

University of Southampton Research Repository ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

UNIVERSITY OF SOUTHAMPTON

Media Fragment Semantics: The Linked Data Approach

by

Yunjia Li

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the
Faculty of Physical Sciences and Engineering
School of Electronics and Computer Science

June 2015

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF PHYSICAL SCIENCES AND ENGINEERING
SCHOOL OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

by Yunjia Li

In the last few years, the explosion of multimedia content on the Web has made multimedia resources the first class citizen of the Web. While these resources are easily stored and shared, it is becoming more difficult to find specific video/audio content, especially to identify, link, navigate, search and share the content inside multimedia resources. The concept of media fragment refers to the deep linking into multimedia resources, but making annotations to media fragments and linking them to other resources on the Web have yet to be adopted. The Linked Data principles offer guidelines for publishing Linked Data on the Web, so that data can be better connected to each other and explored by machines. Publishing media fragments and annotations as Linked Data will enable the media fragments to be transparently integrated into current Web content.

This thesis takes the Linked Data approach to realise the interlinking of media fragments to other resources on the Web and demonstrate how the Linked Data can help improve the indexing of media fragments. This thesis firstly identifies the gap between media fragments and Linked Data, and the major requirements that need to be fulfilled to bridge that gap based on the current situation of presenting and sharing multimedia data on the Web. Then, by extending the Linked Data principles, this thesis proposes **Interlinking Media Fragment Principles** as the basic rationale and best practice of applying Linked Data principles to media fragments. To further automate the media fragments publishing process, a core RDF model and a media fragment enriching framework are designed to link media fragments into the Linked Open Data Cloud via annotations and visualise media fragments on the Web pages. A couple of examples are implemented to demonstrate the use of interlinked media fragments, including the case to enrich YouTube videos with named entities and using media fragments for video classifications. The **Media Fragment Indexing Framework** is proposed to solve the fundamental problem of media fragments indexing for search engines and, as an example, Twitter is adopted as the source for media fragment annotations. The thesis concludes that applying Linked Data principles to media fragments will bring semantics to media fragments, which will improve the multimedia indexing on a fine-grained level and new research areas can be explored based on the interlinked media fragments.

Contents

Nomenclature	xv
Acknowledgements	xvii
1 Introduction	1
1.1 Research Questions	4
1.2 Contributions of the Thesis	5
1.3 Structure of the Thesis	8
1.4 Acknowledgements	11
2 Literature Review	13
2.1 Multimedia Annotations and Synote	14
2.2 Semantic Web and Linked Data	16
2.2.1 Linked Data Technologies and Principles	17
2.2.2 Linked Datasets	18
2.2.3 Named Entity Recognition and Disambiguation	20
2.2.4 Linked Data Publishing Patterns	21
2.2.5 The Infrastructure of Linked Data-Driven Applications	24
2.3 Media Fragments and Multimedia Annotation Ontologies	25
2.3.1 W3C Media Fragment URI	26
2.3.2 Ontology for Media Resource	27
2.3.3 Other Related Standards and Vocabularies	28
2.4 Applications of Media Fragments and Semantic Multimedia Annotation . .	30
2.5 Video Classification	34
2.6 Summary	35
3 Use Case Study and Requirement Analysis	37
3.1 Interlinked Media Fragments Usage on the Web	38
3.1.1 Use Case 1 (UC1): Sharing Media Fragments in Social Networks . .	38
3.1.2 Use Case 2 (UC2): Search Media Fragments Based on User-generated Content	39
3.1.3 Use Case 3 (UC3): Reasoning Based on Timeline	41
3.2 Multi-disciplinary Cases	42
3.2.1 Use Case 4 (UC4): UK Parliamentary Debate	42
3.2.2 Use Case 5 (UC5):US Supreme Court Argument Analysis	43
3.2.3 Use Case 6 (UC6):Improving Political Transparency by Analysing Resources in Public Media	44
3.3 Applying Media Fragments and Linked Data to Existing Applications . .	45

3.3.1	Use Case 7 (UC7): Media Fragments in YouTube	45
3.3.2	Use Case 8 (UC8): Media Fragments in Facebook	47
3.3.3	Use Case 9 (UC9): Publishing Media Fragments in the Synote System	48
3.3.4	Use Case 10 (UC10): Publishing Media Fragments for Edina Me- diahub	49
3.4	Problem Discussion	51
3.4.1	Two Types of “Server”	51
3.4.2	Two Different Concepts of “Video”	55
3.4.3	Three cases of “Annotations”	56
3.5	Summary of Requirements	57
4	Interlinking Media Fragments Principles	63
4.1	Choosing URIs for Media Fragments	64
4.2	Dereferencing URIs for Media Fragment	69
4.3	RDF Description for Media Fragments	74
4.3.1	Choosing Vocabularies to Describe Media Fragments	75
4.3.2	Describe Media Fragments Annotations	76
4.4	Media Fragments Interlinking	78
4.5	Summary of the Principles	81
5	A Framework for Linking Media Fragments into the Linked Data Cloud	83
5.1	RDF Descriptions for Media Fragments	84
5.1.1	Core Model for Media Fragment Enrichment	84
5.1.2	Example Implementation of Core Model for Media Fragment En- richment	86
5.2	Media Fragment Enriching Framework	88
5.2.1	Metadata and Timed-Text Retrieval	90
5.2.2	Named Entity Extraction, Timeline Alignment and RDF Generation	91
5.2.3	Media Fragment Enricher UI	92
5.2.4	Implementation of Media Fragment Enricher Services	92
5.3	Creating Enriched YouTube Media Fragments with NERD using Timed Text	93
5.4	Video Classification using Media Fragments and Semantic Annotations . .	96
5.4.1	Dataset	97
5.4.2	Methodology	101
5.4.3	Experiments and Discussion	103
5.4.3.1	Overall Accuracy	103
5.4.3.2	Breakdown scores per Channel	104
5.4.4	Summary of the Video Classification	107
5.5	Summary	108
6	Visualisation of Media Fragments and Semantic Annotations	111
6.1	Synote Media Fragment Player	111
6.1.1	Motivation	112
6.1.2	Implementation and Evaluation of the smfplayer	113
6.2	Visualise Media Fragments with Annotations	117
6.3	Summary	118

7	Automatic Media Fragments Indexing For Search Engines	119
7.1	Media Fragments Indexing Framework	121
7.1.1	Problem Analysis	121
7.1.2	Implementation	122
7.1.3	Evaluation and Discussion	124
7.2	A Survey of Media Fragment Implementation on Video Sharing Platforms	127
7.2.1	Methodology	128
7.2.2	Survey Results	130
7.2.3	Discussion	132
7.3	Indexing Media Fragments Using Social Media	134
7.3.1	Workflow of Twitter Media Fragment Indexer	135
7.3.2	Implementation and Evaluation of Twitter Media Fragment Indexer	137
7.4	Summary	141
8	Conclusions and Future Work	143
8.1	Conclusions	143
8.2	Future Work	147
8.2.1	Media Fragment Semantics in Debate Analysis	147
8.2.2	Extension of Media Fragment Indexing Framework	149
8.2.3	Visualisation of Media Fragments and Annotations On Second Screen	150
8.2.4	Video Classification with Media Fragments	151
8.2.5	Formal Descriptions of Interlinking Media Fragments Principles . .	151
A	Namespaces Used in the Thesis	153
B	Results of Twitter Media Fragment Monitor Using Twitter Firehose API	155
C	The Formal Definition of Media Fragment Publishing	157
C.1	RDF Documents as Labelled Directed Multigraph	157
C.2	A Mathematics Description for Linked Data Principles	158
C.3	Formalisation of Publishing Media Fragments as Linked Data	161
	Bibliography	163

List of Figures

1.1	The gap between raw multimedia data and interlinked multimedia annotations	3
1.2	Relationship Between Different Chapters in the Thesis	10
2.1	Synote Object Model	16
2.2	Linked Data Cloud at September 2011	19
2.3	Linked Data Publishing Patterns (Heath and Bizer, 2011)	22
2.4	Concept of a Linked Data-driven Web application (Hausenblas, 2009) . .	24
3.1	An example of interlinking media fragments to other resources	40
3.2	Addressing relationships between media fragments using Time Ontology .	41
3.3	Screenshot of Synote System	48
3.4	Edina Mediahub’s architecture	50
3.5	The relationship between Player Server and Multimedia Host Server . . .	53
3.6	The dependencies among requirements	60
4.1	Use Content Negotiation and 303 Redirect to dereference Media Fragments	72
4.2	Media Fragments Interlinking and Publishing Patterns	79
5.1	The graph depicts the MediaFragment serialization and how an Entity and its corresponding text are attached to a MediaFragment though an Annotation.	85
5.2	The graph depicts the MediaFragment serialization and how an Entity and its corresponding Subtitle are attached to a MediaFragment though an Annotation	87
5.3	Architecture of proposed Media Fragment Enriching Framework	89
5.4	Implementation of the Media Fragment Enricher Framework using NERD for metadata extraction, name entity recognition and disambiguation . . .	94
5.5	Distribution of named entities extracted from subtitles for each channel and the summary of their temporal position in the videos	100
5.6	Accuracy Comparison for each Algorithm-Experiment Pair	104
6.1	Screenshot of Synote Media Fragment Player	115
6.2	Screenshot of Media Fragment Enricher	117
7.1	Google Ajax Crawler	122
7.2	Model to improve media fragment presence based on Google Ajax Crawler	123
7.3	Screenshot of Synote replay page	125
7.4	Search results comparison between TED Talks and Synote	126

7.5	Percentage of Page Views for Websites Implementing Media Fragments Out of the Total Page Views for the 59 Websites Investigated	133
7.6	The Workflow of Twitter Media Fragment Indexer	136
7.7	The Landing Page in Twitter Media Fragment Indexer	139
7.8	The Snapshot Page in Twitter Media Fragment Indexer	139
7.9	Searching Media Fragment URIs provided by Twitter Media Fragment Indexer	140
8.1	Screenshot of UK Parliament Demonstration	148
8.2	User generated annotations in UK Parliament Demonstration	148
C.1	An example of interlinking datasets into the linked data cloud	161

List of Tables

2.1	Applications about media fragments and semantic multimedia annotations	31
3.1	Relationships between requirements and research questions	61
5.1	Comparison of Harmonised Model with YouTube, Dailymotion Metadata Schemas	91
5.2	Upper part shows the average number of named entities extracted. Lower part shows the average number of entities for the top 9 NERD top categories grouped by video channels.	95
5.3	Video and Video Metadata Distribution for the Different Channels (<i>ne</i> stands for named entity)	98
5.4	Number of Named Entities grouped by Type for Each Channel	98
5.5	Precision (P), Recall (R) and F-measure (F1) on different channels for the experiments using Logistic Regressions (%), $\lambda = 0.0001$	105
5.6	Precision (P), Recall (R) and F-measure (F1) on Various Channels for the Experiments using K-Nearest Neighbour (%), $k = 20$	105
5.7	Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Naive Bayes (%).	105
5.8	Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Support Vector Machine (%).	106
6.1	Compatibility of Synote Media Fragment Player with different browsers and devices (Test file container MP4, video codec H.264).	116
7.1	Media Fragment Compatibility on Video Hosting Services (Oct. 2013) . .	130
7.2	The Supported Media Fragment Syntax in Different Video Hosting Services (Oct. 2013)	133
7.3	Number of Media Fragment URIs shared in each Website	138
B.1	Twitter Media Fragment Monitor Using Twitter Firehose API	156
C.1	Definition of sets and functions for the formal description of linked data .	157

Listings

2.1	Example RDF Description of Media Fragment Using Ninsuna Ontology .	28
4.1	Examples of YouTube Deep Linking URIs	65
4.2	Example Media Fragment URIs	66
4.3	Example of using ma:locator	77
4.4	N3 Code Clip to Describe a Media Fragment	77
5.1	Example RDF Description of Media Fragment Serialised in Turtle	87
5.2	Example RDF Description of Named Entity Serialised in Turtle	88
5.3	Example RDF Description of Annotations Serialised in Turtle	88
5.4	JSON Format of the Harmonised Metadata Model	90
6.1	Example code of initialising smfplayer	114
7.1	Test Tweets for Twitter Media Fragment Indexer	140
A.1	Namespaces Used in this Thesis	153

Nomenclature

HLS	HTTP Live Streaming
KNN	K-Nearest Neighbour
LG	Logistic Regression
LOD Cloud	the Linked Open Data Cloud
NB	Naive Bayes
NERD	Named Entity Recognition and Disambiguation
NIF	Natural Language Processing Interchange Format
NLP	Natural Language Processing
RDF	Resource Description Framework
SVM	Supported Vector Machine
SRT	SubRip Text format
UA	User Agent
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
W3C-MA	W3C Ontology for Media Resource 1.0 (a.k.a. Media Annotation Ontology)
W3C-MFURI	W3C Media Fragment URI 1.0 (basic)
WebVTT	The Web Video Text Tracks Format

Acknowledgements

I would like to thank my supervisors Dr Mike Wald and Dr Gary Wills for their support towards the completion of this PhD thesis.

To my dear wife Pei Zhang and my son Yichen Li

Chapter 1

Introduction

Globally, consumer Internet video traffic will be 69 percent of all consumer Internet traffic in 2017, up from 57 percent in 2012. This percentage does not include video exchanged through peer-to-peer (P2P) file sharing. The sum of all forms of video (TV, video on demand [VoD], Internet, and P2P) will be in the range of 80 to 90 percent of global consumer traffic by 2017.

—*Cisco Visual Networking Index: Forecast and Methodology, 2012-2017*

Web applications today have been enriched with various multimedia resources and annotations. In a broad sense, multimedia annotations include a resource's own metadata (e.g. creator, created date, copyright information of the multimedia resource) and other resources which annotate this resource for content enrichment purposes (e.g. user generated comments). The success of multimedia storage and sharing applications, such as Flickr¹ and YouTube², has proved that, instead of plain text resources, multimedia resources are being raised to “first class citizens” of the Web. Blogs, wikis and social network applications like Facebook³, which allow sharing multimedia resources from various repositories, generate further massive quantities of annotations stored independently from the original media repositories. According to the study by comScore, in June 2013, 183 million U.S. internet users watched online video content for an average of 21.7 hours per viewer per month⁴. Compared with the same statistic in July 2007, which was 3 hours per viewer per month⁵, the last few years have seen an explosion of video content on the Web.

The term “media fragment refers to the content inside multimedia objects, such as a

¹<http://www.flickr.com>

²<http://www.youtube.com>

³<http://www.facebook.com>

⁴http://www.comscore.com/Insights/Press_Releases/2013/7/comScore_Releases_June_2013_U.S._Online_Video_Rankings

⁵http://www.comscore.com/Press_Events/Press_Releases/2007/09/US_Online_Video_Streaming

certain area within an image, or a ten-minute segment within a one-hour video. Media fragments usually denote the temporal-spatial nature of multimedia resources or different tracks (audio, subtitles, etc.) encoded in the same container. While multimedia resources are becoming easier to store and share in the Web 2.0 era, it also has become more and more difficult to search for and locate specific video/audio content, especially inside multimedia resources. As an analogy, the readers of a textbook find it difficult to read the book if there is no content table and page numbers. Thus the indexing of media fragments is critical for multimedia resources to be transparently integrated into the current Web content. With this goal, many people have been trying to make the video/audio the “first class citizen of the Web, such as the W3C Media Fragment Working Group⁶ and the W3C Media Annotation Working Group⁷. Various standards have been published as W3C recommendations for the associated areas, such as Media Fragment URI 1.0 (W3C-MFURI) (Troncy et al., 2012) and Ontology for Media Resource (a.k.a Media Annotation Ontology, W3C-MA) (Sasaki et al., 2012), to help identify, link, navigate, search and share the multimedia resources on the Web, especially media fragments on the Web. More specifically, Van Deursen et al. (2012) states that people need to be able to:

1. link to and bookmark media fragments
2. visualise and browse media fragments in browsers
3. provide structured annotations for media fragments
4. experience synchronized media and annotations

However, the deep linking into media fragments and making multimedia annotations on a fine-grained level are still not common practice, which leads to insufficient indexing of content inside multimedia resources. While billions of videos are hosted by YouTube, Dailymotion and Vimeo, most results by major search engines for multimedia resources stay at the level of the whole multimedia resource. Some research has been devoted to address media fragments and semantic multimedia annotations, such as MPEG-7 (Martinez, 2004) and Core Ontology of Multimedia (COMM) (Arndt et al., 2007), but there is still no systematic methodology on how media fragments in those major multimedia sharing platforms could be interlinked to other resources on the Web, thus making them more discoverable and accessible.

The Linked Data principles (Berners-Lee, 2006) offer guidelines for publishing linked data on the Web, so that data can be better connected and explored by machine. Using Linked Data, all the resources on the Web will be described using the Resource Description Framework (RDF) (Manola and Miller, 2004), the relationships between resources

⁶<http://www.w3.org/2008/WebVideo/Fragments/>

⁷<http://www.w3.org/2008/WebVideo/Annotations/>

can being expressed by triples in “subject, predicate and object” format without ambiguities. Linked data principles do not restrict the objects they can be applied to, so theoretically raw multimedia data, including media fragments and annotations, can be published as linked data and linked with other resources. However, there are questions as to why Linked Data has not been widely applied to media fragments and what are the obstacles to this. This work explores the Linked Data approach of using semantics with media fragments to link media fragments to other resources on the Web, and publishing structured annotations for media fragments.

