

Multimedia Feature Extraction in the SAPIR Project

Aaron Kaplan¹, Jonathan Mamou², Francesco Gallo³, and Benjamin Sznajder²

¹ Xerox Research Centre Europe, 6 chemin de Maupertuis, 38240 Meylan, France

² IBM Haifa Research Lab., 31905 Haifa, Israel

³ EURIX Group R&D Department, 26 via Carcano, 10153 Torino, Italy

Abstract. SAPIR is a peer-to-peer multimedia information retrieval system that can index structured and unstructured text, still and moving images, speech, and music. The system's feature extraction component, which analyzes documents to prepare them for indexing, is implemented using UIMA. It handles compound documents using an architecture of (potentially nested) splitters and mergers within a UIMA aggregate. For example, the moving image from a video is split into a number of representative video frames, each of which is processed by the same analysis engine used for still images, and then merging the results to form a unified representation of the video. The output of the feature extraction module is a document description in a representation based on the MPEG-7 standard.

1 Introduction

The SAPIR project⁴ brought together nine industrial and academic partners to build a peer-to-peer multimedia search engine that supports search over audio (both speech and music), video, still images, and text, using the query-by-example paradigm. For example, a snapshot taken with a mobile phone can be used to search for images and videos of similar objects, or a short audio recording of some music can be used to search for performances of the same musical work. Multimedia documents, by definition, contain more than one type of information, and the SAPIR query mechanism supports queries over multiple types. For example, one could search for videos that have images similar to a given snapshot and contain given words in the audio.

We used UIMA to implement SAPIR's feature extraction component. This component takes a multimedia document as input, and returns a description of the document in a representation based on the MPEG-7 standard (see Section 3). The description contains features extracted from the different media in the document, and it is used by the indexing component to insert the document into a peer-to-peer distributed index.

The feature extraction system breaks down a compound multimedia document (e.g. a video) into its component parts (e.g. the video's image frames and

⁴ <http://www.sapir.eu/>

audio track), and routes the parts to media-specific analysis engines that do the feature extraction. These analysis engines include:

- video: processes video recordings; performs shot detection, generates a representation of the video’s shot structure, and identifies a representative frame for each shot to be processed by the image annotator.
- image: processes both still images and video frames; extracts five MPEG-7 visual descriptors (ScalableColor, ColorStructure, ColorLayout, EdgeHistogram, HomogeneousTexture).
- music: processes audio recordings and MIDI files; extracts representations of melody, harmony, and rhythm.
- speech: processes audio recordings (which may be stand-alone documents or audio tracks extracted from videos); builds a word confusion network and a phoneme lattice.
- text: processes text, with or without XML or HTML markup; performs tokenization, lemmatization, named entity recognition, and summarization.

We have found no previous descriptions of systems in which UIMA was used to extract features from multiple media types in a single multimedia document. TALES⁵ is a UIMA-based system which performs multimedia mining and translation of broadcast news and news Web sites. For broadcast video news, TALES performs video capture, keyframe extraction, automatic speech-to-text conversion, machine translation of the foreign text to English, and information extraction. However, there is no publicly available description of the UIMA analytics approach.

In Section 2, we will first explain the hierarchical structure, implemented within a UIMA aggregate analysis engine, by which compound multimedia documents are decomposed, and the parts analyzed and then recomposed to form the final representation. Next, Section 3 briefly presents the MPEG-7 standard, describes the MPEG-7 based representation that is the output of our feature extraction system, and explains the corresponding UIMA feature structures. Since this workshop is attached to an NLP conference, we then briefly describe in Section 4 the analysis engines that process natural language, namely the speech and text components. We will not cover the music, image, or video analysis engines in detail here. The music component is based on research described in [1–3]. The image component is built around reference software from the MPEG-7 eXperimentation Model⁶ and the ImageMagick⁷ library; for research on the use of these features for image retrieval, see [4, 3]. The video analysis engine uses the ffmpeg⁸ and MJPEG⁹ open-source libraries; for details, see [5, 3].

