Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Volume 22, No. 1, ISSN 1726-9679
ISBN 978-3-901509-83-4, Editor B. Katalinic, Published by DAAAM International, Vienna, Austria, EU, 2011
Make Harmony between Technology and Nature, and Your Mind will Fly Free as a Bird
Annals & Proceedings of DAAAM International 2011

VIDEO METADATA IN WEB BASED APPLICATIONS

BOICEA, A[lexandru]; CAPITANESCU, I[ulia]; RADOI, C[odrut] D[umitru] & PETRE, M[ihai] - R[azvan]

Abstract: *This paper analyses a technique for extracting metadata regarding YouTube videos that are embedded in various websites. Web pages are defined by their URL and the results are stored in a relational database. The algorithm uses a web crawler for page browsing and HTML tags for metadata discovery.*
Key words: *metadata, video, embed, web, crawler*

1. INTRODUCTION

As the World-Wide-Web is incessantly expanding, the quantity of information one can find on the Internet is continually growing. As a result, relevant information regarding a specific topic is harder to find. (Safari , 2004) This is why the concept of metadata was introduced. Metadata is 'data about data', which means that it describes a certain resource on the Internet.

There are a few things regarding metadata that need to be considered. First of all, the information that is captured by the metadata needs to be defined. This depends on the type of resource and the purpose of the metadata. The second aspect concerns the way metadata is produced. The final problem regards the way metadata is accessed and used. (Ianella & Waugh, 2011) YouTube offers web site developers the possibility of inserting videos into their web sites by just copying a few lines of code into their HTML source. This process leads to the generation of a dedicated area of the page where a user can watch the video directly without having to navigate to the YouTube video page. This particular approach doesn't offer the user any information other than the actual video. Therefore, the search on a certain subject is not very relevant in regard to pages containing embedded YouTube videos.

The idea of this paper is to implement an algorithm that extracts metadata regarding YouTube videos embedded in a certain web page. Information is extracted from the HTML tags of the corresponding YouTube page and is stored in a customized database.

The end purpose is to easily locate relevant videos. The application can also be used for data mining purposes, to compute statistics or establish relationships between various web resources.

2. EMBEDDING YOUTUBE VIDEOS

There are two ways of embedding YouTube videos. The old way begins with the *<object>* tag and only supports Flash playback
(http://www.google.com/support/youtube/bin/answer.py...):

```
<object width="960" height="750"><param name="movie"
value="http://www.youtube.com/v/WApx6lXAwMQ?
fs=1&amp;hl=ro_RO&amp;rel=0"></param><param
name="allowFullScreen" value="true"></param><param
name="allowscriptaccess"
value="always"></param><embed
src="http://www.youtube.com/v/WApx6lXAwMQ?
fs=1&amp;hl=ro_RO&amp;rel=0"
type="application/xshockwave-flash"
allowscriptaccess="always"
allowfullscreen="true" width="960"
height="750"></embed></object>
```

A newer version uses the *<iframe>* tag and supports both Flash and HTML5 video (http://apiblog.youtube.com/...):

```
<iframe title="YouTube video player" width="960"
height="750"
src="http://www.youtube.com/embed/WApx6lXAwMQ"
frameborder="0" allowfullscreen></iframe>
```

Some services only support the older version so the application will search for both types of embedded video code.

3. STAGES OF THE APPLICATIONS

The user is prompted to specify an URL that identifies the page containing embedded YouTube videos. The application will extract metadata for each of these videos. Based on the input address, it scans the page source in search of relevant tags (<iframe> for newer versions of embedded videos and <object> <param name="movie"> for previous ones). The links to YouTube pages containing the videos are discovered in this manner. Identified links are accessed by using a web crawler.

Each YouTube page source is parsed in order to find specific tags, knowing that they contain useful information that will later be added to the metadata database.

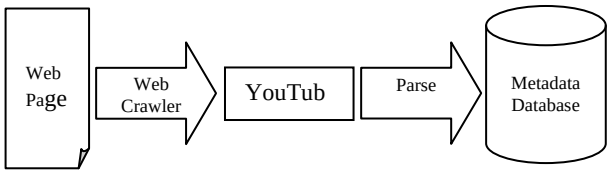

Fig.1. Application stages

4. WEB CRAWLER

The web crawler algorithm has the following phases (Blum et al, 1998):

1.getHtmlSource(url)
  1.1 createHttpWebRequest to access the given URL
  1.2 getHttpWebResponse

1.3 create streamReader from HttpWebResponse
1.4 while ((streamReader.ReadLine()) != null)
1.5 write read line into local file

2.parseURLPageSourceLocally
2.1 get relevant embedded url:
2.1.1 <iframe>
2.1.2 <object> <param name=”movie”>

