CONTENTS

1

METADATA FOR MIXED-MEDIA ACCESS

Francine Chen, Marti Hearst, Don Kimber,
Julian Kupiec, Jan Pedersen, Lynn Wilcox*

Xerox Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304

**FX Palo Alto Laboratory*
3400 Hillview Ave. Bldg 4
Palo Alto, CA 94304

ABSTRACT

In this chapter, we discuss *mixed-media access*, an information access paradigm for multimedia data in which the media type of a query may differ from that of the data. This allows a single query to be used to retrieve information from data consisting of multiple types of media. In addition, multiple queries formulated in different media types can be used to more accurately specify the data to be retrieved. The types of media considered in this paper are speech, images of text, and full-length text. Some examples of metadata for mixed-media access are locations of keywords in speech and images, identification of speakers, locations of emphasized regions in speech, and locations of topic boundaries in text. Algorithms for automatically generating this metadata are described, including word spotting, speaker segmentation, emphatic speech detection, and subtopic boundary location. We illustrate the use of mixed-media access with an example of information access from multimedia data surrounding a formal presentation.

1 INTRODUCTION

Modern document databases contain information in a variety of media; audio, video, and image data are becoming increasingly common companions to plain text. Access to this rich and variegated information is typically accomplished in standard systems through queries over manually supplied topic keywords or descriptive text. Recent work suggests that fully automatic methods operating directly on the data can offer comparable characterizations. Moreover, analyses tuned to particular media can expose structure that can refine and augment queries based on simple keyword information.

We are exploring these issues through a paradigm we call *mixed-media access*, which encourages the user to query in a variety of media, regardless of the media type of the data. This encompasses, for example, spoken access to textual databases, as well as queries that combine cues across the media types present in a complex document.

Special metadata considerations arise within such a paradigm. For our purposes, mixed-media metadata is defined as derived properties of the media which are useful for information access or retrieval. These properties can be derived either in advance or "on the fly". Our focus is on automatically derived metadata for speech, scanned text images, and full-length text.

In a purely textual database, metadata for information access typically consists of indices on word tokens. The state-of-the-art in speech and image recognition is such that we cannot reliably create a word-level transcription for arbitrary speech documents [21] or text images [5]. Therefore, in a multimedia database prepared for mixed-media access it is unrealistic to suppose that a full transcription is available in advance as metadata. We can, however, robustly recognize particular keywords, a process known as *word spotting* [30]. Word spotting produces metadata in the form of time indices of keywords in audio [31], or locations of keywords in a text image [3].

In addition, we can enrich this word-level metadata with information that captures some of the context implicit in particular media. For example, in speech data one important aspect is the identity of the speakers. We can automatically detect speaker changes in audio, in a process which we refer to as *speaker segmentation* [32]. This produces metadata in the form of time indices of the audio segments corresponding to the different speakers. When the speakers are known each segment can be annotated with the identity of the speaker, a process known as *speaker identification.* This information helps characterize the data, and can be stored as metadata and used for indexing.

Another source of information in speech not present in plain text is prosodic information. Prosodic information includes changes in pitch, amplitude, and timing, and is used by a speaker to signal regions of speech that are important. We can automatically detect regions of emphatic speech [4] and note the time indices of the audio segments in which emphatic speech occurs. These regions are another form of metadata and can be used as a method of indexing into a conversation.

A full-text document can often be characterized as consisting of one or more main topics or themes, as well as a sequence of subtopical discussions that take place in the context of the main topic themes. Thus, another source of information for metadata which can be applied to both spoken and written text is that of subtopic extent. We can determine when the discussion within a text changes from one subtopic to the next [10], and then generate indices that indicate which paragraphs or which regions correspond to each subtopic segment. A separate, but related, text analysis task is the assignment of category information to the full text and its subtopical segments [11]. The contrast between discovery of subtopic boundaries and discovery of subtopic categories bears an analogical resemblance to the contrast between discovery of speaker change boundaries and discovery of speaker identity.

In order to illustrate how mixed-media access can be used to retrieve information from multimedia data, we provide an example based on a presentation on Hypercars[1]. The data consists of an audio recording of the talk, scanned images of the slides used in the talk, and text from a paper on Hypercars. Queries can be posed as spoken keywords, text keywords, or natural language text. Spoken keywords can be used to retrieve information from text and audio data, while text keywords and natural language queries can be used to retrieve data from slide images and text. In addition, structure in the audio imposed by different speakers, and structure in the text imposed by topic boundaries, can be used to refine these queries.

In the remainder of this chapter, we first provide more detail about the three media types and their corresponding metadata. We then describe how this metadata can be derived automatically, and finally present an example of the use of such metadata in mixed-media access.

2 CHARACTERISTICS OF THE MEDIA AND THE DERIVED METADATA

Digitized speech and scanned images of text are not easily searched for content. However, they contain information which can be organized to provide easier access. In chapter X, Wechsler and Schauble show how to use phone sequences[2] for indexing audio recordings. In this chapter, we restrict our attention to metadata at the word level. Metadata providing indices to speech includes keyword locations, segmentation of a conversation by speaker, and regions which a speaker highlighted by speaking more emphatically. In text images, keywords and layout may be identified.

Queries may consist of a Boolean expression, which requires searching for a small number of keywords or phrases, or for a particular speaker. In information retrieval, a tradeoff has been observed between searching using only fixed categories and allowing the user access to unlimited vocabulary. The evidence suggests that when category information is combined with free text search, results are improved over allowing

[1] According to the speaker's paper [19], Hypercars are very light-weight passenger cars, constructed of carbon-fiber material, driven by hybrid-electric drives meant to achieve very high gas mileage. The authors write "Far from sacrificing other attributes for efficiency, ultralight hybrids could be more safe, peppy, clean, durable, reliable, quiet, comfortable, and beautiful than existing cars, yet be priced about the same or less."

