

Automatic Discovery of Salient Segments in Imperfect Speech Transcripts

Dulce Ponceleon and Savitha Srinivasan
IBM Almaden Research Center
650 Harry Road, San Jose, CA 95120 USA
E-mail: {dulce,savitha}@almaden.ibm.com

ABSTRACT

This paper addresses the problem of automatic detection of salient video segments for real-world applications such as corporate training based on associated speech transcriptions. We present a novel segmentation algorithm based on automatic speech recognition (ASR) applied to the audio track of the video. Our feature set consists of word n -grams extracted from the imperfect speech transcriptions. We use a two-pass algorithm that combines a boundary-based method with a content-based method. In the first pass, we analyze the temporal distribution and the rate of arrival of features to compute an initial segmentation. In the second pass, we detect changes in content-bearing words by using the content-bearing features as queries in an information retrieval system. The content-based second pass validates the initial segments and merges them as needed. Variations in the structure of the audio/video content, and the accuracy of ASR have an impact on the feasibility of the segmentation task. For realistic data we observe that we can identify content-rich segments of the audio. In the best scenario a high-level table-of-contents is generated and in the worst scenario a *single* salient segment is identified. We illustrate the algorithm in detail with some examples and validate the data with manual segmentation boundaries.

1 INTRODUCTION

The rapidly growing amount of on-line information makes it necessary to support browsing of information where the underlying conceptual structure is revealed. This complements query driven approaches that focus on content specific queries for information retrieval. The existence of both, manual and automatic text categorization schemes on the World Wide Web provide compelling evidence that such schemes are both, important and useful. Despite the connectivity offered by the web, the primary reason that audiovisual data is not ubiquitous yet is the set of

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CIKM'01, November 5-10, Atlanta, Georgia, USA.

Copyright 2001 ACM 1-58113-436-3/01/0011...\$5.00.

challenges encountered in dealing with the unstructured, spatio-temporal nature of video. Therefore, cataloging and indexing of video has been universally accepted [3, 7] as a step towards enabling intelligent navigation, search, browsing and viewing of digital video.

In this paper, we focus on the segmentation of an audio/video source into topically cohesive segments based on automatic speech recognition (ASR) transcriptions. A valuable tool in this domain is the ability to automatically segment the audio/video source and assign a meaningful textual label to the segment. These labeled segments may then be used as a navigational tool, as indices that yield a higher semantic level of indexing in comparison with keyword indexing, and as discriminatory/ranking information towards automatic generation of video/audio summaries.

2 RELATED WORK

Automatic segmentation and labeling of unstructured documents has been extensively studied in the following forums: statistical machine learning, topic distillation on the web and the DARPA sponsored Topic Detection and Tracking (TDT) track. Machine learning literature refers to this task as text categorization, and partitions it into supervised and unsupervised methods. Supervised text categorization refers to the automatic assignment of topics to text collections when sample training data is available for each topic in a predefined topic set. Unsupervised text categorization methods do not use a predefined topic set with sample training data; instead, new documents are assigned topics following an unsupervised training phase. Query driven topic identification, often referred to as topic distillation, has received a lot of attention with the ubiquity of the web [5]. These approaches are based on connectivity analysis in a hyperlinked environment, together with content analysis to generate quality documents related to the topic of the query.

TDT investigates the state of the art in finding new events in a stream of broadcast news stories [1, 5]. The TDT project builds on and extends the technologies of Automatic Speech Recognition and Document Retrieval with three major tasks: (1) segmenting a stream of data into topically cohesive stories. The data comprises news wire and textual

transcriptions (manual, automatic, or both) of audio (2) detecting those news stories that are the first to discuss a new event occurring in the news and (3) given a small number of sample news stories about an event, finding all following stories in the stream. In this context a topic is defined to be “*a seminal event or activity, along with all directly related events and activities*”. The segmentation task is performed on several hundred hours of audio either using the audio signal, or its textual transcription. The tracking task associates incoming stories with target topics defined by a set of training stories that discuss the topic.