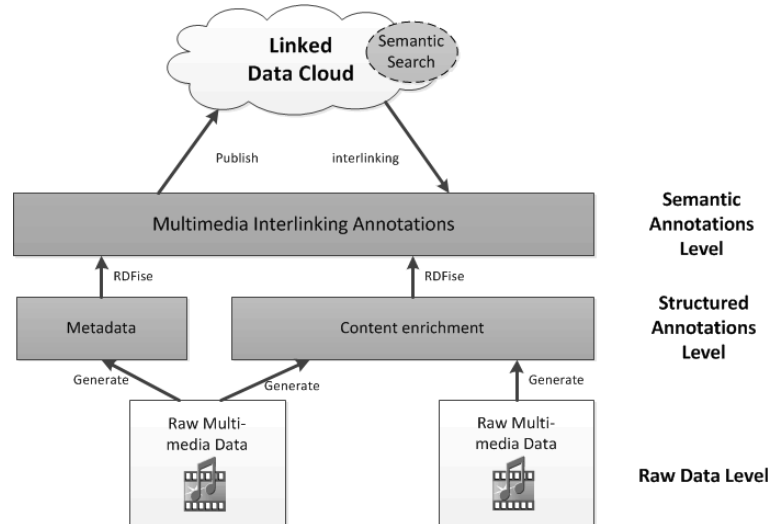


FIGURE 1.1: The gap between raw multimedia data and interlinked multimedia annotations

Figure 1.1 shows the gap between multimedia data and the interlinked multimedia annotations in the Linked Open Data (LOD) Cloud. On the semantic annotations level, all the annotations are represented as machine-readable format. Semantic multimedia annotations differ from traditional data as they require explicit identification for both the parent media resource and for media fragments. Troncy et al. (2012) stated that “enabling the addressing of media fragments ultimately creates a means to attach annotations to media fragments. The annotations are then ready to be further published and integrated with other datasets in the linked data cloud. So two steps are needed to fill the gap: (1) the extraction of structured data from raw multimedia resources and annotations; (2) interlinking them to the LOD Cloud. Many researchers have applied semantic Web technologies to multimedia resources, but most of them worked at the whole multimedia level or did not focus on the multimedia resources shared on the Web. Applying Linked Data principles to media fragments and annotations on the Web is a relatively new topic.

This Chapter will now detail the research questions. Section 1.2 will briefly explain the contributions of the thesis including the related publications resulting from this research. Section 1.3 details the structure of the thesis.

1.1 Research Questions

Generally, this thesis will examine in depth the current situation of multimedia delivery and sharing on the Web, and determine the best practices to apply Linked Data to media fragments, so that media fragments can be better interlinked, visualised and indexed. This thesis will thus address the following research questions.

1. Q1: What are the fundamental requirements and minimum best practice that need to be followed when applying Linked Data principles to media fragments for the current multimedia sharing platforms, so that media fragments can be seamlessly integrated with the Web of Data?
2. Q2: How can media fragments benefit end users on the Web, which include:
 - (a) Q2.1: How to automatically link media fragments and annotations to the LOD Cloud and publish them as structured Linked Data so that semantic-aware agents can index and merge media fragments into the global Web of Data?
 - (b) Q2.2: How to visualise media fragments with semantic annotations?
 - (c) Q2.3: How to improve the presence the media fragments for the major search engines, which are based on crawling through HTML documents? How can semantic markups (see Section 2.2.4) be applied to Web pages (or landing pages) that represent media fragments and annotations?
 - (d) Q2.4: Is it possible to apply structured media fragment annotations to video classification tasks?

Among these research questions, applying Linked Data principles to media fragments (Q1) is the fundamental problem, the solution to which will be mainly based on the analysis of the rationale of Linked Data principles and the nature of media fragments. Answering Q1 will give general guidelines for identifying media fragments with URIs, making the URIs dereferenceable, and describing media fragments using RDF.

The research questions in Q2 are based on Q1. Improving the presence of media fragments on the Web consists of three goals. Q2.1 focuses on publishing structured media fragments and annotations, so that they are exposed to semantic-aware agents, such as semantic indexing services, and can be dereferenced and queried. The media fragments linking can be carried out using several approaches, but this thesis will explore the fully automatic line of attack within the big picture that many videos and their metadata have already been shared online. So the main problem of Q2.1 is to propose a model to describe media fragments, and a framework to automate the media fragment interlinking process. Q2.2 aims at improving the presence of media fragments for end users, i.e. visualising media fragments together with annotations in a sensible way for end users.

Q2.3 is trying to integrate the media fragments with the major search engines, so that the semantic descriptions of media fragments can be crawled and processed by them. As the structured media fragment annotations are not widely available for videos currently online, Q2.4 is actually an attempt to explore whether such data can be used as features to carry out video classification.

1.2 Contributions of the Thesis

Following the research questions detailed in Section 1.1, the main contributions of this thesis are listed below as the answer to each research question.

1. Identify the fundamental requirements of applying Linked Data principles to media fragments by analysing the real world use cases (Chapter 3).
2. **Interlinking Media Fragments Principles** (Chapter 4): the principles specify best practice when considering publishing media fragments as Linked Data.
3. **Core Model of Media Fragment Enrichment** (Chapter 5): the core model reuses well-known vocabularies to describe media fragments and setup connections between media fragments and concepts in the LOD Cloud.
4. **Media Fragment Enriching Framework** (Chapter 5): the framework uses named entity extractors to automate the linking of media fragments with the LOD Cloud for existing video/audio resources on the Web, so that they are indexable by semantic-aware agents.
5. **Synote Media Fragment Player** and **Media Fragment Enricher UI** (Chapter 6): these highlight media fragments with the semantic annotations published by the Media Fragment Enriching Framework.
6. **Media Fragment Indexing Framework** and media fragment indexing using Twitter (Chapter 7): the indexing framework improves the online presence of media fragments by allowing search engines such as Google to index media fragments with minimum change to the multimedia host server. The idea of using Twitter as the source for media fragment annotations makes the indexing framework scalable on the Web.
7. **Applying media fragments and semantic annotations in video classification** (Section 5.4): structured media fragment annotations generated from named entity recognition techniques are introduced as new features for video classification.

Interlinking Media Fragments Principles extend Linked Data principles for media fragments on the Web. The **The Core Model of Media Fragments** specifies what

information needs to be included in the RDF descriptions of media fragments, while **Media Fragment Enriching Framework** designs automatic means to construct instances of **Core Model of Media Fragments** and publish them as Linked Data. Those three contributions offer guidelines, best practice and starting points for developers (either to upgrade existing multimedia platforms or to start new applications) who want to publishing media fragments as Linked Data.

Synote Media Fragment Player, **Media Fragment Enricher UI**, **Media Fragment Indexing Framework** and the video classification tasks further extend the first three contributions to media fragments. All of them enhance and expand the media fragment presence and usage on the Web.

The following papers have been published in conferences and journals as the direct research results of this research. The author of this thesis is the primary author of those papers.

- Li, Y, Troncy, R, Wald, M and Wills, G (2014) Media fragment indexing using social media. In *2nd International Workshop on Linked Media (LiME2014)*, Crete, Greece.
- Li, Y, Rizzo, G, Garcia, J L R, Troncy, R, Wald, M and Wills, G (2013) Enriching media fragments with named entities for video classification. In *First Worldwide Web Workshop on Linked Media (LiME-2013)*, Rio de Janeiro, Brazil, 13-17 May 2013.
- Li, Y, Rizzo, G, Troncy, R, Wald, M and Wills, G (2012) Creating enriched YouTube media fragments With NERD using timed-text. In *11th International Semantic Web Conference (ISWC2012)*, Boston, USA, 11-15 Nov 2012.
- Li, Y, Wald, M, Omitola, T, Shadbolt, N and Wills, G (2012) Synote: weaving media fragments and linked data. In *Linked Data on the Web (LDOW2012)*, Lyon, France, 16 Apr 2012.
- Li, Y, Wald, M and Wills, G (2012) Applying linked data in multimedia annotations. *International Journal of Semantic Computing*, 6, (3), 289-313.
- Li, Y, Wald, M and Wills, G (2012) Let Google index your media fragments. In *WWW2012 Developer Track*, Lyon, France, 16-20 Apr 2012.
- Li, Y, Wald, M and Wills, G (2011) Applying Linked Data to Media Fragments and Annotations. In *ISWC 2011: 10th International Semantic Web Conference*, Bonn, Germany.
- Li, Y, Wald, M and Wills, G (2011) Interlinking Multimedia Annotations. In *Web Science 2011*, Koblenz, Germany, 14-18 Jun 2011.

- Li, Y, Wald, M, Wills, G, Khoja, S, Millard, D, Kajaba, J, Singh, P and Gilbert, L (2011) Synote: development of a Web-based tool for synchronized annotations. *New Review of Hypermedia and Multimedia*, 17(3), 1-18.

The author also wrote or co-authored some other papers, the content of which include multimedia, semantic Web and Open Data, but they are not directly related to the topic of this thesis:

- Wald, M, Li, Y, Cockshull, G, Hulme, D, Moore, D, Purdy-Say, A and Robinson, J (2014) Synote second screening: using mobile devices for video annotation and control. In *Home 14th International Conference on Computers Helping People with Special Needs (ICCHP 2014)*, Paris, France, 09-11 Jul 2014.
- Wald, M, Li, Y, Draffan, E A and Jing, W (2013) Synote mobile HTML5 responsive design video annotation application. *UACEE International Journal of Advances in Computer Science and its Applications (IJCSIA)*, 3(2), 207-211.
- Wald, M, Li, Y, Draffan, E A and Wei, J (2013) HTML5 video on mobile browsers. *International Conference on Information Technology and Computer Science (IJITCS)*, 9(3), 61-67.
- Wald, Mike, Li, Yunjia, Draffan, E.A. and Wei, Jing (2013) HTML5 video on mobile browsers. *International Conference on Information Technology and computer Science (IJITCS)*, 9, (3), 61-67.
- Li, Y, Draffan, E A, Glaser, H, Millard, I, Newman, R, Wald, M, Wills, G and White, M (2012) RailGB: using open accessibility data to help people with disabilities. In *International Semantic Web Conference 2012*, Boston, USA, 11-15 Nov 2012.
- Wald, M and Li, Y (2012) Synote: Important Enhancements to Learning with Recorded Lectures. *ICALT 2012*.

The survey conducted in Section 7.2 is partially published as a blog post “Are the Videos We are Watching Online Media Fragment Ready?”⁸. The author also developed several open source software programmes as the research results that are freely available online:

- Synote⁹: Synote implements the Media Fragment Indexing Framework in Chapter 7. It is also underpins the implementation of Media Fragment Enriching Framework(Chapter 5).

⁸<http://goo.gl/qJse87>

⁹<https://github.com/yunjiali/Synote>

- Synote Media Fragment Player¹⁰: A client-side Media Fragment Player, which is part of the Media Fragment Enriching UI (Section 6.1), but it could be used stand-alone.
- Media Fragment URI Loose¹¹: A fork of Thomas Steiner’s Media Fragment URI Parser¹². It adds more support for parsing the media fragment string defined in different video sharing platforms, such as YouTube and Dailymotion.
- Media Fragment Enricher¹³: Including all the implementations in Chapter 6. The author of this thesis is the main developer of this software.
- Nerd4node¹⁴: A node.js client for NERD. The author of this thesis worked with Giuseppe Rizzo to develop this software.

The author of the thesis also joined some competitions and received awards as the result of this research work:

- Winner of ICT Initiative of the Year in prestigious THE Award (Nov, 2011)
- Participant of Semantic Web Challenge in ISWC2012 (Oct, 2012)

1.3 Structure of the Thesis

The rest of this thesis will be presented in the following seven chapters:

Chapter 2 covers the background of media annotations, Linked Data and media fragments. The history of multimedia annotations and legacy multimedia annotation applications will be explored. Many standards and projects closely related to the research questions will be also be reviewed. As a specific topic, the literature about video classification will be presented in the last section. Video classification is an extensive topic, so the review will focus only on the algorithms using higher-level annotations and semantic-related features. The gap addressed by this thesis will be explained through an analysis of the limitations of the previous work.

Use cases are very important at each stage of this research. In Chapter 3, several use case studies are conducted to reveal how applications in different domains could benefit from the exposure of media fragments together with the semantic Web. Some of the case studies also involve technical analysis by applying linked data to existing multimedia

¹⁰<https://github.com/yunjiali/Media-Fragment-Player>

¹¹<https://github.com/yunjiali/Media-Fragments-URI-Loose>

¹²<https://github.com/tomayac/Media-Fragments-URI>

¹³<https://svn.eurecom.fr/mediafragment>, hosted in Eurecom, need login

¹⁴<https://github.com/giusepperizzo/nerd4node>

applications. Some of them are domain-specific problems that can be tackled by publishing media fragment annotations. Following the use case studies, Chapter 3 presents in-depth discussion of the requirements derived from case studies. Seven requirements are identified, and they will form the outline of the rest of the thesis, i.e. each of the remaining chapters will resolve some of the requirements listed in Chapter 3.

Thus, Chapters 4, 5 and 6 will present the solutions of applying linked data principles to media fragments. Chapter 4 discusses the fundamental best practice, named Interlinking Media Fragments Principles, which will be universally applicable to applications that want to publish media fragments into the LOD Cloud. Based on the four linked data principles, the discussion will consider applying each principle to media fragments and how the relevant requirements can be fulfilled. This chapter will also summarise the possible interlinking methodologies for media fragments.

The Interlinking Media Fragments Principles discussed in Chapter 4 focuses on the higher level rationale of the principles, rather than their actual implementation details. Chapter 5 will take a step forward by designing a Core Model of Media Fragment Enrichment, which reuses existing vocabularies to model the interlinking media fragments. The Media Fragments Enriching Framework will also be described to automate the media fragment interlinking process, based on named entity extractions. The core model and the framework introduced in this chapter are general design solutions to implement the Interlinking Media Fragments Principles, and could be extended or modified if the principles were applied to some domain-specific areas.

To demonstrate the real use case of the Core Model of Media Fragment Enrichment and Media Fragment Enriching Framework, Chapter 5 will then introduce two examples of using the automatically interlinked media fragments. Section 5.3 uses the framework to enrich YouTube videos and publish the annotations as Linked Data. Section 5.4 bootstraps a new research area of applying the media fragments and named entities generated from the Media Fragment Enriching Framework into online video classification tasks. In this example, a number of named entities extracted from video subtitles, their types and their appearance in the timeline are exploited as features for classifying online videos into different categories. Different machine learning algorithms are applied to the classification tasks and the results will reveal some interesting findings.

Chapter 6 will focus on the visualisation of the media fragments. To integrate the media fragments with the current HTML5 player, the Synote Media Fragment Player was designed to highlight both temporal and spatial dimensions of media fragments. After that, an implementation of the Media Fragment Enrichment UI is presented as part of the Media Fragment Enriching Framework, which displays media fragments with annotations in a synchronised manner.

Chapter 7 will talk about the problems and solutions to media fragment indexing in “traditional Web search engines. By looking at current use and sharing of multimedia

resources online, the conclusion is drawn that the major obstacle for search engines to index media fragments is that there is no individual HTML page for media fragments and their related annotations, so the search engines will not be able to distinguish the “media fragment page from the video landing page. Based on this finding, Chapter 7 describes the Media Fragment Indexing Framework, which creates a snapshot page for each media fragment and uses the architecture of Google’s Ajax Crawler to make them indexable. To scale up this method and acquire more media fragment annotations, Sections 7.2 and 7.3 propose the collection and filtering of the media fragment data from social media, such as Twitter, as the input data to the Media Fragment Indexing Framework. A demonstration has been built to show this and the initial evaluation results reveal that media fragments can be successfully indexed by Google using this method.

Conclusions are given in Chapter 8 together with a list of contributions of this research. The conclusion mainly reflects what has been learned during this research and the major achievements of providing the solutions. There are several areas that the media fragment semantics can apply to, where future research will be needed to clarify the problems and detail the solutions.

To make the relationships between different chapters clear, Figure 1.2 demonstrates the main contributions in each chapter and their relationships.

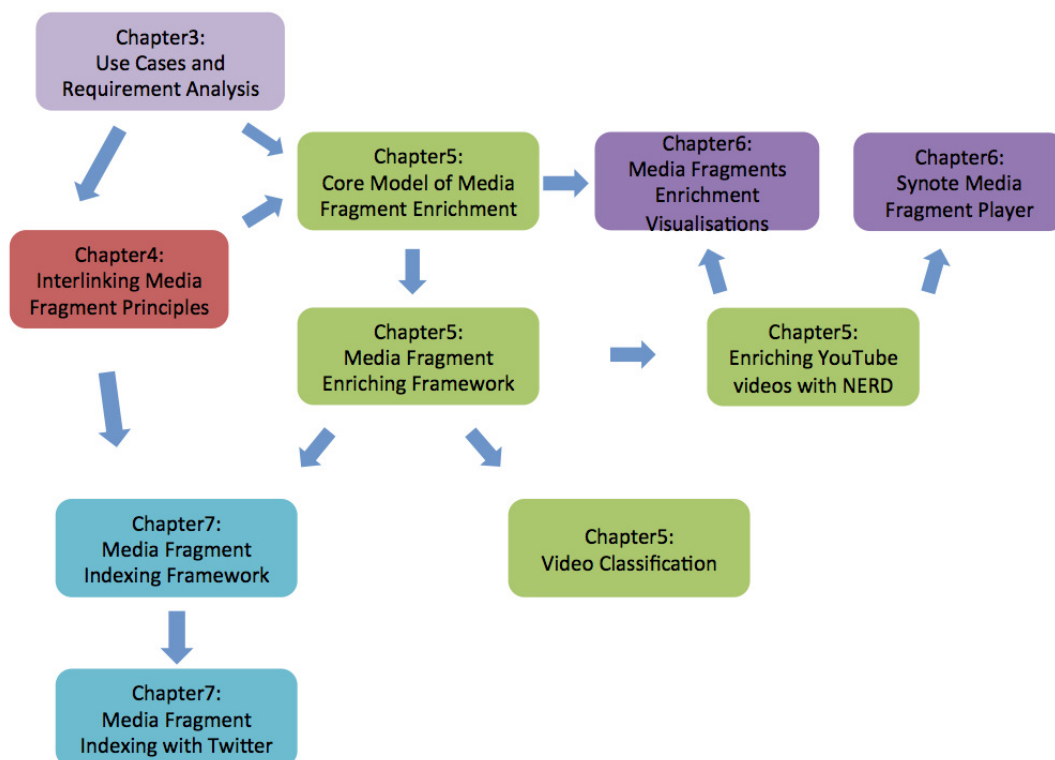


FIGURE 1.2: Relationship Between Different Chapters in the Thesis

1.4 Acknowledgements

The research work conducted in Chapter 5 was mainly designed and implemented by the author of the thesis. However, some implementations and evaluations were carried out as joint work with Giuseppe Rizzo, García, José Luis Redondo and Raphaël Troncy in Eurecom, France and they are partially supported by the European Union's 7th Framework Programme via the project LinkedTV (GA 287911). The research results of those Sections and Chapters were first published in Poster and Demo Session at the International Semantic Web Conference 2012, Boston, USA (Li et al., 2012), and the First international workshop on Linked Media (LiME2013) on World Wide Web Conference 2013, Rio, Brazil (Li et al., 2013). The author of this thesis is the primary author of both papers.

