UAIC: Participation in VideoCLEF Task

Tudor-Alexandru Dobrilă, Mihail-Ciprian Diaconaşu, Irina-Diana Lungu, Adrian Iftene

UAIC: Faculty of Computer Science, "Alexandru Ioan Cuza" University, Romania
{tudor.dobrila, ciprian.diaconasu, diana.lungu, adiftene}@info.uaic.ro

Abstract. This year marked UAIC[1]'s first participation at the VideoCLEF competition. Our group built two separated systems for tasks "Subject Classification" and "Affect Detection". For first task we created two resources starting from Wikipedia pages and pages identified with Google and used two tools for classification: Lucene and Weka. For the second task we extract the audio component from a given video file, with FFmpeg codec. After that, we computed the average intensity for each word from the transcript, by using Fast Fourier Transformations to analyze the sound. A brief description of our system components is given in this paper.

1 Introduction

VideoCLEF[2] offers cross-language classification, retrieval and analysis tasks on a video collection containing documentaries and talk shows.

In 2009, the collection extended the corpus used for the 2008 VideoCLEF pilot track. Task participants were provided with video data along with speech recognition transcripts, archival metadata, shot segmentation and shot-level keyframes. Two classification tasks were evaluated: "Subject Classification", which involves automatically tagging videos with subject labels, and "Affect and Appeal", which involves classifying videos according to characteristics beyond their semantic content. The track was coordinated by Dublin City University (IE) and Delft University of Technology (NL).

Our team participated in the following tasks: in *Subject Classification* (in which participants must tagging automatically videos with subject labels such as 'Archeology', 'Dance', 'History', 'Music', etc.) and in *Affect Detection* (in which participants must select keyframes using a combination of video and speech/audio features and these selected keyframes should represent the semantic content of the video, e.g., an episode of a documentary).

The way in which we classified a video accordingly to it transcript is described in Section 2, while Section 3 is concerned with presentation of details related to the extraction of keyframes. Last Section presents conclusions regarding our participation in VideoCLEF 2009.

---

[1] "Al. I. Cuza" University

[2] VideoCLEF: http://www.cdvp.dcu.ie/VideoCLEF/

2 Subject Classification

In order to classify a video according to its transcripts we perform the following steps:

- **Step 1**: For every category we extract from Wikipedia and Google web pages related to it;
- **Step 2**: From documented extracted at Step 1 we extract only relevant words and count the number of appearance for them. For every category we build resources with relevant words and number of appearances and normalize at 1000 the sum of number of appearances for every category;
- **Step 3**: Similar with Step 2 we extract and count the relevant words from video transcripts;
- **Step 4**: Video classification using extracted words from Step 3 in category clusters built at Step 2. In classification process we used combinations between results offered by Lucene and Weka tools, using resources obtained from Google or from Wikipedia, or both.

Details related to previous steps are presented below.

2.1 Extract Relevant Words from Wikipedia

First of all, we found an URL pattern for each relevant article of each category. Using this pattern we identified the most important pages from Wikipedia. The source pages for these pages are retrieved directly from Wikipedia Server creating direct connections for each page and then these source pages are save into a single file according to each category.

Second of all, for each such a file, the XHTML/HTML tags are eliminated from the files and we save only the paragraphs (the information contained in the tags).

Third of all, the stop words and punctuation signs are eliminated; and all words are transformed to lower case.

Next, we lemmatize all remaining words and for each lemma we count the number of appearances.

In the end we normalize to 1000 for each category the sum of number of appearances. This step was necessary because initial for some categories like Music the number of relevant pages was very high, and the sum of number of appearances was also very high in comparison with other categories. Without normalizing a transcript with a word from Music category is automatically classified in Music category.

2.2 Extract Relevant Words from Google

This part is similar with part performed on Wikipedia with few differences. One of differences is related to fact that from relevant pages we extract only words from <keywords> tag. The second main difference is the fact that we split the content of the <keyword> tag after comma separator and in this way we considers important for

one category a succession of few words. In this way we search the context in which relevant words for one category appear.

2.3 Lucene

Lucene is a high performance, scalable Information Retrieval (IR) library. It allows adding indexing and searching capabilities to applications. Lucene is a mature, free, open-source project implemented in Java (Hatcher, E. and Gospodnetic, 2005).

