JANUS-II: Towards Spontaneous Spanish Speech Recognition

Puming Zhan, Klaus Ries, Marsal Gavalda
Donna Gates, Alon Lavie, and Alex Waibel

Interactive Systems Laboratories
Carnegie Mellon University
Pittsburgh, PA 15213
Email: {zhan,ries,marsal,dmg,alavie,ahw}@cs.cmu.edu

ABSTRACT

JANUS-II is a research system for investigating various issues in speech-to-speech translations and has been implemented for speech-to-speech translations on many languages [1]. In this paper, we address the Spanish speech recognition part of JANUS-II. First, we report the bootstrap and optimization of the recognition system. Then we investigate the difference between push-to-talk and cross-talk dialogs, which are two different kinds of data in our database. We give a detail noise analysis for the push-to-talk and cross-talk dialogs and present some recognition results for the comparison. We have observed that the cross-talk dialogs are harder than the push-to-talk dialogs for speech recognition, because they are more noisy than the latter. Currently, the error rate of our Spanish recognizer is 27% for push-to-talk test set and 32% for cross-talk test set.

1. Introduction

Today, most of the state of the art speech recognition systems are based on Hidden Markov Model (HMM) techniques, which were first applied to speech recognition about twenty years ago. The HMM based systems are quite successful on reading speech recognition task. However they are far from satisfactory for spontaneous speech recognition. Much research in this field has been directed to the recognition and understanding of spontaneous speech in recent years. Compared to reading speech, spontaneous speech usually contains much more noises and disfluencies, such as human noise, background noise, simultaneous speaking, mispronunciations and repetitions. Therefore, it is well know that the spontaneous speech is much harder than the reading speech for speech recognition. In this paper, we report how to bootstrap and improve JANUS-II speech engine for spontaneous Spanish speech recognition. Then we analyze the disfluencies of the push-to-talk and cross-talk dialogs, and compare their performance in speech recognition.

2. Database

The JANUS system is built for and evaluated on the appointment scheduling task. The details of this Database, including English, German, Korea, Japanese and Spanish data, can be found in [1]. The Spanish Database consists of two different kinds of data: push-to-talk dialogs and cross-talk dialogs. More than a half of the data in the database are cross-talk dialogs. Although they are all human to human dialogs, these data are recorded in very different styles. Briefly, in push-to-talk recording, two speakers have to interface with a computer and push the "return" key to speak, so that simultaneous speaking can be avoided. In the cross-talk recording, two speakers can interrupt each other at any time, so that simultaneous speaking is possible. Table 1 is a summary of the database used for development of the Spanish speech recognizer.

training set	push-to-talk	cross-talk
utterances	1090	7740
words	42142	73617
words per utt	38.6	9.5
hours	5	7

Table 1: Spontaneous Spanish Scheduling Task Database

On average, there are 38.6 words per utterance for the push-to-talk dialogs and 9.5 for the cross-talk dialogs, indicating that the length of cross-talk utterance is, in general, much shorter than the push-to-talk utterance. Because of the lack of the training data, we use the push-to-talk and cross-talk dialogs together to train the acoustic models, but keep an individual test set for each of them. The push-to-talk test set consists of 13 dialogs, three male and four female speakers, which contains 86 utterances. The cross-talk test set consists of 6 dialogs, three male and three female speakers, which contains 117 utterances. The test vocabulary consists of 3911 unique words in the training set. For both test sets, the out of vocabulary word rate is 1.6%.

3. Preprocessing

The feature we are using is Perceptual Linear Predictive (PLP) coefficients, which are generated based on [3]. The speech signal is sampled with 16KHz rate. After passing through a preemphasis filter and Hamming window as usual, 128 points FFT spectrum is calculated. The FFT spectrum

is integrated with a critical band in Bark-scale, and operated with the cube 0.33 cubic-root amplitude compression. After such kind of perceptual processing, 21 coefficients are obtained and used to generate 13 LPC coefficients and then 13 LPC-Driven Cepstrum coefficients. We combine 13 cepstrum coefficients with its Delta and Delta-Delta coefficients together to generate a 39-dimension feature vector. Finally, this feature vector is transformed by a 39x39 matrix which is generated by Linear Discriminant Analysis [2]. The first 16 components of the transformed vector are kept as the final feature vector. Our experiments showed that the above PLP is better than the Mel-Frequency-Scale Coefficients (MFSC). The word accuracy with PLP feature is about 1.5% better than that with MFSC feature.