Chapter 2

Literature Review

This chapter examines the background to the research questions put forward in Section 1.1, and covers the history of multimedia annotations, as well as the concepts, standards and the state of art technologies of media annotations, Linked Data and media fragments.

On one hand, a lot of research work has addressed the concept of media fragments and associate them with automatic or user-generated annotations before the introduction of the Semantic Web and Linked Data. This work needs to be revisited, so that their relationships to the current Web of data can be clarified. On the other hand, even though Linked Data has been widely applied in many areas, media fragments and annotations have rarely been published into the Linked Data Cloud, especially for the existing major multimedia sharing platforms. Generally, there is a gap between media fragments and Linked Data, which are addressed by research questions Q1 and Q2. This chapter will show that, while semantic technology has been increasingly applied to search engine optimisation, it is still a blank area on how the Linked Data approach could optimise the indexing of media fragments on large scale, which is also a research question of this thesis. The remainder of this chapter is arranged as follows.

Sections 2.1 and 2.2 will introduce multimedia on the Web and semantic Web technologies separately. Section 2.1 will go through legacy multimedia annotation applications, including an Open Hypermedia system named Synote, which a lot of research work in this thesis has extended, while Section 2.2 will focus on the current state of the semantic Web and Linked Data technologies. The literature mentioned in those sections are general background that will be used in this thesis. Next, the literature about the combination of multimedia, media fragments and Linked Data is discussed, including an overview of the current standards on media fragments and multimedia annotations (Section 2.3), followed by related media fragments and Linked Data implementations (Section 2.4). Section 2.5 reviews some machine learning algorithms applied to video classification as background for Section 5.4.

2.1 Multimedia Annotations and Synote

Annotation for multimedia is a means of marking up the objects and scenes in audio-video streams in order to facilitate the interpretation and understanding of its content. The idea of annotating a multimedia resource and media fragments was well developed even before the Web era. Work on annotations in the hypertext and hypermedia world dates back to the development of Memex by Vannevar Bush (Bush, 1945). Englebart in the 1960s developed NLS/Augment (Engelbart, 1963), which provided its users with a collaborative, multi-window system with video conferencing, where every medium could be linked. When Hypertext and Hypermedia were introduced, many systems were geared toward the initial design of creating hypertext enabled content, such as Microcosm (Davis et al., 1992) and CoNoter (Davis and Huttenlocher, 1995).

Multimedia annotations are also essential parts of some early standards as a means of metadata association and content enrichment. Metadata association methods use specific metadata models to build a structure to support features such as content search. Examples of such systems are EXIF¹, ID3² and MPEG-7 (Martinez, 2002).

Instead of making the annotations for the whole multimedia resource, some systems were designed to deal with temporal behaviour in Hypermedia. The Videobook system (Ogawa et al., 1990) combines temporal media composition with linking. Users can navigate through the links embedded in the construction of composite multimedia nodes. The Firefly Document System (Buchanan and Zellweger, 1992) allows authors to specify and manipulate temporal relationships within media segments at a high level rather than as timings.

These multimedia annotation systems usually limit themselves to the boundary of specific applications; thus the annotations are restricted to a group of people who can make the annotations and share them outside the applications. With more and more multimedia resources being made available on the Web, and the improved ability to gather annotations from users through collaboration in the Web 2.0 era, the content enrichment function of annotations start to gain more focus and many Web-based applications emerge as collaborative annotating tools for documents on the Web. The Mimicry system (Bouvin and Schade, 1999) is one such effort that allows authors and readers to link to and from temporal media (video and audio) on the Web. The system uses Open Hypermedia integration to provide multi-headed linking facilities. Annotea (Kahan et al., 2002) is another example of a hypermedia system to create annotations in web documents. It uses the RDF model to define Web annotations. The relations defined in the model are “body and “related, but additional user-defined relation types can be added. One of the examples of Annotea implementation is Vannotea (Schroeter et al., 2006).

¹<http://www.exif.org/Exif2-2.PDF>

²<http://www.id3.org/id3v2.4.0-structure>

Before the explosion of online video sharing platforms such as YouTube, the displaying of multimedia resources on the Web, especially videos, relied on Java Applets or other plugins for the Web browsers, such as Windows Media Player³, Quicktime⁴ and Acrobat Flash⁵. The annotations, especially user-generated annotations on the Web, were very limited and difficult to visualise together with the videos. With the advent of Web 2.0, annotations have been modified in the form of comments, tags, bookmarks and folksonomies, and it has become extremely easy for users to act as multimedia contributors to the Web. Web applications such as Overstream⁶, Tegrity⁷ and Ponopto⁸ enable users to enter text synchronised with media on the Web.

Synote is a Web-based Open Hypermedia annotation system, which is trying to address the problem that a given fragment of audio or video is difficult to link, index and search (Li et al., 2011a). The system introduces temporal synchronization points in its conceptual model and highlights the transcript, notes and slides, which annotate a certain audio or video fragment, along with replaying of the audio or video. Synote also provides a search facility, which allows users to search for synchronised transcript, tags and notes. Selecting a search result will lead users directly to that time point in the audio or video.

Figure 2.1 shows the backend object model of Synote. Annotation and Resource are the main entities in the object model. An instance of annotation has a “source resource”, which is a video/audio, and “target resource”, which are text based resources such as tags and transcript, or an array of resources, such as an array of images as slices. The Synpoint (abbreviation of Synchronisation Point) saves the anchors from both source and target resources to indicate which parts of the two resources are related respectively. Usually, the anchor for the “source resource” is the start and end time of the video/audio, and the anchor for “target resource” is the character count or the index of the array.

Those applications take the advantage of Web and enhance the sharing and distribution of annotations. Some of them even use annotations to index part of the multimedia resources within the applications. However, they are still not ready for the vision of the “Web of data” considering the emerging of semantic Web and Linked Data. The data structure of multimedia and annotations is usually application-dependent and the annotations are rarely published and reused in machine-readable format. As a result, those datasets are isolated from each other and hardly discoverable on Web, which makes it more difficult to index part of the multimedia resources on large scale.

Synote is a typical Web 2.0 application and in this research work. Chapter 3 presents it as an example to demonstrate how the gap between Web 2.0 and Web 3.0 for multime-

³<http://windows.microsoft.com/en-GB/windows/windows-media-player>

⁴<http://www.apple.com/uk/quicktime/download/>

⁵<http://get.adobe.com/flashplayer/>

⁶<http://overstream.net>

⁷<http://www.tegrity.com>

⁸<http://panopto.com>

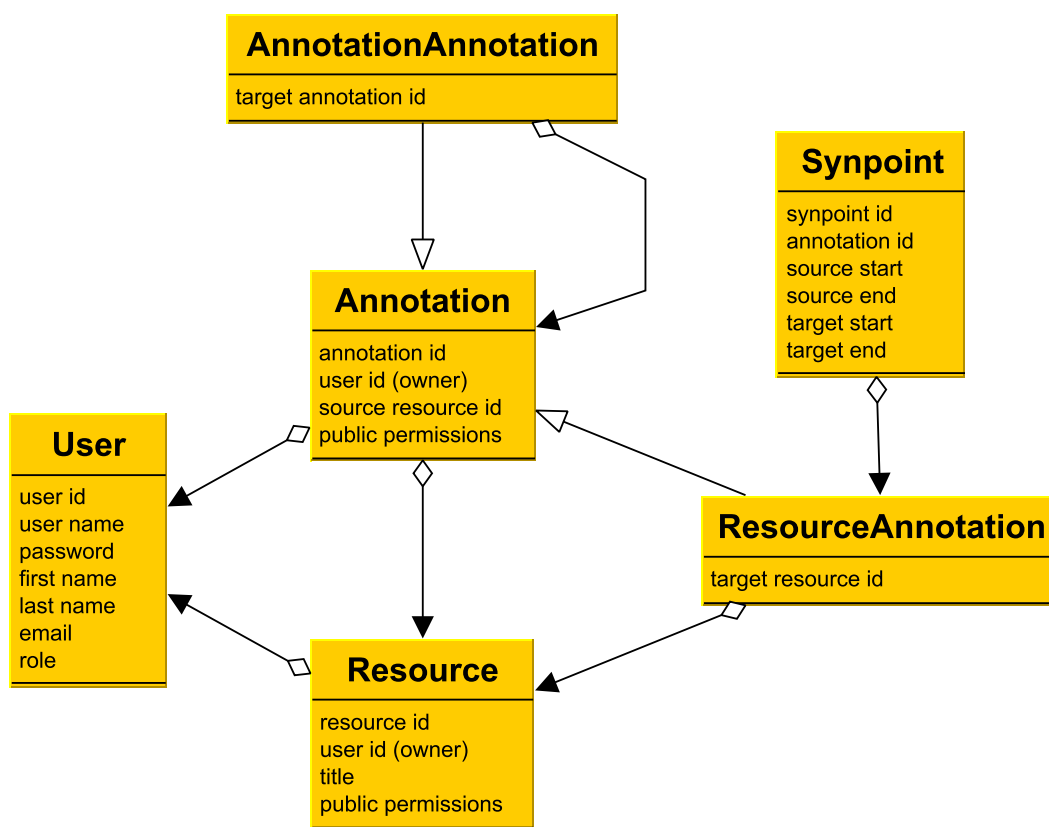


FIGURE 2.1: Synote Object Model

dia and media fragments applications can be bridged using the research results in this thesis, i.e. how the media fragment annotations in Synote can be published as Linked Data. Furthermore, the design of the Synote Object Model is similar to the idea of Open Annotation Model(Haslhofer et al., 2012) and inspires the media fragment annotation model designed in Chapter 5. In Chapter 6, the implementation of Synote Media Fragment Player is also inspired by the user interface of Synote.

2.2 Semantic Web and Linked Data

After more than 20 years of growing, the World Wide Web has gradually stepped into the Semantic Web or Web 3.0 era (Hendler, 2009). The vision of a Semantic Web was first expressed in 2001 (Berners-Lee et al., 2001) as exploring machine-readable data on the Web instead of HTML documents. Semantic Web technologies, recommended by the W3C, are trying to enrich the unstructured or semi-structured data on the Web with semantics and build connections between them in order to achieve better search and query.

2.2.1 Linked Data Technologies and Principles

The Linked Open Data (LOD) initiative⁹, which represents the first large scale collaborative effort to implement a Web of Data, describes a series of methods for publishing structured data using semantic Web technologies and other related standards¹⁰. The basis of Linked Data resides in the Resource Description Framework (RDF) (Manola and Miller, 2004), where both informational and non-informational resources are identified by uniform resource identifiers (URIs) (Berners-Lee et al., 2005) and the relationships between data are represented in triples of subject, predicate and object. The semantic Web requires vocabularies being defined when creating RDF data. Such vocabularies in the semantic Web field are called ontologies. Ontology, from the field of artificial intelligence, is defined as a specification of a conceptualization (Gruber, 1993). It is a logical theory accounting for the intended meaning of a formal vocabulary (Guarino, 1998). In a semantic Web, vocabularies defined with RDF Schema (RDFs) (Brickley and Guha, 2004) and the Web Ontology Language (OWL) (Smith et al., 2004) can be used to describe relationships among data within the RDF. Once the data are available in the form of RDF, the SPARQL query language (Prud'hommeaux and Seaborne, 2008) allows users to create SQL-like queries to query RDF datasets. The vision of linked datasets can be considered as a large knowledge-base, where anyone can make contributions by publishing machine-readable data into public Web space. The RDF datasets can then also be indexed and merged by an RDF crawler, such as Sindice¹¹, through shared URIs and shared vocabularies. In this way, Linked Data enables machines to automatically discover more data from the data they already know.

To achieve the final vision of Linked Data, four rules must be followed when publishing Linked Data on the Web (Berners-Lee, 2006):

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information using the standards such as RDF and SPARQL
4. Include links to other URIs, so that they can discover more things

These four rules, instead of focusing on the ontology level, simplify the data publishing process.

While the Linked Data technology is becoming mature, its principles have never changed over the years. The initial idea of Linked Data did not target a specific type of data, but

⁹<http://esw.w3.org/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

¹⁰<http://linkeddata-specs.info/>

¹¹<http://sindice.com>

as the representations and usage of multimedia content is obviously different from plain text and numeric data, the application of Linked Data principles need to be extended to this specific case. This problem will be addressed in Chapter 3

2.2.2 Linked Datasets

Many existing data repositories on the Web have been re-structured and published as Linked Data. By September 2011, nearly 300 major datasets had been published with around 31 billion triples¹². As can be seen from the linking open data cloud diagram (Figure 2.2), a network of knowledge interlinking covers topics from media, publications, government and life science, etc. DBpedia¹³ is one of the most famous datasets, which have been acting as a central hub of Linked Data. DBpedia extracts structured information from Wikipedia and make it available as Linked Data for reuse and query. The DBpedia knowledge-base describes more than 3.64 million things (in DBpedia v3.7) ranging from persons and places to concepts in different scientific areas such as biology and chemistry. Other important datasets include data.gov.uk¹⁴, Linked GeoData¹⁵, Music Brainz (DBtune), etc. RKBExplorer, developed by the University of Southampton, harvests raw data from many research institutions and repositories, such as ACM, IEEE, EPSRC, etc, and publishes them in RDF format (Glaser et al., 2008). The integrated datasets are then visualised by RKBExplorer to reveal the relationships between different publications and projects.

The eGovernment initiative (Shadbolt et al., 2011) is trying to use semantic Web technologies to add significant value to government related data. The recent efforts by the UK government, data.gov.uk, is a model for adopting Linked Data as recommended practice to improve transparency of local government data and the interaction between citizens and government in the UK, as well as around the world (Shadbolt and Hall, 2010). The datasets already published in the UK's Public Sector Information (PSI) include: government expenses, crime rate, parliament information. Most of the datasets in data.gov.uk use geographic data as the key to tie other datasets together and a single point of access is established for all public UK datasets.

Some government data, such as energy consumption, have been published within the EnAKTing project¹⁶, which is trying to address fundamental problems in “achieving an effective web of Linked Data”. This project concentrates on four key research challenges: build ontologies quickly in a large-scale manner which are flexible and adaptive (Shadbolt et al., 2006) effectively query the unbounded Web of data; enable browsing, navigation through distributed datasets; build deliverable real world applications for people to use

¹²<http://www4.wiwiiss.fu-berlin.de/lodcloud/state/>

¹³<http://dbpedia.org>

¹⁴<http://www.data.gov.uk>

¹⁵<http://linkedgeo.org>

¹⁶<http://enakting.org>

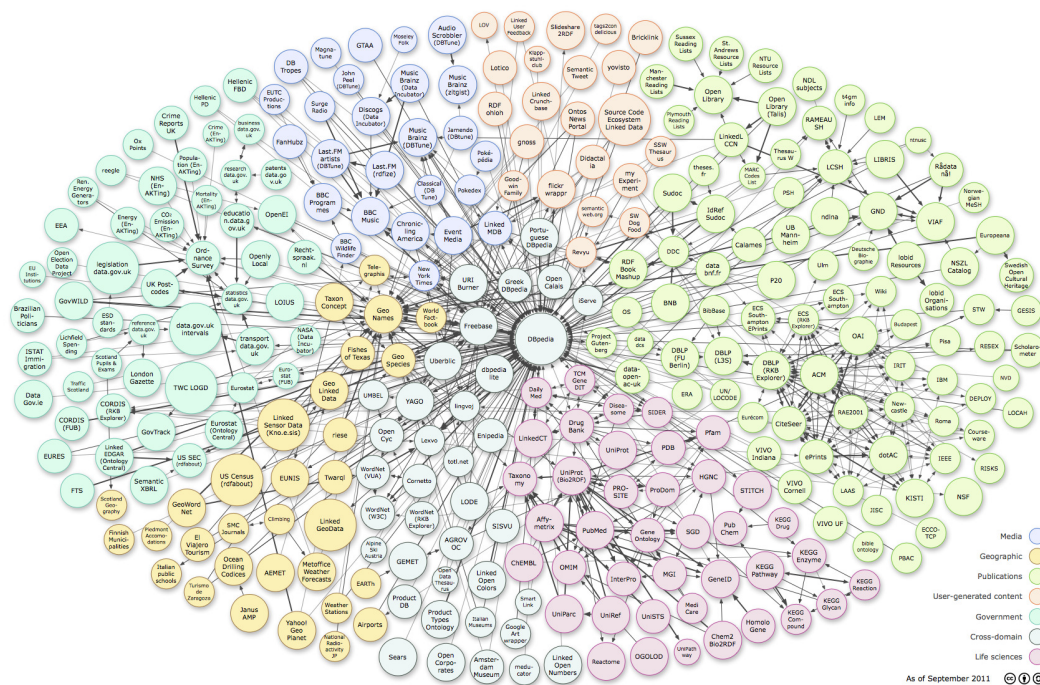


FIGURE 2.2: Linked Data Cloud at September 2011

the datasets on the Web. EnAKTing has developed quite a few services on top of different datasets, such as backlinking service, SameAs service¹⁷, GeoService, temporal reasoning service, etc.

There have been many interesting results since the PSI was published as Linked Data. Analysing the final data from different public sectors, David McCandless sorts out the “cost for average British taxpayer per day” categorised by essentials, travel, banking charges, going out, etc (Shadbolt and Hall, 2010). UK Dentists¹⁸ is an iPhone application which uses the location information of NHS dentists in data.gov.uk and iPhones built-in GPS to help users find the nearest NHS dentist quickly. Police.uk¹⁹ is another application which uses the crime rate data in data.gov.uk. Given a postcode or address, Police.uk can provide users the crime and policing information in that area.