3. for each embedded YouTube video
3.1 if (found link type <iframe>)
3.1.1 replace "embed/" with "watch?v="
3.1.2 browse to www.youtube.com
3.2 if (found link type object> <param name=”movie”>)
3.2.1 replace "v/" with "watch?v="
3.2.1 browse to www.youtube.com
3.3 similar step 1; using identified link - write in different file
3.4 similar step 2; identify tags like:
+ span id=”eow-title”
+ <a id=”watch-username”>
+
+ etc.
3.4.1 write data into database

5. YOUTUBE PAGE PARSING

The YouTube page source is parsed to find specific tags, in order to extract metadata regarding the video. Tags and the specific information are listed below:

- Title →
- Author → <a id=”watch-username”>
- Watch count →
- Likes →
- Dislikes →
- Upload date →
- Description →
- Category → <span id=”eow-category”
- Tags →

Information about various user comments can also be extracted from the YouTube web page by searching for the tag:

This is a list of all the comments for a certain video. This list can be ordered by the number of likes. Another option would be to only store the most popular comment.

6. DATABASE STRUCTURE

The database consists of four tables linked together by auxiliary tables that resolve the “many-to-many” relationship problem, as shown in Fig.2.

“Video” holds all the necessary metadata regarding a video: a direct link, its title, who uploaded it, when it was uploaded, the number of likes and dislikes, how many times it was watched, a description of the video and the category that it belongs to.

If more information is needed, additional data regarding a video can be found in the following tables: “Comment” contains the most “liked” comments of the video, “Tag” displays tags associated with each video and “Source_Page” holds the URL and the description of the input page.

Although we could have used the URL as the identifier for the *Source_Page* table, it was better to create a new column that is a numeric id in order to make an easier connection to the *Source_Page_Video* table.

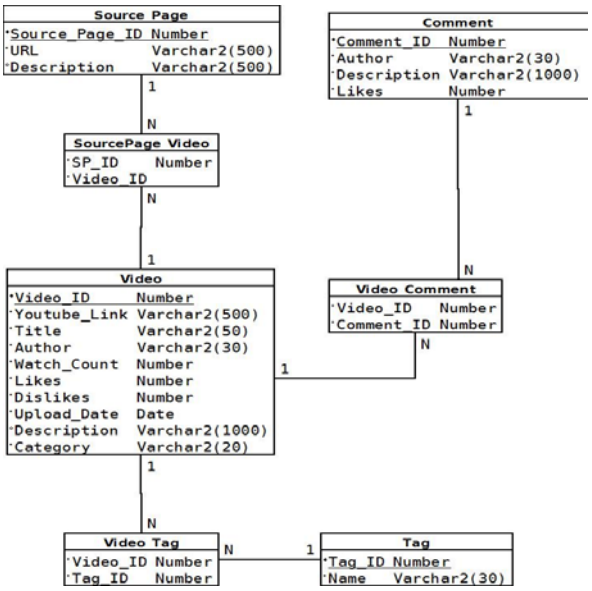

Fig.2. Database diagram

7. CONCLUSIONS

This paper emphasizes the need of using metadata regarding embedded YouTube videos for search optimization. Its purpose was to obtain as much information as possible in order to describe the videos on a certain web page and store them in a database for future usage or processing. The main idea is that all the information you need is contained in the HTML source of the YouTube page.

Using this approach on a large scale could lead to further standardization of video embedding and metadata sets for video hosting web sites. In addition to this, Internet browsers could be modified in order to display such metadata for videos embedded in the currently displayed web page.

This approach has one very important limitation: it is restricted to YouTube videos by the URL format and the extracted metadata set.

Hence, the next step is to extend metadata extraction for other types of web resources (images, audio etc.) in order to easily locate relevant pages related to a certain subject.

Future development may include extending the application to support other types of video embedding offered by different video hosting web sites. The application can be further extended with data-mining algorithms for statistic computing. (Ungureanu & Boicea, 2008)

8. REFERENCES

Blum, T., Keislar D., Wheaton J. & Wold E.(2011) *Writing a Web Crawle in the Java Programming Language, Available on:* http://java.sun.com/developer/technicalArticles/ThirdParty/WebCrawler/ *Accessed: 2011-05-13*

Iannella R. & Waugh A.(2011) *Metadata: Enabling the Internet Available on:* http://archive.ifla.org/documents/libraries/cataloging/metadata/ianr1.pdf *Accessed: 2011-05-13*

Safari M.(2004) *Metadata and the Web*, Available *on:* http://www.webology.ir/2004/v1n2/a7.html *Accessed: 2011-05-13*

Ungureanu D. & Boicea A.(2008) *A Depth First Search Algorithm for Mining Intertransaction Association Rules*, The 3rd International Conference on Software and Data Technologies, ICSOFT ‘08, pp 148-153, Porto 2008

***(2011)http://www.google.com/support/youtube/bin/answer.pyanswer=171780&expand=UseOldEmbedCode#oldcode Embed a YouTube video *Accessed on: 2011-05-13*