Instead to index files corresponding to categories created at previous steps from Google and Wikipedia, we created another files from these files in which every word appear by a number proportional with associated number from corresponding file. In this way the Lucene score will be higher if the word from associated file to categories has a higher number of appearances.

2.4 Weka

Weka[3] (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato. The Weka workbench (Witten and Frank, 2005) contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to this functionality.

2.5 Submitted Runs

We submitted 4 runs described below:

Table 1: Subject Classification: Characteristics of Runs

Run ID	Tools and Resources used
Run 1	• It uses only Lucene for classification • Like resources are used both resources obtained from Wikipedia and Google
Run 2	• It uses only Weka for classification • Like resources are used only resources obtained from Google
Run 3	• It uses only Weka for classification • Like resources are used only resources obtained from Wikipedia
Run 4	• It uses Weka and Lucene for classification • Like resources are used both resources obtained from Wikipedia and Google

[3] Weka: http://www.cs.waikato.ac.nz/ml/weka/

3 Affect Detection

At this part, our work is based on the assumption that a narrative peak is a point in the movie where the narrator raises his voice within a given phrase, in order to emphasize a certain idea. This means that a group of words is said more intensely than the way previous words are said and, since this applies in any language, we were able to develop a language independent application.

This is why our approach is based on two aspects of the video: the sound and the ASR transcript.

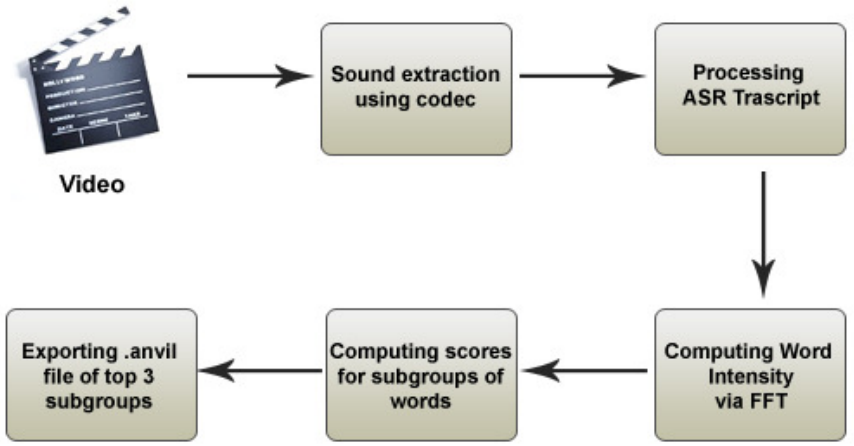

Figure 2: UAIC system used in Affect Detection task

The first step is the extraction of the audio from a given video file, which we accomplished by using FFmpeg[4] codec. We then computed the average intensity for each word from the transcript, by using Fast Fourier Transformations (FFT[5]) to analyze the sound.

[4] FFmpeg: http://ffmpeg.org/
[5] FFT: http://en.wikipedia.org/wiki/Fast_Fourier_transform

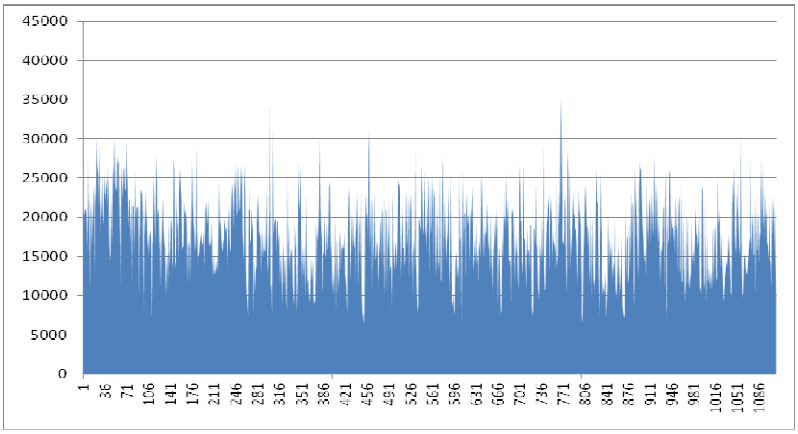

Figure 1: Graph where the X-axis represents the words indexes and the Y-axis the average intensity of the words for **BG_36926.cinepak.avi**

We then computed a score for any group of words (which spanned between 5 and 10 seconds) based on the previous group of words. The score is a weighted mean of several metrics.

