CLASSIFICATION OF CLEAN AND NOISY BILINGUAL MOVIE AUDIO FOR SPEECH-TO-SPEECH TRANSLATION CORPORA DESIGN

Andreas Tsiartas[1], Prasanta Kumar Ghosh[2], Panayiotis Georgiou[1], Shrikanth Narayanan[1]

[1]Signal Analysis and Interpretation Laboratory,
Department of Electrical Engineering,
University of Southern California, Los Angeles, CA 90089, USA
[2]Department of Electrical Engineering,
Indian Institute of Science (IISc), Bangalore, 560012, India
tsiartas@usc.edu, prasantg@ee.iisc.ernet.in, georgiou@sipi.usc.edu, shri@sipi.usc.edu

ABSTRACT

Identifying suitable sources of bilingual audio and text data is a crucial part of statistical Speech to Speech (S2S) research and development. Movies, often dubbed in other languages, offer a good source for this purpose; but not all data are directly usable because of noise and other audio condition differences. Hence, automatically selecting the bilingual audio data that are suitable for analysis, and training S2S systems for specific environments becomes crucial. In this work, we extract bilingual speech segments from movies and aim at classifying segments as clean speech or speech with background noise (i.e. music, babble noise etc.). We examine various features in solving this problem and our best performing method delivers accuracy up to 87% in discriminating clean and noisy speech in bilingual data.

Index Terms— bilingual movie audio clean speech detection, audio segmentation

1. INTRODUCTION

Due to the statistical nature of Speech-to-speech (S2S) translation systems, bilingual data have played a significant role in their research and development, for example, bilingual parallel audio has been shown to be important for the translation of paralinguistic cues [1, 2, 3]. Researchers have focused on both manual and automatic data collection approaches for the design of S2S translators. Such bilingual data not only include spoken utterances in the source language along with their interpretation in the target language but also text translation of speech transcriptions. Automatically acquired data could contain speech segments that are not suitable due to low Signal to Noise Ratio (SNR) levels and, thus, reduces the usefulness of the data. For this reason, additional research is needed to automatically distinguish low SNR from high SNR bilingual speech signals which that are suitable for S2S translation design.

Examples of manually obtained bilingual speech and transcript data include the Europarl [4] and the news commentary corpus[1]. In addition to manually collected data, many approaches have been proposed to automatically collect and align bilingual transcriptions. A key component of the automatic algorithms was to model the variability and noise in the alignment of bilingual transcriptions. Such algorithms have been often used to align movie subtitles. For example, Tsiartas et al. [5] focused on aligning speech in movies with subtitles. Sarikaya et al. [6] selected subsets of bilingual subtitle transcriptions by removing noisy pairs and showed BLUE score [7] improvements on a large-scale S2S system.

Bilingual text transcriptions lack additional information that resides in the audio that may contain important linguistic (e.g., prosody) and paralinguistic (e.g., affect) information for modeling speech translations. Hence, beyond text bilingual data, researchers have been collecting audio bilingual data such as, for example, the DARPA TRANSTAC domain data [8] and the Basic Travel Expression Corpus (BTEC) [9]. In addition to manually collected audio, researchers have proposed approaches to extract bilingual audio data automatically from existing sources. In our past work [10], we had proposed a method to segment bilingual audio from movies and align the segments with the corresponding subtitles.

However, the aforomemntioned method did not distinguish between the quality of bilingual speech (clean or noisy) but instead focused if detecting just the presence of speech. Movie data contain a wide variation in the audio quality and, hence, automatic data selection becomes critical. For this classification task, we use movies that are dubbed in at least two languages and propose an approach to classify the bilingual parallel audio as noisy segments of speech (i.e, background music, gunshots etc) or clean speech. To solve this problem, we exploit the fact that the noise in the two channels is acoustically correlated but the speech signals are not correlated since they are in two different languages. For this purpose, we design a set of diverse features and evaluate their performance on a data set annotated by humans.

This paper is structured as follows. In section 2, we present the collected data used in this work. In section 3, we describe the proposed features used in discriminating low and high SNR speech regions. In section 4, we present the experimental setup. Section 5 discusses the experiments and results of our approach and, finally, in section 6, we summarize the results of this work and provide some future directions.