The focus of this paper is on the use of UIMA for composing single-medium analysis engines into a multimedia aggregate. The main scientific contributions of the SAPIR project are described elsewhere—see citations throughout the text.

⁵ http://domino.research.ibm.com/comm/research_projects.nsf/pages/tales.index.html

⁶ <http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm>

⁷ <http://www.imagemagick.org/>

⁸ <http://ffmpeg.org/>

⁹ <http://mjpeg.sourceforge.net/>

2 Multimedia Splitting and Merging

A multimedia document is composed of parts that have different media types. For example, a video is composed of still frames and an audio track. The same basic media type may occur in different kinds of multimedia documents. For example, a frame of a video and a photograph are of the same basic media type, and as far as our feature extraction algorithms are concerned it makes no difference whether a given image comes from a snapshot or a frame of a video. We therefore thought it desirable to have a single UIMA analysis engine for each media type, e.g. a single image annotator that processes both video frames and still photographs.

We implemented this idea by defining components called *splitters* and *mergers* that decompose and recombine compound documents. A splitter is a CAS Multiplier that accepts a CAS of one media type, and outputs the original CAS plus one or more CASes of different media types. For example, the moving image splitter takes a moving image CAS (the difference between a video and a moving image will become clearer shortly) as input, and outputs that CAS plus a number of still image CASes representing selected frames of the moving image. Each splitter has a matching merger, which receives all of the CASes output by the splitter after they have been processed by other components, and assembles the extracted information into an appropriate structure in the original CAS. In our example, the moving image merger copies image features from the individual image CASes into a feature structure array in the original moving image CAS. Only the original CAS (the moving image CAS in the example) leaves the aggregate, so seen from the outside the aggregate behaves like a normal analysis engine, not a CAS multiplier. A splitter and a merger are composed in an aggregate with the appropriate feature extraction modules and a custom flow controller, which directs a CAS to the appropriate delegate (or directly to the merger) based on its media type.

The split/merge structure can be applied recursively. Figure 1 illustrates the structure of our aggregate analysis engine for video. At the top level, the video splitter generates one CAS for the moving image and another for the audio track. The moving image CAS is processed by the moving image analysis engine, which is itself a split/merge aggregate as described above. The audio CAS is processed by the speech aggregate, which is a traditional aggregate that performs speech-to-text transcription followed by text processing on the resulting transcript. (The speech and text components will be described in more detail in Section 4.)

Note that the flow of information inside the video aggregate forks and then joins. As always, the custom flow controller routes CASes to the appropriate annotators within the aggregate.

The split/merge approach allows a clean separation of concerns. It allows the image annotator to process frames extracted from videos, without knowing anything about the structure of videos or the feature structures used to describe them. To add, for example, functionality for processing web pages composed of text and embedded images, we needn't modify the image annotator to add a case

for handling web page CASes. Instead, we can write a new web page splitter and merger, and compose them with the original, unmodified image annotator.

The argument for the split/merge structure at the top level of the video aggregate is perhaps not as strong as it is at the level of the moving image aggregate. An alternative would be to put the moving image and audio parts into different views in the original video CAS, and use SOFA mappings to bind the moving image and speech analysis engines to the appropriate views. Splitting the image and audio parts into different CASes might give some advantage in terms of paralllizing the processing flow, but we have not yet pursued this idea.