4. JANUS-II Spanish speech recognition system

4.1. Speech engine

The JANUS-II Spanish speech engine is based on Continuous Density Hidden Markov Model (CDHMM). We use Gaussian-Mixture density as the output probability of each CDHMM's state. The mixture-order, called codebook size in semi-continuous density HMM, is chosen according to the amount and separability of the training samples which are aligned to this mixture density. Thus we can make sure that every component of the mixture density has enough samples for its training, and meanwhile the mixture-order is reasonable large in order to keep the model's accuracy. We use the Viterbi algorithm for acoustic model training and update the parameters of the best matched component of Gaussian-Mixture density. For recognition, we use the standard JANUS-II decoder which includes three passes, i.e. Tree-pass, Flat-pass and Lattice-pass [1, 6]. All results we report in this paper are obtained from the Flat-pass, with which word accuracy is about 1.5% - 2.5% better than the Tree-pass and 0.5% - 1.5% lower than the Lattice-pass. We have used two different trigram language models. One is generated with the standard LM Tools at CMU, which is based on the standard backoff algorithm. The other is generated by using the Knesey/Ney backoff algorithm [4]. We found that Knesey/Ney's algorithm gives us about 4% error reduction. Therefore we keep using this language model in the paper.

4.2. Bootstrapping the context dependent phone models

When porting a existing speech recognition system towards a new language, the first thing to be done is to choose a suitable speech units of the target language as acoustic models. Spanish is a phonological language and its phoneme numbers is in the same range as English, so we first choose Context-Independent (CI) phones as speech units for acoustic models, then extend the CI phone models into Context-Dependent (CD) phone models. We use 40 CI phones as the CI acoustic models for the Spanish speech recognition system. The initial parameters of the Spanish acoustic models are obtained from the corresponding acoustic models of JANUS-II English speech recognizer. From CI to CD acoustic phone models, the within-word-triphones are used without position tag. We simply choose the triphones according to their frequencies in the training set. There are two typical methods for context-dependent model clustering: the data-driven algorithm and decision-tree algorithm. The purpose to cluster the triphones is to make the acoustic models cover more context-dependent information and at the same time keep the number of acoustic models in a reasonable level so that we can train them well with the limited data. Therefore, there is always a trade-off between accuracy and robustness of the acoustic models in the condition of limited training data. Compared to choosing the triphones according to their frequencies in the training set, the data-driven method did not give us substantial improvement. Because our training and testing data are restricted in the same domain, the scheduling task, we can get better result by just using the high frequent triphones as acoustic models. In Table 2 and Table 3, we present some results based on different number of CI to CD phone models and different dimensions of the PLP feature vectors. All these results were obtained with the push-to-talk test set.

phones	48(CI)	245	421	596	684
WA	61.2%	67.5%	71.3%	72.3%	72.1%

Table 2: Word accuracy with different number of triphones

In Table 2, we keep using a 16-dimensional PLP feature vector, full covariance matrix in the Gaussian Mixture density and Trigram language models. The triphone numbers in Table 2 includes 8 special phones as the noise models. It shows that 596 triphones as acoustic models give us the best word accuracy. The word accuracy with different number of acoustic models depends on the acoustic model's complexity and the amount of training data. With more data, we may get better word accuracy with more acoustic models.

Dimension	eigen-ratio	Diagonal	Full
12	44.3%	66.7%	67.8%
16	53.2%	71.0%	72.3%
20	61.6%	71.7%	71.9%
24	69.9%	72.2%	71.5%

Table 3: Word accuracy with different dimensions of feature and different type of covariances

In Table 3, we keep using 596 triphones as acoustic models in the recognizer. The table shows that the best result is obtained from the system which uses a 16-dimensional feature vector and full covariance. The dimension of the feature vector was increased from 16 to 24, the word accuracy also had slight increase for the system with diagonal covariance, but decrease for the system with a full covariance. We need to mention that we compare the diagonal and full covariance system based on the principle of **same parameter num-**

bers. The principle means that we choose the mixture numbers to keep the two systems have nearly the same number of parameters in their acoustic models. For example, in the case of using 16-order feature vector in Table 3, the average number of Gaussian mixtures for each acoustic model is nine in the diagonal covariance system, and two in the full covariance system. Because of symmetry of the covariance matrix, there is only a little extra computational complexity in the full covariance system. Therefore we choose 16-order PLP feature, 596 triphones and full covariance in the Gaussian mixture density as our standard system for the remaining experiments. The sum of eigenvalues of the diagonal LDA matrix is a measure of separability of the classes with which the LDA matrix is generated. Table 3 shows that if we choose the first 16 coefficients from the 39-order feature vector, we can keep 53.2% separability of the original vector. Actually, projecting a feature vector to a lower space with the LDA technique usually leads to a better performance, because with limited data, we can get more robust model in the reduced space.