Prior to the first TDT task in 1997, the work on text segmentation was based on semantic word networks, vector space techniques from information retrieval [8], and decision tree induction algorithms. After the TDT pilot study, several new techniques were successfully applied to text segmentation. One approach was based on treating topic transitions in text stream as being analogous to speech in an acoustic stream. Classic Hidden Markov Model (HMM) techniques were applied in which the hidden states are the topics and observations are words or sentences. A second approach has been to use content-based local context analysis (LCA) where each sentence in the text is run as a query and the top 100 concepts are returned. Each sentence is indexed using offsets to encode positions of the LCA concepts and these offsets are used as a measure of vocabulary shifts over time. A third approach has been to combine the evidence from content-based features derived from language models, and lexical features that extract information about the local linguistic structure. A statistical framework called feature induction is used to construct an exponential model that assigns to each position in the text a probability that a segment boundary belongs at that position. In general, clustering methods such as agglomerative clustering have been used for the segmentation task [6]. Initially, a fixed length window is considered to be a cluster, and a similarity score is computed for all pairs of neighboring clusters. If the most similar pair of clusters meets a threshold, the two clusters are combined to form a new cluster. This process is repeated until no pairs of neighbors meet the similarity threshold. Applications that incorporate some form of automatic video categorization based on an analysis of the speech transcripts have been focused on broadcast news content. The Informedia Digital Video Library includes a supervised topic-labeling component where a kNN classification algorithm is used to categorize incoming stories into one of 3000 topic categories [7]. An HMM approach has been shown to be better than a naive Bayesian approach for the classification of news stories into a static set [12].

From our survey of the literature, much of the research addresses topic discovery for large document collections. The problem we address bears the largest similarity to the TDT segmentation task. However, there are several

important differences that are unique to our problem domain: TDT is fed with a relatively homogeneous corpus of broadcast news audio. The notion of a *story* and therefore, the associated segment is relatively well defined. In contrast, our content comes from various distributed learning and corporate training videos, where the duration of audio ranges between 15 and 90 minutes each. Segmentation based on cohesiveness of topics can be subjective - and is not as unambiguously defined as in news stories. Initial TDT results on imperfect transcripts obtained from speech recognition have not been as good as those on carefully transcribed broadcast news text. Further, the accuracy of our speech recognition transcripts vary from 35-60% word error rate depending on fidelity of audio, background noise, and professional versus amateur speaker.

Our approach is similar to the approaches used in TDT in that we combine content-based methods with boundary-based methods to segment imperfect speech transcripts. However, our approach is unsupervised, and requires no training data. Additionally we target the identification of salient segments. Our novel contributions are as follows: we use word n-grams consisting of noun phrases as our feature set to alleviate the problem of noisy features due to inaccurate speech transcriptions. We consider the temporal proximity of the features, and the changes in the rate of arrival of the features as parameters to trigger the first pass of the segmentation. In the second pass, we use a content-based method similar to local context analysis to merge the segments. The rest of this paper is organized as follows: section 3 describes background in speech recognition, section 4 describes our algorithm in detail, and section 5 illustrates the algorithm using examples.

3 SPEECH RECOGNITION SYSTEMS

Speech recognition systems output the most probable decoding of the acoustic signal as the recognition output, but keep multiple hypotheses that are considered during the recognition process. The multiple hypotheses at each time, often known as N-best lists, provide grounds for additional information for retrieval systems. Recognition systems generally have no means to distinguish between correct and incorrect transcriptions, and a word-lattice representation (an acyclic directed graph) is often used to consider all hypothesized word sequences within the context. The path with the highest confidence level is generally output as the final recognized result. It is often known as the 1-best word list. Speech recognition accuracy is typically represented as word error rate (WER) defined to be the sum of word insertion, word substitution and word deletion errors divided by the total number of correctly decoded words. It has been shown that WER can vary between 8-15% and 70-85% depending on the type of speech data and tuning of the recognition engine [6, 9]. The 8-15% error rates typically correspond to standard speech evaluation data and

the 70-85% corresponds to “real-world” data such as one-hour documentary and commercials. Retrieval on transcripts with WER of 8-30% have been reported to yield an average precision of 0.6-0.7. However, for real-world audio with high WER of 70-80%, the precision and recall have been reported to drop dramatically to 0.17 and 0.26 respectively [6]. The NIST sponsored Text Retrieval Conference (TREC) has implemented a Spoken Document Retrieval track to search and retrieve excerpts from spoken audio recordings using a combination of automatic speech recognition and information retrieval technologies. The TREC Spoken Document Retrieval task has conducted a set of benchmark evaluations and has demonstrated that the technology can be applied successfully to query audio collections. The best retrieval results report a precision between 0.6 and 0.7, and yield 82-85% overall performance of a full-text retrieval system.