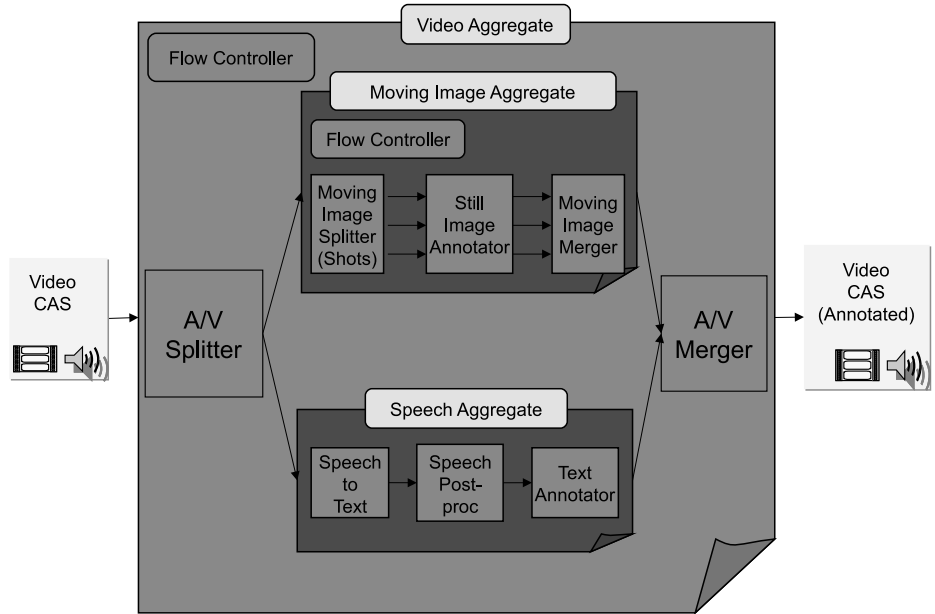


Fig. 1. The video aggregate

After processing by the video aggregate, a video CAS contains the following information: a temporal decomposition of the video into shots; for each shot, the start time and duration, the URL of a still image that represents the shot, and five MPEG-7 visual descriptors extracted from that image; a word confusion network representing the results of speech-to-text processing; a textual transcription of the WCN annotated with lemmata, named entities, and temporal offsets; and a summary of the text.

3 Feature Representations

It was decided at the outset of the SAPIR project to use the MPEG-7 standard [6, 7] to represent extracted features and other metadata for all media.

MPEG-7 provides an extremely rich XML-based formalism for representing the structure and contents of multimedia documents. Despite its expressiveness, the standard did not cover all of the types of features we intended to extract, in particular for text and music, so we defined some SAPIR-specific extensions [8].

Extracted features are first represented and manipulated as UIMA feature structures, and then in a final step the fully annotated CAS is transformed into an MPEG-7 description. Initially we hoped to create the UIMA type system definition, and the code for translating between formats, automatically. The MPEG-7 representation is defined using XML Schema, and in principle it would be possible to map XML Schema to a UIMA type system definition automatically. We attempted to do this by going via Ecore: eclipse provides automatic mapping of XML Schema to Ecore, and UIMA's `Ecore2UimaTypeSystem` goes from Ecore to UIMA. We encountered several problems with this approach. First, `Ecore2UimaTypeSystem` is not widely used, and is thus not as mature or well-tested as other parts of UIMA; we encountered a number of bugs in this code (the bugs we have discovered so far have since been fixed). Second, the full MPEG-7 standard is very large, and the corresponding UIMA type system was too big to be loaded into memory. Had we continued to pursue this approach, the next step would have been to prune the XML Schema definitions down to just the part of the standard that we actually use, which is only a small fraction of the total. In the end, we did not pursue this approach. We gave up on automating the mapping, and simply defined a UIMA type system by hand, and wrote code for translating UIMA feature structures to the subset of MPEG-7 that we use.

4 NLP Components

We will now present the feature extraction modules that handle natural language.

4.1 Spoken Information Retrieval

Search in spoken data is an emerging research area currently garnering a lot of attention from the natural language research community. We have developed a UIMA analysis engine that incorporates the state-of-the-art asset developed by IBM Research in the area of Automatic Speech Recognition (ASR) [9].

