Improving Acoustic Models with Captioned Multimedia Speech

Photina Jaeyun Jang and Alexander G. Hauptmann

Computer Science Department
Carnegie Mellon University
Pittsburgh, Pennsylvania, 15213
U.S.A.
{jaeyun,alex}@cs.cmu.edu

Abstract

Speech recognition can be used to create searchable transcripts for audio indexing in digital video libraries. Large amounts of hand-transcribed speech training data are required to build or improve acoustic models of highly accurate speech recognition systems using current technologies. We present a technique to use television broadcasts with closed-captions as a source for large amounts of automatically extracted and accurately transcribed speech for improving acoustic models. The errorful closed captioned text is aligned with the also errorful speech recognition output and matching segments are used with each corresponding audio segment as acoustic training data to improve the speech recognition system. Our technique automatically extracted 131.4 hours of transcribed speech and improved the word error rate of our currently best speech recognition system (Sphinx-III) from 32.82% to 31.19%. A speech recognizer trained exclusively on 70.7 hours of this automatically transcribed speech produced a word error rate of 32.7%.

1. Introduction

We first give a brief overview of the Informedia Digital Video Library System [Christel et al, 1996]. Then we describe the sources of speech and transcription information and the process for extracting reliable transcribed speech from TV broadcasts in the next section, followed by sections describing the acoustic model training procedure, the speech decoding process and finally the results and analysis.

1.1. The Informedia Digital Video Library

The Informedia Digital Video Library Project [Christel et al, 1996] at Carnegie Mellon University is creating a digital library of text, image, video and audio data, whose entire content can be searched for rapid retrieval of material relevant to a user query. Through the integration of technologies from the fields of natural language understanding, image processing, speech recognition and information retrieval, the Informedia Digital Video Library system allows a user to explore the multimedia data both in depth and in breadth. Speech recognition is a critical component in the automated library creation process. Automatic speech recognition is used to transcribe the audio portion of the video data stored in MPEG format in the library [Hauptmann & Wactlar 1997, Witbrock & Hauptmann, 1997]. This speech recognizer generated transcript forms the basis for the main text search and retrieval functions.

1.2. Speech Recognition

For all current speech recognition systems, it is important to have as much representative transcribed speech as possible to train the acoustic models in order to maximize the performance of the speech recognition system [Bourlard et al, 1996]. Besides extensive speech data collection, we also face the bottleneck of manually annotating or transcribing the collected speech corpora. For supervised learning systems, such as neural networks or Hidden Markov Model (HMM) based speech recognition systems, a highly accurate transcription of the training data is required. Manual speech transcription is very tedious and expensive but still subject to human errors. The currently largest available collection of accurately transcribed speech data consists of almost 200 hours of broadcast news provided by the Linguistic Data Consortium [LDC, 1998], but this data is still not error free.

We are presenting a method for solving both problems of expensive speech data collection and expensive human annotation of speech. Every day, there is new speech data available on television along with human transcribed closed-captions. The challenge is to make use of this unreliable data for building high quality acoustic models.

2. Sources of Speech and Preliminary Transcriptions

The Informedia Digital Video Library records 1.5 hours of TV broadcasts daily in MPEG format, together with closed-captioning, if available. To date, the library contains over 1400 hours of video. While it would be possible for us to directly use broadcast TV 24 hours a day to extract our data, for replicability we have used only the video archived in the Informedia Digital Video Library. For this experiment we utilized 709 hours of initial raw data from the library.

For comparison, we also examined 100 hours of manually transcribed training data provided by the Linguistic Data Consortium [LDC 1998], which was used as a baseline contrast as well as “seed data” in our experiments. After eliminating silences and untranscribed sections, the LDC 1997 HUB-4 data was reduced to 66 hours of transcribed speech.

2.1. Extracting Audio Samples

From each stored MPEG, we extract the MPEG audio stream. The MPEG audio is uncompressed and downsampled to 16 kHz. The audio is further processed into mel-frequency coefficients to represent the audio signal as feature vectors of 12 values every 10 milliseconds. Two sources of transcriptions are available for each MPEG audio: the broadcast closed-captions and the initial output of the SPHINX speech recognizer.

