Real-Time Rich-Content Transcription of Chinese Broadcast News

Daben Liu, Jeff Ma, Dongxin Xu, Amit Srivastava, Francis Kubala

BBN Technologies
10 Moulton Street, Cambridge, MA 02138
{dliu, jma, dxu, asrivastava, fkubala}@bbn.com

ABSTRACT

This paper describes the recent development of an Audio Indexing System for Chinese (Mandarin) broadcast news. Key issues of the three major components: automatic speech recognition, speaker identification and named entity extraction are addressed. The Chinese-language-specific challenges are discussed and our solutions are described. The recognition accuracy of the final system is comparable to the best-known state-of-the-art systems, while the throughput is below real-time. The accuracy of the speaker identification and named entity extraction is comparable to our English system. The Chinese system currently runs 24x7 on a satellite feed of CCTV-4 broadcast news data.

1 INTRODUCTION

Over the last few years, BBN has developed an Audio Indexing System based on the Rough'n'Ready system for the real-time indexing and browsing of broadcast news in English and Arabic [1][2]. This system records streaming audio from TV and Radio broadcasts and automatically produces annotated metadata information in real-time. We have recently extended the capability of this audio indexing system to support the Chinese language.

In this paper, we provide a description of the Chinese Audio Indexing System followed by a description of the technical challenges involved and the solutions proposed for each major component in this system. We also provide performance evaluation numbers for each of the components and compare them to published baselines where available.

2 SYSTEM DESCRIPTION

A block diagram of the Chinese audio indexing system is shown in Figure 1. The audio server takes audio input from TV, radio broadcast or satellite and converts it into a digital audio stream. Speaker change detection component detects speaker changes in the continuous audio stream, segmenting the audio into homogenous speaker turns. Each turn is then converted into a word sequence by the speech recognition component. The speaker identification component recognizes the speaker name for each speaker turn. The speaker clustering component then groups the unlabeled turns based on speaker entity. Finally, the named entity extraction component identifies names of people, places, and organization in the word sequence. The end result is a textual document annotated with rich content.

Audio is passed to the audio encoder to be converted into Real Media format and stored in the audio archive. The indexed documents are stored in the text database. Both databases can be accessed through Web browsers over the network. The rich-content document can then be viewed, searched and retrieved, and the audio can be selectively replayed, through the Web browser.

3 AUTOMATIC SPEECH RECOGNITION

The speech recognition component is derived from the English Rough’n’Ready system [1] that originates from the 10x real-time BBN Byblos system [5]. Various fast algorithms [1][4] were implemented to enable the system to run under real-time. Some of the technologies used in the English system can be directly used in the Chinese system without modification due to their language-independent nature. In this section, we focus on the language dependent problems that were addressed in developing the Chinese ASR system.

3.1 Word segmentation

Words are the basic units in any ASR language model training. In the Chinese written language, words are not separated by space as is done in English or Arabic. Moreover, the definition of word boundaries in written Chinese is ambiguous. All the Chinese words are composed of characters. Each character by itself can be a word. To confuse the situation further, a new word can be constructed by compounding two or three other words. Given a big training corpus, hand-labeling the word boundaries is both expensive and time-consuming. Also two different annotators may label the boundaries differently. As a result, finding an efficient way of segmenting Chinese text into words is an important first step.

Given the significant ambiguity, we believe that there is no benefit in developing a complex process for word segmentation. For the sake of expediency, we implemented a simple longest-first method of word segmentation based on a given lexicon. For each sentence in the training corpus, we combine characters from left to right and only keep the longest combination that represents a word in the lexicon. A character is left as a single-character word if it does not combine with adjacent character to match an entry in the lexicon.
We used the 50K lexicon from LDC in this simple procedure to segment a 25-hour acoustic training transcription set, and compared the result with the manual segmentation of the same data, also provided by LDC. There were about 6% word boundaries that were different. A typical difference between the two segmentation methods is shown as follows:

Manual segmentation: 这是 中央 电视台 报道 的
Longest-first: 这是 中央电视台 报道 的

We believe that differences like the one shown above would have little effect on the final character error rate (CER).

3.2 Lexicon design

To find the optimal lexicon size, we experimented with three different lexicons. In all our lexicons, we include all the characters from the GB2312-80 Level-1 and Level-2 characters covering more than 99% of all the characters used in Chinese. Any characters in the training transcripts that are not present in GB2312-80 set are also added to the lexicon.