[1]Made available for the workshop shared task http://www.statmt.org/wmt10/

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2. DATA COLLECTION

For the purpose of these experiments, we collected 5 movies containing audio and subtitles in English and French and we down-mix all channels to one channel for each language. Then, we use the approach proposed in [10] to segment the parallel streams of audio into multiple aligned bilingual speech segments. This generates a corpus of 490 bilingual segments. An overview of the the audio segmentation and alignment tagging is shown in Fig. 1. These were manually tagged into clean and noisy bilingual speech audio. In the next stage of annotation, we classified segments that contained no background noise as clean bilingual speech and segments with even some noise as noisy bilingual speech. Overall, we obtained 27% clean English-French speech segments with the other 73% tagged as noisy speech segments.

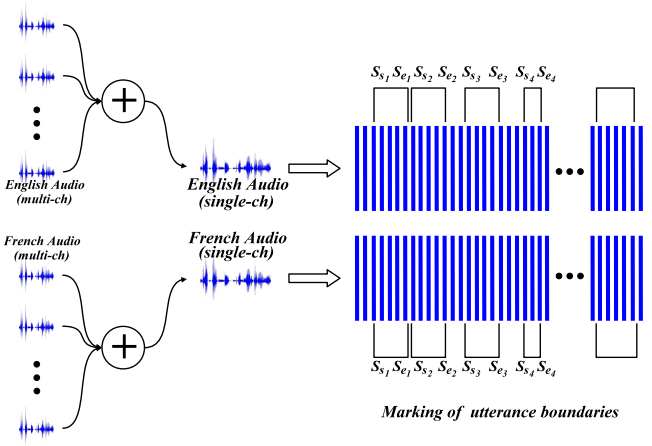

Fig. 1. An illustration of the automatically segmented bilingual audio streams for English and French. S_{s_i} and S_{e_i} denote the begin and end sample indices for the i^{th} segment

3. PROPOSED FEATURES

In this section, we aim to design the features that capture information that discriminate bilingual noisy and clean speech segments based on the SNR levels. To design these features, we need to understand some important properties of the bilingual speech audio. Firstly, the bilingual segment pair contains speech in two different languages in two separate signals. The speech signal may or may not contain noise. Noise can be background music, background babble noise and in general any non-speech audio signal including noise that can be much smaller in duration than the speech segment. Acoustically, noise is similar in both audio streams. In some cases, noise in one audio stream can be a shifted, scaled and maybe filtered version of the noise in the other audio stream. Using the above-mentioned properties, we construct features that capture the spectral correlation (due to the acoustic similarity) between the two audio streams. In addition, we use the first audio stream to predict the second audio stream, thus, estimate the noise and measure the energy ratio between the estimated noise and both audio streams.

3.1. Spectral correlation (SC)

In order to approximate the acoustic and perceptual proximity of the bilingual audio streams, we use the mel-frequency cepstral coefficients (MFCC) [11] to represent the audio signal. Suppose for the i^{th} segment there are R^i frames. To define the spectral correlation, we first concatenate R^i consecutive frames' L-dimensional MFCC feature vector (excluding the DC coefficient). Thus, for each segment, say segment i, we have two vectors (one for each language) of dimension $I^i = R^i L$. Hence, we define the two I^i MFCC feature vectors as $C_{L_1}(i)$ and $C_{L_2}(i)$ for the i^{th} segment. Using these two vectors, we compute the correlation coefficient. The reason we are using the spectral correlation is to capture the spectral similarity of the streams while keeping the feature robust to any scalings and short-term shifting of any of the two audio streams.