[2] Phones are the basic sound units of speech, somewhat similar to the alphabet for text.

either alone [20, 17]. If the set of keywords is fixed, metadata based on keyword locations can be pre-computed and stored for later indexing. This is efficient, in that keyword spotting can be done off-line, but restrictive, in that the available query terms are limited to a pre-defined set of keywords. It is not currently possible to reliably generate precomputed word-level metadata that supports unrestricted vocabulary searching over image or audio data. However, it is possible to support this search style by spotting for keywords "on the fly".

2.1 Speech

Audio is by nature a time-sequential media. Portions of audio data can be accessed by specifying starting and ending times for a desired segment. However, indexing solely by time interval is restrictive; the development of sophisticated speech analysis techniques allows for attribute-based interval specification, such as locating the portion of an audio stream in which a comment was made by a particular speaker.

Speech can be analyzed in different ways. One is in terms of the presence of specific keywords in the speech [30]. Another is in terms of the identity of the speakers in the audio [32]. A third is in terms of prosodic information [4], which can be used by a speaker to draw attention to a phrase or sentence, or to alter the word meaning. This information can be exploited to obtain metadata for access to audio.

Metadata describing keywords consists of the keyword and its starting and ending time in the audio. Because the process of spoken keyword identification is not perfectly accurate, a confidence score is also a part of the metadata representation.

Metadata describing the identity of the speakers consists of the name of the speaker, and a list of the time intervals during which the speaker talks. Speaker-independent indices are also maintained by taking note of when one speaker finishes talking and either silence or another speaker follows. In this case, starting and ending times for each of the different speakers in the audio are recorded, but only symbolic names are attached to the speakers, for example speaker A, speaker B, etc. This allows for retrieval of logical speaker units even when the identity of the speakers are not known. Additionally, silence and non-speech sounds can be identified and stored as metadata.

Metadata describing emphasized regions of speech consists of the time indices of the intervals where the speech was emphatic, as well as a measure of certainty that the particular region did indeed contain speech which the speaker intended to emphasize.

2.2 Text Images

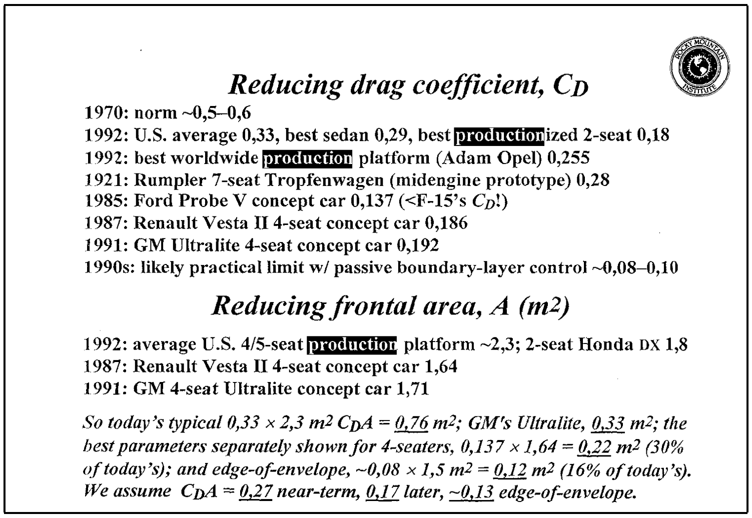
Text image data can be characterized in terms of the layout structure of the document, *e.g.*, columns and paragraphs [12], the semantic information contained in the document [8], and by the words in the document. However, a reliable word-level transcription of arbitrary pages containing text is not yet possible. Therefore, rather than use a word-level transcription, we characterize image data by the location and identity of keywords, which can be stored as metadata. As in the audio example, the representation may also include a score indicating the degree of confidence in the identification of the keyword.

2.3 Full-length Text

Full-length texts are natural language expressions of sufficient length to exhibit topical substructure. For example, a magazine article will be composed of numerous sections each illuminating aspects of the overall topic. Often these sections will be demarked by author provided typographical annotations, perhaps in a markup language such as SGML [26]. However, author provided subtopic markup is neither always available nor always reliable.

Full-length text shares the same basic representation as shorter text forms, such as titles and abstracts: words. Therefore standard mechanisms for text indexing, such as inverted indices [24], can act as metadata. For retrieval based on spoken multi-word queries, index mechanisms which support search with proximity constraints are also useful. In addition, current work in computational linguistics allows for the assignment of additional information at the word token level, *e.g.*, part-of-speech tags [6] and morphological derivation [7].

Full-length texts can also be segmented at topic and subtopic boundaries. Algorithms that detect subtopic structure can either partition the text or allow overlap among multi-paragraph units. In both cases, the metadata consists of indices indicating which paragraphs or which regions of tokens correspond to each subtopic segment. Additionally, information that characterizes the content of the subtopics and the main topics can serve as useful metadata [11]. Automated determination of main topic content is an active area of research [18].



Reducing drag coefficient, C_D

1970: norm ~0,5–0,6
1992: U.S. average 0,33, best sedan 0,29, best productionized 2-seat 0,18
1992: best worldwide production platform (Adam Opel) 0,255
1921: Rumpler 7-seat Tropfenwagen (midengine prototype) 0,28
1985: Ford Probe V concept car 0,137 (<F-15's C_D!)
1987: Renault Vesta II 4-seat concept car 0,186
1991: GM Ultralite 4-seat concept car 0,192
1990s: likely practical limit w/ passive boundary-layer control ~0,08–0,10

Reducing frontal area, A (m^2)

1992: average U.S. 4/5-seat production platform ~2,3; 2-seat Honda DX 1,8
1987: Renault Vesta II 4-seat concept car 1,64
1991: GM 4-seat Ultralite concept car 1,71

So today's typical 0,33 × 2,3 m^2 C_DA = $\underline{0,76}$ m^2; GM's Ultralite, $\underline{0,33}$ m^2; the best parameters separately shown for 4-seaters, 0,137 × 1,64 = $\underline{0,22}$ m^2 (30% of today's); and edge-of-envelope, ~0,08 × 1,5 m^2 = $\underline{0,12}$ m^2 (16% of today's). We assume C_DA = $\underline{0,27}$ near-term, $\underline{0,17}$ later, $\underline{\sim 0,13}$ edge-of-envelope.