5. Optimization of the LDA transformation matrix

Linear Discriminant Analysis (LDA) is a traditional technique for pattern recognition [2]. It has been used in speech recognition systems as a preprocessing method for several years. But how to embed the LDA matrix into the training process of speech recognition is still a open problem [5]. The LDA transformation matrix is created based on the classes of the patterns. The goal is to build a linear transformation matrix, with which the feature can be projected into its subspace, and meanwhile keep or increase the separability of the patterns. In speech recognition system, the LDA matrix is usually generated based on the label file which contains the alignment of the training samples with the acoustic models, hence the phone classes. Once the LDA matrix is built, it is rarely updated, because the current training algorithm (Viterbi or Baum algorithm) does not include training the LDA matrix. Obviously, this is not optimal. The LDA matrix is entirely based on the alignment of the training samples. Every time the acoustic model is updated in training process, the alignment is changed. Therefore, the LDA matrix should be updated too. The training algorithm can be divided into two steps: dynamic match and model update. The alignment of the training samples is obtained in the dynamic match step, and model is updated based on the alignment. Our idea is to insert a LDA matrix updating step between the dynamic match and model update step. After the dynamic match or force-alignment, we first update the LDA matrix according to the alignment, then update the models. Finally, the new models are obtained by projecting the updated models into a new space based on the new LDA matrix. [5] gives a rigorous algorithm for the LDA optimization, but did not get significant improvement. Compared to it, our method is simple and suboptimal. But we got 5%-7% error reduction from it.

6. Comparison of push-to-talk and cross-talk dialogs

In this section, we first analyze the noise distribution of the two databases, then describe the noise model generation and compare the performance of the two database in speech recognition.

6.1. Noise analysis

Generally speaking, noises fall into three classes: (a) distortion of the recording equipment, such as channel and microphone distortion; (b) human and nonhuman made noises which occur exactly between the real words, such as /LS/ /H#/ W1 W2 /MM/ W3 /EH/..., where /LS/ is lip smack, /H#/ is breathing; (c) human and nonhuman made noises which occur at the same time that the user is speaking to the recognizer, i.e. the noises overlap with the real speech signal, such as /BEGIN-LAUGH/ W1 W2.../END-LAUGH/; Most of the noises in class (c) are background noises, and as of our knowledge, there is no existing very effective method dealing with them. Using special microphone is one of the ways to reduce the effect of such background noises, but it is limited by practical environment. Besides, there is a lot of false-starts, repetitions, mispronunciations, and simultaneous talking in the spontaneous speech, which heavily affect the speaking rate, amplitude and prosody, and are very difficult to handle. In our system, we use the Mean-Subtraction technique to eliminate the channel distortions, i.e. the noises in class (a). Table 4 contains some statistical analyses of the noises in class (b) and (c). Ratio1 is **noises/(words + noises)**, where the noises are those in class (b). We did not include false-starts into the noise count, because we treated them as real words. But we included the mispronounced words into the noise rate calculation, because they are one of the worst noises and very hard to be recognized. Ratio2 is **words-covered-by-noise/words**, which gives us a kind of measurement for the noises in class (c). The **words-covered-by-noise** was counted according to the noise marks in the transcription files. Only the real words which are between the noise beginning and ending marks were counted, and the noises in class (b) were excluded, because they were already considered in Ratio1.

Database	Utts	Words	Ratio1	Ratio2
push-to-talk	1090	42142	17.94%	9.8%
cross-talk	7740	73617	19.32%	30.5%

Table 4: Statistics of the noises

Table 4 illustrates that there is no significant difference between the Ratio1 of the cross-talk dialogs and push-to-talk dialogs. But we noticed that among 19.32% noise rate of the cross-talk dialogs, the mispronunciation rate is 2.67%, which is significantly higher than 0.87% mispronunciation rate of the push-to-talk dialogs. Obviously, the Ratio2 of the cross-talk dialogs is much higher than that of the push-to-talk

dialogs. Among 30.5% rate of noise covered words in the cross-talk dialogs, the rate of words covered with simultaneous speaking is 5.2%, compared to zero in the push-to-talk dialogs.