Hence, the Spectral Correlation (SC) of the i^{th} segment is defined as:

$$SC(i) = \frac{\sum_{j=1}^{I^i} \left(C_{L_1}(i,j) - \bar{C}_{L_1}(i)\right)\left(C_{L_2}(i,j) - \bar{C}_{L_2}(i)\right)}{\sqrt{\sum_{j=1}^{I^i} \left(C_{L_1}(i,j) - \bar{C}_{L_1}(i)\right)^2 \sum_{j=1}^{I^i} \left(C_{L_2}(i,j) - \bar{C}_{L_2}(i)\right)^2}}$$

where the mean is defined as: $\bar{C}_{L_1}(i) = \sum_{j=1}^{I^i} \frac{(C_{L_1}(i,j))}{I^i}$

Thus, by definition, the closer the acoustic similarity of the two vectors is, the closer $SC(i)$ will be to 1. Thus, a high value of $SC(i)$ indicates that similar noise in the two streams is significantly more than the speech signals.

3.2. Noise to Speech and Noise Ratio (NSNR)

The Noise to Speech and Noise Ratio (NSNR) aims to capture the ratio between the noise that is common in the two channels and the amount speech that is present. For each segment separately, we denote the audio of language L_1 and L_2 as $\mathcal{S}_{L_1}$ and $\mathcal{S}_{L_1}$ respectively. Moreover, we assume the following signal model for the audio signals $\mathcal{S}_{L_1}$ and $\mathcal{S}_{L_1}$: $\mathcal{S}_{L_1} = X_{L_1} + N$ and $\mathcal{S}_{L_2} = h * X_{L_2} + h * N$ where * represents the convolution operator. X_{L_1} and X_{L_2} are the speech signals in the audio streams of language L_1 and L_2 respectively. N is the noise in the audio stream of L_1 and h is a filter. In addition, we assume X_{L_1}, X_{L_2} and N are uncorrelated. To verify this uncorrelated assumption, we computed the correlation coefficient between such signals and we found that the correlation coefficient is very close to 0 (The average correlation coefficient is of the order 10^{-4}). We define NSNR for the i^{th} segment as:

$$\begin{aligned} NSNR &\triangleq \frac{|E\{(h*\mathcal{S}_{L_1})\mathcal{S}_{L_2}\}|}{E\{(h*\mathcal{S}_{L_1}+\mathcal{S}_{L_2})^2\}} \\ &= \frac{|E\{UC+(h*N)^2\}|}{E\{(SSN+4UC-2(h*X_{L_1})(h*X_{L_2})\}} \\ &= \frac{|E\{(h*N)^2\}|}{E\{SSN\}} \end{aligned}$$

where $SSN = (h*X_{L_1})^2 + (h*X_{L_2})^2 + (2h*N)^2$

$UC = (h * X_{L_1})(h * X_{L_2}) + (h * X_{L_1})(h * N) + (h * N)X_{L_2}$ and $E\{UC\} = 0$ because X_{L_1}, X_{L_2} and N are uncorrelated.

For signal X of size K, $E\{X\} = \frac{\sum_j X(j)}{K}$

The above expansion and analysis of the NSNR feature reveal that NSNR takes values closer to 0 when the SNR is very high in both audio streams. On the other hand, NSNR takes values closer to 1 if SNR is very low. From the definition of the NSNR, we need to know $\mathcal{S}_{L_1}$, $\mathcal{S}_{L_2}$ and h. $\mathcal{S}_{L_1}$ and $\mathcal{S}_{L_2}$ are directly known from the data, since they are simply the time domain samples of each bilingual audio segment. However, the filter h is unknown and, thus, we need to estimate h from the data for each bilingual segment separately.

3.3. Filter estimation

To estimate the filter h, we propose two approaches. The first approach is simplistic and faster and assumes that the filter acts on the signal by scaling and shifting it. The second approach tries to estimate a time-varying Least Mean Squares (LMS) filter using the normalized LMS [12] approach.

3.3.1. Scaling and Shifting Filter (SSF)

In this case, the assumption is that the filter h is only shifting and scaling the signal. To estimate this signal, we use regions in which there is only noise. Such regions are returned by the algorithm described in [10]. The segments between consecutive speech regions are expected to be noise only. For example, the i^{th} segment has a left and right noise-only region. As Fig. 1 shows the left noisy region is between $S_{e_{i-1}}$ and S_{s_i} and the right noisy region of segment i is between S_{e_i} and $S_{s_{i+1}}$.