Except for the datasets published by PSI, the Linked Data has been made available in many other domains. Some higher education institutions have already published their university information as Linked Data. The Open University published its research, podcast and course information in the Linking University Content for Education and Research Online (LUCERO) project²⁰. University of Southampton Open Data²¹ have published a wide range of information related to courses, daily teaching, school services, organisations, transportation, etc. For both institutions, the published datasets can be

¹⁷<http://sameAs.org>

¹⁸<http://data.gov.uk/apps/uk-dentists>

¹⁹<http://www.police.uk>

²⁰<http://data.open.ac.uk>

²¹<http://data.southampton.ac.uk>

freely accessed through SPARQL endpoints or downloaded as RDF files. The Britains national map agency, Ordnance Survey²², also published their 1:50 Scale Gazetteer, Code-Point Open and the administrative geography gazetteer for Great Britain as Linked Data. Meanwhile, Ordnance Survey hosts a SPARQL endpoint²³ to query and reuse their datasets.

While the number of datasets and the size of the triples published on the Web is growing rapidly, it is clearly noticeable that multimedia and user-generated content (such as multimedia annotations) have hardly been published into the LOD considering that Internet video traffic was 57% of all consumer traffic in 2012. No major datasets have claimed to publish media fragments and annotations as Linked Data. This leads to the need to research the potential demand and the technical barriers to publishing media fragment annotations into LOD.

2.2.3 Named Entity Recognition and Disambiguation

Natural language processing (NLP) techniques have been an essential component in the area of Information Extraction (IE). At the early stage, those techniques focus on atomic identification of word or phrases, i.e. named entities. Later on, the focus moves to the classification of named entities into predefined categories by using various classification techniques. These named entities are further linked to real world objects using web identifiers, called Named Entity Disambiguation. The final disambiguation of the named entities is linked to a particular knowledge base, where the identifiers are predefined, so the choice of knowledge bases usually exerts a great influence on the disambiguation task.

With the development of the semantic Web, many semantic knowledge bases have emerged that contain a large number of concepts that correspond to real-world entities. Except for the named entities, these knowledge bases also developed exhaustive classification schemes, according to which the named entities are classified. Some widely recognised semantic knowledge bases are YAGO (Suchanek et al., 2007), DBpedia (Auer et al., 2007) and Freebase (Bollacker et al., 2008). To perform the named entity recognition and disambiguation tasks, many tools, such as DBpedia Spotlight (Mendes et al., 2011), OpenCalais²⁴ and Zenmanta²⁵, have been developed to extract named entities from plain text, categorise them according to the taxonomies defined in knowledge bases, and disambiguate them using URIs. Even though each of these tools use different training data and algorithms to provide named entity extraction services, they all provide similar output with a set of named entities.

²²<http://ordnancesurvey.co.uk>

²³<http://data.ordnancesurvey.co.uk/datasets/os-linked-data/explorer/sparql>

²⁴<http://www.opencalais.com/>

²⁵<http://www.zemanta.com/api/>

Most of the named entity extractors have their output data models exposed via Web APIs. However, those data models vary from extractor to extractor, which limits the interoperability among the extracts. To solve this problem, NLP Interchange Format (NIF) (Hellmann et al., 2012) has been proposed as an interoperable model for the information exchange between NLP tools. NIF includes a core ontology of classes and properties to describe the relations commonly used in NLP tools, such as strings, text, documents, etc. It also defines the URI patterns used to refer to a named entity in a document. This has been reused here in the implementation of the Media Fragment Enrichment Framework in Section 5.2.4.

NIF solves the interoperability problem among NLP tools, but the output of those tools is not harmonised as they all have their own taxonomy to model the types of named entities. So Rizzo and Troncy (2012) have proposed a Named Entity Recognition and Disambiguation (NERD) framework to unify the types of these extractors. NERD framework also developed a NERD ontology, which is a set of mappings established manually between the taxonomies developed by different named entity extractors. There are 9 top-level classes defined in the NERD ontology: *Thing*, *Amount*, *Event*, *Function*, *Location*, *Organization*, *Person*, *Product* and *Time*. *Thing* in NERD is used as the fallback option if NERD cannot find a specific type that a named entity belongs to. The NERD is used as the fall-back option if NERD cannot find a specific type that a named entity belongs to. The NERD framework not only makes it possible to analyse and compare quantitatively the advantages and disadvantages of each extractor on documents in different domains, but also accumulates the major extractors output into a single Web API, so that developers can access multiple extracting services easily. Much research work here, in Chapter 6 and Section 5.4, is also highly reliant on NERD to harmonise named entity extraction results from different tools.

Plain text, such as title, descriptions, tags, comments, etc., has been widely available in major video sharing platforms for the enrichment of multimedia resources. So it is now possible to obtain those text documents through Web APIs and use different named entity extractors to apply semantic structures to them. In addition, text documents like SRT and WebVTT associate text blocks with the timeline of the multimedia resources, so it is easy to automatically align named entities extracted from that text block with the media fragments that the text block corresponds to. This is the basic rationale for the design of the core RDF model and the framework of automatic media fragment publishing.

2.2.4 Linked Data Publishing Patterns

Since much effort has been devoted to publishing and consuming various linked datasets, some summaries have been made as guidelines for Linked Data developers. Heath and Bizer (2011) have summarised the common Linked Data publishing steps that develop-

ers have to take from choosing URIs to the final testing and discovering Linked Data according to different situations. Heath and Bizer (2011) have put forward five major patterns in which Linked Data could be published. It has been pointed out that developers should not totally abandon “existing data management system and business applications”, but add an “extra technical layer of glue to connect these into the Web of Data”. Figure 2.3 shows the patterns of Linked Data publishing based on different data and storage types.

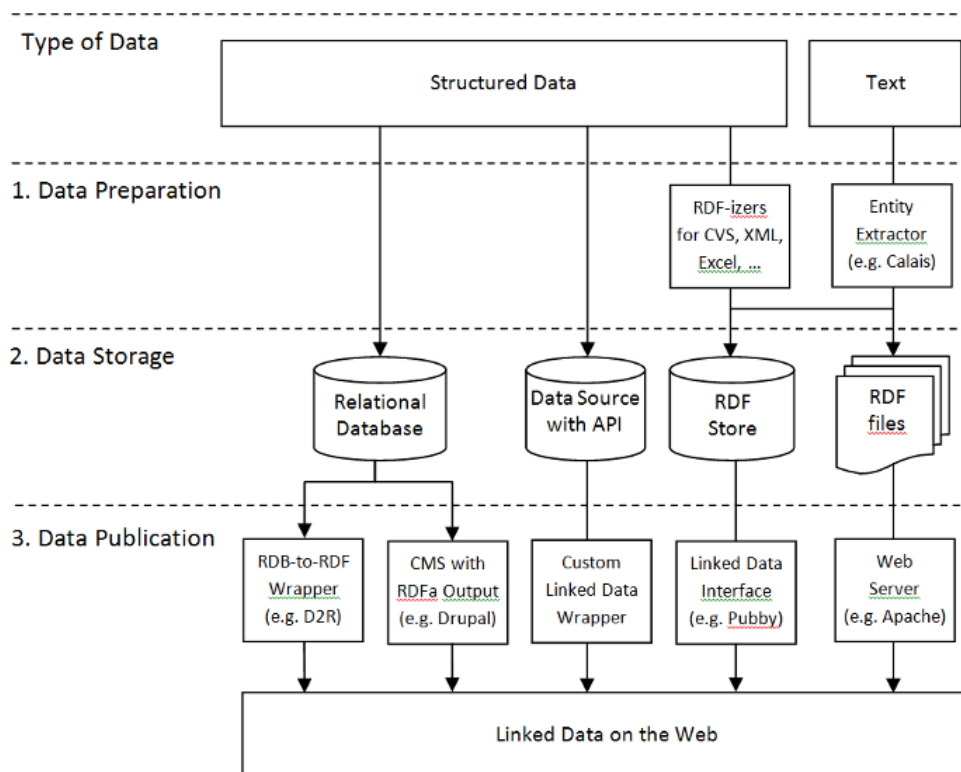


FIGURE 2.3: Linked Data Publishing Patterns (Heath and Bizer, 2011)

Serving static RDF documents sometimes is the most straight-forward way to publish Linked Data. Developers could create the RDF files manually or dump the data as static files from the data sources. Some famous datasets follow this pattern, such as DBpedia²⁶ and LinkedGeoData²⁷. RDF information could also be saved in a triple store, such as 4store (Harris et al., 2009), for easy management and access. The Web server, which serves RDF documents, can be totally independent of the legacy applications. However, a major problem in this case is that the RDF documents will not be up-to-date. This is not acceptable in situations where the data, such as the real-time temperatures in a day, must be published at runtime.

RDF could also be embedded in HTML files using the semantic markup or “Rich Snippets” technologies (Steiner et al., 2010). The Rich Snippets feature is currently built on

²⁶<http://dbpedia.org>

²⁷<http://linkedgeo.org>

open standards or community-agreed approaches, such as RDFa (Adida and Birbeck, 2008), Microformats²⁸ and HTML5 Microdata (Hickson, 2012). All of these technologies use semantic mark-ups, which can be embedded into current Web pages. Then traditional search engines can easily recognise the structured data embedded in the HTML documents and highlight that bit of information in the search results. Major search engines Google, Bing²⁹ and Yahoo!³⁰ have proposed vocabularies, schema.org, for publishing structured data using Microformats. The shared vocabularies have been maintained under schema.org³¹. For multimedia resources, schema.org has defined vocabulary to describe “ImageObject”, “AudioObject” and “VideoObject”.

Another common way of publishing Linked Data is serving RDF documents directly from the relational database, i.e. RDB-to-RDF. Many tools are designed for this purpose, such as D2R server (Bizer and Cyganiak, 2006), OpenLink Virtuoso (Erling and Mikhailov, 2007), Triplify (Auer et al., 2009) and Sparqlify (Ermilov et al., 2013). These tools often require developers to provide a mapping between database schema and the vocabularies. Usually, SPARQL endpoints and a view of the dataset will also be provided by the tools to access the RDF document.

Applications that expose their data through Web services or Web APIs, can write a wrapper to convert the API data to RDF. Examples of these wrappers are FlickrTM Wrapper³² and Slideshare2RDF server³³.

To ease the publishing process for government data, Tim Berners Lee proposed a 5-star process (Berners-Lee, 2006) to encourage data owners, especially government, share their data on the Web:

1. 1 star: make your stuff available on the Web (whatever format) under an open license
2. 2 stars: make it available as structured data (e.g., Excel instead of image scan of a table)
3. 3 stars: use non-proprietary formats (e.g., CSV instead of Excel)
4. 4 stars: use URIs to denote things, so that people can point at your stuff
5. 5 stars: link your data to other data to provide context

In the 5-star process, an open licence underpins everything that will be published, because it is important to guarantee that every piece of data is protected by law. From the

²⁸<http://microformats.org>

²⁹<http://bing.com>

³⁰<http://www.yahoo.com>

³¹<http://schema.org>

³²<http://www4.wiwiiss.fu-berlin.de/flickrwrapp/>

³³<http://linkeddata.few.vu.nl/slideshare/>

developers point of view, this process is trying to break down the task into achievable steps so as to lower the barrier of applying semantic Web to different areas. As can be seen from the description of each step, the raw data available from government are tables, scanned images, CSV files, MS Excel files, etc., in which the main form of data are statistics (revenue, crime rate, etc.) and strings (post code, country names, street names, etc.). From 3 to 4 star and 5 star, the RDF designer usually needs domain specific knowledge to design an appropriate ontology and link current data to other datasets.

It can be seen that the choice of a certain pattern depends on the type of raw data that is going to be published and the ease of technical implementation. To publish media fragments and annotations, likewise the raw data first needs to be extracted from the multimedia and delivered as RDF, depending on the nature of the raw data. However, the current patterns have not yet specified this process for multimedia data, which this research will address.

2.2.5 The Infrastructure of Linked Data-Driven Applications

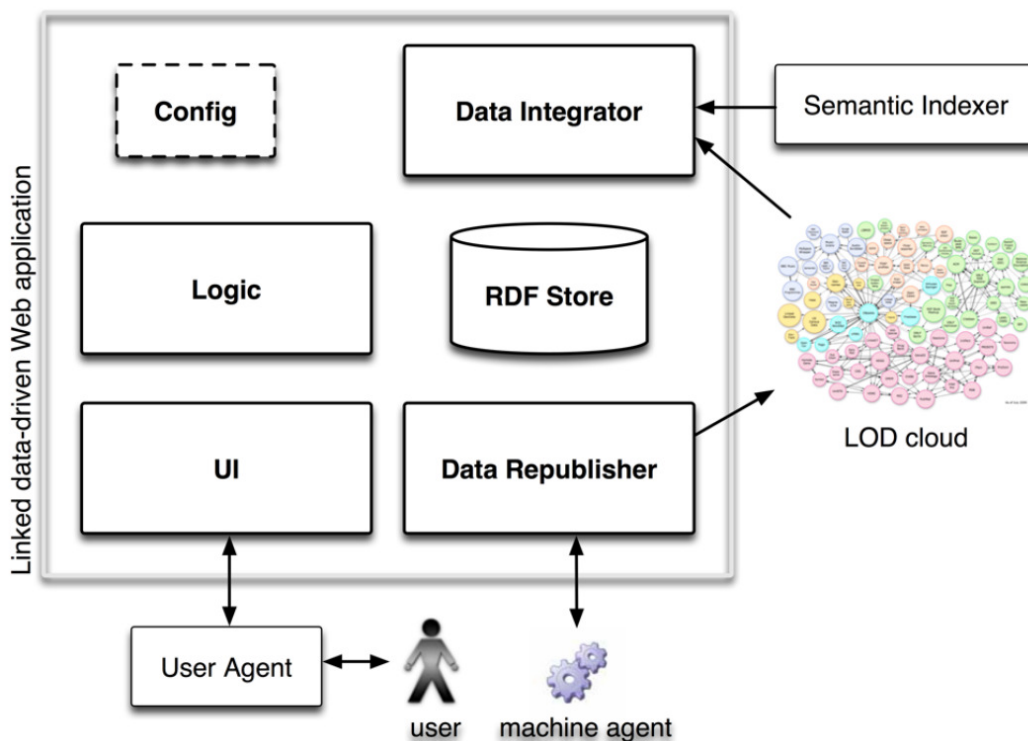


FIGURE 2.4: Concept of a Linked Data-driven Web application (Hausenblas, 2009)

While the Linked Data publishing pattern varies from application to application, the essential components in each application's infrastructure are roughly the same. Hausenblas (2009) has identified four major components for linked-data driven applications (Figure 2.4):

- A local RDF store, able to cache results and act as a permanent storage device to track users, etc.
- Some logic (a controller) and UI modules implementing the business logic, the User Interface (UI) and the interaction parts of the application.
- A data integration component, focusing on fetching Linked Data from the Web of Data
- A republishing component that eventually exposes parts of the application's (interlinked) data on the Web of Data.

The data store, business logic and UI are traditional components for Web applications, which is summarised by (Leff and Rayfield, 2001) as the Model-View-Controller (MVC) pattern. The two main differences that linked-data driven applications have, are data integration and republishing components. The structure of the data store might also be (but not necessarily) different, as the applications need to serve RDF data.

The infrastructure of Linked Data driven multimedia applications should be similar to other applications. However, as the delivery of multimedia resources on the Web is different from the normal HTML pages and the annotations of the multimedia are not necessarily hosted in one domain, the infrastructure needs to be extended, especially in the data integration component.

2.3 Media Fragments and Multimedia Annotation Ontologies

Compared with the hypertext Web, multimedia provides enriched data with multiple representations. The notion of media fragments and multimedia annotations can be tracked back to the early research of hypermedia and the Web in 1990s. The Amsterdam Hypermedia Model (AHM)(Hardman et al., 1994) was developed to compliment the limitations of hypertext systems, which are not broad enough to support the relationships of temporal data. AHM defined “a general framework that can be used to describe the basic constructs and actions that are common to a wide range of hypermedia systems”. After two decades, when online multimedia sharing has become more common on major Web 2.0 applications, multimedia is still a “foreigner” of the Web due to its complexity and multi-dimensional nature. In the semantic Web era, exposing the higher and lower semantics of multimedia content and promoting them into “first-class citizen” of the Web has become more urgent, as the multimedia has yet to join the Web of Data. Many communities have been involved in developing models and standards to make multimedia resources more accessible on the Web. This section reviews the major models and standards in this area.

2.3.1 W3C Media Fragment URI

The W3C Media Fragment Working Group of the Video in the Web Activity³⁴ have collected a wide range of use cases for using media fragments and have proposed Media Fragments URI 1.0 (basic) (Troncy et al., 2012) (W3C-MFURI)³⁵ and Ontology for Media Resource 1.0 (Sasaki et al., 2012), which underpin the interlinking of media fragments and annotations in the Linked Data era. Media Fragments URI 1.0 (basic) is a W3C recommendation, which supports the addressing of image, audio and video along two major dimensions: temporal and spatial. Two more dimensions, track and id (such as chapter 1, section 3, etc) are further defined in a W3C working draft named Media Fragments 1.0 URI (advanced)(Troncy et al., 2011). The information about each dimension is encoded in URIs using hash fragments following a certain format. The main motivation for Media Fragments URI 1.0 is that by parsing the URI fragments, a byte stream of media sub-part can be returned through HTTP with the collaboration of UAs, proxy and original web server, instead of the whole media entity. Some issues regarding URI dereferencing and multimedia representations using W3C-MFURI in Linked Data have also been identified (Hausenblas et al., 2009a).

W3C-MFURI is designed to deal with the closed character of multimedia resources on the Web. It expects the cooperation of “smart servers” and proxy caches to deliver media fragments via HTTP protocol (Mannens et al.). A number of approaches exist for the delivery of multimedia content on the Web: HTTP Progressive Download, Real-time Streaming, and HTTP Streaming. Van Deursen et al. have compared those protocols and pointed out the advantages and issues of each protocol. Real-time Streaming Protocols³⁶ are “stateful” (compared with “stateless” HTTP) and have been implemented by Adobe’s Real Time Messaging Protocol (RTMP) and Microsoft Media Server protocol (MMS). RTSP is not a major concern of W3C-MFURI as the implementations are not based on HTTP. The HTTP Progressive Download optimises the multimedia download process by allowing the multimedia file to be replayed even while the file is not fully downloaded. The native HTML5 video/audio player and many video sharing platforms, such as YouTube, implement this approach. HTTP Streaming use the HTTP protocol to deliver small chunks of video to the client every time, so it can be treated as a series of HTTP progressive downloads. There are two main implementations of HTTP Streaming: Microsoft Smooth Streaming³⁷ and Apple Inc.’s HTTP Live Streaming (HLS)³⁸. Unfortunately, on the server side, major vendors of multimedia hosting services have not yet implemented the functions required to deliver media fragments following the issue of Media Fragments 1.0 URI (advanced).