2.2. Closed-Caption Transcripts

Closed-captions are frequently broadcast together with the video. Closed-captioned data contains on average 17% word error rate in the broadcast news domain. In addition to spelling errors, insertions and omissions, the captions usually are completely absent if there is text on the screen or during commercial breaks. Occasionally the human captioner will also fall behind and "rephrase" the speech to summarize what had been spoken, in order to catch up with the realtime captioning process. A high word error rate in the transcription is likely to corrupt the existing acoustic model, decreasing rather than increasing the accuracy of the speech recognition system.

2.3. Speech Recognizer Hypotheses

A second independent source of transcription data is the initial output of a speech recognizer. We used the lower accuracy, but faster SPHINX-II speech recognition system to get a quick and rough speech recognizer transcript of the video data Since a large amount of data has to be processed, recognizer speed is more important than quality of transcript in this pass. The initial speech recognition word error rate in this broadcast news domain were approximately 30%~60%.

2.4. Extracting Accurately Transcribed Speech Segments

To obtain an accurate transcription of an audio stream, we align the closed-captions and the speech recognition and select segments where sequences of three or more words are the same in both recognizer output and closed captions. Figure 1 shows a selection of the words "I think there are" from the audio stream, using the associated speech recognition and closed captions. The effect of this selection is to verify the closed captions using the speech recognition as an independent source. This can be viewed as a form of mutual confirmation or as a binary confidence measure. The corresponding sequences function as the training corpus for supervised learning in any training data dependent speech recognition systems. The quality of the extracted training segments were evaluated by a human and judged to be correct with a word error rate of 8.6%. Our initial training corpus comprised 709.8 hours of video, with associated closed-captions. From this training corpus we extracted 18.6 % (131.4 hours) of the data as usable transcriptions.

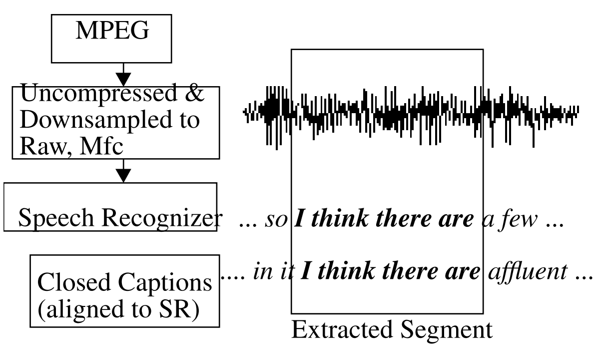

Figure 1: Extracting an audio segment with a reliable transcription from the (MPEG) video using the closed captions aligned with a rough speech recognition transcript.

3. Acoustic Model Training

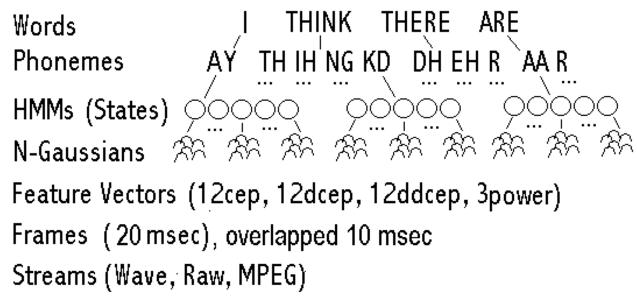

Figure 2: Hierarchy of HMM-based acoustic model training units.

The process of training HMM-based acoustic models [Lee, 1990] for the Sphinx-III speech recognition system is shown in Figure 2. Words are translated into phoneme sequences. Each phoneme is represented by a 5-state HMM, with predefined transitions. A state transition is triggered by a 10 millisecond frame of observations in the speech signal. The vector of observations at each frame is computed through 39 features derived from the raw audio signal sampled at 16bits/16kHz. The raw samples are converted into 12 mel-frequency cepstral coefficients per frame. The 12 differenced cepstral coefficients and 12 differenced differenced cepstral coefficients, together with the 3 power coefficients form a vector of 39 observed features at each 10 msec frame. Each HMM state is modeled through a mixture of N gaussian densities (means and variances). During training we derive the gaussian density means and variances that best model each state. Since there are 5 states for each triphone, where a triphone is a phoneme considered in the context of its left and right phoneme neighbor, there are more possible states than we can reasonably train. We limit the number of states (senones), where different states from different phonemes can be tied together, if they are sufficiently similar, and modeled as one of the limited number of senones.