Figure 1 Result of spotting for "production*"

3 EXTRACTION AND USE OF METADATA

This section describes implemented techniques for the automated extraction of the kinds metadata described in Section 2.

3.1 Word-Image Spotting

Word-image spotting refers to the task of detecting and locating user-specified keywords and phrases in images of text. Several systems for spotting whole words in scanned images of text have been developed. In noisy document images, where optical character recognition systems perform poorly [14] [15], these systems have been found to locate keywords more accurately than the option of performing optical character recognition (OCR) and then searching for keywords as text strings. However, the image-based word-spotting systems require as input an image that has been correctly segmented into words.

A word-image spotting system developed at Xerox PARC detects words in a variety of fonts within lines of imaged text. The user-specified keywords and phrases can be partially specified, similar to a simple "grep", but over images of text [3]. For each word identified as a keyword by the word-image spotter, the location of the word in the image can be stored as metadata. Figure 1 shows the result of spotting for “production*”, where “*” represents a wildcard, in one of the digitized overhead transparencies used in a formal presentation on Hypercars. Note that the alternate word forms “productionized” and “production” were detected. The spotted characters composing “production” are highlighted. Although the search is for a partially specified keyphrase, the location of the entire word could be highlighted by configuring the word-image spotter to identify interword space.

The PARC word-image spotter is based on the use of multi-resolution image morphology [1] to identify bounding boxes of text lines, and hidden Markov modeling to identify specific words within a text line. Each text line bounding box is normalized to a standard height, and the width of the bounding box is scaled proportionally, producing a gray-scale image. This scaling permits recognition of words in a variety of fonts in a range of sizes. A feature vector characterizing a scaled bounding box is derived from the columns of pixel values within that bounding box.

A prespecified set of keywords, as is commonly used in word spotting systems, is not required. Instead, for each keyword or keyphrase specified by the user in a query, a hidden Markov model (HMM) is created “on the fly” from pre-trained character models. Another pre-trained HMM is used to model the data which are not part of a keyword or phrase. The non-keyphrase model coarsely represents the columns of pixels in a bounding box. A non-keyphrase model composed of all characters and symbols connected in parallel could be used instead, but would be much more computationally expensive.

The models are trained on data labeled with the characters appearing in each line of text and with the location of each line of text, but not the location of each character. Baum-Welch training [23] is used to estimate the parameter values of the models. To detect keywords or phrases, the keyphrase models and non-keyphrase model are connected in parallel to create a spotting network. Keyphrases within a bounding box are identified using Viterbi decoding [23] on the spotting network. The detected keywords and phrases and their locations in a text image can then be used as metadata. A fixed set of keywords can be spotted and stored as metadata in a preprocessing step. As users specify additional keyword and phrase searches during access, indices to these search terms can be added to the metadata. In this approach the lexicon is not limited, in contrast to the approach of performing OCR followed by indexing, which is susceptible to errors caused by words being “out of vocabulary.”

3.2 Audio Word Spotting

Audio word spotting is the ability to locate keywords or phrases in previously recorded speech. It differs from isolated word recognition, in which words to be recognized must be spoken in isolation, and continuous speech recognition, in which each word in a continuous stream must be recognized. Word spotting generates metadata in the form of time indices for the beginning and ending of keywords. This provides indexing by keywords into long audio files, thus allowing retrieval of specific information without the need to listen to the entire recording.

Certain word spotting systems assume there are a fixed set of keywords to be spotted in continuous speech from many different talkers. An example is the operator assisted telephone call task in [34], where spotting for only five keywords is required. Such systems are based on whole word models, and require training data for each of the keywords from a large database of speakers. They are thus appropriate in tasks for which a small number of fixed keywords suffice. Other speaker-independent keyword spotting systems are based on large vocabulary continuous speech recognition. For example, the system proposed by SRI [29] uses the $Decipher^{TM}$ large-vocabulary speech recognition system to transcribe the speech, and any keywords that occur in the transcription are hypothesized. A drawback of this approach is that certain keywords, for example proper names, are unlikely to be included in the vocabulary of the recognizer.

As an alternative to the above speaker-independent word spotting systems is the interactive system developed at Xerox PARC [31]. The system is speaker-dependent, so that the audio is restricted to speech from a single talker. When word spotting is to be performed, the talker simply speaks the keyword or phrase to be located. Alternatively, a keyword can be manually excised from a recording. There are no linguistic assumptions, so that the word spotting system is multi-lingual. In addition, spotting can be performed for non-speech sounds such as music or laughter.

The PARC word spotting system uses a hidden Markov model (HMM) to model arbitrary, user-defined keywords in the context of continuous speech [30]. Training the HMM consists of two stages: an initial, static stage in which statistics for a given talker are learned and a model for the non-keyword speech is obtained, and a second, dynamic stage in which the keyword model is trained as the system is in use. Data for the static training stage consists of an arbitrary segment of the talker's speech. The dynamic training stage is novel in that it requires only a single repetition of a keyword; thus, there is no distinction between keyword training and word spotting.

The search technique for locating instances of a keyword in continuous speech is a "forward-backward" search which uses peaks in the *a posteriori*, or forward [23], probability of the keyword end state to detect potential keyword endpoints. State probabilities are then recursively computed backwards to find a peak in the keyword start state. In this way, a score for the keyword is obtained in addition to the starting and ending times, which helps to prevent false alarms. This search is efficient, in that backtracking is only required when a keyword is hypothesized.

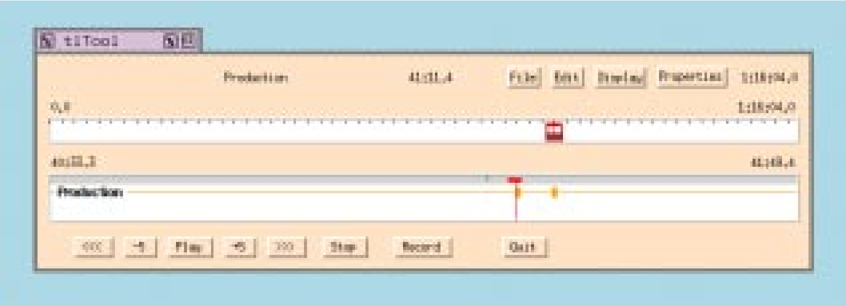


Figure 2 Audio browser with keyword 'production' highlighted

Figure 2 shows a display of the audio portion of the formal presentation on Hypercars. The user specified an audio query for the keyword "production" by locating an instance of the word in an initial portion of the audio. The keyword search was performed to locate all other instances of this keyword. The intervals corresponding to the times when the keyword "production" was spotted are highlighted. By listening to the audio in the vicinity of these intervals, information on the production of hypercars can be obtained.