Example of results for **BG_36926.cinepak.avi :**

Table 2: Results for **BG_36926.cinepak.avi**

Interval	Score	Metrics
456.21 - 464.52	2.400728805400962	**WORDS** : 11 **AVG** : 18040.010846944315 **STDEV** : 4564.3711633004605 **KURTOSIS** : 0.9652762936344059 **Q1** : 15247.984478723254 **Q3** : 20143.514196512962
429.49 - 438.03	2.154955493176883	**WORDS** : 6 **AVG** : 15511.604737251293 **STDEV** : 7446.017829654594 **KURTOSIS** : 4.414204969079757 **Q1** : 11919.996752204017 **Q3** : 14682.031805791958
129.15 - 135.51	1.8945333420277617	**WORDS** : 21 **AVG** : 17622.175516302843 **STDEV** : 7708.211025959182 **KURTOSIS** : 1.7932215422413296 **Q1** : 11538.565026101109 **Q3** : 19575.974092855566

We then considered only the top 3 scores, which were exported in *.anvil* format for later use in Anvil Player. An example of output is in next table:

Table 3: Output example for "BG_36926.cinepak.avi" video

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
<annotation>
  <head>
    <specification src="Segment.xml"/>
    <video src="BG_36926.cinepak.avi"/>
    <info key="coder" type="String">Example</info>
  </head>
  <body>
    <track name="Segment" type="primary">
    <el end="396.6" index="0" start="389.42">
      <attribute name="Label">peak</attribute>
    </el>
    <el end="117.85" index="1" start="112.7">
      <attribute name="Label">peak</attribute>
    </el>
    <el end="241.23" index="2" start="234.02">
      <attribute name="Label">peak</attribute>
    </el>
    </track>
  </body>
</annotation>
```

We submitted 3 runs with following characteristics:

Table 4: Affect Detection: characteristics of runs

Run ID	Metrics Used When Computing The Score
Run 1	• Ratio of Current Group and Previous Group Means of Intensities • Ratio of Current Group and Previous Group Quartile Coefficients of Dispersion
Run 2	• Ratio of Current Group and Previous Group Means of Intensities • Ratio of Current Group and Previous Group Quartile Coefficients of Dispersion • Ratio of Current Group and Previous Group Coefficients of Variation
Run 3	• Ratio of Current Group and Previous Group Means of Intensities • Ratio of Current Group and Previous Group Coefficients of Variation

In total, 60 hours of assessor time were devoted to creating the reference files of the narrative peaks for the 45 Beeldenstorm episodes used in the VideoCLEF 2009 Affect Task. Three assessors watched each of the 45 test files and marked their top three narrative peaks using the Anvil tool.

The results are presented in next table:

Table 5: Runs Evaluation

Run ID	Point based scoring	Peaks based scoring 1	Peaks based scoring 2	Peaks based scoring 3
Run 1	33	26	7	2
Run 2	41	29	10	3
Run 3	33	24	7	2

4 Conclusions

This paper presents the UAIC system which took part in the VideoCLEF 2009 competition. Our group built two separated systems for tasks "Subject Classification" and "Affect Detection".

For *Subject Classification* task we created two resources starting from Wikipedia pages and pages identified with Google search engine. These resources are then used by Lucene and Weka tools for classification.

For *Affect Detection* task we extract the audio component from a given video file, with FFmpeg codec. After that, we computed the average intensity for each word from the transcript, by using Fast Fourier Transformations to analyze the sound. In the end, like final result in this task we considered only the top 3 values obtained in previous step.

Acknowledgements

We also like to give a special "thank you" to those who helped from the very beginning of the project: our colleagues from second year group 1 B.

References

1. Hatcher, E. and Gospodnetic, O.: Lucene in action. *Manning Publications Co*. (2005)
2. Witten, I. H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd Edition. Morgan Kaufmann, San Francisco. Retrieved 2007-06-25. (2005) http://www.cs.waikato.ac.nz/~ml/weka/book.html.