone languages using pitch information on the main vowel of each syllable," in *Proc. ICASSP*, 2001, vol. 1, pp. 61–64.

[10] B. Xiang, L. Nguyen, X. Guo, and D. Xu, "The BBN Mandarin broadcast news transcription," in *Proc. Interspeech*, 2005, pp. 1649–1652.

[11] M.Y. Hwang, X.D. Huang, and F. Alleva, "Predicting unseen triphones with senones," in *Proc. ICASSP*, 1993, pp. 311–314.

[12] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," in *Proc. Interspeech*, 2005, pp. 2125–2128.

[13] X. Lei, M.Y. Hwang, and M. Ostendorf, "Incorporating tone-related MLP posteriors in feature representation for Mandarin ASR," in *Proc. Interspeech*, 2005, pp. 2981–2984.

[14] N. Morgan, B. Chen, Q. Zhu, and A. Stolcke, "Trapping conversational speech: Extending trap/tandem approaches to conversational telephone speech recognition," in *Proc. ICASSP*, 2004, pp. 537–540.