3.3 Text Access via Spoken Queries

In Section 3.2 we described how spoken utterances can be used as input to a search of audio media. Spoken utterances can also be used as input to text search. This allows a user to utter spoken words and be presented with portions of text containing those words. This mode of input could be especially natural in settings where the user is also providing speech input for audio access. The relevant metadata for text retrieval given speech input is an inverted index structure capable of supporting proximity constraint searches.

Text access via spoken query necessarily involves large vocabulary speech recognition. Typically this requires a language model which constrains the recognition task by disallowing, or assigning very low probability to most word sequences [13]. This language model is often constructed from statistical analysis of a large amount of training text taken from the application domain. An alternative approach, however, is to use word proximity information as metadata which provides an implicit language model.

In the simplest form, the access is achieved by performing standard large vocabulary word recognition and using the recognized words as input to a text search engine. Recognition errors, however, may cause the search to fail to locate correct regions of text. To address this problem, the speech recognizer can be configured to output multiple hypothesis for each spoken word. For example if the single word "president" was spoken, the recognizer output might be the list ("precedent", "prescient", "president", ...) rank ordered by the estimated likelihood that each of the words had been spoken. The text can then be searched for regions containing a word from this list. This increases the chance of locating the desired text, but of course at the expense of locating undesired regions. However, when the spoken query contains multiple words from a phrase occurring within the text data, very often the only regions of text containing one word from each hypothesized wordlist are those in which the actual spoken words occur. For example if the spoken phrase contains the words "president" and "kennedy", resulting say in hypothesized wordlists ("precedent", "prescient", "president", ...) and ("kennerty", "kennedy", "kemeny, ...) then with very high probability the only regions of text containing words from each list, within close proximity, will in fact contain "president Kennedy." The reason is simply that other combinations of the words do not commonly co-occur.

In general we have observed that the intended words of a spoken query tend to co-occur in text documents in close proximity whereas word combinations that are the result of recognition errors are usually not semantically correlated and thus do not appear together. We refer to this as *semantic co-occurrence filtering.* Note that in exploiting this principle, the text proximity metadata can be interpreted as providing an implicit language model used with the recognizer.

At Xerox PARC, we have implemented a text retrieval system using semantic co-occurrence filtering. [16]. Our system is based on speaker dependent, isolated word phonetic recognizer, although this is not an inherent requirement.

3.4 Speaker Segmentation

In speaker segmentation, the audio is partitioned into intervals, with each interval containing speech from a single speaker. The metadata derived from this consists of starting and ending times for each speaker, as well as the identity of the speaker. Pauses, or silence intervals, as well as non-speech sounds such as music or applause, can also be identified for use in indexing. A speaker index provides the capability to access portions of the audio corresponding to a particular speaker of interest, or to browse the audio by skipping to subsequent speakers.

The basic framework for segmentation of the audio is an HMM network consisting of a sub-network for each speaker and interconnections between speaker sub-networks [32]. Speaker segmentation is performed using the Viterbi algorithm [23] to find the most likely sequence of states and noting those times when the optimal state sequence changes between speaker sub-networks. The speaker sub-networks used here are multi-state HMMs with Gaussian output distributions. In addition to modeling speakers, sub-networks are also used to model silence and non-speech sounds such as a musical theme.

In applications where the speakers are known *a priori*, and where it is possible to obtain sample data from their speech, segmentation of the audio into regions corresponding to the known speakers can be performed in real time, as the speech is being recorded. This is done by pre-training the speaker sub-networks using the sample data, and then using the Viterbi algorithm with continuous traceback for segmentation. Real-time speaker segmentation is useful, for example, in video annotation systems where annotations are made during the recording process [27].

When no prior knowledge of the speakers is available, unsupervised speaker segmentation is possible using a non-real-time, iterative algorithm. Speaker sub-networks are first initialized, and segmentation is achieved by iteratively using the Viterbi algorithm to compute a segmentation, and then retraining the speaker sub-networks based on the computed segmentation. It is necessary for the iterative segmentation algorithm to have good initial estimates for the speaker sub-networks. One way of obtaining the initial estimates is by hand-labeling a portion of the audio for each of the speakers. Experiments indicate that 30 to 60 seconds of data per speaker is sufficient [32]. However, hand-labeling by speaker can be difficult, particularly when the number and identity of the speakers are unknown.

Another method for initializing speaker clusters is agglomerative clustering. The data is first divided uniformly into 3 second intervals. The distance between each pair of intervals is computed, and the closest pair of intervals is merged. This process is repeated until a desired number of clusters is obtained, or until the merge distance exceeds a fixed threshold. This provides a course segmentation of the data, accurate only to the length of the original 3 second intervals. However, the iterative resegmentation algorithm can be performed using this as the initial clustering to obtain more accurate segmentation.

Figure 3 shows the audio portion of the presentation on Hypercars segmented according to speaker. The segmentation was obtained by first using the unsupervised clustering to obtain an initial set of five clusters. These clusters were hand-labeled as “Announcer”, “Speaker”, “Audience”, “Applause”, and “Silence”. We refined this initial segmentation by using it as training data for the iterative resegmentation algorithm. Each of the resulting clusters is displayed on a different track. In addition to the usual play, fast forward and reverse options, the audio browser provides skip buttons to skip forward to the next speaker, or backwards to the previous speaker. Speaker buttons provide the capability to play audio corresponding to the individual speakers.