4 SALIENT SEGMENT SELECTION ALGORITHM

We introduce the framework for our algorithm by first identifying the types of boundary conditions between segments as well as the density within a segment. We examine the speech data from several speakers - both, professional and amateur. Figure 1 illustrates two broad categories of speakers with different speech patterns; 1a shows a relatively homogeneous rate of arrival of features where the words are being spoken at a uniform rate. 1b shows a non-uniform rate of arrival of features where we first see a rapid onset of features, followed by a relatively slow pace, and finally, followed by another rapid onset. Firstly, segments with a high rate of arrival of features are considered to be content-rich. This is a reasonable assumption since ngrams features are generally content bearing. Secondly, we consider the change in rate of arrival of features as a *potential* indicative of shifts in topic. This is based on assumption that when the speaker starts a new topic, he is likely to use technical terms or content-bearing words at a high rate. However, it is also possible to observe a sharp change in the rate of arrival of features, while discussing the same topic. The second pass of our algorithm addresses this issue.

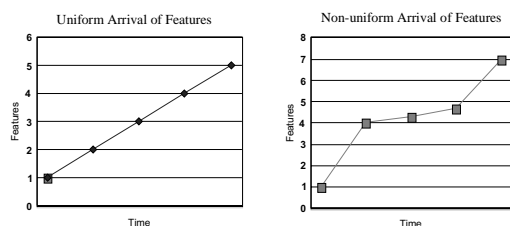


Figure 1: Speaker Categories and Speech Patterns

Our algorithm consists of the following steps:

1. Feature Extraction: Extract word-ngram features using local and global frequency counts from the entire transcript.
2. Information Retrieval: Run each word-ngram as a query against a combined word and phonetic index to obtain the times of occurrence and corresponding probabilistic retrieval scores [11].
3. First pass segmentation: Combine the arrival rate of the features with the observed gaps in the temporal distribution of the features to compute the initial segment boundaries.
4. Second pass segmentation: Validate or update initial segment boundaries using a content-based method by examining the temporal distribution of features *within* each segment and the intersection of the support features set of contiguous segments. Segment Ranking: probabilistic ranking of segments based on the density of the segment, the rank of every feature and the score given by the retrieval step for each feature occurrence.

4.1 Feature Extraction

We select a single feature set of high frequency word ngrams along a time line. The high frequency word ngrams are identified in the speech transcript text by a cascaded sequence of natural-language processing tools and a technical term extractor. The literature on multiword automatic term recognition agrees that multiword terms are mainly noun phrases containing adjectives, nouns and occasionally prepositions [10]. Semantically, multi-word technical terms refer to important concepts defined or discussed in the document and in the domain covered by the corpus as a whole. Linguistically, they form a subset of the noun phrases occurring in the document: they are lexical, that is, their meaning is not usually derivable from the meanings of their component words (e.g., "landing gear" or "carbon fiber"). These phrases do not usually undergo other noun-phrase variations, such as insertion or deletion of modifiers ("landing" or "the carbon" are not likely to appear). Technical terms usually consist of one or more simple modifiers preceding the head noun ("computing power", "control system") and infrequently contain one or more prepositional phrases following the head ("fly by wire"). Conjunctions ("and" or "or") verbs and adverbs in technical terms are rare. The algorithm that identifies technical terms scans the document tokens for all sequences of words that match grammatical structures in which technical terms tend to appear. We use the regular expression in [10] to extract such technical terms or ngrams. In words, a candidate term is a multi-word noun phrase; and it either is a string of nouns and/or adjectives, ending in a noun, or it consists of two such strings, separated by a single preposition. The part of speech of each word is determined by lookup in an online dictionary

of English. Since many words in English have more than one part-of-speech, the procedure may extract sequences that are not really noun phrases, such as "price rose". The feature selection represents the original speech transcript as a sparse, single dimensional vector of word ngram features along the temporal dimension. The temporal component of the word ngrams implicitly models events such as non-speech or music that tend to be longer when a topic change occurs.