We started with the 50K lexicon provided by LDC as part of the Chinese acoustic training corpus. We then added missing characters from the GB2312-80 characters and from the training data to generate our baseline 50K lexicon. For our second lexicon, we downloaded a 122K lexicon from the Web and augmented it by adding to it words from the LDC 50K lexicon that were not already included in this new lexicon. Finally, for the sake of comparison, we created a lexicon of only Chinese characters. This eliminates the requirement for word segmentation during training. It should be noted that the common belief is that character-based language models do not utilize the context information efficiently. We made this 7K-character lexicon to study the effect of character-only models on CER as compared to word-based models.

Table 1 shows the CER result for using the three different dictionaries:

<table>
<thead>
<tr>
<th>Lexicon</th>
<th>CER (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7K -- character only</td>
<td>25.5</td>
</tr>
<tr>
<td>50K -- LDC + missing characters</td>
<td>18.2</td>
</tr>
<tr>
<td>122K - from Web</td>
<td>18.5</td>
</tr>
</tbody>
</table>

Table 1. Character-error-rate (CER) with different lexicon

It is clear, from the above results, that a word-based lexicon is important. On the basis of these results, we decided to continue with the 50K dictionary for the rest of our experiments.

Even though the character-only system performs the worst, a larger dictionary does not necessarily give the best performance. We do not fully understand this behavior in the result at this time and we intend to revisit the problem later in the course of development.

3.3 Acoustic modeling

We obtained 25 hours of annotated Chinese broadcast news from LDC as part of the Hub4 Mandarin training corpus. Using a standard acoustic training corpus allows us to compare our results with those reported by other research sites.

3.3.1 Pitch feature

Chinese is a tonal language. It has been shown that including tonal information in the lexicon improves the recognizer performance by 5% relative [8]. Adding tonal information, such as pitch, directly to the speech feature, which presumably increases the discriminative power, should provide even more gains.

We applied the pitch feature as a regular dimension of the feature space similar to [7]. For each speech frame, which was 29 milliseconds long, the autocorrelation of the residue error from linear prediction was computed. The frequency corresponding to the second-largest peak was determined as the pitch value for the frame. A threshold was used on the relative amplitude of this second peak to determine if the frame was voiced or unvoiced. Pitch corresponding to unvoiced frames was reset to zero.

The extracted pitch values were then passed through a low-pass filter to generate a smoothed contour over time. To remove the discontinuity between voiced and unvoiced regions, an iterative decay function was implemented as follows:

\[ P_{t+1} = P_t + (A_{pk} - P_t) \cdot \eta \text{ if } P_{t+1} = 0 \]

where \( P_t \) is the pitch value at frame \( t \). \( A_{pk} \) is the average pitch on the current speaker turn, and \( \eta \) is the decay factor which is a value between 0 and 1.

The first and second-order derivatives of pitch were computed. These three pitch-related features were normalized within a speaker turn to have zero mean and unit variance. The final feature space consisted of 48 dimensions, including 14 cepstral features with their first and second-order derivatives, normalized energy, first and second-order derivatives of energy, and the 3 pitch-related features.

3.3.2 Chinese-specific modeling

It is straightforward to use the same context-dependent phoneme-based modeling in our English system for Chinese ASR. LDC even provides a 50K dictionary with the phonetic baseform for each word. Several existing systems [8] [12] have adopted this form of modeling.

However, Chinese is a syllable-based language. Each character is a monosyllable, with or without an initial consonant part (Initial) and always with a vowel part (Final)[3]. It seems more natural for Chinese to use Initial/Final (I/F) acoustic modeling. Moreover, the widely used Pinyin standard uses I/F to represent Chinese characters. The I/F pronunciation of any new words can be looked up in a publicly available Chinese dictionary, whereas creating phonetic spellings requires special linguistic knowledge.

Another interesting, but less commonly used modeling scheme was proposed in [7], which balanced the number of initials and tonal finals. In contrast to the usual I/F system, it includes the glide (first vowel of the regular final), if there is any, into the initials rather than to the finals. We modified this method by further distinguishing the last vowels of finals with glides from those finals without glides, as illustrated in Table 2. We call the new method balanced Initial/Final (B-I/F).