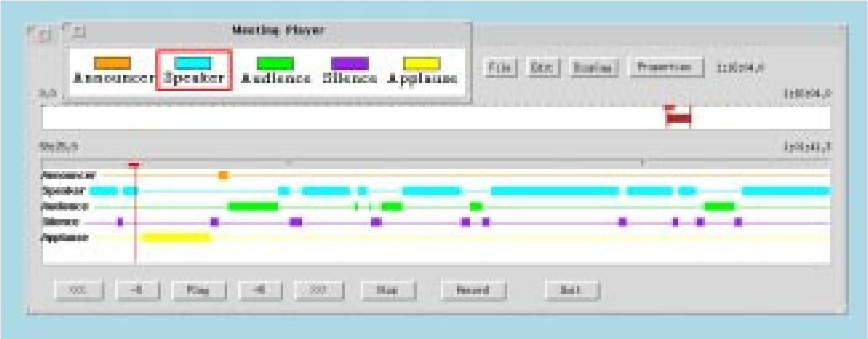


Figure 3 Audio browser with speaker segmentation

3.5 Emphatic Speech Detection

By modifying the pitch, volume, and timing, that is, the prosodics of speech, a talker can convey syntactic and semantic information, in addition to the spoken words. Prosodics can be used to alter the meaning of words, to signal whether a sentence is a statement or question, or to indicate a phrase boundary. Butzberger *et al.* [2] used prosodic information to classify isolated words as a statement, question, command, calling, or continuation. Wightman *et al.* [33] combined the use of prosodic information and word recognition information to identify intonational features in speech. Based on prosodic information, metadata can be created identifying when a question was asked, and identifying phrase boundaries for use as endpoints for presentation.

Prosody is also used in natural, conversational speech to give more emphasis to some words and phrases. When making a point, the spoken words are given greater and more frequent emphasis. This prosodic information can be exploited to serve as indices to regions of possible interest.

Emphatic speech has been found to be characterized by prominences in pitch and volume. To estimate pitch, the fundamental frequency (F0) of the glottal source is computed. To locally estimate speaking volume, energy in a short duration of the speech signal is computed. In our work at PARC, emphatic speech is identified by matching the set of prosodic features computed from a speech signal against an HMM network designed to model different prosodic patterns [4]. Prosodic features were selected which contain information to capture emphatic prominences; these features include F0, energy, change in F0, change in energy, and voicing to indicate vocalic regions.

To identify emphasized speech, syllable-based HMMs are created to model different patterns of emphatic speech. Separate models are created for unemphasized speech, which has a relatively flat prosodic pattern, for background noise, and for pauses.

A network modeling variations in emphasis is created by connecting the models of emphasized speech, unemphasized speech, and background noise in parallel. An optional pause is allowed between each of the models. Viterbi decoding [23] is used to find the best path through the network. When the best path passes through an emphatic speech model, the time indices are recorded as an emphatic region.

Regions with a high density of emphatic speech are more likely to contain parts of a conversation which a speaker wished to highlight. The time indices of these regions are stored as metadata indicating regions of possible interest for browsing.

3.6 Subtopic Boundary Location

Both automatically-identified and author-identified structural information is important for locating information in full-text documents. The structure of expository texts can be characterized as a sequence of subtopical discussions that occur in the context of one or a few main topic discussions. Subtopic structure is sometimes marked by the author in technical texts in the form of headings and subheadings. When author-identified structure is available, indices corresponding to SGML markup can be easily generated, therefore this discussion focuses only on automatically-generated structural information.

For the cases in which texts consist of long sequences of paragraphs with very little structural demarcation, we have developed an algorithm, called TextTiling, that partitions these texts into multi-paragraph segments that reflect their subtopic structure [10]. This algorithm detects subtopic boundaries by analyzing the term repetition patterns within the text. The main idea is that terms that describe a subtopic will co-occur locally, and a switch to a new subtopic will be signaled by the ending of co-occurrence of one set of terms and the beginning of the co-occurrence of a different set of terms. In texts in which this assumption is valid, the central problem is determining where one set of terms ends and the next begins. Figures 4 and 5 show the results of TextTiling the Hypercars article. The larger peaks in the graph represent a relatively large amount of lexical cohesion [9] among the words within the sentences within the peak. The valleys represent breaks in lexical cohesion between adjacent text blocks. The algorithm's success is determined by the extent to which these simple cues actually reflect the subtopic structure of the text.

The core algorithm has three main parts: tokenization, similarity determination, and boundary identification. Tokenization refers to the division of the input text into individual lexical units. The text is also grouped into 20-word adjacent token-sequences,

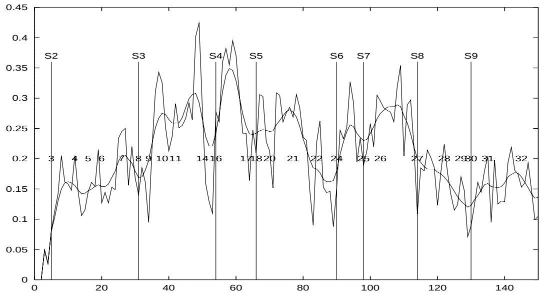


Figure 4 Result of TextTiling on the Hypercars article, first 150 token-sequences. The x-axis represents sentence numbers, the y-axis represents a measure of similarity between adjacent text blocks, and internal numbers indicate the locations of paragraph boundaries within the text. The vertical lines indicate topic boundaries marked by the algorithm.

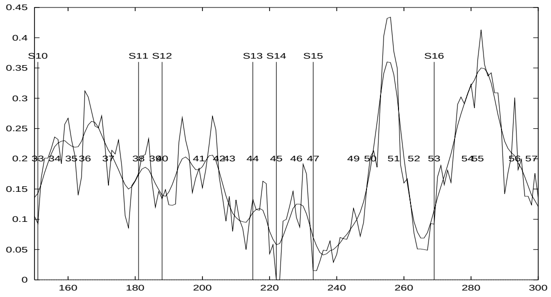


Figure 5 Result of TextTiling on the Hypercars article, second 150 token-sequences.

ignoring sentence boundaries in order to avoid length normalization concerns. A record of the locations of paragraph boundaries is maintained.