4.2 Information Retrieval

The information retrieval phase of the algorithm enables us to find the time stamps of all occurrences of each word-ngram feature in the video. A combined word and phonetic retrieval system retrieval based on the probabilistic formulation of term weighting using phone confusion data in a Bayesian framework supports this phase [15]. We generate a time-aligned word transcript and a phonetic transcript of the input audio using the IBM speech recognition system with a broadcast news language model. Two sets of indices are created - a word index and a phonetic index using overlapping quadphone sequences as the subword indexing unit. During retrieval, the word index is searched using the stemmed query word excluding stop words, and the phonetic index is searched using the phonetic representation of the non-stemmed query word including stop words. The search results from both indices are combined to return a single ranked list of time offsets into the audio/video stream.

4.3 First Pass Segmentation

An analysis of the arrival times of features in a video reveals that segments within a video are characterized by an almost constant rate of arrival. Therefore, we use differences in the arrival of features to drive our segmentation. This is done in several ways. We first identify gaps where there are no features. If the gap is *sufficiently large* we assume this signifies a segment boundary. Sufficiently large is a relative term and here we mean relative to the average gap in the arrival of features. To determine the average gap we compute the time difference between the arrival of the first feature and that of the last and divide by the number of features. Let G_i denote the gap between the arrival of the i -th feature and the $(i+1)$ -th feature then we consider a gap sufficiently large if

$$G_i > Kg ,$$

where g is the *average gap* and K is a constant determined by doing a frequency analysis of the gaps in the arrival of features. A typical set of frequencies indicates that a choice of five would be suitable. What we are seeking when determining K is a gap size that appears to be an outlier. Having made an initial segmentation of the audio/video we can apply the same algorithm to the segments to identify new segments. The average gap in the arrival of features is bound to be smaller for at least one segment and is typically

smaller for all segments since sparse sections have been eliminated. Consequently, the test for a sufficient gap may reveal new segments even if the same relative constant K is still used. It may be worth re-analyzing the segments to determine constants for each segment, however the results presented here do not incorporate this.

While some gaps between segments may have no features sometimes there is a protracted transition in which an occasional feature will arrive. These transition periods may be at the beginning and ends of the segment we have tentatively identified and may be elsewhere. Such transitions are characterized by an arrival rate much slower than in segments. To identify such transition periods we examine the beginnings and end of the segments identified so far and refine the start and end of the segment by eliminating any period in which the gap between features is greater than $.aK_k g_k$, where k denotes the k -th segment, g_k the average arrival rate for that segment and a is a constant, say 0.6, that defines the size of the transition gap. To identify those that may lie elsewhere we search for sparse arrival rates. This may be viewed as generalizing the definition of a gap to include j arrivals in a period greater than $mg_k j$. Obviously m needs to be smaller than K . We need only do the search for $j=1$ since larger sparse gaps with more features can be determined by extending a gap with a single feature. It is important that the search for *new* sparse sections is performed after all pure gaps have been identified and the end and beginning of those segments have been refined.

Having exhausted the search for gaps or sparse sections we turn to changes of the *rate of arrival* to search for new segments. Typically in a segment the rate of arrival of features is relatively constant. If the arrival rate changes abruptly and the magnitude of the change is significantly we denote the point of change as being the end of one segment and the beginning of another. To qualify to be a change of segment several characteristics must be true. If the arrival rate is generally erratic we conclude no further segmentation is possible. We make a similar conclusion if there is a change but it is brief. In mathematical terms, we seek data that is closely approximated by linear splines with free knots. The knots are the breaks between the segments. To identify such segments we compute a least-square fit of a linear function to the segments we have identified. No further investigation is made if it is a close fit. If the fit is not close but the arrival rate is erratic (this may be deduced by checking for changes in sign of consecutive residuals) we also do no further segmentation. Finally the segment has to be sufficiently long. It is reasonable to assume that we are interested in segments of a minimum length, say a couple of minutes. Consequently, such segments contain a minimum number of data points, say $(x+1)$. This implies that only segments with $2x$ or more date points are

considered. We could apply a general-purpose algorithm to determine the knots [11], but opted to use our own algorithm. The objective function of a general-purpose algorithm is to find the most accurate fit whereas what we want is to determine the knots. The objectives may seem identical but they are not. Consider the case in which a segment can be divided into two, and one half behaves much more like a linear function than the other. The overall error may be reduced if the knot is moved within the second section since this is the hard part to fit and making the interval shorter helps.