6.2. Noise model generation

The noise acoustic models described in this section are for the noises in class (b). We count the human and nonhuman made noises in the training database and pick up several sorts of noises according to the rank of their frequencies to assign special acoustic models for them. Two general noise models are used for the remaining human and nonhuman noises. We found that the high frequency noises in the push-to-talk and cross-talk dialogs are almost the same, though their noise rates are different. The major difference is that there are a lot of Key-Click noises in the push-to-talk dialogs and no such noises in the cross-talk dialogs. Thus we use the same noise models for both data. We also assign a general acoustic model to those words which were mispronounced (most of them were pronounced incompletely, i.e. some phonemes in the word were not pronounced). Table 5 gives the word accuracy with respect to different number of noise models. The results depend on the database size

No. of Noise Models	push-to-talk	cross-talk
2	69.2%	64.4%
4	70.5%	65.7%
8	72.3%	67.4%
11	71.7%	67.1%

Table 5: Word accuracy with different No. of noise models

and the statistical distribution of the noises. In our case, the best word accuracy is obtained with 8 noise models. We tried to merge some noises which likely have similar voice together, such as, /MM/ /NN/, and also tried to use more noise models, but did not get significant improvement. We found that some noises, which have low amplitude and short duration, such as lip smacks, glottal noises, do not affect the performance of the system very much, though they occur with high frequency. The noises which have high amplitude and long duration, such as laugh, mispronunciations, EH or HUH with long duration, and long silence, heavily affect the performance of the system.

6.3. Comparison in speech recognition

In this section, we give a comparison of the word accuracy for the push-to-talk and cross-talk dialogs. Table 6 contains the results of speech recognition for the push-to-talk and cross-talk test set. We use different number of acoustic models in the recognition systems for this experiment, but keep using 8 noise models according to the results in Table 5.

Table 5 and Table 6 indicates that because of the high noise and disfluency rate, as we showed in Table 4, the word accuracy of the cross-talk dialogs is consistently lower than

Triphones	push-to-talk	cross-talk
48(CI)	60.2%	51.2%
245	66.5%	61.0%
421	68.7%	65.0%
596	73.5%	68.8%

Table 6: Word accuracy with different acoustic models

that of the push-to-talk dialogs regardless of the number of acoustic models and noise models.

7. Conclusion

In this paper, we have reported the development of JANUS-II Spanish speech recognition system, and given a detail analysis and comparison of the push-to-talk and cross-talk database in speech recognition. The error rate of the system has been reduced from around 70% at the beginning to the current 26.5%.

8. ACKNOWLEDGMENTS

The work reported in this paper was funded in part by grants from the US Department of Defense. The author wish to thank all members of the Interactive Systems Laboratories in University of Karlsruhe and Carnegie Mellon University, especially Ivica Rogina, Bernhard Suhm, Torsten Zeppenfeld, Martin Maier and Monika Woszczyna, for their active support.

9. REFERENCES

1. A.Waibel, M.Finke, D.Gates, M.Gavalda, T.Kemp, A.Lavie, L.Levin, M.Maier, L.Mayfield, A.McNair, I.Rogina, K.Shima, T.Sloboda, M.Woszczyna, T.Zeppenfiled, and P.Zhan. Jnaus-ii – advances in spontaneous speech recognition. *ICASSP-96*, 1996.
2. Fukunnaga and Keinosuke. *Introduction to statistical pattern recognition.* Academic Press, Boston, 1990.
3. Hynek Hermansky, Nelson Morgan, Aruna Bayya, and Phil Kohn. Rasta-plp speech analysis technique. *ICASSP-92*, 1:121–124, 1992.
4. Reinhard Knesey and Hermann Ney. Improved backing-off for m-gram language modeling. *ICASSP-95*, pages 181–184, 1995.
5. E. Gunter Schukat-Talamazzini, Joachim Hornegger, and Heinrich Niemann. Optimal linear feature transformations for semi-continuous hidden markov models. *ICASSP-95*, pages 369–372, 1995.
6. M. Woszczyna and M.Finke. Minimizing search errors due to delayed bigrams in real-time speech recognition system. *ICASSP-96*, 1996.