After tokenization, adjacent pairs of blocks of token-sequences are compared for overall lexical similarity. Token-sequences are grouped together into blocks to be compared against an adjacent block of token-sequences. Actual paragraphs are not used because their lengths can be highly irregular, leading to unbalanced comparisons.

Similarity values for adjacent blocks are computed for every token-sequence gap number. Boundaries are determined by changes in the sequence of similarity scores. The token-sequence gap numbers are ordered according to depth scores, that is, the sum of the heights of the plot on either side of the token-sequence gap. Segment boundaries are assigned to the token-sequence gaps with the largest corresponding depth scores, adjusted as necessary to correspond to true paragraph breaks. The cutoff for boundary assignment is a function of the average and standard deviations of the depth scores for the text under analysis. A boundary is drawn only if the depth score exceeds a threshold, and this threshold is a function of the properties of the graph, and so scales with the size of the document.

The TextTiling procedure when run on the Hypercars article *Reinventing the Wheels* produces 17 TextTiles, or subtopic segments, the contents of which can be glossed as:[3]

S1	Title, Table-of-Contents, and Abstract
S2	The Movement Towards Alternative Automobiles
S3	Ultralight Material is the Solution to Steel
S4	How Hypercars Differ from Electric Cars
S5	The New Idea: Combining Hybrid Fuel with Lightweight Material
S6	Lightweight Material Leads to Cheaper Accessories
S7	Lightweight Material Leads to Large Savings
S8	GM's 1991 Prototype
S9	The Role of Alternative Fuels
S10	Why Composites are Better than Steel
S11	Continuation of Theme S10
S12	Changes in Competitive Strategies
S13	Most Automakers Cannot Accept these Changes
S14	Continuation of Theme S13
S15	The Role of Oil in American Politics
S16	Gas Taxes and Related Issues
S17	Innovative Public Policy Ideas

[3] The settings used were: token-sequence size set to 20, 2 rounds of smoothing with a smoothing width equal to 3, blocksize set to 6. The gloss is provided for illustrative purposes and is not done automatically.

The paper has 60 paragraphs and 6 sections explicitly marked with section headings. Although the algorithm is not instructed to take note of the section headings, it nevertheless placed boundaries at four of them (tiles S3, S5, S10, and S16). The algorithm missed one section heading by one short paragraph (after tile 11), and missed the last heading by one long paragraph (after tile S15). In two cases (tiles S11 and S14), the breaks suggested by the algorithm are probably extraneous, since in both cases the subject matter continues from the previous section.

3.7 Summary of Metadata

In this chapter we have discussed how metadata can be derived from various types of media. Table 1 summarizes the types of metadata discussed in this chapter and the media which they serve to index. The column labeled "Media" gives the media type of the document, the column labeled "Metadata" describes the metadata used to index the media, and the column labeled "Extraction" shows the method for obtaining the metadata.

Media	Metadata	Extraction
Image	Keyword	Word-image spotting [3]
Audio	Keyword Speaker ID Emphasis	Word spotting [30] Speaker Segmentation [32] Emphasis Detection [4]
Text	Subtopic Boundary Inverted Index	TextTiling [10] Co-occurrence Filtering [16]

Table 1 Types of media indexed, metadata used for indexing, and how each is extracted.

4 USE OF METADATA IN MIXED-MEDIA ACCESS

Word-oriented information is present in full-length text, text image, and speech data, hence the user may expect to approach such data with a degree of uniformity. In particular, this may include accessing the media via queries in the same media type, for example, keyword spotting in speech using a spoken keyword. It also includes situations in which the media type of the query is different from that of the data, for example a text query to an image database. The metadata used must be flexible enough to accommodate each useful combination. This section discusses an example of mixed-media access.

4.1 Hypercar Forum

The multimedia data considered in this example is based on a formal presentation on hypercars, and includes recorded audio, scanned slides, and online text for an article on hypercars [19]. In order to make full use of this material, we need to be able to retrieve information from each of these media. Using mixed-media access, this retrieval can be done simultaneously.

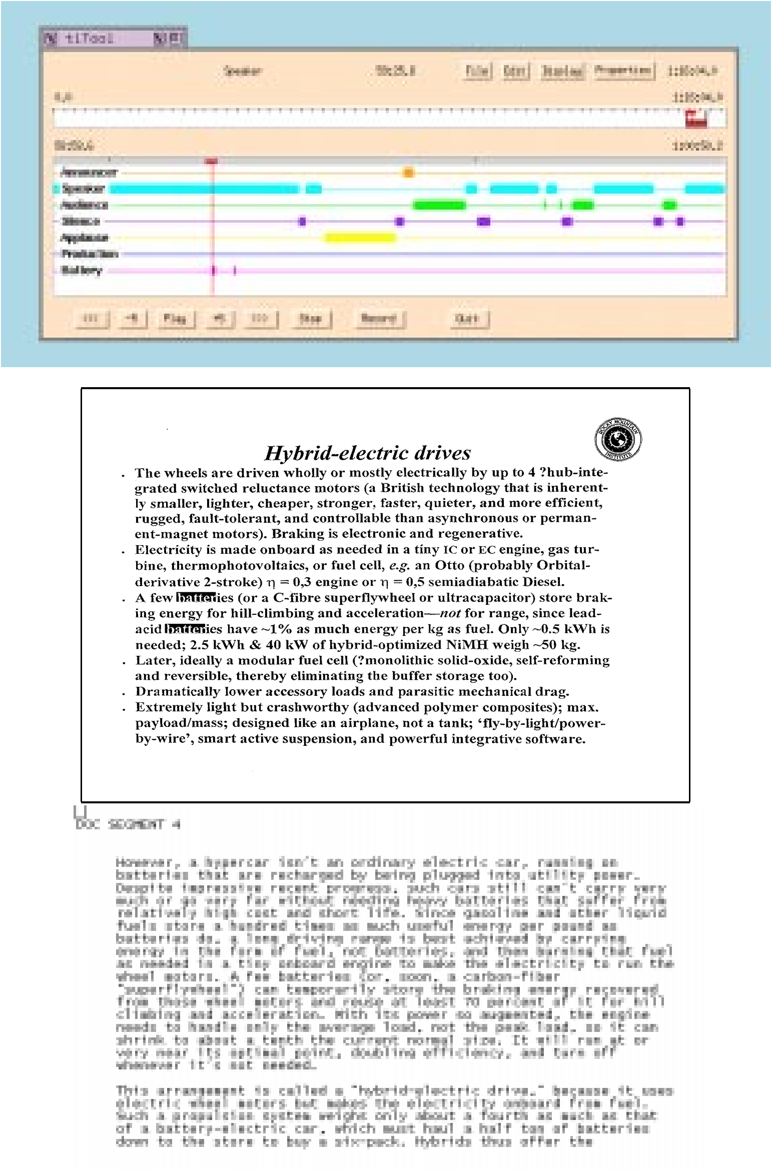
Metadata for the audio in the form of speaker segmentation is pre-computed. As discussed previously, the speakers in the audio are defined as "Announcer", "Speaker", "Audience", "Silence" and "Applause". "Announcer" is the person who introduces the speaker prior to the talk. "Speaker" is the person giving the talk. "Audience" is not actually a single speaker, but all members of the audience who asked questions. Thus queries involving the speaker can specify whether information from the speaker, announcer, or audience is desired. In addition, a search can be made for a non-speech sound, in this case applause. This is useful for locating likely highlights of the talk, as well as finding its beginning and end.