The configuration of our acoustic model training is identical to the one of the official 1997 HUB-4 DARPA evaluation [Seymore et al, 1998]. We used 6000 senonically tied states, where each state consists of a mixture of 20 gaussian densities. The acoustic model was trained on context-dependent phonemes (triphones).

Our acoustic model adaptation process consists of four basic steps: Forced Alignment, Codebook generation with Clustering & Gathering, and Model Reestimation using the Baum-Welch algorithm. We built fully-continuous hidden markov models (HMM) for the acoustic model of the Sphinx-III speech recognizer.

3.1. Forced Alignment

Forced alignment models filled pauses explicitly as phonemes, to prevent these non-speech noises from corrupting the training of actual phonemes. Secondly, the forced alignment stage segments the speech data into state segments to align for Baum-Welch training for the later model reestimation stage. The forced alignment thus assigns each feature vector in the training data segment to a single state in the HMM model. We force-align the transcript with its speech data to obtain a more adapted transcript to the speech data by inserting noise phones, silences, filler words, or filled pauses between the content words. Figure 3 shows an example of an original transcript and the insertion of non-speech sounds in the force-aligned transcript.

Original Transcript: I THINK THERE ARE
Force-aligned Transcript: <s> I ++SMACK++ THINK
<sil> THERE ARE ++INHALE++ </s>

Figure 3: Transcript before and after force-alignment

Forced alignment may reject training utterances where the feature vectors cannot be adequately aligned to a sequence of states, e.g., if the number of phonemes in the training transcript is much larger than in the audio feature vector. This rejection reduced the amount of training data from the baseline LDC HUB-4 data from 66 hours to 62.8 hours. The automatically extracted "CCTRAIN" data was reduced from 131.4 hours to 111.5 hours. The data yield from the original 709.8 hours was therefore 15.6%. Together the amount of usable data totalled 174.3 hours available for further acoustic model training.

3.2. Codebook Generation

There are two steps in the codebook generation: *gathering* the feature vectors together and *clustering* them into different densities. During the gathering step, we aggregate all the feature vectors of all the frames in the training data. The clustering step uses K-means clustering to partition the training vectors into different gaussian densities. Each density is represented by its mean and variance.

3.3. Model Reestimation

As is typical for HMM-based speech recognition systems, we use the Baum-Welch algorithm, also known as the "forward-backward" algorithm to model the observations in the training data through the HMM parameters. During this step we reestimate the mixing weight, transition probabilities and mean and variances parameters. After each Baum-Welch reestimation iteration, a normalization step takes place. This combined Baum-Welch and normalization iteration is repeated until parameter convergence has been achieved.

4. Speech Decoding

The speech decoding was done with Viterbi search, which finds the state sequence that has the highest probability of being taken while generating the observation sequence. Since the purpose of this experiment was to evaluate the effectiveness of the new acoustic model, only a single search pass was performed. To get optimal performance from the system, multiple passes can be used, where maximum likelihood linear regression can be performed between passes for speaker adaptation. The 1997 HUB-4 evaluation system consisted of three such decoding passes with acoustic adaptation steps between each pass. For each pass A* beam search is applied for the Viterbi decoding pass and DAG search is applied for the best path pass of the Viterbi word lattice. Finally, N-best hypotheses are generated and rescored.

4.1. Language Model

In addition to the acoustic model, the decoding system uses a language model which reflects the likelihood of different word sequences. The language model is trained on broadcast news language model training data and built using a 64k vocabulary [Seymore et al, 1998]. By increasing the contribution of the language model relative to the acoustic model, we make the system less sensitive to acoustic errors and more sensitive to word transition probabilities. In general the recognition accuracy is highest when decoding with N-best rescoring, followed by DAG best-path search and then the single 1-best Viterbi decoding. Multipass decoding is applied to obtain higher accuracy with contributions by better search through the hypothesis lattice and with smoothed language models. Our baseline HUB-4 system has an overall WER of 24% when all passes are applied [Seymore et al, 1998].