Now, to compute h using the SSF estimation, we first compute the correlation coefficient for the noisy regions before and after the speech segment by varying the shift index M as follows:

$$CCbefore(i,M) = \frac{\sum_{j=S_{e_{i-1}}+M}^{S_{s_i}} V_{j-M}^{L_1} V_j^{L_2}}{\sqrt{\sum_{j=S_{e_{i-1}}+M}^{S_{s_i}} \left(V_{j-M}^{L_1}\right)^2 \sum_{j=S_{e_{i-1}}}^{S_{s_i}} \left(V_j^{L_2}\right)^2}}$$

and

$$CCafter(i,M) = \frac{\sum_{j=S_{e_i}+M}^{S_{s_{i+1}}} V_{j-M}^{L_1} V_j^{L_2}}{\sqrt{\sum_{j=S_{e_i}+M}^{S_{s_{i+1}}} \left(V_{j-M}^{L_1}\right)^2 \sum_{j=S_{e_i}}^{S_{s_{i+1}}} \left(V_j^{L_2}\right)^2}}$$

where $V_j^{L_k} = \mathcal{S}_{L_k}(j) - \bar{\mathcal{S}}_{L_k}(j)$

We define the maximum correlation coefficient by

$$MCC(i) = \max(\max_M(CCleft(i,M)) \max_M(CCright(i,M)))$$

The optimal shift (delay of h) for the i^{th} segment is the value M that corresponds to the $MCC(i)$ value. To compute the scale, we select the noise region (left or right) which corresponds to $MCC(i)$ value. Then, the scale factor is computed as the ratio of the energy between the noisy region L_1 and the noisy region L_2. Thus, we construct h and compute NSNR.

3.3.2. Least Mean Squares Filter (LMSF)

In this case, we relax the assumptions of h and we let h to be any filter. To estimate the filter, we use the normalized LMS algorithm as described in [12]. Note at this point that we include the left and right noise regions of $\mathcal{S}_{L_1}$ and $\mathcal{S}_{L_2}$ in the input and target signals to get better estimates of the filter h. We denote the extended signals as SN_{L_1} and SN_{L_2}. Since at each step of LMS we are minimizing the distance between SN_{L_1} and SN_{L_2}, the filter will be such that SN_{L_1} will track SN_{L_2}. Ideally, the error of the LMS will be $h * X_{L_2}$ and ,thus, the output will be $h * SN_{L_1}$. To define the iterative Normalized LMS, we need to define first the truncated versions of SN_{L_1} and SN_{L_2}. We define $SN_{TL_1}(n)$ the truncated signal starting at sample n. The signal is truncated to have length equal to h and n is used to shift the truncated signal. The input and target signals to Normalized LMS are SN_{TL_1} and SN_{TL_2} respectively. Next, the iterative Normalized LMS to estimate h in the $m^{th}+1$ iteration is performed in the following manner:

$$h^{m+1} = h^m + \frac{\mu(SN_{TL_2} - h^m \cdot SN_{TL_1}) \cdot SN_{TL_1}}{||SN_{TL_1}||^2}$$

After finding h, we use h, $\mathcal{S}_{L_1}$ and $\mathcal{S}_{L_2}$ to compute NSNR.

4. EXPERIMENTAL SETUP

To identify, align and segment speech and noisy speech regions, we used the algorithm described in [10]. Furthermore, we have used the same parameters values optimized in [10], since we are working on the same data set.

After getting the segments, we extracted the various features described in section 3. For the computation of SC, we computed 12 MFCCs (excluding the DC coefficient). For computing the filter h using the SSF method, we have searched M in the range -800:800 and, thus, searching correlations of 1601 values which means we are searching for the best shift within 100ms in a 16kHz audio signal. This is a reasonable assumption given the grounding of the audio channel to the video stream; additionally, this helps in constraining the computational cost.

