Let Google Index Your Media Fragments

Yunjia Li, Dr Mike Wald and Dr Gary Wills
{yl2,mw,gbw} @ecs.soton.ac.uk
School of Electronics and Computer Science
University of Southampton
Agenda

• Media Fragments
• Problems of media fragments indexing
• The model to let Google index media fragments
• Discussion
• Conclusions
Find Charlie

If you search “Charlie” in Google and find the following pictures in the result list...

This is Charlie!
Find Charlie (cont)

• Again, where is Charlie?

You may expect that the search engine can tell you where exactly is Charlie.
Media Fragment

• Denote the inside content of multimedia resources

• Four dimension defined in Media Fragment 1.0 spec
  – Temporal dimension (10s to 20s of a 20mins long video clip)
  – Spatial dimension (a rectangle area in an image)
  – Track dimension (different sound tracks or captioning for different languages)
  – Named dimension (the combination of the above three)
Current Situation

- Multimedia uploading, sharing, tagging is easy
- Searching WHOLE multimedia resource is easy
- But searching PART of multimedia resource is difficult
  - Annotations are not linked to media fragments
  - Linked using javascript
  - no unique page for each media fragment
  - Example
What we want to do...

Title: The next Web of open, linked data
Presenter: Tim Berners Lee

- 06:02: Linked Data Principles
- 07:28: Grassroot picture
- 08:21: DBpedia
- 09:15: Government Data

Gimme the FRAGMENT, not the WHOLE video!
The Difficulties

- The landing page is not search-engine-friendly
  - Everything is on the same page and the notion of media fragment is not explicitly embedded in HTML
  - Ajax content is ignored (e.g. Youtube interactive trans)
  - No semantic descriptions, no preview in search results
The Difficulties (cont)

That’s it!
Google’s Ajax Content Crawler

- The Crawler is designed to index Ajax content
- Replace token “#!” in URLs with “_escaped_fragment_”

3. Server maps from ugly URL to pretty URL: www.example.com/page?query#!key=value
4. Server creates HTML snapshot for pretty URL
1. Crawler maps from pretty URL to ugly URL: www.example.com/page?query&_escaped_fragment=key=value
2. Crawler requests ugly URL
5. Server returns HTML snapshot
6. Crawler processes HTML snapshot, extracts pretty URLs

*Diagram from https://developers.google.com/webmasters/ajax-crawling/docs/getting-started
The Key Ideas in Solution

• The fragment information must be included in the URL
  – Syntax: W3C Media Fragment 1.0 Specification

• Prepare two sets of pages for every media fragment
  – Share the same landing page for users
  – the SEO page for search engine

• Landing page keeps the original user interaction
  – Highlight media fragments on opening

• SEO page
  – ONLY includes annotations of the media fragment
  – Embed rich snippet
The Solution

1: Submit pretty URL replay/1#!t=3,7 to the crawler
2: Crawler asks server for replay/1?_escaped_fragment_=t=3,7
3: Redirect the request to the snapshot page generated by the server. The snapshot page only contains annotations and Microdata for “#t=3,7”
4: The snapshot page is returned to the crawler with URL replay/1#!t=3,7
5: A user searches keyword “Terrace Theater”
6: Google includes replay/1#!t=3,7 in the search results
7: The user click the link and ask for the document at replay/1#!t=3,7
8: The server returns the landing page containing both “Terrace Theater” and “Linked Data”
9: The landing page highlights the media fragment by start playing from 3s to 7s
Introduction of Synote

• We implemented the model in Synote system

• User can generate annotations and synchronise them with audio-visual resources

• Synote doesn’t store video, audio, image files

• Synote stores:
  – The URL references to video, audio image files online
  – User generated annotations and synchronisation points

• Single Resource: Tag, Note, Slide, etc

• Four categories of compound resources: Multimedia, Transcript, Synmark (tags, description), Presentation Slides
Demo Time!

Synote v.s. TED Talks
All kinds of conceptual things, they have names now that start with HTTP
Discussion

• This is a hack! It now only works for Google

• The Media Fragment URI syntax is slightly modified

• Hashbang URL “#!” has many side effects [1]

• Microdata is embedded in snapshot pages for each media fragment

• Keep another set of snapshot pages for SEO and do not need to abandon the existing landing pages

• A better solution could be adding vocabularies about media fragment to schema.org

Workload

- For existing applications, you need
  - Includes hashbang in your url
  - URL mapping for “#!” and “_escaped_fragment_”
  - Programmes to generate the snapshot page
  - Highlight the fragment when the page is opened
Conclusion

• Initial attempts to improve the online presence of media fragments

• Future work
  – For more search engines, Yahoo!, Bing
  – Consider other aspects of SEO

• More media fragments could be published to benefit search
Questions?