5. Evaluation Results

In the following experiments, we only decoded with a *single pass* Viterbi search and without MLLR (Maximum Likelihood Linear Regression) acoustic adaptation, since we intended to estimate the improvement achieved on the acoustic model only compared to the baseline model, where single pass decoding is sufficiently indicative. The word error and word error rate (WER) is computed as follows:

Word Error = # Insertions + # Deletions + # Substitutions
Word Error Rate =Word Error / Total # of words in Test set

The test set for our results came from the 1996 DARPA HUB-4 broadcast news development set (Dev'96 test set) [Pallett, 1997], with 409 utterances totalling 16456 words. The Dev'96 test set used for decoding can be classified into the following multiple acoustic conditions:

F0: Baseline Broadcast Speech
F1: Spontaneous Broadcast Speech
F2: Speech Over Telephone Channels
F3: Speech in the Presence of Background Music
F4: Speech Under Degraded Acoustic Conditions
F5: Speech from Non-Native Speakers
Fx: All other speech

Figure 4: HUB-4 Acoustic Conditions

Different acoustic models suitable for reduced bandwidth speech are used for the telephone (F2) conditions. We excluded the F2 condition because we are improving a full-bandwidth mixture gaussian model, not a narrow-bandwidth-telephone model.

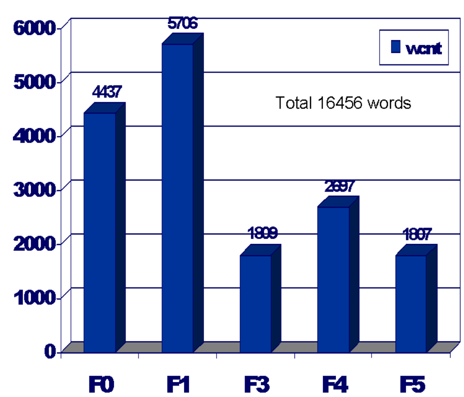

Figure 5: Word counts in each acoustic condition in our Dev'96 test set.

The different acoustic conditions within the given Dev'96 test set were not evenly distributed. The number of words in each acoustic category varied widely as shown in Figure 5 above.

Figure 6 shows the overall result statistics (including all acoustic conditions together) on the Dev'96 test set for both the baseline HUB-4cmu system and our improved system (CCTRAIN) which was built by estimating the acoustic model parameters from both the HUB-4 training data and our automatically derived training corpus based on closed-caption data and initial recognition transcripts. For all three different language model weights, our CCTRAIN acoustic model achieved improvements over our baseline system.

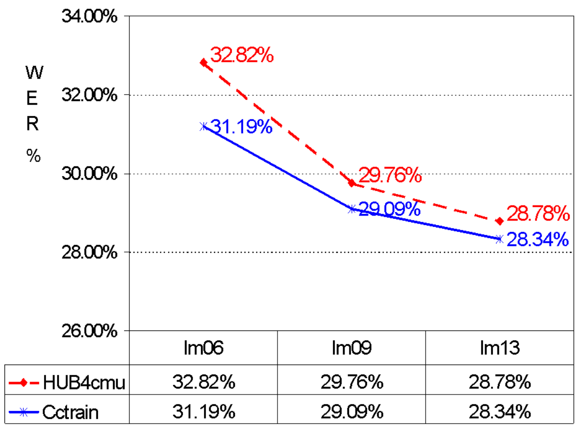

Figure 6: Recognition Word Error Rate (WER) of single pass speech decoding for the HUB-4cmu (upper dashed line) baseline models compared to the CCTRAIN (lower solid line) models for the Dev'96 test set, with three different language model weights.

The absolute percentage decrease is depicted in Figure 7 again with different language model weights of 6, 9 and 13.

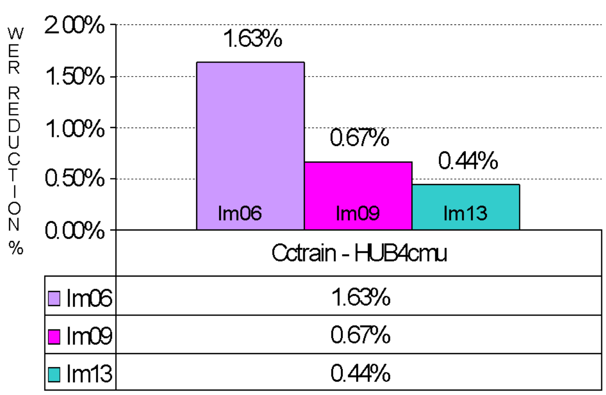

Figure 7: The absolute *difference* in Word Error Rate (WER) with single pass speech decoding between the CCTRAIN system and the HUB-4cmu baseline system using three different language model weights.