We apply the following algorithm to promising segments. Segments with 3x or fewer data points are identified. Such sections can have at most one knot. Separate linear least-squares fits are made to the first half and second half of the data. It follows if a free knot exists with a good linear fit to both sections then one of the halves must be a good fit. Consequently, if both fits are poor no further segmentation is made. If both fits are good then the knot is at the midpoint. Otherwise, the knot lies in the half with the poor fit. To identify where we extrapolate using the good fit to check whether it fits the adjacent data point. If it does, the fit is extended to include that data point and the process is repeated. The process must terminate since we have already ascertained a good fit to the whole data is not possible. A similar process may be applied to longer segments. For example, if a segment is 4x or less but greater than 3x then it can have at most two knots. The segment is divided into three equal parts and three fits made. One fit must be valid if a knot exists and by extending the valid fit at least one knot is identified and the segment reduced to the one knot case.

4.4 Second Pass Segmentation

The first-pass of the algorithm computes all possible candidate segments of the video content. That is, it defines segmentation at its finest granularity. After this step no further splitting of segments is performed. The second step, the content-based method, can be viewed as a validation step to improve the initial segmentation. The goal is to remove boundaries (spuriously introduced in the first step) between adjacent segments that are topically very similar. This content-based method examines the distribution of features *within* each segment. Intuitively, if the set of features that occurred in adjacent segments are nearly the same this indicates that these segments are covering the same topic, and they should be merged into one topically cohesive segment. For each segment, we define the *support set* to be the set of features that occurred in that segment. We analyze the intersection of the support set corresponding to contiguous segments, taking into account that not all features bear the same weight, and that the phonetic retrieval engine will provide different scoring for the different occurrences of a feature in the transcript. For

example, if two segments share 80% of the top three features with high retrieval scoring, then they should be merged. However, if the intersection of the support sets mainly lies on low-ranked features, we conservatively keep the segment boundaries.

4.5 Segment Ranking

We compute a relevancy score associated with each segment that provides a measure of both, the saliency of the topic discussed in the segment, and the confidence associated with the topic. The saliency is computed based on the temporal proximity of the word ngram features. The confidence associated with the topic is based on the confidence level associated with the recognition of the word ngram feature. The confidence of an ngram word feature is based on term-weighting corresponding to each feature observed in the transcript which specifies the probability of occurrence of the feature in the audio [13]. For a given segment, S_i , we define the relevance score to be:

$$R(S_i) = \left(\frac{1}{L_i}\right) \sum_{k=1}^{N_i} c(f_k) \cdot \left(\sum_{j=1}^{m_k} p(t_{kj})\right), \text{ where}$$

S_i is the i -th speech segment,

$R(S_i)$ is the rank of the i -th speech segment,

L_i is the length of the i -th speech segment,

f_k is the k -th feature within segment S_i ,

$c(f_k)$ is the relevance rank for feature f_k with respect to all the features in the transcript,

$p(t_{kj})$ is the probabilistic score given by the retrieval engine for the j -th time offset computed for query f_k ,

N_i is the total number of features in segment S_i , and

m_k is the total number of matches for feature f_k in S_i .

The formula above gives a score for each segment, where the highest scoring segment corresponds to the most important topics discussed in the video. In practice we observe that the scoring is typically dominated by the segment density.

5 EXPERIMENTAL VALIDATION

It is necessary to observe the differences between the TDT tasks and evaluation data, and our task and evaluation data in order to evaluate the results. The required condition of the TDT segmentation task was based on a 10000-word decision deferral period, i.e. the story boundary could be detected up to 10000 words after the onset of the new story. In contrast, our video lengths range from 6000 words to 30000 words total. We have been working with a corporate training data set that is being used in a realistic environment for distributed learning. Table 1 lists our test data statistics. To maintain consistency in our findings, we report results on an augmented test collection based on corporate

training/distance learning previously used to evaluate spoken document retrieval [13]. The test data consisted of two classes of videos. The first, with 35% WER, corresponds to high fidelity recording, professional speaker/non-spontaneous speech, general domain, quiet conditions). The second with 65% WER, corresponds to low fidelity recording, amateur speaker/spontaneous speech, scientific/technical domain, and noisy conditions.