<table>
<thead>
<tr>
<th>Type</th>
<th>Syllables(Pinyin)</th>
<th>Initials</th>
<th>Finals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular</td>
<td>Lan2</td>
<td>L</td>
<td>an2</td>
</tr>
<tr>
<td>Initial/Final</td>
<td>Lian2</td>
<td>L</td>
<td>ian2</td>
</tr>
<tr>
<td>Balanced</td>
<td>Lan2</td>
<td>L</td>
<td>an2</td>
</tr>
<tr>
<td>Initial/Final</td>
<td>Lian2</td>
<td>Li</td>
<td>An2</td>
</tr>
</tbody>
</table>

Table 2. The different unit assignment between regular Initial/Final and balanced Initial/Final

(Note that an2 and An2 are two different finals)
To find the most suitable modeling scheme, we built all the three systems.

For the phonetic system, we used the LDC provided phonetic spellings. There are a total of 76 phonemes out of which 51 are tonal vowels. For the I/F system, we used the same LDC lexicon (50K) but changed the spellings to use Pinyin. There are 28 initials. Seven of them are used as pseudo-initials for characters that only have finals, so that all the characters have the same C-V structure in their spellings [9]. There are 147 tonal finals. For the B-I/F system, we used 64 initials and 95 finals.

Note that we only used 4 tones because there was very little training data for the 5th-tone or un-toned vowels, which were consequently mapped to the third tone. To make a fair comparison, we tuned the systems so that they all ran at approximately the same speed. The comparison result is shown in Table 3.

![Table 3. Character-error-rate (CER) and real-time factor (xRT) comparison with different modeling](image)

We were surprised to observe that the B-I/F modeling was significantly better than the other two. One explanation for this could be that the inclusion of glides helped to improve the modeling on initials, which mostly consist of consonants.

### 3.4 Language Modeling

The language model training data consists of two parts. The first is the Chinese news corpus released by LDC. The sources include China Radio, People’s Daily, and Xinhua News. There are around 240 million characters in this corpus. The second part is the TDT2 and the TDT3 Corpus, also released by LDC for Topic Detection and Tracking evaluation. The sources in this data include Voice of America (VOA), Zao-Bao News (ZBN), and Xinhua News. There are 17 million characters in this corpus.

Prior to estimating the language models, we used the word segmentation algorithm described in section 2 and the 50K LDC dictionary to segment the text corpus. The acoustic training transcriptions were included with the other language model training data with a weight factor of 10. The final language model consists of about 50K unigrams, 6.4M bigrams, and 16.1M trigrams.

### 3.5 Decode with online speaker adaptation

The speech recognition component uses a 2-pass fast-matching decoder, which includes the forward fast-match pass, and a backward N-best generation pass. Finally, an N-best re-scoring pass is applied to generate the final 1-best output.

We have recently added real-time online speaker adaptation to the decoder. We removed speaker clustering before the adaptation since it is a non-causal process. Instead, we ran adaptation using partially available information from a speaker turn. We ran adaptation in all three passes. Forward pass used an incremental adaptation that utilized the previous utterance result for the adaptation of the current utterance within a speaker turn. The backward pass used the result from the forward pass for adaptation and the re-scoring pass used the result from the backward pass. This streamlined process enables the implementation of the adaptation algorithm with very low latency.

### 3.6 Experiment results

All our tests were run on a dual_processor 1.2GHz Pentium III with 2G of RAM and Windows 2000 operating system. We used Microsoft Visual C++ 6.0 as our compiler for all the executables. The real-time factor was measured by using elapsed wall-clock time from the start of audio feeding to the end of the last word emitted.

#### 3.6.1 Accumulative improvements

Our test data set consists of Hub4 1997 development data, Hub4 1997 and 1998 evaluation data, all adding up to about 3 hours. From the initial baseline phonetic system, we have gradually improved the performance, as shown in Table 4. The character-error-rate (CER) results are the average error rate across all the three sets.

![Table 4. Character-error-rate (CER) and real-time factor (xRT) for different systems](image)

We can see from the table that online adaptation has consistently provided about 7% relative gains over the un-adapted systems. The speed degradation due to adaptation is, however, only 0.1xRT. Pitch feature improves the system by another 3% relative. Using the B-I/F system gives an additional 4% relative improvement. It is pleasing to see that all the gains are additive to the overall improvement, which totals to 17%.