We see that higher language model weights make it harder to estimate the degree of improvement on the acoustic model. This is due to the increasing contribution of the language model to the speech recognition accuracy during the decoding process. The two figures show that the overall speech recognition accuracy is improved by adding our CCTRAIN corpus to the manually transcribed HUB-4 training data.

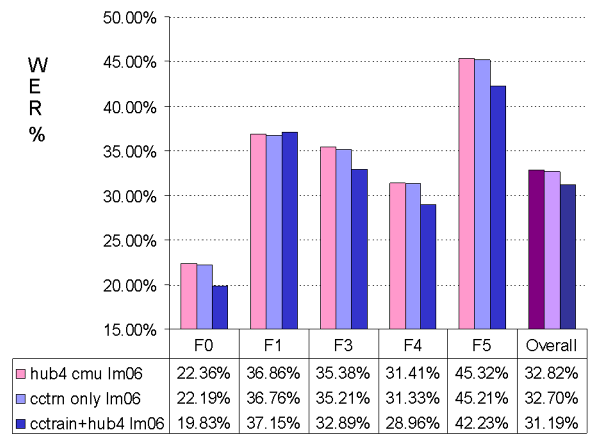

Figure 8: Word Error Rate of CCTRAIN built from 174 hours of speech consisting of our extracted segments and HUB-4 data (3rd bar of each column), HUB-4cmu built from 62 hours of HUB-4 data (1st bar of each column) and CCTRAIN built from 70.7 hours of extracted segments only (2nd bar of each column) of training data, for each of the different acoustic conditions: F0 = clean broadcast speech, F1 = spontaneous speech, F3 = speech with background music, F4 = speech under degraded acoustic conditions and F5 = speech from non-native speakers.

Figure 8 shows the word error rate for the individual acoustic conditions. The 2nd bar of each column, labeled CCTRAIN-only shows the word error rate of the CCTRAIN system built only from 70.7 hours of our extracted segments without any other data. Thus the number of hours of training data are roughly comparable to the baseline HUB-4 system, which word error rate is shown in the 1st bar using 62 hours of manually transcribed data. The CCTRAIN model shown in the third bar of each column was trained on 174 hours of both automatically derived and manually transcribed training data. We see that the CCTRAIN model provides superior results on F0, F3, F4 and F5. The overall WER is shown in the last column. The CCTRAIN model, which outperforms our baseline HUB-4cmu system was obtained by simply increasing the amount of training data extracted with our automatic method.

Different acoustic conditions result in widely varying word error rates, reflecting the difficulty of the condition for speech recognition as well as the varying amounts of training data in each condition. For this and subsequent figures, the language model weight was set to 6 to demonstrate the effect of the acoustic component, with relatively little influence provided by the language model.

6. Analysis

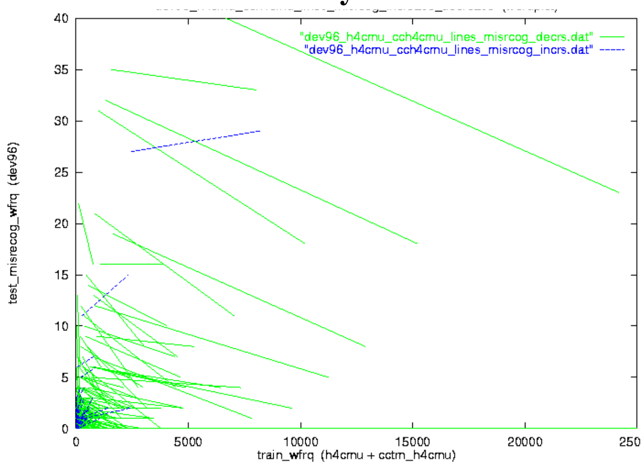

Figure 9: Effect of word frequency in the different training corpora (x-axis) on the number of errors (y-axis). Each word in the test data is plotted based on its errors in the HUB-4cmu baseline recognition given its frequency during HUB-4 training (left point of each line) compared to the CCTRAIN frequency and recognition errors for that word (right point of each line). An ascending line shows an increase in the error rate for a word (17.4% of the Dev'96 test set words), while a descending line reflects an improvement in error rate by the CCTRAIN model (82.6% of the Dev'96 test set words).