Metadata for the online text, namely the TextTile segmentation of the text into topic regions, is also precomputed. This allows the relevant passages of text to be retrieved in response to a keyword query. All other metadata used in this example is computed "on the fly".

Figure 6 illustrates the response to a mixed media query consisting of the textual query for the keywords "battery" or "batteries", to be searched for in the online text and slide image data, an audio query for the keyword "battery", to be searched for in the audio data, and a specification that the keyword be spoken by "Speaker". Since the mixed-media query was over three types of media, relevant portions of each of these media are displayed.

The audio browser displays a timeline of the audio portion of the meeting, with different speakers corresponding to different tracks on the timelime. There is also a track for the specified keyword "battery"[4]. Since the query specified the keyword "battery" spoken by "Speaker", the cursor is positioned to the first occurrence of "battery" by "Speaker". The user can listen to the portion of the audio in the region where the speaker mentioned the word battery.

[4] In addition, there is a track for the keyword "production" which was specified in a previous query.



Hybrid-electric drives

- **The wheels are driven wholly or mostly electrically by up to 4 ?hub-integrated switched reluctance motors (a British technology that is inherently smaller, lighter, cheaper, stronger, faster, quieter, and more efficient, rugged, fault-tolerant, and controllable than asynchronous or permanent-magnet motors). Braking is electronic and regenerative.**
- **Electricity is made onboard as needed in a tiny IC or EC engine, gas turbine, thermophotovoltaics, or fuel cell, *e.g.* an Otto (probably Orbital-derivative 2-stroke) η = 0,3 engine or η = 0,5 semiadiabatic Diesel.**
- **A few batteries (or a C-fibre superflywheel or ultracapacitor) store braking energy for hill-climbing and acceleration—*not* for range, since lead-acid batteries have ~1% as much energy per kg as fuel. Only ~0.5 kWh is needed; 2.5 kWh & 40 kW of hybrid-optimized NiMH weigh ~50 kg.**
- **Later, ideally a modular fuel cell (?monolithic solid-oxide, self-reforming and reversible, thereby eliminating the buffer storage too).**
- **Dramatically lower accessory loads and parasitic mechanical drag.**
- **Extremely light but crashworthy (advanced polymer composites); max. payload/mass; designed like an airplane, not a tank; 'fly-by-light/power-by-wire', smart active suspension, and powerful integrative software.**

DOC SEGMENT 4

However, a hypercar isn't an ordinary electric car, running on batteries that are recharged by being plugged into utility power. Despite impressive recent progress, such cars still can't carry very much or go very far without needing heavy batteries that suffer from relatively high cost and short life. Since gasoline and other liquid fuels store a hundred times as much useful energy per pound as batteries do, a long driving range is best achieved by carrying energy in the form of fuel, not batteries, and then burning that fuel as needed in a tiny onboard engine to make the electricity to run the wheel motors. A few batteries (or, soon, a carbon-fiber "superflywheel") can temporarily store the braking energy recovered from those wheel motors and reuse at least 70 percent of it for hill climbing and acceleration. With its power so augmented, the engine needs to handle only the average load, not the peak load, so it can shrink to about a tenth the current normal size. It will run at or very near its optimal point, doubling efficiency, and turn off whenever it's not needed.

This arrangement is called a "hybrid-electric drive," because it uses electric wheel motors but makes the electricity onboard from fuel. Such a propulsion system weighs only about a fourth as much as that of a battery-electric car, which must haul a half ton of batteries down to the store to buy a six-pack. Hybrids thus offer the

Figure 6 Result of search for the keywords "battery" and "batteries" in different media. The audio browser shows speaker segmentation and keyword locations. For the slides, the single search term "batter*" was used; the identified keywords in the slide are highlighted. A portion of one TextTile containing the keyword is shown.

The image data query for “battery” or “batteries” was expressed as a search for “batter*”, where “*” is a wildcard. The response to this query is to display the slide containing the specified character sequence “batter*”. In this case, the slide is titled “Hybrid-electric drives”. Finally, the query for “batteries” in the online text produces a window view into the full-length text, displaying one of the TextTiles (subtopical segments of text) containing the specified keyword. In this case, the fourth segment is displayed, in which hypercars are contrasted to electric cars running on batteries.

5 SUMMARY

Multimedia databases are typically accessed through text queries, often referring to manually-assigned keywords. Recently developed methods provide ways to automatically generate metadata for audio, images, and text that enable more natural access modes such as mixed-media access.

Acknowledgements

We would like to thank Jeanette Figueroa for her enthusiasm and administrative support.

REFERENCES

[1] D.S. Bloomberg, “Multiresolution Morphological Approach to Document Image Analysis,” In Proc. of the International Conference on Document Analysis and Recognition, Saint-Malo, France, September 1991.