At this time, we do not use a formal evaluation measure to evaluate the output of our segmentation algorithm. This is primarily because unlike broadcast news stories - we do not

have an unambiguous definition of what constitutes a segment. We observed subjectivity in the manual segmentation performed by different users. We compare our results with a manually computed segmentation. We illustrate the complete algorithm and contrast it with manual segmentation results for a representative video of 60-minute duration with a WER of 40%. We begin with some sample speech recognition transcript to provide an idea of what 40% WER translates into. For illustration purposes, Figure 2 shows text corresponding to approximately two minutes of audio from this particular video used.

Number of Videos in Test Collection	6
Total Duration of Videos	6 hours
High fidelity Recording with Professional Speaker (35% WER)	4 hour
Low fidelity Recording with Amateur Speaker (65% WER)	2 hours

Table 1: Test Data Statistics

Speech Transcript
 early room northwestern minded states and move benjie news was about a month and the first flying bonds a boost in the months to boeing hope that this plan to himself and wish you were the success of the company into the twenty-first century it had taken ten thousand people before used to design and build the plant use in the latest technology including two of the largest engines on to any airline in the world but the plane had never left the great john cash but chief test pilot who take the triple seven to be here for the first time if all went well and not a first play its are always risky their onlytwo people aboard the pilots who both carry parish and on the ground all the chief engineers were on hand to help the plane developed problems with any of its systems are structures among the people with most of state that were phil condit president of the boeing company and allen the last meter for and engineer in to with live and breathe and the planned for more than five years.

Caption Text
 In the northwestern United States, a group of engineers was about to monitor the first flight of Boeing’s newest airliner. Therewas a lot at stake. Boeing hoped that this plane, the Triple Seven, would assure the success of the company into the 21st century. It had taken ten thousand people nearly four years to design and build the plane using the latest technology, including two of the largest engines on any airliner in the world. But the plane had never left the ground. John Cashman, the chief test pilot would take the Triple Seven into the air for the first time if all went well. Okay. Ladies and gentlemen, it is now your duty to move back behind the barricades so the airplane can taxi northbound. Please move backward. First flights are always risky. There are only two people on board. The pilots. And both carry parachutes. On the ground, all the chief engineers were on hand to help if the plane developed problems with any of its’ systems or structures. Among the people with the most at stake, were PhilCondi, president of the Boeing Company and Allan Mulally, leader of the engineering team, who had lived and breathed the plane for more than five years.

Figure 2: Sample Speech Recognition Transcript

Automatic Speech Recognition Transcript Features	Manual Full-Text Transcript Features
landing gear, first flight, control system, computing power	Dangerous test, velocity minimum, airline world, nose gear
Electronic control, people on board, 21st- century, first time	air supply, total weight, electronic control, carbon fiber
engine shutdown, john cashman, computer code, long time	line of computer code, flow pattern, grease fire
Passenger entertainment system, test flight, united airline	half line, higher thrust, hydraulic line, duct tape, flight test team

Table 2: Word-ngram features

Table 2 shows the word-ngram features selected by our feature extractor from both, the speech recognition transcript and original full-text transcript that was created manually. The first column of Table 2 shows typical word-ngrams generated by our feature extractor from ASR transcriptions. They are sorted by relevancy. We observed that in general, these features are a subset (except a couple of outliers) of those generated from the corresponding manual full-text transcription. We extracted a total of 21

features from the ASR transcript and 50 features from the associated manual transcript. Just for illustration, the second column of Table 2 shows some of those additional features obtained from the manual transcripts. We observe that typically for our category of videos the top 20-30 features are sufficient to detect salient segments, hence we can use ASR transcripts. It is not uncommon to see that features in the second column share words with features in the first column (i.e. “line of computer code” and “flight

test team”). In those cases, the phonetic retrieval for “computer code” and “flight test” will detect the occurrences of those additional features.