Figure 9 shows words that were present in both the training corpus as well as the test set. The x axis represents the number of times a word was present in the training set and the y axis is the number of errors for that same word in the test set. Each word is represented by two points, connected by a line. The first point represents the number of training instances for this word in the HUB-4 training corpus and the number of errors for this word during the HUB-4cmu baseline recognition. The second point is based on the CCTRAIN corpus. The CCTRAIN point also is determined by the number of training instances in the CCTRAIN corpus on the x axis and the number of errors for that word during the CCTRAIN decode on the y axis. Since the CCTRAIN training corpus included all the HUB-4 words as well as new training material, the CCTRAIN point is always on the right end of a line, while the HUB-4cmu point is on the left end of a line. We divided the data into two sets of words, those words where the errors decreased with more training data and those where the errors increased after adding the additional CCTRAIN data. The descending (gray) lines show words which were recognized with fewer errors in the CCTRAIN model while ascending (boldface) lines show words which were recognized worse after CCTRAIN. 17.4 % of the words increased in error and 82.6 % of the words showed improvement in error with CCTRAIN models.

Figure 10 below shows all the words in the test data, plotted as a function of their frequency in the training data. To show the trend of the individual points, a line was plotted that represents the average word error rate in histogram bins, where each bin combines 30 adjacent frequencies. From this trend line in Figure 10, we see that at about 2000 words, the number of training instances no longer helps to reduce the word error rate. Up to about 2000 words, each additional training word helps to reduce the recognition error

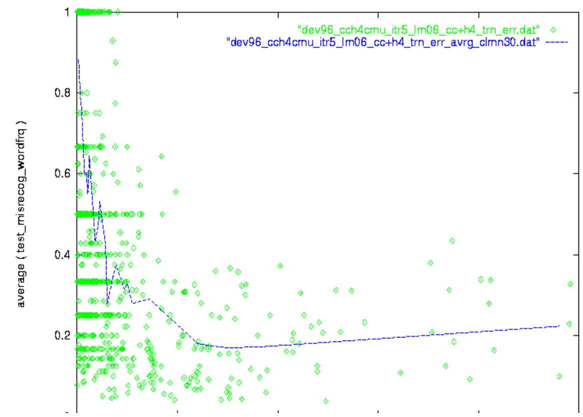

Figure 10: The word error rate in the CCTRAIN condition for each word in the Dev'96'96 test set, plotted as a function of the number of training instances for that word. The 'trend' line shows the averaged error rate for words within a histogram bin of 30 similarly frequent words.

7. Conclusion

Automatically deriving large amounts of accurately transcribed speech training data can help improve the speech recognition accuracy for an already highly optimized, state-of-the-art speech recognition system. There is a potentially unlimited amount of this data freely available from TV broadcasts 24 hours every day. Our analysis also indicates that it would be sufficient to *selectively* train on limited instances of each word for acceptable recognition performance.

REFERENCES

1. Bourlard, H., Hermansky H., Morgan, N. "*Copernicus and then ASR Challenge -- Waiting for Kepler*", Proceedings of the DARPA Speech Recognition Workshop, 1996.
2. Christel, M., Kanade, T., Mauldin, M., Reddy, R., Stevens, S., Wactlar, H., "*Techniques for the Creation and Exploration of Digital Video Libraries*", Multimedia Tools and Applications (Volume 2), Borko Furht, editor. Boston, MA: Kluwer Academic Publishers, 1996.
3. Hauptmann, A.G. and Wactlar, H.D., "*Indexing and Search of Multimodal Information"*, International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1997.
4. LDC 1998: http://morph.ldc.upenn.edu/ldc/frame.html
5. Lee, K.F., "*Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition*", Waibel and Lee(eds), Readings in Speech Recognition, Morgan Kaufmann, 1990.
6. Pallett, D. "*Overview of the 1997 DARPA Speech Recognition Workshop*", Proceedings of the DARPA Speech Recognition Workshop, 1997.
7. Seymore, K., et al "*The 1997 CMU Sphinx-3 English Broadcast News Transcription System*", Proceedings of the 1998 DARPA Speech Recognition Workshop, 1998.
8. Witbrock, M.J. and Hauptmann, A.G., "*Artificial Intelligence Techniques in a Digital Video Library*", Journal of the American Society for Information Science (JASIS), 1997.