³⁴<http://www.w3.org/2008/WebVideo/>

³⁵In this thesis, we will use the abbreviation “W3C-MFURI” to refer to URIs that follow the W3C Media Fragments URI 1.0 syntax, while “media fragment URIs” refers to generic URIs as names of media fragments.

³⁶<http://www.ietf.org/rfc/rfc2326.txt>

³⁷<http://www.iis.net/expand/SmoothStreaming>

³⁸<http://tools.ietf.org/html/draft-pantos-http-live-streaming-01>

The first (and probably the only) server-side implementation of Media Fragments 1.0 URI (advanced) is NinSuna(Deursen et al., 2009). NinSuna (Metadata-driven media adaptation and delivery) is a format-independent RDF Model-driven media delivery system. The system can “ingest” different formats of multimedia resource and save information about temporal, spatial and track dimensions to semantically represented data blocks. When delivering the media resource, one or more data blocks can be returned to the user agent (UA) through HTTP Live Streaming according to the Range header the in HTTP request as well as the adaptive context of the UA. As NinSuna model is partially implemented in OWL, the metadata and media fragments can be further linked to other resources. The evaluation results of the server-side implementation of Media Fragments 1.0 URI (advanced) can be found at Van Deursen et al. (2010).

For the client-side implementation, both Firefox³⁹ and Google Chrome⁴⁰ (actually WebKit⁴¹) have partially implemented the temporal fragment syntax defined in W3C-MFURI. Tim De Mey and Davy Van Deursen implemented a media fragment player on Android devices named Fraggy⁴². The player can validate the URI and highlight both temporal and spatial media fragment axes. There are also other client-side implementations of W3C-URI, such as Ninsuna media fragment player⁴³ and xywh.js⁴⁴ developed by Thomas Steiner, but they are based on javascript and thus are only “polyfill” for the browsers where functions are not implemented natively.

Compared with other efforts, which try to encode media fragments as part of the URI, the syntax of MFURI is very simple and easy to implement. However, the specification mainly targets on the informational resources on the Web, i.e. a audio or video file accessible via HTTP. It has not been made clear yet that how MFURI be reused to represent as non-informational multimedia resources and how to make the MFURI dereferencable following Linked Data Principles, which will be the major concern of Chapter 3.

2.3.2 Ontology for Media Resource

Another important W3C standard related to media fragments is the Ontology for Media Resource (a.k.a Media Annotation Ontology, W3C-MA)(Sasaki et al., 2012) developed by Media Annotation Working Group. The goal of the ontology is to provide a core set of properties to describe multimedia resources and map the descriptions of multimedia resources from different standards to this set of properties. W3C-MA presents the core properties with an abstract ontology using RDF/OWL OWL so that it could be widely applied to describe multimedia resources on the Web. The notion of media

³⁹<http://lists.w3.org/Archives/Public/public-media-fragment/2011Nov/0017.html>

⁴⁰<http://lists.w3.org/Archives/Public/public-media-fragment/2012Jan/0021.html>

⁴¹https://bugs.webkit.org/show_bug.cgi?id=65838

⁴²<https://play.google.com/store/apps/details?id=be.mmlab.fraggy>

⁴³<http://ninsuna.elis.ugent.be/MFPlayer/html5>

⁴⁴<http://tomayac.github.io/xywh.js/>

fragment is clearly defined in this ontology as *ma:hasFragment* (the reverse property is *ma:isFragmentOf*) and the range of this property is a media fragment URI following the W3C-MFURI standard.

The NinSuna Ontology⁴⁵ extends W3C-MA and defines richer properties to describe the properties of a “Media Fragment Object”. For example, the values of different dimensions encoded in media fragment URIs can be expressed as in Listing 2.1. Other properties are also defined to describe the lower-level features of a multimedia resource, such as *nsa:codingProfile* and *nsa:maxBitrate*. Based on the NinSuna Ontology, a semantic-driven media decision-taking engine was developed to automatically select the multimedia fragment content to be delivered to the client, based on the available context information (Van Lancker et al., 2013). The decision-making algorithm is presented in N3Logic (Berners-Lee et al., 2008).

```
<http://example.org/1.mp4#t=5,12>
  a nsa:TemporalFragment , ma:MediaFragment ;
  nsa:temporalEnd "5.00"^^xsd:float ;
  nsa:temporalStart "12.00"^^xsd:float ;
  nsa:temporalUnit "npt" ;
  ma:isFragmentOf <http://example.org/1.mp4>.
```

LISTING 2.1: Example RDF Description of Media Fragment Using Ninsuna Ontology

To promote the aggregation ability of annotations created by different systems, the Open Annotation Data Model (OA) (Haslhofer et al., 2012) is proposed by W3C Open Annotation Community Group⁴⁶. This model defines a RDF/OWL vocabulary for representing the annotation relationships of digital resources. The notion of media fragment is deeply integrated into the Open Annotation Data Model and the model can thus be broadly used for any media fragment annotations, even though the standards initial focus was on biomedicine science.

2.3.3 Other Related Standards and Vocabularies

Exposing temporal and spatial fragments for multimedia resources on the Web is not a new idea and many standards have been proposed in both academic and industrial domains. To name a few, MPEG-7 (Martinez, 2004), Synchronized Multimedia Integration Language (SMIL)⁴⁷, Timed Text Authoring Format 1.0⁴⁸ and SVG⁴⁹ have defined XML based syntax to represent temporal or spatial fragments within images, audio and video. However, the descriptions of temporal and spatial dimensions in those standards are divided into several attributes, thus the media fragment is not represented by a sin-

⁴⁵<http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna#>

⁴⁶<http://www.w3.org/community/openannotation/>

⁴⁷<http://www.w3.org/TR/SMIL/>

⁴⁸<http://www.w3.org/TR/2006/CR-ttaf1-dfxp-20061116/#timing>

⁴⁹<http://www.w3.org/TR/SVG/>

gle URI, so they cannot be applied under Linked Data principles as they are non-URI based mechanisms.

At the early stage of Semantic Web, an ontology called OntoMedia(Jewell et al., 2005) was developed to provide a meaningful set of vocabularies to annotate the vast collections of media already existed on the Web. The original target of OntoMedia was to model cultural, textual fiction and film data. At the high level of abstraction, entities and events are defined as the two basic classes and this ontology could be applied to describe the events (person, time, location, etc) happening in a multimedia resource.

There are also URI-based standards, which have been proposed to use URI query, URI fragment (hash URI) or slash URI, to encode various domains of media fragments into URIs. Continuous Media Markup Language (CMML)⁵⁰ is a markup language to represent non-overlapping temporal segments for an audio or video file. CMML is part of the RFC3533 Ogg⁵¹ bitstream format designed for client-server architecture of Web. CMML defines Temporal URI, which specifies time intervals in URI queries and fragments for time-based web resources (Pfeiffer et al., 2005), as well as tracks URI and named fragments.

MPEG-21(Bormans and Hill, 2002) Part 17 also specifies a normative URI fragment syntax to address fragments in MPEG compatible files. MPEG-21 Part 17 defines a powerful syntax to identify temporal, spatial and tracks from an MPEG bitstream. Despite the expressive syntax, it is too complex to fully follow the standard and it is only applicable to MPEG-related MIME types. Core Ontology for Multimedia (COMM) aims to add formal semantics to MPEG-7 (Arndt et al., 2007), but again, the ontology relies on the MPEG-7 description and is not compatible with other formats. The EnAKTing project develops reusable URIs to represent temporal entities (Correndo et al., 2010). They applied slash namespaces and developed the concept of Linked Timelines, which adopts the OWL time ontology⁵² to describe temporal entities and relationships between them. The proposed URIs for Linked Stream Data also applied slash namespaces to include real time and space information in URIs together or separately.

Multimedia Metadata Ontology (M3O) (Saathoff and Scherp, 2010) provides a framework to describe and annotate complex multimedia resources. It fills the gap between the structured metadata models, such as SMIL and EXIF⁵³, and semantic annotations. M3O can be integrated with various standards to provide semantic annotations, with some further development. The key concept of M3O is separation of information objects and information realisations. Based on this idea, M3O produces a core set of ontology design patterns, including annotation, decomposition, collection and provenance patterns. M3O is very expressive in that it separates the information object and its

⁵⁰<http://annodex.net/TR/draft-pfeiffer-cmml-03.txt>

⁵¹<http://tools.ietf.org/html/rfc3533>

⁵²<http://www.w3.org/2006/time#>

⁵³<http://www.exif.org/Exif2-2.pdf>

realisation, which has not been included in W3C-MA or OA. But the models and patterns themselves are very complex and abstract for implementation. In addition, M3O depends on the provision of mappings in order to be embedded in other languages, such as SMIL, which needs a lot of extra work.

In the recently established LinkedTV project⁵⁴, the Linked Media Principles has been suggested as (Nixon, 2013a):

- Web media descriptions need a common representation of media structure
- Web media descriptions need a common representation of media content
- Web media descriptions need to use a media ontology which supports description of both the structure and content of media
- The descriptions of media in terms of common representations of structure and content are the basis for deriving links across media on the Web (Linked Media)

The Linked Media Principles actually call for a universal representation of both lower level structure and higher level content of the media resources, so that links can be created across the Web. Hausenblas et al. (2009a) divided the methods for establishing links among multimedia annotations into four categories: totally manual, collaborative editing, semi-automatic and totally automatic. Manual methods mainly refer to user-contributed linking (Halb et al., 2007) and game-based interlinking, where end users can generate high quality links with fun. This thesis actually solves parts of the problems in Linked Media Principles.

2.4 Applications of Media Fragments and Semantic Multimedia Annotation

The early work of using semantic Web technology to annotate multimedia started around 2000 in the AKT project⁵⁵. The CoAKTinG project (Buckingham Shum et al., 2002), which aims to support collaboration in e-Science, released some toolkits to help annotate real-time meetings with ontological mark-ups. The events in the multimedia stream can be automatically captured and annotated with semantic annotations. These projects represent early attempts at managing continuous multimedia resources through semantic annotations. But the semantic annotations had not reached a degree to which various multimedia resources on the Web could be massively linked together at that time because of the lower popularity of continuous multimedia resources and the less developed semantic Web technologies.

⁵⁴<http://linkedtv.eu>

⁵⁵<http://www.aktors.org/akt/>

With the development of Linked Data, many multimedia authoring applications on the Web have implemented their URIs for media fragments and managed to relate resources from various datasets for discovery and reasoning. Bizer et al. (2007) introduced two general ways of setting RDF links: manually or automatically. In the manual way, developers need to know which datasets they want to link to with the assistance of official CKAN datasets registry⁵⁶ or an RDF crawler like Sindice (Tummarello et al., 2007). Then developers need to set up the links manually. The automatic way usually needs an algorithm to map items in both datasets. Ideally, a one-to-one relationship can be generated if the item in one dataset can be found with no ambiguity in the other one. As the data in the BBC have connections with a variety of domains, they have to provide a content categorisation system called CIS (Kobilarov et al., 2009) to select the best matched resource.

In the rest of this subsection, a set of applications about media fragments and semantic multimedia annotations is listed and how the denotation of media fragments are used in each system is explained.

Table 2.1: Applications about media fragments and semantic multimedia annotations

Semantic Wiki and MetavidWiki	Semantic Wiki, which contains multimedia objects and allows annotation of part of the multimedia objects, is the chief manner of collaborative interlinking. MetavidWiki extends the famous semantic MediaWiki ⁵⁷ and supports interlinking between temporal fragments. Semi-automatic method is quite similar to the applications which create tags with controlled vocabulary. Users will be given some suggestions and asked to accept, modify, reject or ignore the suggestions. Automatic interlinking can be realised by simply analysing the content of datasets. However, it has been suggested that after the automatic interlinking generation, a community needs to review and modify the links to improve the accuracy.
Annnotation (Lambert and Yu, 2010)	Annnotation is a tool to handle the input annotations from users developed by the Open University in the UK. The annotations are saved in the RDF quad store with users' own privacy and provenance data. All the videos and annotations in the store are assigned with globally unique URIs so that they can be published in the Linked Open Data cloud.

⁵⁶<http://ckan.net/group/lodcloud>

⁵⁷http://metavid.org/wiki/MetaVidWiki_Features_Overview

SugarTube (Lambert and Yu, 2010)	SugarTube browser is another Linked Data application developed by the Open University to help learners navigating through resources using Linked Data. When doing a term search, SugarTube can invoke RESTful services provided by DBpedia, Linked GeoNames and Open University Learning Resources RDF Repositories to get semantically relevant data about the searched terms.
Yovisto.com (Waitelonis and Sack, 2009)	Yovisto.com hosts large numbers of recordings of academic lectures and conferences for users to search in a content-based manner. Some researchers augment Yovisto open academic video search platform by publishing the database containing video and annotations as Linked Data . Yovisto uses the Virtuoso server (Erling and Mikhailov, 2007) to publish the videos and annotations in the database and MPEG-7, COMM to describe multimedia data.
Europeana ⁵⁸	Europeana is a platform for users to share and annotate multimedia resources about culture and history all over Europe. Europeana integrated LEMO multimedia annotation framework (Haslhofer et al., 2009), in which media fragments are published using MPEG-21 vocabulary. LEMO has to convert existing video files to an MPEG compatible version and stream them from the LEMO server. LEMO also derived a core annotation schema from Annotea Annotation Schema ⁵⁹ in order to link annotations to media fragments identifications.
SemWebVid(Steiner, 2010)	SemWebVid is an application which can automatically generate an RDF description for video resources. The original plaintext description is provided by user-generated metadata or speech recognition. Then the plaintext description is analysed by multiple Natural Language Processing (NLP) Web services, such as OpenCalais API ⁶⁰ and Zemanta API ⁶¹ . The entities extracted from the NLP services are mapped to each other and finally merged together as the RDF description of the video resources.

⁵⁸<http://www.europeana.eu>

⁵⁹<http://www.w3.org/2000/10/annotation-ns#>

⁶⁰<http://www.opencalais.com/>

⁶¹<http://www.zemanta.com/>

The EU NoTube project(Aroyo et al., 2011)	Used semantic web technologies to link TV channels' data with LOD resources . The semantic data was used to further exploit the complex relations between users' interests and the background information of TV programs.
LinkedTV project	As a follow up of NoTube, the LinkedTV project has developed a linked service infrastructure as an entry point to search and link media resources with individual concepts (Nixon, 2013b). A semantic tool, ConnectME, is also developed for enriching online videos with Web content (Nixon et al., 2012).
Vox Populi Bocconi et al. (2008)	Vox Populi implements a model to automatically generate video documentaries. In Vox Populi, media fragments are applied to generate video documentary through the annotation structure. The concepts associated with media fragments could be free-text, predefined keywords, a taxonomy, a thesaurus or an ontology.
The BBC World Service archive (Raimond et al., 2013)	BBC World Service archive has a large collection of audio resources that have little or no metadata and subtitles. Raimond et al. (Raimond et al., 2013) developed a system to use semantic Web to automatically link the audio content in the historical archive to the BBC live news . The system is driven by concept extractions from semantic named entity recognition tools, topic extraction from live BBC News subtitles and user validations of the results.

Some other applications also have functions that allow users to share part of a multimedia resource. YouTube has launched facilities to annotate temporal and spatial parts of a video clip. Users can right click on a playing YouTube video and “copy video url at current time. The copied URI will have a URI fragment starting with “t=XXs, and YouTube can play from that time point. In November 2013, YouTube introduced a new feature that allows users to “tag a spatial area at a temporal point. Each user created tag has a URL pointing to a landing page hosted by Clickberry⁶² then this link can be shared via Facebook and Twitter. Here is an example of the tag URI:

<https://clickberry.tv/video/6dafe30e-dcb8-44b8-8190-32be8249a297>

Similarly, Synote also provides a get current position url function, which encodes temporal information into a URI (Li et al., 2009). Google predicts that Rich Snippets will provide richer video search results. With the help of a media fragments URI, the faces in

⁶²<http://clickberry.tv>

a film clip fragment (still image or moving images) can be identified with a single URI, which allows other users to link their resources to this particular media fragment (Steiner et al., 2010). Google also suggests that the use of video markup formats, such as Facebook Share and Yahoo!SearchMonkey RDFa⁶³, be used to improve the search results for videos.

While many of the systems listed above allow users to associate semantic annotations with media assets or even media fragments, there is no agreed way to share those annotations across systems, especially for systems that work offline and distribute the annotations in other media, such as CD and DVD. Hardman et al. defines nine “canonical processes, representing the highest level abstraction of the processes, to improve the interoperability of semantic-enriched multimedia production (Hardman et al., 2008). The nine processes are: premeditate, create media asset, annotate, package, query, construct message, organize, publish and distribute.

2.5 Video Classification

This section explores the machine learning techniques used in different video classification tasks, which serve as background knowledge to the new algorithms that will be introduced in Section 5.4. In-depth research on video classification algorithms is not the aim of this thesis, nor developing new algorithms to improve the accuracy of a specific video classification task. The focus is on the algorithms that are related to video classification using multimedia annotations, especially user-generated annotations for online video sharing platforms.

Automatic video classification has usually been treated as a supervised classification task using lower-level features from multimedia analysis or higher-level textual features. A survey of automatic video classification shows three modalities mainly being used: text, audio and visual (Brezeale and Cook, 2008). Many classification algorithms have involved two or more modalities. For example, Borth et al. (2009) combined tags and visual features using a weighted sum fusion.