The information produced by this analysis engine is used in a novel scheme for information retrieval from noisy transcripts. The scheme uses additional output from the transcription system to reduce the effect of recognition errors in the word transcripts [10]. Although ASR technology is capable of transcribing speech to text, it suffers from deficiencies such as recognition errors and a limited vocabulary. For example, noisy spontaneous speech is typically transcribed with an accuracy of 60% to 70%. In some circumstances where there are noisy channels, foreign accents, or under-trained engines, accuracy may fall to 50% or lower. Our scheme shows a dramatic improvement in the quality of searches being

conducted within transcript information [11]. To overcome the limitations and high error rate associated with phonetic transcription and queries for terms not recognized by the ASR engines, we have developed a new technique that combines phone-based and word-based search. When people search through speech transcripts and query for terms that are outside the vocabulary domain on which the engine is trained, the engine may not return any results. The “out of vocabulary” (OOV) terms are those words missing from the ASR system vocabulary. Although phonetic transcription constitutes an alternative to word transcription for OOV search, they suffer from high error rate and are therefore not a viable alternative. We have developed algorithms specifically for fuzzy search on phonetic transcripts, thereby overcoming this problem [12–14].

4.2 Text Processing

The text analysis engine provides tokenization, lemmatization, sentence boundary detection, recognition of dates and person and place names, and summarization, for English text. It is based on the Xerox Incremental Parser (XIP) [15], a tool that performs robust and deep syntactic analysis. XIP provides mechanisms for identifying major syntactic structures and major functional relations between words on large collections of unrestricted documents (e.g. web pages, newspapers, scientific literature, encyclopedias). It provides a formalism that smoothly integrates a number of description mechanisms for shallow and deep robust parsing, ranging from part-of-speech disambiguation, entity recognition, and chunking, to dependency grammars and extra-sentential processing. Named entity recognition relies on, and is also part of, the general parsing process [16]. Measured over entities of all types, the named entity recognition system has a precision of 94% and recall of 88%.

The summarizer uses sentence, lemma, and name annotations produced by the linguistic analysis, as well as other internal XIP information such as anaphoric information, to rank the sentences of the document by informativeness and choose the most informative ones to include in the summary. Summaries can be used to facilitate browsing of results retrieved for a query.

Needless to say, the text processing functionality works better on “clean” text documents than on automatically transcribed speech. We have not attempted a formal evaluation, but our impression is that the quality of lemmatization and named entity recognition when applied to transcribed speech is degraded but remains acceptable, whereas the quality of summaries generated from speech is generally too poor to be useful, as recognition errors and errors in sentence boundary detection compound the already difficult summarization problem.

While for some media the input to the text processing module is transcribed from speech, in other pipelines the original document is textual. An analysis engine based on the open source `nekohtml`¹⁰ and `xerces`¹¹ libraries prepares XML and HTML documents, including ill-formed HTML as it is often found

¹⁰ <http://nekohtml.sourceforge.net/>

¹¹ <http://xerces.apache.org/>

on the web, for processing by the parser, which expects plain text. It creates a plain-text view in which the markup tags have been removed, and adds UIMA annotations to preserve the alignment between the plain-text view and the original view, so that at a later stage annotations added to the plain-text view by the text analysis engine can be copied back to the original view, with the offsets adjusted appropriately. Information about the location of tags is also used to influence sentence boundaries—for example, a sentence will not be allowed to span a location in the plain-text view that corresponds to a <p> (paragraph break) tag in the original view.

5 Conclusions and Future Work

To support multimedia indexing and query-by-example search, the SAPIR project has developed a feature extraction system for multimedia documents that contain combinations of text, images, video, speech, and music. The system is implemented in UIMA, using a pattern of splitters and mergers in which a multimedia CAS is split into multiple simpler CASes containing one media type each, which are processed and then recombined in order to generate a representation of the original, compound document. In this paper we have described the system architecture and some of the design decisions behind it, as well as the speech transcription and text processing modules. Descriptions of the other feature extraction modules can be found in the references.

Indexing and search in SAPIR are distributed over a peer-to-peer network, but the current version of the feature extraction subsystem is not. Since it is built on UIMA it could of course be distributed using UIMA's distributed processing functionality, but only over a network with a fixed set of nodes known ahead of time. It would be interesting to explore how UIMA could be adapted to working in a peer-to-peer network, in order to distribute the computational load of feature extraction among all participants.