[2] J.W. Butzberger Jr., M. Ostendorf, P.J. Price, and S. Shattuck-Hufnagel. “Isolated word intonation recognition using hidden Markov models.” In Proc. International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, New Mexico, April 1990.

[3] F.R. Chen, D.S. Bloomberg, and L.D. Wilcox. “Detection and location of multicharacter sequences in lines of imaged text.” Journal of Electronic Imaging, 5(1):37-49, 1996.

[4] F.R. Chen and M.M. Withgott. “The use of emphasis to automatically summarize a spoken discourse.” In Proc. International Conference on Acoustics, Speech and Signal Processing, San Francisco, California, March 1982.

[5] S. Chen, S. Subramaniam, R.M. Haralick, and I.T. Phillips. “Performance Evaluation of Two OCR Systems.” In Proc. Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, April 1994.

[6] D. Cutting, J. Kupiec, J. Pedersen, and P. Sibun, “A Practical Part-of-Speech Tagger,” The 3rd Conference on Applied Natural Language Processing, Trento, Italy, 1991.

[7] *Tools for Morphological Analysis*, M. Dalrymple (ed.), Center for the Study of Language and Information, Stanford, California , 1987.

[8] A. Dengal. “The role of document analysis and understanding in multimedia information systems.” In Proc. International Conference on Document Analysis and Recognition, Tsukuba Science City, Japan, October 1993.

[9] M.A.K. Halliday and R. Hasan, *Cohesion in English*, Longman, London, 1976.

[10] M.A. Hearst. “Multi-paragraph segmentation of expository text.” In Proc. 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, 1994.

[11] M.A. Hearst. “Using Categories to Provide Context for Full-Text Retrieval Results.” In Proc. RIAO 94, Intelligent Multimedia Information Retrieval Systems and Management, Rockefeller, New York, 1994. To appear.

[12] D.J. Ittner and H.S. Baird. “Language-Free Layout Analysis.” In Proc. International Conference on Document Analysis and Recognition, Tsukuba Science City, Japan, October 1993.

[13] F. Jelinek. “Self-Organized Language Modeling for Speech Recognition”. In *Readings in Speech Recognition*, A. Waibel and K.F. Lee, eds. Morgan Kaufmann, San Mateo, California, 1990.

[14] S. Khoubyari and J.J. Hull. “Keyword Location in Noisy Document Images.” In Proc. Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, April 1993.

[15] S. Kuo and O.E. Agazzi. “Machine vision for keyword spotting using pseudo 2d hidden markov models.” In Proc. International Conference on Acoustics, Speech and Signal Processing. Minneapolis, Minnesota, April 1993.

[16] J. Kupiec, D. Kimber, and V. Balasubramanian. “Speech-based retrieval using semantic co-occurrence filtering.” In Proc. ARPA Human Language Technology Workshop, Plainsboro New Jersey, March 1994.

[17] F. Lancaster, “Vocabulary Control for Information Retrieval, Second Edition,” *Information Resources,* Arlington, VA, 1986.

[18] David D. Lewis and Philip J. Hayes, *ACM Transactions of Office Information Systems*, Special Issue on Text Categorization, 12(3), 1994.

[19] A. Lovins and L. Lovins, “Reinventing the Wheels”, *Atlantic Monthly*, January 1995.

[20] K. Markey, P. Atherton, and C. Newton, “An Analysis of Controlled Vocabulary and Free Text Search Statements in Online Searches,” *Online Review* Vol. 4, pp. 225-236, 1982.

[21] R.D. Peacocke and D.H. Graf. “An Introduction to Speech and Speaker Recognition”. Computer, Vol. 23, No. 8, August, 1990.

[22] L.R. Rabiner, R.W. Schafer. *Digital Processing of Speech Signals.* Prentice-Hall Inc.: Englewood Cliffs, New Jersey, 1978.

[23] L.R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Application". Proc. IEEE, Vol. 77, No. 2, February 1989.

[24] G. Salton, *Automatic text processing : the transformation, analysis, and retrieval of information by computer*, Addison-Wesley, Reading, Massachusetts, 1988.

[25] Jürgen Schürmann, Norbert Bartneck, Thomas Bayer, Jürgen Franke, Eberhard Mandler, and Matthias Oberländer. "Document analysis–from pixels to contents." In Proceedings of the IEEE, Vol. 90, No. 7, July 1992.

[26] International Organization for Standardization. "Information Processing, Text and Office systems, Standard Generalized Markup Language (SGML), International Standard; 8879" 1986.

[27] K. Weber and A. Poon. "Marquee: A Tool for Real-Time Video Logging". Proc. CHI '94, ACM SIGCHI, April 1994.

[28] M. Wechsler and P. Schauble. "Metadata for Content Based Retrieval of Speech Recordings". Chapter X in this book.

[29] M. Weintraub. "Keyword-Spotting Using SRI's $Decipher^{TM}$ Large-Vocabulary Speech-Recognition System". Proc. International Conference on Acoustics, Speech and Signal Processing, Minneapolis, Minnesota, April 1993.

[30] L.D. Wilcox and M.A. Bush. "Training and search algorithms for an interactive wordspotting system." Proc. International Conference on Acoustics, Speech and Signal Processing, San Francisco, California, March 1992.

[31] L.D. Wilcox, I. Smith, and M.A. Bush. "Wordspotting for Voice Editing and Audio Indexing." Proc. CHI '92, ACM SIGCHI, Monterey, California, May, 1992.

[32] L.D. Wilcox, F.R. Chen, D. Kimber, and V. Balasubramanian. "Segmentation of speech using speaker identification." Proc. International Conference on Acoustics, Speech and Signal Processing, Adelaide, Australia, April 1994.

[33] C.W. Wightman and M. Ostendorf. "Automatic recognition of intonational features." Proc. International Conference on Acoustics, Speech and Signal Processing, San Francisco, California, March 1992.

[34] J.G. Wilpon, L.R. Rabiner, C.H. Lee, E.R. Goldman. "Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models". IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 38, No. 11, November 1990.