Moreover, using LMSF to estimate h, we had to optimize the learning rate, μ, and filter size, $|h|$. On a development set, by using grid search we picked the parameters that maximize the average K-Nearest Neighbor K-NN performance for $K = 1-20$. We computed the performance for $\mu = [0.1\ 0.01\ 0.001\ 0.0001]$ and filter size $|h| = [30\ 80\ 250\ 800]$. We found that the average K-NN performance was maximized for a filter size of 80 and learning rate $\mu = 0.001$. To get better estimates of the h filters for each bilingual segment, we run two iterations over the same segment. This helps the Normalized LMS algorithm to converge if it did not converge during the first iteration. Of course, more iterations one runs, the better the estimate and convergence of the filter; however, extra iterations over the same segments increase the computational cost significantly.

In all experiments that K-NN is involved, we used Mahalanobis [13] distance as a distance function. Also, in order to strengthen the results of our work, we run a 5-fold cross-validation in all experiments. The split of train/test is 60%/40% and the results reported are the average of the folds.

5. EXPERIMENTS, RESULTS AND DISCUSSION

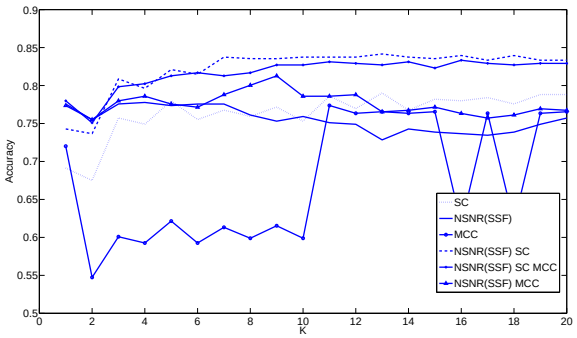

Fig. 2. This figure shows the K-NN classifier versus the accuracies by varying K and combining various features.

Fig. 2 shows the performance of the features considered namely the SC feature, NSNR, and the Maximum Cross Correlation (MCC) value. In addition, we used various combinations of features to test, understand and verify which features contain complementary information. Using the features isolated, NSNR and SC have similar (75%-80%) performance across different values of K. However, it is interesting to observe that by combining the NSNR and SC features, we see an increase in classification accuracy. This suggests that these two features contain complementary information. While the latter contains information about spectral similarity and totally ignores the short-term phase differences and amplitude related information, the former only takes into account phase shifts and scalings into computing the similarity of the two signals. Including all the features in the classifier, the selection accuracy is 83%. While accuracy represents an operating point closer to the priors of the data, it is interesting to note that at Equal Error Rate (EER) the error for NSNR is much lower than the SC and MCC features (32%, 38% and 37% respectively). The experiment with the lower EER (30%) was when combined all features. All EER results correspond to $K = 11$ of KNN which is the optimal point of the development set.

Since the NSNR feature depends on the accurate estimation of h, it is interesting to compare the performance of different approaches in estimating h and, in addition, we consider the performance of different combinations of features using SC, NSNR, NSNR with least means squares filter estimation (LMSF) and (LMSF)-Extended in which we are using more iterations to estimate the filter h. For the (LMSF)-Extended, we are using 10 iterations over each segment for estimating each time the filter h. As shown in Fig. 3, the filter estimated with (LMSF)-Extended gives better convergence characteristics with the highest performance among all features. It is interesting to see that by combining NSNR(LMSF)-Extended with NSNR(SSF), SC and MCC, we get the best performance (84%-87%). This fact suggests that those features complement some missing information from NSNR(LMSF)-Extended. Also, due to the computational costs to estimate higher order filters for the least means square filter (i.e, the size of filter $|h|$), we did not experiment with filters higher than 50ms. This might be one reason that longer term information is not captured by NSNR(LMSF) features. The results also suggest that some of that information is captured by the the rest of the features namely NSNR(SSF), SC and MCC.

Overall, our method depends on a set of training data so that the algorithm learns from human annotations. The results indicate that the features provide discrimination up to 87% using the KNN [14] classifier for the data set considered. In addition, the NSNR features which are motivated by Signal-to-Noise ratio ideas have been the best performing.