For the classification of online videos, some studies have brought the semantic Web into the text mining. Wang et al. (2012) proposed using extrinsic semantic context extracted from video subtitles with lower-level video streams. Cui et al. (2010) proposed a framework that extracts content features from training data to enrich the text-based semantic kernels, which is used in the Supported Vector Machine (SVM) classifier. For all these algorithms, removing ambiguity from words plays an important role. So external knowledge sources, such as WordNet (Bentivogli et al., 2004), are usually involved for knowledge extraction and word sense disambiguation. The temporal feature is applied in many algorithms since it is an important attribute for videos. Hence, Niebles et

⁶³<http://www.google.com/support/webmasters/bin/answer.py?answer=162163>

al. used temporal correlations in videos to detect audio-visual patterns for classifying concepts (Niebles et al., 2010).

The viewing behaviour of users and their (online) social interactions are sometimes used for improving the accuracy of video classifications. YouTube co-watch data was used for training in Zhang et al. (2011). The results demonstrated that the proposed method has superior performance when there is not enough manually labelled data available. Filippova and Hall (2011) categorised YouTube videos based on textual information, especially user-generated comments. Subtitles are also identified as an important resource to provide new features for video classification on the Web (Huang et al., 2010). Katsioulis et al. (2007) explored an unsupervised approach for semantic video classification by analysing subtitles. They used WordNet as the external knowledge sources for named entity disambiguation and they also suggested that the “subtitles of each segment can be processed with the support of domain ontologies” in order to improve the classification results.

As the world's largest video sharing platform on the Web, YouTube has also developed automatic algorithms to classify videos into different YouTube channels, based on named entity recognition and some simple machine learning algorithms (Simonet, 2013). The approach proposed in the paper completely ignores the lower-level image and audio content, and relies on user-generated metadata associated with the videos and different channels, such as title, description and keywords provided by the users when they create channels and upload videos. As the textual metadata is usually sparse and noisy, some additional refinements need to be made before applying the machine learning algorithms and named entity recognition. The methodology has three steps to map a video into a channel: (1) mapping videos to semantic entities; (2) mapping semantic entities to taxonomic categories; (3) mapping channels to taxonomic categories (by combining both previous steps). The results show a precision of around 95% for the automatic classification, covering over 85% of the channels.

2.6 Summary

The introduction to the state of the art found that Linked Data, multimedia annotations and media fragments, are mature in both theoretical research and practical implementations. However, few publications and implementations address the application of Linked Data to media fragments and the seamless integration of media fragments with the Web of Data. Some of the theoretical research focuses on just one or two Linked Data principles, such as designing media fragment URIs (W3C-MFURI), dereferencing (NinSuna), describing media fragments with RDF (W3C-MA, Annotea and MPEG-21), and interlinking media fragments (Europeana, SemWebVid and NoTube). Some at the practical level developed working systems to interlink and visualise media fragments, such

as Yoviso.com, Europeana, SemWebvid and YouTube. However, there is hardly any research to bridge and integrate the theoretical principles and working implementations in a systematic manner, and to offer guidelines and best practice for media fragments publications. Chapters 4 5 and 6 will address this issue.

In addition, the current applications seldom consider using the Linked Data to improve the indexing of media fragments for major search engines, which is important for refining the online presence of media fragments. Efforts such as schema.org enhance video/audio indexing as a whole, but such attempts are not currently working for media fragments, and the function of indexing and searching media fragments has not yet reached the Web scale. The Media Fragment Indexing Framework in Chapter 5.4 will propose a mechanism to solve this problem. For video classification, named entity disambiguation has been widely applied. However, no previous research has considered media fragments as a feature and combined it with other feature sets. Section 5.4 will conduct an experiment to bring the new feature set, based on media fragments, into video classification tasks.

Chapter 3

Use Case Study and Requirement Analysis

Chapter 2 identified a theoretical gap in the process of publishing media fragments as Linked Data, i.e. media fragment semantics, and using it to improve the indexing of part of the multimedia resources. This Chapter will detail the requirements to bridge this gap through a series of case studies. The purpose of presenting use cases is to extract the key requirements to fulfil the scenarios and how users from different domains could benefit from the media fragment semantics.

Sections 3.1 and 3.2 present use cases that will benefit from implementing media fragment semantics, including some multi-disciplinary use cases where media fragments semantics can help to solve research problems in other areas. As this research is underpinned by the current state of the art, it is inevitable that several case studies on existing multimedia applications are included. They are either research platforms (such as Synote and Edina Mediahub¹), or major multimedia sharing platforms (such as YouTube and Facebook). It is necessary to investigate existing platforms as they are the major data providers of raw media fragments and annotation. Section 3.3 therefore analyses a couple of major multimedia platforms and research applications to determine their requirements for publishing media fragments as Linked Data. All the use cases refer back to both the formal and the informal (such as blogs and wiki) relevant literature.

Section 3.4 then analyses in-depth the underlying problems that prevent the application of media fragments and Linked Data to the current Web of data. Finally, Section 3.5 summarises the requirements that fulfil the gap between media fragments and Linked Data.

¹<http://jiscmediahub.co.uk>

3.1 Interlinked Media Fragments Usage on the Web

The three use cases presented in this section implement sharing, indexing and reasoning on media fragments. The requirements derived from these use cases are actually specific implementations of Linked Data principles in the media fragments domain.

3.1.1 Use Case 1 (UC1): Sharing Media Fragments in Social Networks

Video sharing has become a common function for social network applications. With this function, users on Facebook and Twitter can share the videos they like and make comments when sharing them. Sometimes, there is only a certain part of the video that the users want to highlight instead of sharing the whole video. For example, Bob finds a YouTube video about Web Platform Docs², and quite to his surprise, Sir Tim Berners-Lee, the inventor of World Wide Web, was introduced as a “Web Developer”. Tim Berners-Lee is interviewed in this video between 1m30s and 1m33s, and the phrase “Web Developer” appears on the left-bottom corner. He wants to share this media fragment on Facebook, but he does not want his friends to watch the whole video. So using W3C-MFURI syntax, he attaches the URI hash `t=00:01:30,00:01:33&xywh=40,220,200,50` at the end of the YouTube URI, which corresponds to the temporal span 1m30 to 1m33 and the spatial area taken by “Web Developer”. Bob’s friend Jack sees Bob’s update about this video. Jack clicks the video and wants to see if there is anything interesting. The video player embedded in Facebook then starts playing the video directly from 1m30s and stops on 1m33s. What’s more, the phrase “Web Developer” is enclosed by a rectangle that highlights the area Bob wants everyone to notice.

Bob also sends a Tweet to share this media fragment. His Tweet message looks like this:

```
Check this. Tim Berners-Lee is just a “Web Developer”?  
http://www.youtube.com/watch?v=Ug6XAw6hzaw  
#t=00:01:30,00:01:33&xywh=40,220,200,50
```

Bob’s follower Charlie clicks on the link and the browser opens the YouTube page. YouTube player then start playing the video from 1m30s and stopped on 1m33s and the “Web Developer” area is circled and highlighted.

This use case demonstrates that media fragments are very useful where the video is very long and users only want to bookmark a certain section of it. Media fragments will make the video sharing on social networks easier and more efficient. To implement such a function, at least two aspects need to be considered.

²<http://www.youtube.com/watch?v=Ug6XAw6hzaw>

First, *the media fragment information from regularly used domains (such as temporal and spatial domains) need to be encoded in the URI*. This has also been reflected by the requirement defined by the W3C Media Fragment Working Group³. Meanwhile, it is also a non-functional requirement that *the syntax of the media fragment URI needs to be simple so that it can be constructed and parsed easily*. This requirement actually indicates the trade-off between “expressiveness” and “user-friendliness”. One example is MPEG-21 mentioned in Section 2.3.3. MPEG-21 Part 17⁴ defines a very comprehensive syntax to build a fragment identifier. It can express moving regions in a video sequence and the objects in a 3D image. However, according to the W3C Media Fragment Working Group’s wiki⁵, no implementations of such syntax yet been identified by at least the members of the working group.

Secondly, by clicking on the media fragment URI, users want to view directly the fragment encoded in the URI. This means either the URI pointing to a real multimedia file or to some webpage that replays the video; thus *the media fragment need to be highlighted directly when opening the URI in the browser*. This user experience has been identified by many applications, such as MetaVid (see W3C Media Fragment Work Group wiki⁶ and YouTube as explained in Section 2.4. Technically, this requirement can be divided into two situations. In HTML5 webpages, when the media fragment URI points to a real audio or video file, it is up to the specific browsers to highlight the fragment as the HTML5 native audio or video player will be used to replay this resource. For example, Firefox has implemented the highlighting of temporal dimension of MFURI⁷. Another situation is with YouTube and Dailymotion, where the media fragment URI actually points to a landing page instead of a real file. In this case, the application owners can implement their own scripts to make the audio/video player highlight the media fragment.

3.1.2 Use Case 2 (UC2): Search Media Fragments Based on User-generated Content

Steve is a student watching a lecture recording made by Bob about the Semantic Web. Between the 80th second and 90th second, Bob mentioned the term Linked Data principles, which Steve could not understand. He adds a comment “What are Linked Data principles?”. Bob, the lecturer, on seeing Steve’s comment, relates this media fragment to a media fragment of Alice’s lecture recording (between the 20th second and 500th second), in which she further explains Linked Data principles. It is not necessary that the two recordings are located in the same repository thanks to the fact that both media

³<http://www.w3.org/TR/media-frags-reqs/#media-fragment-requirements>

⁴<http://mpeg.chiariglione.org/standards/mpeg-21/fragment-identification>

⁵<http://www.w3.org/2008/WebVideo/Fragments/wiki/MPEG-21>

⁶http://www.w3.org/2008/WebVideo/Fragments/wiki/Use_Cases_Discussion#Media_Browsing_UC

⁷https://bugzilla.mozilla.org/show_bug.cgi?id=648595

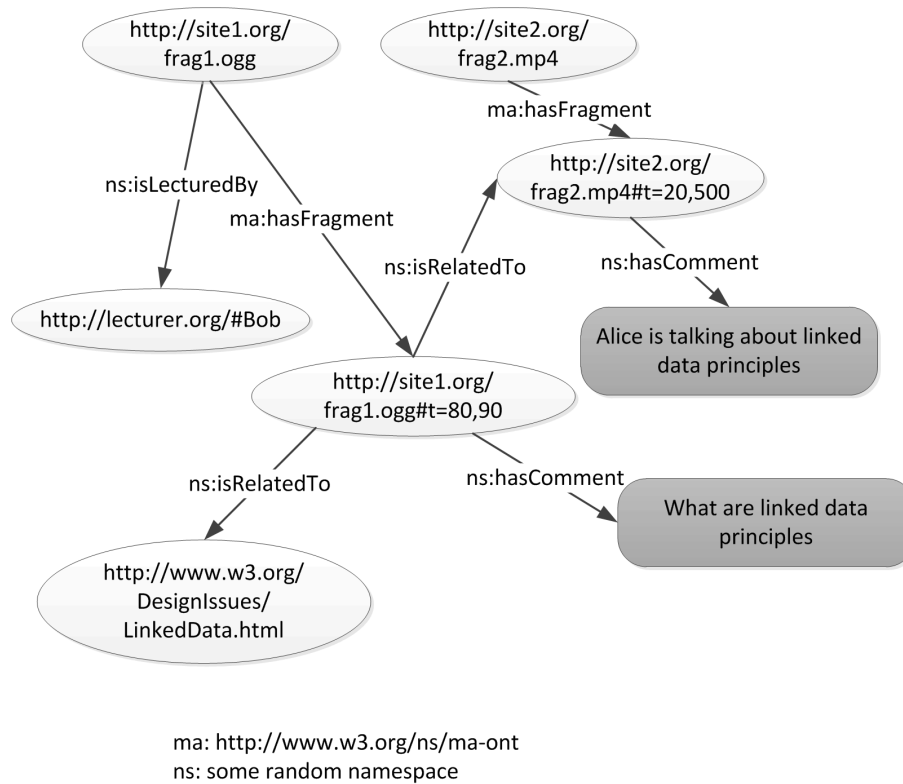


FIGURE 3.1: An example of interlinking media fragments to other resources

fragments are exposed as URIs. Bob also links the official W3C page about Linked Data to this media fragment, so that Steve, as well as other students with similar questions, can do some further reading. More importantly, when searching "Linked Data principles", both Bob and Alice's lectures will be listed in the search result and the result can also point to the exact fragment where the term "Linked Data principles" is mentioned, even though the whole lecture may be about some other topic. Figure 3.1 presents the whole RDF graph representing the relationships described in this use case.

While the keyword search has been widely accepted and become a habit of users, it is straight forward thinking to *search media fragments through the plain-text annotations associated with them*. The notion of "annotation" is very vague and its design and implementation depends on the individual application. However, when considering merging media fragments with the LOD Cloud, such annotation relationship should be modelled by ontologies. This requirement has been identified by W3C Media Fragment URI 1.0 (basic) as "Example uses (of media fragment URI) are ... the annotation of media fragments with RDF". Likewise, the Open Annotation Model in "Fragment URIs Identifying Body or Target"⁸.

⁸<http://www.openannotation.org/spec/core/core.html#FragmentURIs>

3.1.3 Use Case 3 (UC3): Reasoning Based on Timeline

Alice is watching a lecture about how to disassemble computer hardware. She adds a comment “cut the power” to explain what happens between 80th second and 600th second. Bob watches the same recording and creates a note “open the mainframe box” to the time span of 700th second and 800th second. Steve is not sure which should be done first: “cutting the power” or “open the mainframe box”. As the media fragment to which Alice’s annotation attaches occurs before the one to which Bob’s annotation attaches, the system can do the reasoning and give the answer to Steve to “cut the power before open the mainframe box”. Figure 3.2 demonstrates this use case. There are many

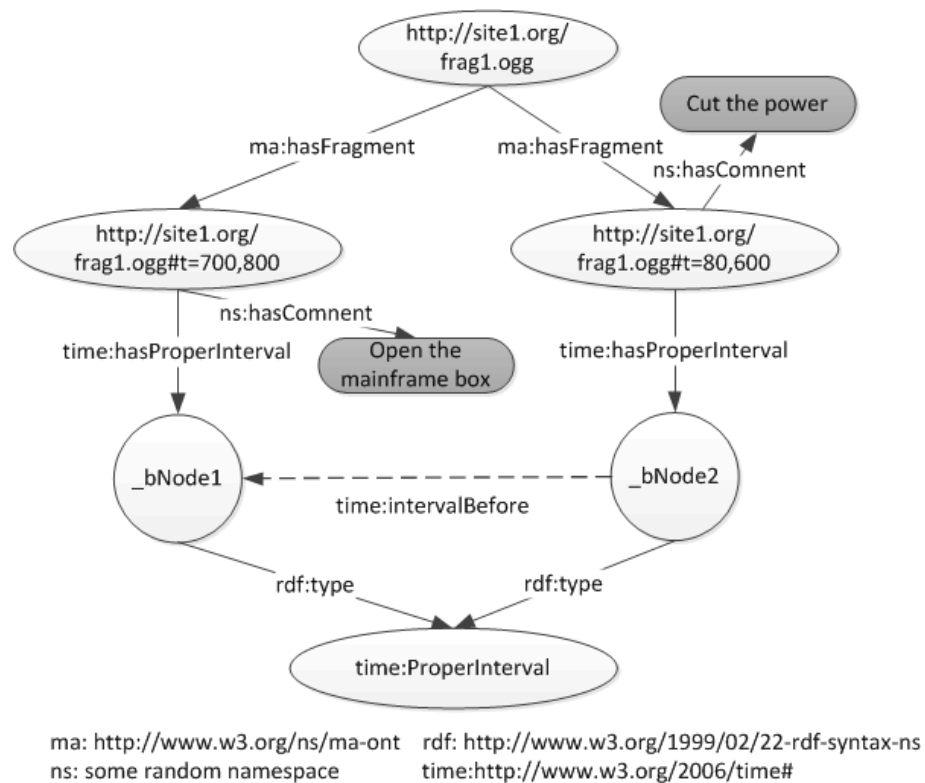


FIGURE 3.2: Addressing relationships between media fragments using Time Ontology

new predicates related to temporal and spatial dimensions of media fragments that can be further explored. For example, J F Allen summarises the relationships between time intervals as: before, equal, meets, overlaps, during, starts and finishes (Allen, 1983). Based on UC3, when text annotations are associated with media fragments, the possible time or spatial relationships between annotations can be inferred automatically. There are many videos trying to present a procedure where things must be done in a certain order, such as doing chemistry experiments, cooking a dish, using electronic equipment, starting a car, etc. Similar relationships can also be found in the spatial dimension as “within or around an area of something”. This function is very useful in reasoning to answer questions like “A should happen before B”, even though user-generated annotations do not explicitly say that.

This use case reflects that about time-based reasoning defined in Time Ontology⁹ (also see Section 2.3.3). While the media fragment URI encodes media fragment information from different dimensions into a URI hash, it is necessary to *bring in other vocabularies to explicitly model the detailed value of each dimension*, such as the relative start and end time or the real physical time. Then reasoning rules can be setup to deduce the results in UC3. This has already been implemented by NinSuna ontology.

3.2 Multi-disciplinary Cases

Section 2.4 introduced various applications that apply media fragments in other domains, such as cultural heritage (Europeana) and e-learning (SugarTube). This section presents multi-disciplinary cases related to politics, law, and public media domains. Even though the domains differ, media fragments still enhance existing research methodologies in those domains, such as saving effort of finding support materials or showing insight into the audio-visual data at a smaller granularity.

Generally speaking, the annotations can be divided in two categories: lower level multi-media metadata annotations (such as codec information, resolution, and duration of the resource), and higher level annotations related to the semantics of the multimedia content (such as who is in the picture and user-generated tags, descriptions, etc.). Higher level annotations are usually more useful in multi-disciplinary data analysis since the actual content of the annotation is more important.

3.2.1 Use Case 4 (UC4): UK Parliamentary Debate

In recent years, the UK Parliament has taken great steps to make available a quite remarkable range of documents. The debate videos and verbatim Hansard records¹⁰ of Parliamentary debates, business sessions and reports from both public bill and select committees are now routinely available online. Many volunteer communities, such as TheyWorkForYou.com¹¹, provide applications for users to search and analyse Parliamentary data based on the Hansard records. However, except for the basic functions provided by Hansard and the volunteer communities, students and researchers in politics and public relations are looking for means of demonstrating how these verbatim records can be better contextualised to deepen the understanding of those who read them. They are looking for the interlinking among those verbatim records with relevant materials and documents located elsewhere on the Web in order to help facilitate better and more meaningful public engagement with them. So it is necessary to publish media fragments

⁹<http://www.w3.org/TR/owl-time/#examples>

¹⁰<http://www.parliament.uk/business/publications/hansard/>

¹¹<http://www.theyworkforyou.com>

of the debate videos and interlink the fragments with other relevant resources on the Web.