#### 3.6.2 Comparison to Hub4 Benchmarks

Both IBM and Dragon systems have participated in the 1997 and the 1998 Hub4 Chinese broadcast news transcription evaluation [10]. They used the same set of acoustic training data as we used. IBM’s 98 system significantly increased its language model training data by adding Chinese newspapers corpus. We had additional data from TDT2 and TDT3, which were 17 million characters more than that used by Dragon’s system. The informal comparison of our system to their official results is in Table 5.

![Table 5. Comparison of the character-error-rate (CER) to the Hub4 Benchmark results from 1997 and 1998](image)

Note that the numbers quoted for IBM and Dragon are the official evaluation numbers reported at the year of evaluation and are over 3 years old. These are the best known state-of-the-art numbers officially reported. We would expect these systems to perform much better nowadays. The goal of this comparison is just to see if our preliminary system is close to the official benchmarks. The results clearly show that we are already in the ballpark while running faster than real-time. Note that the Hub4 97-98 systems ran much slower than real-time [11][12].
4 SPEAKER IDENTIFICATION

In the Chinese system, we used the same speaker identification system as had earlier been used in English [1].

We selected 222 speakers from the training data that had more than 2 minutes of speech. Most of them are merely labeled as “announcer_x” or “reporter_x” in the training data. Only 78 of them are labeled with real names. We use these 78 speakers (46 males and 32 females) as our targets and the rest were treated as background speakers.

The system was evaluated on the same test set used in speech recognition. The decision threshold was optimized on some held-out data and selected such that the total number of errors is minimized. We achieved a segment error rate of 5.4%, which is comparable to the English system. The system can correctly recognize the target speakers in 94.6% of all the test segments.

5 CHINESE NAMED ENTITY EXTRACTION

The named entity extraction (NE) component is modeled on the English version of the Audio Indexer [1]. However, since various assumptions in the English tools are no longer valid for Chinese, we had to modify the system to make it work for Chinese.

The biggest challenge we encountered in this work is acquiring the training data for Chinese NE. The ambiguity of Chinese word boundaries introduces difficulties in consistently labeling names. Existing automatic segmentation algorithms have a fairly high error rate (5% or more), and require annotators to fix segmentation mistakes before marking names. We decided to annotate and train on unsegmented data, avoiding the problem of correcting segmentation errors while annotating. The corresponding loss in accuracy from using unsegmented data is about 2 points of F-measure.

175K words of Xinhua News Agency data, and 30K words from the Zao Bao News (Singapore) were annotated and used in Chinese NE training. The resulting F-measure on Xinhua News test data, with no speech recognition errors, is 87%. The result is comparable to that of the English system when the loss due to unsegmented data is factored in. An error analysis was conducted by a Chinese annotator on a sample of 200 errors generated by the named entity training data. It was found that 36% of the errors were caused by word ambiguity. For example, “Zhong” could mean “center” or the first character of “China”. Additional training and/or using a word segmentation system would probably solve the problem.

6 CONCLUSION

We have developed a real-time rich-content audio indexing system for Chinese broadcast news. The character-error-rate of the speech recognition component is 16.8% at 0.9xRT on hub4 test sets. The system also detects speaker changes, speaker names, and named entities. The system is currently running daily and indexing CCTV-4 news from a satellite receiver. Figure 2 shows the typical output of the indexing system displayed in the Internet Explorer. Any portion of the audio can be played back by selecting the corresponding text.

This is an on-going work. We will continue to improve the system performance. We will also enrich the audio indexing system by adding topics, stories, and other useful information that can be extracted from audio.

ACKNOWLEDGEMENT

This work was supported by the Defense Advanced Research Projects Agency under contract N66001-00-C-8008. The views and findings contained in this material are those of the authors and do not necessarily reflect the position or policy of the Government and no official endorsement should be inferred.

REFERENCES


Figure 2. Rich-content output of Chinese audio indexing system is shown in MS Internet Explorer. The document is organized in speaker turns, with the left column showing the speaker names, or gender with speaker cluster IDs if the speakers are not in the target list. The right column shows text with named entities displayed in different colors.