6 Acknowledgements

The SAPIR project was funded by the European Commission Sixth Framework Programme. We would like to thank all of the SAPIR participants, and in particular Fabrizio Falchi and Paolo Bolettieri for the still image analysis engine, Nicola Orio and Riccardo Miotto for the music analysis engine, and Walter Allasia, Mouna Kacimi, and Yosi Mass for helpful discussions and comments.

References

1. Di Buccio, E., Masiero, I., Mass, Y., Melucci, M., Miotto, R., Orio, N., Sznajder, B.: Towards an integrated approach to music retrieval. In: Proceedings of the Fifth Italian Research Conference on Digital Library Management Systems, Padua, Italy (2009)

2. Miotto, R., Orio, N.: A music identification system based on chroma indexing and statistical modeling. In: Proceedings of the 9th International Conference of Music Information Retrieval, Philadelphia, USA (2008) 301–306
3. Kaplan, A., Bolettieri, P., Falchi, F., Lucchese, C., Allasia, W., Gallo, F., Mamou, J., Sznajder, B., Miotto, R., Orio, N., Brun, C., Coursimault, J.M., Hagège, C.: Feature extraction modules for audio, video, music, and text. Combined deliverables 3.2, 3.3, 3.4, 3.5, SAPIR (December 2008) <http://www.sapir.eu/deliverables.html>.
4. Stanchev, P., Amato, G., Falchi, F., Gennaro, C., Rabitti, F., Savino, P.: Selection of MPEG-7 image features for improving image similarity search on specific data sets. In: Proceedings of the 7-th IASTED International Conference on Computer Graphics and Imaging (CGIM 2004), ACTA Press (August 2004) 395–400
5. Allasia, W., Falchi, F., Gallo, F., Kacimi, M., Kaplan, A., Mamou, J., Mass, Y., Orio, N.: Audio-visual content analysis in P2P networks: The SAPIR approach. In: Proceedings of the 19th International Workshop on Database and Expert Systems Applications (DEXA 2008), IEEE Computer Society (September 2008) 610–614
6. International Organization for Standardization: Information technology - multimedia content description interface. ISO/IEC 15938
7. Manjunath, B.S., Salembier, P., Sikora, T., eds.: Introduction to MPEG-7: Multimedia Content Description Interface. Wiley (2002)
8. Kaplan, A., Falchi, F., Allasia, W., Gallo, F., Mamou, J., Mass, Y., Miotto, R., Orio, N., Hagège, C.: Common schema for feature extraction (revised). Deliverable 3.1, SAPIR (December 2007) <http://www.sapir.eu/deliverables.html>.
9. Soltau, H., Kingsbury, B., Mangu, L., Povey, D., Saon, G., Zweig, G.: The IBM 2004 conversational telephony system for rich transcription. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). (March 2005)
10. Mangu, L., Brill, E., Stolcke, A.: Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language* **14**(4) (2000) 373–400
11. Mamou, J., Carmel, D., Hoory, R.: Spoken document retrieval from call-center conversations. In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2006) 51–58
12. Mamou, J., Ramabhadran, B., Siohan, O.: Vocabulary independent spoken term detection. In: SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM (2007) 615–622
13. Mamou, J., Mass, Y., Ramabhadran, B., Sznajder, B.: Combination of multiple speech transcription methods for vocabulary independent search. In: Search in Spontaneous Conversational Speech Workshop, SIGIR 2008. (2008)
14. Ramabhadran, B., Sethy, A., Mamou, J., Kingsbury, B., Chaudhari, U.: Fast decoding for open vocabulary spoken term detection. In: NAACL-HLT. (2009)
15. At-Mokhtar, S., Chanod, J.P., Roux, C.: A multi-input dependency parser. In: Proceedings of the Seventh IWPT (International Workshop on Parsing Technologies), Beijing, China (October 2001)
16. Brun, C., Hagege, C.: Intertwining deep syntactic processing and named entity detection. In: ESTAL 2004, Alicante, Spain (October 2004)