For example, when an MP participating in a debate mentions some specific piece of evidence given by a witness to one of the select committee inquiries into a crime case, some vocabulary can be used to link this media fragment with the evidence found in the select committee report. It will save considerable time for users, who want to find out where the report is mentioned, than going through the whole debate video. It will also be useful for those who are watching this video fragment to find out the supporting evidence for the debate. Similarly, when MPs mention some key actors in the debate, such as former members of the police, journalists or other MPs, the URIs of these people can be linked to this media fragment.

The publishing of fragments of debate video and transcripts enables users online to put the Parliamentary documents into a far richer informational context and makes them easier to search. It can help students to better contextualise parliamentary records as part of their studies and come to a fuller understanding of the role of parliament in political life.

3.2.2 Use Case 5 (UC5):US Supreme Court Argument Analysis

The records coming from The Oyez Project at Chicago-Kent College¹² in the form of audio and transcripts of oral arguments in the United States Supreme Court cover the period October 1955 to June 2011. The data elements include an existing ontology and an implementation of RDF. The archive now has about 10 000 hours of audio aligned at the sentence level, with all speakers identified, over a 50+ year period (about 110 million words) in valid and well-formed XML.

Scholars of the U.S. legal system believe that the oral arguments provide valuable insight into the final decision-making. During the oral arguments, justices often reveal their thoughts and predilections, including the fact that they may need additional information before deciding. Scholars want to know if the thoughts in these oral arguments finally translate into actions, i.e. do the justices' actions during these arguments manifest themselves overtly when they reach final legal and policy decisions?

Currently, scholars only try to answer this question with in-depth case studies of one or two landmark decisions, by examining systematically a small handful of cases and by examining how the justices' behaviour at oral arguments affects their votes in the case. Oyez Corpus itself is not sufficient to address the research question because there is a lack of technology to analyse thousands of cases with audio and transcripts and link them together across cases or to the decision-making data in other repositories.

¹²<http://www.oyez.org>

The publishing of audio fragments and annotations (transcript in this case) will allow the automatic connection of multimedia information (including text, audio, pictures, videos) across the cases, and viewing them together in any number of applications. All the cases can be modelled by certain ontologies and the structured data will be saved as RDF. Important relationships can be set up among the Oyez Corpus, official U.S. Supreme Court dataset ¹³ and related communities' reports. Scholars will be able to discover portions of one oral argument (either justices' questions or attorneys' answers) associated with other oral arguments years or decades earlier or later. But more important is the ability to combine that information in entirely new ways and showing connections, differences and similarities providing the radical development to be able to search across cases and repositories. Further, part of what Linked Data does is exposing information that is often hidden and can help scholars analyse and infer, to better understand how Supreme Court oral arguments affect decisions justices make.

3.2.3 Use Case 6 (UC6):Improving Political Transparency by Analysing Resources in Public Media

TV programmes, such as new reports, political discussions and interviews, exert a tremendous influence on political transparency in the UK. Political figures need to be responsible for what they have said to public media as they will be monitored by the public. However, it is currently still difficult to automatically analyse the archives of such programmes and find the answer to questions such as: did some political figure make a promise some time ago that he did not meet later; did someone in the government refer to a figure that was actually wrong? As the most influential broadcasting company in the UK, the BBC has a large volume of archives of political programmes with transcripts segmented by speakers, such as Question Time¹⁴. Natural language processing tools can be applied to analyse the transcripts, extract important concepts (semantic annotations) from statements speakers have made, and categorise them by key concepts, such as law, economy, foreign affairs, NHS, migration. Furthermore, using Linked Data, each important statement and semantic concept in the programme will be linked to a fragment of the video archive. Then users can search by speakers, categories and plain text, and watch the video fragments as proof of the statement. With the help of video metadata, statements and media fragments can be visualised along the real-world timeline. Similar to the demonstration of TimelineJS¹⁵, users can easily navigate through the timeline and spot whether some government or political figure's statements made at different times are inconsistent.

Compared with traditional annotations, semantic annotations can tell a computer how media fragments are related to other data and how these relationships can be processed

¹³<http://scdb.wustl.edu/>

¹⁴<http://www.bbc.co.uk/programmes/b006t1q9>

¹⁵<http://timeline.verite.co/>

automatically. So in those multi-disciplinary cases, the *meaning of annotations need to be contextualised with the research questions and the annotation relationship will be initialised following domain specific vocabularies.*

To generate such semantic annotations, the entities need to be extracted from *the raw multimedia resources first and they should be time-aligned with media fragments in order that the segments can be searched through the annotations.*

Even though not expressed explicitly, all three use cases mentioned require *the provision of time-aligned text with media fragments in order to search audio or video segments.* Such alignment has sometimes been provided by the data source, such as Oyez, or through end users, such as UC2. Other technologies such as speech recognition can generate such alignment automatically. In addition, *the published media fragments and annotations need to be further processed and interlinked with external resources on the Web,* in order to help multi-disciplinary research, which is related to the fourth Linked Data principle. The external resources include, but are not limited to, social media, general linked datasets (such as DBpedia), and domain-specific datasets (such as LinkedGeoNames).

3.3 Applying Media Fragments and Linked Data to Existing Applications

Most multimedia sharing applications have made those annotations available on the whole multimedia level, but they are yet to expose sufficient semantic annotations for media fragments. This leads to difficulty in processing complex searches of media fragments and thus providing accurate search results. Section 2.4 presented work on publishing existing video resources as Linked Data, such as SemWebVid and NoTube. This subsection will now elaborate several use cases to apply Linked Data principles to existing multimedia applications and suggest what publishing patterns they could use to publish that Linked Data. Many multimedia sharing platforms have already made their multimedia and text resources available through means such as Web APIs. When publishing media fragments, all the use cases follow the core rule that this function will, preferably, be an extra layer on top of the existing systems, instead of developing new systems from scratch (Heath and Bizer, 2011).

3.3.1 Use Case 7 (UC7): Media Fragments in YouTube

YouTube is currently the largest video sharing platform on the Web and was ranked by Alexa in April 2014 as the third most popular website¹⁶. Every YouTube video has

¹⁶<http://www.alexa.com/siteinfo/youtube.com>

a unique id, for example “Wm15rvkifPc” and when watching this video, users need to open this URL:

```
http://www.youtube.com/watch?v=Wm15rvkifPc
```

The webpage dereferenced by this URL does not point to an actual video file, but a landing page with a Flash video player embedded, which requests the video content from the YouTube streaming server. Apart from the video player, the landing page also contains other content that is related to the video, such as video descriptions, and comments. YouTube implements its own syntax, which is partially complied with the W3C MFURI specification, to encode an offset time for video replay. For example, YouTube will start playing the video from 120 seconds onwards if “#t=120” is attached to the landing page URL:

```
http://www.youtube.com/watch?v=Wm15rvkifPc#t=120
```

This function is implemented by detecting the time offset information in the URL and controlling the player to start playing from 120 seconds through the YouTube video player API¹⁷. There are several other syntaxes that YouTube accepts to encode time offset (see Table 7.2 in Chapter 7), but the implementation to highlight such a media fragment is the same.

YouTube allows users to upload closed caption (cc) to the videos and one video can have closed captioning in different languages. If closed captioning is available on the landing page, users can either select it in the status bar of the video player or open it in a separate panel underneath the video player. For the latter, users can click the transcript block and the video player will jump to the start time of that transcript block. The time-aligned transcript has also been used for searching within the YouTube domain. For example, when searching “tremendous gift” with “Closed Caption” feature, some results show an option of “start playing at search term”. However, if the same search term is input to a Google search and restricted to the YouTube domain, no search result shows the URL with time offset, which means the transcript has not been indexed by Google and associated with a media fragment in YouTube.

YouTube’s method of sharing videos is very common, i.e. users cannot download the video directly from the YouTube website and all the videos are streamed from the YouTube server and displayed on a landing page. Dailymotion and Vimeo also follow this method. While this gets around the legal issues of distributing creative work and saves bandwidth for delivering video, the concept of video in these cases is actually blurred and sometimes even misleading.

¹⁷https://developers.google.com/youtube/js_api_reference#Playback_controls

Since the real video file is hidden from direct access by YouTube, the landing page is treated as the “YouTube video” itself when someone refers to a video on YouTube. Thus the URI with $t=120$ can be considered as a media fragment URI in a broader sense. However, strictly speaking, the landing page URI is only an HTML page with a video player embedded, so it should not be considered as a “video” at all. Therefore, the hash fragment attached at the back of the landing page URI is only a HTML anchor in the page. It can, of course, follow the syntax defined in MFURI, but it is still not strictly a media fragment URI. The problem is how the YouTube video can be referred to at a media fragment level, because there is a dilemma that no solid and stable URI is pointing to an informational video file in YouTube, while the landing page cannot represent the video itself. Even worse, semantically, all the annotations, either metadata of the video or the annotations displayed on the landing pages, are actually annotations of the videos instead the landing pages.

This dilemma is a very important design issue in the semantics of media fragments. From the YouTube use case, it is essential for existing multimedia applications to *create two sets of URIs to distinguish real audio/video file locations from the landing pages*.

3.3.2 Use Case 8 (UC8): Media Fragments in Facebook

Facebook is the largest online social network website, founded in February 2004 by Mark Zuckerberg and a group of fellow Harvard students. In March 2013, Facebook had 1.11 billion users. Facebook filed for a \$5 billion IPO on 1 February 2012 and valued the company at \$104 billion in July 2014¹⁸.

Users of Facebook can post images and share YouTube videos in their timeline. Facebook hosts all the images uploaded by users through a Content Delivery Network (CDN) service. When viewing the pictures, users can tag a certain area in the picture with a person’s name. In other words, Facebook allows users to annotate image fragments with people’s names. YouTube videos embedded in users Facebook pages are still streamed from the YouTube domain. The YouTube player is embedded within the Facebook page as an HTML “iframe”. Even though all the comments and other formats of annotations are stored on Facebook, all the video contents are still hosted by YouTube.

The Facebook use case indicates that the annotations of multimedia in the Web 2.0 era are very different from the traditional annotations when audio and video are recorded and distributed in other media, such as DVD and TV programmes. Web brings the flexibility that the annotations can be generated independent of the original multimedia resource in a massive manner. Likewise for YouTube, many user-generated annotations, such as timed-comments or transcripts, are stored external to the actual video files.

¹⁸<http://www.statisticbrain.com/facebook-statistics/>

Therefore, when publishing multimedia resources and annotations in Facebook as Linked Data, *the annotations need to be linked to the URIs that represent the real images or videos instead of the landing pages, and the hosts of multimedia resources need to make sure those URIs are dereferenceable.*

3.3.3 Use Case 9 (UC9): Publishing Media Fragments in the Synote System

Synote¹⁹ (Figure 3.3) is a Web-based multimedia annotation application, which addresses the issue of associating text-based notes or images with a media fragment through the time-line. User-generated annotations can be synchronised with multimedia resources, which are located in other repositories and referred to via URIs. The synchronisation information is saved in Synotes local database. Synote enables media fragments to be searched by indexing the text-based annotations associated with the media fragments.



FIGURE 3.3: Screenshot of Synote System

Compared with Use Cases 7 (YouTube) and 8 (Facebook), where the external developers are denied permission to change their programs directly, Synote is open source and the architecture of the application is available to be tailored for Linked Data purposes. Synote has been developed using the Grails framework²⁰ and is a Browser-Client Web application. The backend uses Hibernate²¹ Object/Relational Mapping (ORM) to re-

¹⁹<http://www.synote.org>

²⁰<http://www.grails.org>

²¹<http://www.hibernate.org>

trieve data from the database via Java domain objects. The front-end web pages are written in Groovy Server Pages (GSP). The Synote Player, which is used to display annotations, transcripts, images in a synchronised manner, is written in jQuery²².

Synote is a typical application that refers to videos from different resources and store annotations only within the application for bookmarking and searching. So it has no control of the original multimedia resources or annotations attached to those resources outside Synote. If Synote wants to contribute its data to LOD, it first needs to *find the correct audio or video URIs that the annotations within Synote will link to*. Then *different patterns can be applied to Synote to publish its annotations*, which are linked to the original audio or video URIs. It is preferable that those patterns *minimize the changes to the existing Synote system, while publish the linked datasets in efficient ways*.

It is possible to either use RDB-to-RDF services or embed Rich Snippets in Synote Player. Building an RDB-to-RDF service (using D2R server for example) is the most straightforward way to publish media fragments and nothing in the original Synote System needs to be changed. However, as much plain-text content is generated by users and speech recognition software, it is better if semantic entities can be extracted from the plain-text and linked to media fragments. RDFa can be used to implement this function. Some server-side programs need to be added to the GSP pages in order to build media fragment URIs, which should be the same as the ones published using the D2R server. The client-side Synote Player can use DBpedia spotlight API to automatically generate semantic entities based on user-generated plain-text annotations and transcripts. In this way, the content of media fragments are exposed to semantic entities, so that the semantic search will be able to link the entities to specific media fragments.

3.3.4 Use Case 10 (UC10): Publishing Media Fragments for Edina Mediahub

Edina Mediahub²³ provides a single access point for many multimedia archives, such as Independent Television Network (ITN). Some of the archives are publicly available, but some of them can only be accessed by JISC member institutions²⁴. Mediahub enables searching and exploring over 3 500 hours of items and 50 000 images from different multimedia repositories. The metadata in Mediahub is collected from different repositories using different schema. An integration service precedes operations to look-up and create links based on the metadata collected from other repositories. As Figure 3.4 shows, the links are saved in a local data store.

The links are indexed by SOLR²⁵) and exposed to search and retrieval by a URL²⁶

²²<http://www.jquery.com>

²³<http://jiscmediahub.co.uk>

²⁴<http://www.jisc-collections.ac.uk/Catalogue/FullDescription/index/1012>

²⁵<http://lucene.apache.org/solr/>

²⁶<http://www.loc.gov/standards/sru/>

(SRU) interface for searching services and Open Archives Initiative Protocol for Metadata Harvesting²⁷ (OAI-PMH) API, for external metadata harvesting services. The SOLR database can be queried by HTTP and returns an internal metadata format for the user interface. Each multimedia resource in the JISC MediaHub has a unique landing page to display the resource itself, together with the resources metadata. The actual video or audio files are also downloadable. The JISC MediaHub currently has a total of 3112 recordings containing properly defined segments (12917 segments in all), and each segment is also annotated by text, such as title and description. In order to achieve better indexing of the multimedia resources, the segments, like other kinds of metadata in the JISC MediaHub, need to be published for discovery and reuse. Currently, segment information has not been included in OAI-PMH or the SRU interface, so Linked Data are one possible means to publish the metadata and media fragments.

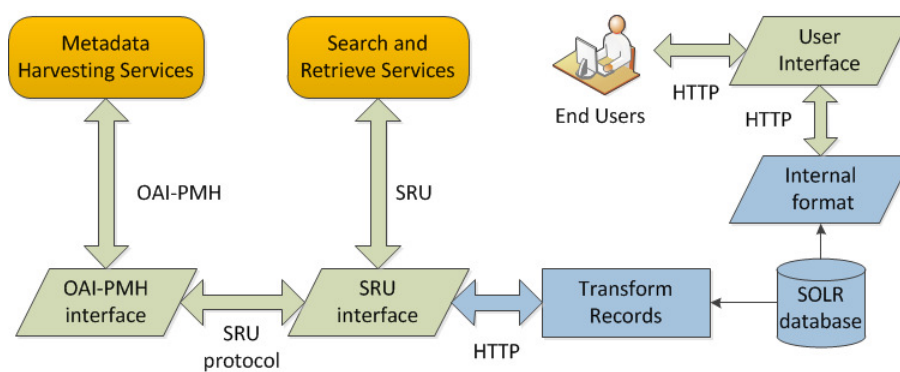


FIGURE 3.4: Edina Mediahub's architecture

Different from the use cases of Synote and YouTube, Edina Mediahub *has local storage for all the multimedia files and annotations, and each file has a unique URI to access it*. Meanwhile, Edina has landing pages for images and videos, and metadata is exposed via APIs. Some other multimedia services, such as archive.org²⁸, share a similar approach.

Generally speaking, there are two patterns for publishing metadata in Edina MediaHub: static publishing and dynamic publishing. As the archives in the JISC MediaHub seldom change and normal users are not allowed to make annotations freely, the static publishing patterns could be considered for this situation. All the metadata in the SOLR database could be dumped in XML format with media fragment information included. Then scripts could be developed to build media fragment URIs and convert the XML data into RDF format. The static RDF files could be hosted along with the JISC MediaHub application, and then metadata for new recordings need to be collected again using the script. This method will not affect the existing application at all. In addition, the new server, which provides Linked Data, could easily control the access to metadata behind proper authentication.

The dynamic publishing patterns are also applicable. A D2R server can be used to

²⁷<http://www.openarchives.org/pmh/>

²⁸<http://archive.org>

publish the metadata directly from the data store. Using a D2R server will not change the existing JISC MediaHub application. Another dynamic way of publishing is by adding media fragments to the original OAI-PMH and SRU interface. Embedding RDFa or Microformats is also possible. The video segments have already been visualised on the Web page, so the page structure will not change much.

3.4 Problem Discussion

In the previous Section, ten use cases were discussed and each use case revealed some requirements regarding linking the media fragment data to the Linked Data Cloud. This section will combine those case studies and analyse the gap between media fragments and Linked Data, revealing the key problem of why the gap is difficult to fill at the moment.

The analysis will clarify the concepts, both abstract or technological, that should be distinguished in order to eliminate confusion when talking about Linked Data principles in the media fragments field.

3.4.1 Two Types of “Server”

From the case studies, it is obvious that, compared with natural language and numeric values, multimedia resources and media fragments have a different nature, which deserves further consideration.