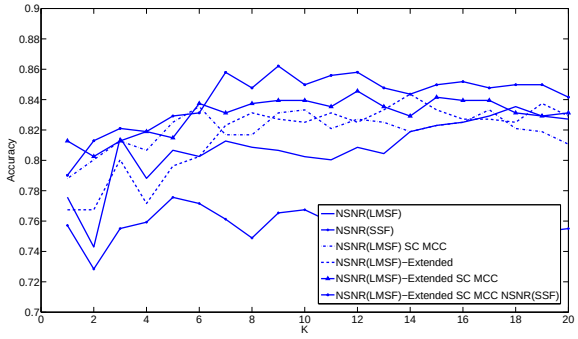

Fig. 3. This figure shows the K-NN classifier versus the accuracies by varying K and combining various features. In particular, the main focus is to compare different approaches in estimating the filter h which relates the source with the target noise.

6. CONCLUSIONS AND FUTURE WORK

In this work, we focus on identifying clean bilingual speech signals by exploiting the relation between the background noise in two audio streams. We proposed various features to capture this information. The first feature captures the spectral correlation (SC) of the bilingual audio streams and aims to measure their relationship by spectral similarity. The second feature, called the Noise to Speech and Noise Ratio (NSNR) aims to model the relation using a signal plus noise model of two audio streams. NSNR requires an estimation of a filter h and we have proposed two methods to estimate h which vary in speed and performance. Our best performing approach delivers accuracies up to 87% in classifying clean and noisy speech.

For future work, we aim to use phonetic information in each language to improve the performance of identifying clean and noisy speech regions. We also want to employ an automatic speech recognizer to align the text with the high SNR speech signals.

7. REFERENCES

[1] A. Tsiartas, P. Georgiou, and S. Narayanan, "A study on the effect of prosodic emphasis transfer on overall speech translation quality," in *Proc. IEEE ICASSP*. IEEE, 2013.

[2] A. Tsiartas, P. G. Georgiou, and S. Narayanan, "Toward transfer of acoustic cues of emphasis accross languages," in *Proc. Interspeech, Lyon*, 2013.

[3] K. Takatomo, S. Sakriani, T. Shinnosuke, N. Graham, T. Tomoki, and N. Satoshi, "A method for translation of paralinguistic information," *Proceedings IWSLT 2012*, 2012.

[4] P. Koehn, "Europarl: A parallel corpus for statistical machine translation," in *Proceedings of the tenth machine translation summit*, 2005, vol. 5.

[5] A. Tsiartas, P. Ghosh, P. G. Georgiou, and S. Narayanan, "Context-driven automatic bilingual movie subtitle alignment," in *Procedings of Interspeech*, Brighton, UK, 2009, pp. 444–447.

[6] R. Sarikaya, S. R. Maskey, R. Zhang, E. Jan, D. Wang, B. Ramabhadran, and S. Roukos, "Iterative Sentence–Pair Extraction from Quasi–Parallel Corpora for Machine Translation," in *Procedings of Interspeech*, Brighton, UK, 2009, pp. 432–435.

[7] K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics (ACL)*, 2002, pp. 311–318.

[8] C. Schlenoff, BA Weiss, M.P. Steves, G. Sanders, F. Proctor, and A. Virts, "Evaluating Speech Translation Systems: Applying SCORE to TRANSTAC Technologies," in *Proc. of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*, 2009.

[9] T. Takezawa, E. Sumita, F. Sugaya, H. Yamamoto, and S. Yamamoto, "Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world," in *Proceedings of LREC 2002*, 2002, vol. 1, pp. 147–152.

[10] A. Tsiartas, P. Ghosh, P. G. Georgiou, and S. Narayanan, "Bilingual audio-subtitle extraction using automatic segmentation of movie audio," in *the Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, Prague, Czech Republic, 2010, pp. 5624–5627.

[11] M. Xu, L. Y. Duan, J. Cai, L. T. Chia, C. Xu, and Q. Tian, "HMM-based audio keyword generation," *Advances in Multimedia Information Processing - PCM 2004: 5th Pacific Rim Conference on Multimedia.*, 2004.

[12] Simon Haykin, *Adaptive Filter Theory (4th Edition)*, Prentice Hall, 2001.

[13] Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.

[14] Cover T. and Hart P., "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.