Figures 3 and 4 show the temporal and content-based distribution of the word n-grams features. The retrieval step generates for every feature time offsets for all the occurrences of such features. We sort those time offsets in increasing order and denote d_k to be the k-th element in the sorted list. Figure 2 plots those offsets, d_k (k-th element in

the y-axis), along time. In this case about 150 time offsets were generated for all 21 extracted features. The four curves in this figure show the time offsets for different number of features considered (5, 10, 15, and 21). Several observations can be made from the graphs. There are four connected (dense) segments. Most of the gaps do not get populated and do not close as we increase the number of features being considered. We observe varying arrival rates of features resulting in at least three distinct segments.

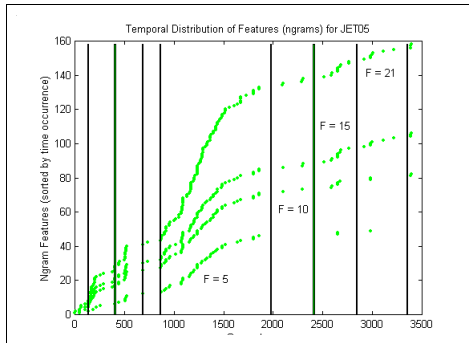


Figure 3: Temporal Distribution of Features

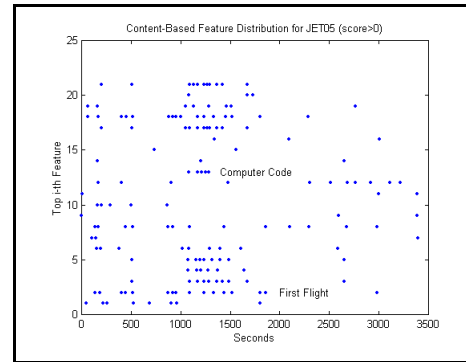


Figure 4: Content-based Distribution of Features

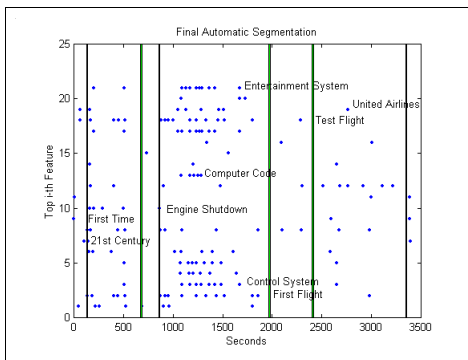


Figure 5: Second Pass Segmentation

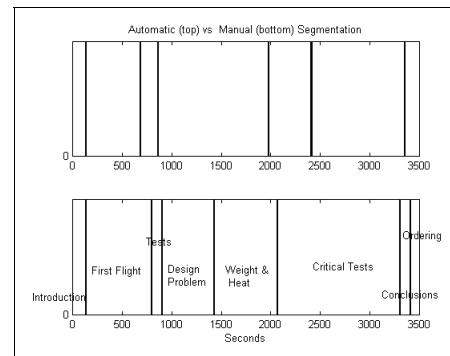


Figure 6: Automatic vs. Manual Segmentation

Dense segments are nearly linear. There are also sparse segments. Figure 4 shows the same offsets along time, but in this case the y-axis represents the i-th feature and they are sorted by relevancy. For example, the 2-nd and 13-th horizontal line contains the projections of all time offsets for features “First Flight” and “Computer Code” respectively. This graph also shows that the segment from 800-2000 seconds is densely populated by the top 10 features and that the support set for the interval 3400 to 3500 seconds contains features 6, 7 and 9.

Figure 3 shows the initial segmentation imposed over the temporal distribution of features. Figure 5 shows the final segmentation results imposed over the content-based feature distribution. Figure 6 shows the results of manual segmentation for the video “The making of the 21st Century Jet”. It can be seen that the first segment in Figure 4 corresponds to the first segment in Figure 6, manually labeled as “Introduction”. It is also possible to automatically generate candidate labels, however we do not describe it in this paper. The label automatically generated for this segment were “Corporation for Public Broadcasting” and “21st Century”, which constitute the

introduction of the program. Notice that since the word “introduction” does not appear in the ASR transcript it is not possible to generate such label. The second and third segment in Figure 3 are strong segments since they contain most top 10 features with high scores from the phonetic retrieval engine. The content-based step merges these segments because the intersection of their support set share the same top 10 features. This segment does not exactly correspond to the “First Flight” manual segment in Figure 6. However a difference of this order can be found in two manually generated segmentations. The automatically generated labels are (in order) “First Flight” and “First Time”. The density of the fourth segment in Figure 5 reveals that is a salient segment and it corresponds to two segments in Figure 6 (“Design Problem” and “Weight and Heat”). Given the slope and temporal distribution of features within this segments our algorithm can only detect one segment in this interval. Finally (after 2000 secs) we detect several sparse segments in contrast to the manually defined single segment “Critical Test”. It is challenging for our algorithm to match the manual segmentation since several miscellaneous topics are being discuss and a human is able to group them as a “set of tests”.

Finally, figure 6, shows the automatic and manual segmentation. The results of both segmentations are aligned to facilitate the comparison. It can be observed that the automatic boundaries corresponding to the most salient segments are well defined, that is, they do not differ significantly from the manually generated boundaries. Notice that even segmentations generated by two different human beings will not fully match.

6 CONCLUSIONS AND FUTURE WORK

We have made a first attempt at automatic segmentation of speech transcriptions from real-world corporate training video content. We have developed a two-pass algorithm based on a combination of content-based methods and boundary-based methods. The first pass creates initial segments based on the temporal distribution of features and their rate of arrival. The second pass validates the initial segments by using the features as queries in an information retrieval system. Initial results on a test video collection look encouraging - and indicate that while a full table of contents may not be practical, detection and labeling of salient segments within a video is achievable. We believe this makes an important contribution to the field of automatic segmentation of video since these techniques can facilitate higher, semantic levels of information retrieval. In terms of future work, we have several enhancements of the algorithm that must be evaluated as well as an algorithm

that generates candidate labels for the most salient segments (at least). This labeling will be useful for the automatic generation of table of contents, summarization and browsing.

REFERENCES

- [1] Allan, J., et al., Topic Detection and Tracking Pilot Study Final Report, *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*, February 1998.
- [2] Bach, J.R., et al., Virage image search engine: An open framework for image management, *Proc. of SPIE Storage and Retrieval for Still Images and Video Databases IV, Vol. 2670*, IS&T/SPIE, February 1996. <http://www.virage.com>
- [3] Bharat, K., and Henzinger, M., Improved Algorithms for Topic Distillation in Hyperlinked Environments, *Proc. of ACM SIGIR 1998*.
- [4] Eichmann, D., et al., A cluster-based approach to tracking, detection and segmentation of Broadcast News, TDT Evaluation, NIST's 1999.
- [5] Fiscus, J.G., et al., TDT Evaluation, NIST's 1998.
- [6] Hauptmann, A.G., Speech Recognition in the Informedia Digital Video Library: Uses and Limitations, *Proc. of ICTAI-95 7th IEEE Int. Conf. on Tools with AI*, Washington, DC., 1995.
- [7] Hauptmann, A.G., and Lee, D., Topic Labeling of Broadcast News Stories in the Informedia Digital Video Library Digital Libraries '98, *Proc. of ACM Conf. on Digital Libraries*, Pittsburgh, PA, June, 1998.
- [8] Hearst, M.A, Multi-paragraph Segmentation of Expository Text, *Proc. of the ACL*, 1994.
- [9] Johnson, et al., Spoken Document Retrieval for TREC-7 at Cambridge University, *Proc. of the 7th Text Retrieval Conference (TREC-7)*, 1998.
- [10] Justeson, J.S. and Slava K., Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text, in *Natural Language Engineering*, 1, pp 9-27, 1995.
- [11] Loach, P.D. and Wathen, A.J., On best least-squares approximation of continuous functions using linear splines with free knots, *IMA J. Numerical Analysis*, 11, pp. 393-409, 1991.
- [12] Schwartz, R., et. al., A Maximum Likelihood Model for Topic Classification in Broadcast News, Eurospeech, *Fifth European Conf. on Speech Communication and Technology*, September 1997.
- [13] Srinivasan, S. and Petkovic, D., Phonetic Confusion Matrix Based Spoken Document Retrieval, *Proc. of SIGIR-2000*, July 2000.