Not alphabetic

Unlike natural language resources, image, audio and video do not have countable discrete units, such as words, phrases and sentences. So extracting meanings of a multimedia resource directly from the binary signal is difficult. Thus, when analysing multimedia content, multimedia annotations are a very important resource to extract the semantics. (UC1, UC2, UC3, UC4, UC5 and UC6)

Multiple Dimensions

Compared with plain text, multimedia data has multiple dimensions, such as temporal and spatial. Multimedia resources can also contain many tracks, such as subtitles in different languages. (UC3, UC7 and UC8)

Various Codecs and Metadata Standards

Many codecs and metadata standards co-exist on the Web. Unlike Web standards (HTML, CSS and javascript), there is still no dominant codec (especially video codecs) that can be supported by all major user agents²⁹. The media servers

²⁹ “major user agents” means the latest version of Internet Explorer, Firefox, Google Chrome, Safari, Opera and their mobile versions.

usually only deliver the multimedia using some of the codecs and no major user agent claims that it supports all codecs. Nearly all the multimedia platforms need to deal with this problem by building an extra layer to deliver the most reasonable multimedia format to the client, such as YouTube. So the URIs of media fragments need to be format-dependent and the RDF description of the media fragments should be compatible with the major metadata standards (if not all of them) used on the Web. (UC7, UC8, UC9 and UC10)

Different Delivery and Visualisation Techniques

Unlike viewing a webpage directly, audio-visual objects are embedded in the webpage using players that are capable of decoding and controlling the replay of the multimedia file. In most video sharing applications, the original video files are not accessible as URIs and a landing page with an embedded player (Flash, Silverlight or HTML5 native player) is provided to replay the video. (UC7 and UC8)

In addition, the investigation of how the multimedia resources are indexed and shared on the Web is also important to the solution of the problem. Years before the explosion of multimedia sharing on the Web, research on media fragments focused on exposing the closed annotations, such as tracks and video segments, within the multimedia file, and server-side features of retrieving a certain time offset into the video without delivering the whole video. As a result, many standards are delivered encoding metadata into the multimedia file, such as MPEG-21 and EXIF mentioned in Chapter 2. But these efforts are inadequate for the full interlinking and indexing of media fragments in the semantic Web era, when large numbers of user-generated annotations are made available externally in Web 2.0 applications. For example, users can upload an interactive transcript to YouTube³⁰ (UC7) and then tag some person in a Facebook photo³¹ (UC8). This differs from self-contained metadata in that, when a person is tagged in a photo, an annotation is created and connected to the media fragment, but the annotation is saved in an external source (UC8 and UC9). Both kinds of annotations should be considered when applying Linked Data principles to media fragments.

It must be made clear that, strictly speaking, the term “Media Fragment URI” in Figure 3.5 actually refers to the “Media Fragment Identifier” on the **Multimedia Host Server**, not the “Landing page identifier” on the **Player Server** in Figure 3.5. However, if the landing page is all about an audio or video, i.e. the landing page represents and only represents the audio or video resource, the landing page identifier can include a fragment that represents a certain part of the actual audio or video.

For example, the following YouTube video URI

<http://www.youtube.com/watch?v=Wm15rvkifPc>

³⁰<http://goo.gl/tinMj>

³¹<http://www.facebook.com/help/photos/tag-photos>

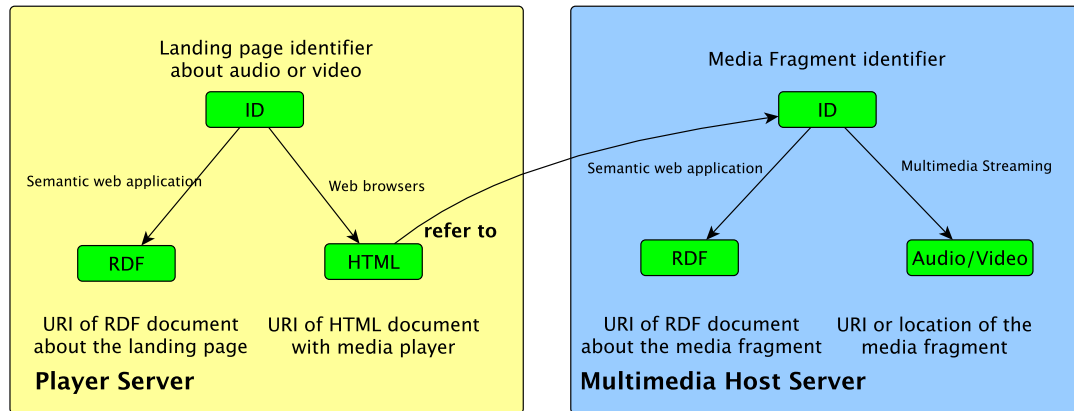


FIGURE 3.5: The relationship between Player Server and Multimedia Host Server

is a landing page and the following URI

`http://www.youtube.com/watch?v=Wm15rvkifPc#t=120`

is a landing page identifier representing the video from 120 seconds onwards. As the real video file is hidden from direct access by YouTube, the above landing page is treated as the YouTube video itself. Thus the URI with $t=120$ can be considered as a Media Fragment URI in a broader sense. Knowing these strict and broader definitions of “Media Fragment URI” is important to understand the next discussions.

Whereas in the traditional video/audio production process, video/audio resources are edited by a group of people and distributed in a controlled manner, in the Facebook and Synote use cases, making multimedia annotations on the Web has become more collaborative, with the sharing of video/audio resources in different applications. Multimedia resources could be hosted by one server, but referred to by some HTML document (also called “replay page” or “landing page”) on another server. For example, users can share a YouTube video (The Multimedia Host Server) on their Facebook timeline (Player Server). Even though, from a technical point of view, an embedded inner page (iframe in HTML) is used as the media of interaction, users still think they are on Facebook instead of YouTube. When they comment on the video, the comments are saved in the Facebook domain instead of the YouTube domain. According to a survey conducted by socialbakers³², between 22nd and 27th, January 2013, a group of users shared 3684 YouTube videos on Facebook, while they only uploaded 458 videos as native Facebook videos. This implies that most page administrators in Facebook would prefer to share YouTube videos on Facebook instead of uploading native Facebook videos. So when users make comments about a YouTube video, those user-generated annotations are saved separately from the servers which host the multimedia files. The similar case also

³²<http://goo.gl/2d0Zx>

exists in Twitter (use cases in Section 3.1.1), where the tweet text can be considered as an annotation of the media fragments (UC1).

Usually, the landing page is where the end users watch the video from a user agent (UA), instead of typing the URL of a video file directly into the address bar of the UA and watching the video there. No matter whether the landing page is within or outside the host domain of the multimedia file, the multimedia file is replayed in a context, where title, keywords, comments and interactive transcript are displayed on the same page as the multimedia. The landing page also controls the audio-visual representation of the media fragments via the embedded player to provide the interactive transcript function.

So the semantic annotations of media fragments could come from two types of server (Figure 3.5): (1) **Multimedia Host Server**: The applications, which offer multimedia streaming service, publish media fragments with metadata they have. The server has to make sure the RDF description of media fragments is accessible by semantic Web applications; (2) **Player Server**: The applications, which embed multimedia players, refer to the media fragments on a **Multimedia Host Server** and interlink them with their own annotations. Usually, the annotations generated by users from **Player Server** are different from the lower-level multimedia metadata and reflect users' interests in different areas. **Player Server** is actually interlinking annotations to the media fragments in another place, but instead of providing an RDF description for the multimedia file itself, they need to provide their own RDF representation for the annotations hosted on their own servers.

It must be emphasised that the **Multimedia Host Server** and the **Player Server** are not necessarily in different domains. For example, YouTube has its own video repository but it also has a replay page which embeds the video and allows users to make annotations. However, if the **Player Server** and the **Multimedia Host Server** are in different domains, the **Player Server** will have no control of what could be dereferenced via the media fragment identifier. The data published on both servers can be about the same property of a multimedia resource. An mp4 file can embed the title information using MPEG-7, but when the mp4 file is played in another application, it may be given another title. This is acceptable because vocabularies describing multimedia resources, such as W3C-MA, do not require that all the metadata come from the file itself. For example, the subtitles in W3C-MA could be an external link of the subtitle file or a track media fragment embedded in the file³³.

From the discussion above, it is safe to conclude that applying Linked Data principles to media fragments must consider both the **Player Server** and the **Multimedia Host Server** for the requests from both traditional user agent and semantic Web applications. This is the real condition that multimedia resources are shared and annotated on the Web, which is different from the producing and distribution of multimedia in other

³³<http://www.w3.org/TR/mediaont-10/#example2>

domains such as TV and film.

3.4.2 Two Different Concepts of “Video”

As Figure 3.5 shows, a multimedia resource usually presents itself on a **Player Server** while the binary data is delivered from a **Multimedia Host Server**. However, when people refer to a video on the Web, they are usually unaware of the difference between a “video object” and the “location of the video”. The Media Fragment Identifier

`http://www.example.com/test.ogv#t=5,12`

is actually an information resource, i.e. a real electronic file accessible by the URL, instead of a “video object”. In Linked Data, there is a common “misunderstanding” between, for example, a URI referring to a webpage about a person and a URI referring to a physical person. Furthermore, a physical person can have more than one webpage about himself or herself on the Web. In this sense, the URL of the location of the video is only a representation of a video object on the Web and one “video object” theoretically can have many representations on the Web with different codecs, bitrates and resolutions. Section 4.1.3 of Heath and Bizer (2011) and Section 4 of Sauermann and Cyganiak (2008) suggest that the Linked Data application need to prepare at least three URIs:

- a URI for the real-world object itself. For example, the URI for Yunjia Li as a person at the University of Southampton is:
`http://id.ecs.soton.ac.uk/person/12034`
- a URI for a related information resource that describes the real-world object and has an HTML representation. For example, Yunjia Li’s home page in HTML is:
`http://www.ecs.soton.ac.uk/people/yl2`
- a URI for a related information resource that describes the real-world object and has an RDF/XML representation. For example, the RDF representation of Yunjia Li is:
`http://rdf.ecs.soton.ac.uk/person/12034`

The multimedia file on the Web is not a “real-world” object and neither does it have an HTML representation. As discussed in UC7 and UC8, many user generated annotations are meant to be linked to the real video files, instead of the landing pages, but the real video URIs are not always available, especially for video streaming services. At the same time, a video can be reused in many different places, i.e. it can be embedded in many landing pages with annotations associated to it from each landing page. So a best practice or standard solution should be proposed in the Interlinking Media Fragments

Principles to distinguish the original video with the reuses, meanwhile preserving the annotations to the real video instead of the landing pages.

3.4.3 Three cases of “Annotations”

Figure 3.5 shows that both the **Multimedia Host Server** and the **Player Server** can contain annotations related to media fragments. According to the third Linked Data principle, when a HTTP URI is dereferenced, the description of the resource needs to be provided using RDF. Since the use of a video/audio resource is divided into two stages (hosting and referencing) in the real world as shown in Figure 3.5, it is necessary to investigate what RDF descriptions can be provided by the **Multimedia Host Server** and the **Player Server**. Generally, there are three cases of “Annotations”, depending on whether the original video/audio files are publicly accessible.

- Case 1: Annotations for video/audio files on the **Multimedia Host Server**
- Case 2: RDF description for video/audio objects on the **Player Server** when the referred video/audio source is NOT available
- Case 3: RDF description for video/audio objects on the **Player Server** when the referred video/audio source is available

If the **Multimedia Host Server** is only a multimedia streaming server, all the description it can provide about the video/audio resource is the metadata or whatever data that has been embedded in the resource itself, unless the video/audio resources are further processed to extract more information from it (this will be discussed in Section 4.4).

Currently, as most **Multimedia Host Servers** focus on the streaming of multimedia resources, they may not be able to handle media fragments (as discussed in Section 4.1), or provide RDF descriptions for media fragments and make them dereferenceable. However, those RDF descriptions are still useful as some of the information is initially embedded in the multimedia resource itself and can hardly be obtained from anywhere else, such as the copyright information. Another useful use case of providing RDF description for media fragments is adaptive video streaming based on reasoning on the RDF descriptions, which has been implemented in NinSuna (Van Lancker et al., 2013). With the RDF descriptions of the technical properties of the videos, such as framesize and framerate, and the profile of the requesting client, the NinSuna server can make decisions on which video format and which media fragments should be delivered.

On the **Player Server**, annotations of the multimedia resources can be provided based on the information on the landing pages, i.e. the information hosted on their own domains. For example, the landing page of YouTube video displays all the information about the video, such as its title, description, tags, comments and related videos.

Something in common between Cases 1 and 2 is that both of them are the gateway to access the video/audio resources host in their own domains and they are supposed to be the primary resources to provide RDF descriptions about media fragments and make them dereferenceable. However, many consumers of the video/audio resources are likely to be the ones who reuse those resources. For example, one can embed a video in a Tweet and add some comments on the video (UC1). The use case of Synote (UC9) also shows that one can reuse or refer to the video in different domains and generate further annotations from a third-party application. The annotations generated in this reuse process may have some overlaps and will be domain specific. For example, example.com hosts a video animation *http://www.example.com/bigbuck.mp4*. Children love the characters in the animation and make some annotations on the characters in example2.com, where the *bigbuck.mp4* is referred. On another application example3.com, animation designers may refer to this video as an example of animation production. At the same time, both example2.com and example3.com can create their own metadata, such as the title, tags and descriptions, of the video. So, in this sense, the annotations in **Case 3** is a combination of **Case 1** and **Case 2**.

The discussion in this section revealed three key aspects of the current status of multimedia distribution on the Web. The multimedia resources can be hosted in one domain and get reused in other domains. During this process, at least two kinds of server (Multimedia Host Server and Player Server) are involved, and both of them can provide annotations for media fragments. Furthermore, the differences between video files and the landing pages should be distinguished in order that the annotations are correctly linked to the video files. Those key aspects will be reflected in the summary of requirements in the next section and will affect the discussions of Interlinking Media Fragments Principles in Chapter 4

3.5 Summary of Requirements

As has been listed in Section 2.2.1, the four Linked Data principles are:

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names
3. When someone looks up a URI, provide useful information using the standards such as RDF and SPARQL
4. Include links to other URIs, so that they can discover more things

Combining the use case studies with the problem discussion in Section 3.4, the requirements for publishing media fragments into the Linked Data cloud can be summarised into the following seven specifics.

R1 A media fragment must be named by a URI and the URI:

- R1.1: must be able to encode different dimensions in different formats (UC1)
- R1.2: must be independent of the multimedia file format (UC7, UC8 and UC9)
- R1.3: syntax should be easy to be implemented and parsed (UC1)

R2 The URI of the actual multimedia file and the URI of the landing page for the multimedia file should be distinguished, and both URIs need to be dereferenceable (UC7, UC8, UC9 and UC10)

R3 Media fragments described with the following information in RDF:

- R3.1: The detailed value of each dimension (UC2, UC3)
- R3.2: Domain-specific vocabularies used to contextualise media fragments if necessary (UC4, UC5 and UC6)
- R3.3: Annotations must be linked to the URIs that represent the real multimedia resources instead of the landing pages (UC7 and UC8)
- R3.4: Annotations must be linked to the URIs that represent media fragments instead of the whole multimedia resource (UC2 and UC3)
- R3.5: The publication of annotations should consider both cases of Multimedia Host Server and Player Server, making sure both of them can publish their own annotations.

R4 The published media fragments and annotations need to be further processed and interlinked with external resources on the Web (UC4, UC5 and UC6)

R5 When publishing media fragments and annotations, an extra layer should be added for the existing systems. This layer should:

- R5.1: enhance the reuse of media fragments
- R5.2: make sure of backward compatibility, i.e. using the media fragment URI should not break the functionality of the old system (UC1)
- R5.3: permit a system that is not media fragment ready to discard the fragment information encoded in the URI (UC1)
- R5.4: use publishing patterns to automate this publishing process (UC9 and UC10)

R6 When a media fragment URI is opened in a browser, the media fragment should be displayed properly, including:

- R6.1: highlighting the dimensions encoded in a media fragment URI by the media fragment player (UC1)

- R6.2: highlighting directly the media fragment and annotations, if any, when opening the URI in the browser (UC1)

R7 It should be possible to index media fragments through the annotations associated with them (UC1, UC2)

Some of the requirements, especially the requirements regarding fragment definition (R1), URI dereferencing (R2), treating the URI as an anchor to annotations (R3), media fragment visualisation (R6) and searching (R7), have already been included in various places, such as “Use cases and requirements for Media Fragments”³⁴ and the use case discussion within W3C Media Fragment Working Group³⁵. However, the working group focuses more on providing solutions for R1 and R2, leaving more research needed for the rest of the requirements. Another uniqueness of these requirements compared with other standards and implementations is that Section 3.4 analyses in depth the common misunderstandings of the concepts related to publishing media fragments. The distinguishing of those concepts in a systematic way was the key leading to the requirements developed in this chapter.

Of course, as a precondition to realise all the requirements, especially R3 to R7, the raw annotations must exist beforehand and should be time-aligned with media fragments in order to publish them with the media fragments and so benefit searching. How to generate such annotations and align them with media fragments is out of the scope of this thesis.

In all those requirements, R1 to R4 are directly linked to the Linked Data principles and they are the main obstacles to applying Linked Data principles to media fragments and annotations. R5 is a non-functional requirement that ensures backward compatibility when bringing media fragments onto existing multimedia platforms. There might be varieties of solution for R1 to R4; however, R5 is a measure of whether the solution is practical for current multimedia platforms. R6 and R7 focus on the visual representation and searching functions related to media fragments respectively.

The dependencies among R1 to R7 are illustrated in Figure 3.6. The choosing of URI and dereferencing depends on R5, which means the solution of R1 and R2 should minimise changes to existing systems and make media fragments integration seamless. R3, particularly R3.3 and R3.5, will depend on R1 and R2, in that the video URIs on the Multimedia Host Server and the Player Server need to be distinguished in the RDF description of media fragments. When R3 and R4 are ready, i.e. media fragments and annotations are published as Linked Data, it is possible to further visualise them and index them for searching.

³⁴<http://www.w3.org/TR/media-frags-reqs/>

³⁵http://www.w3.org/2008/WebVideo/Fragments/wiki/Use_Cases_Discussion

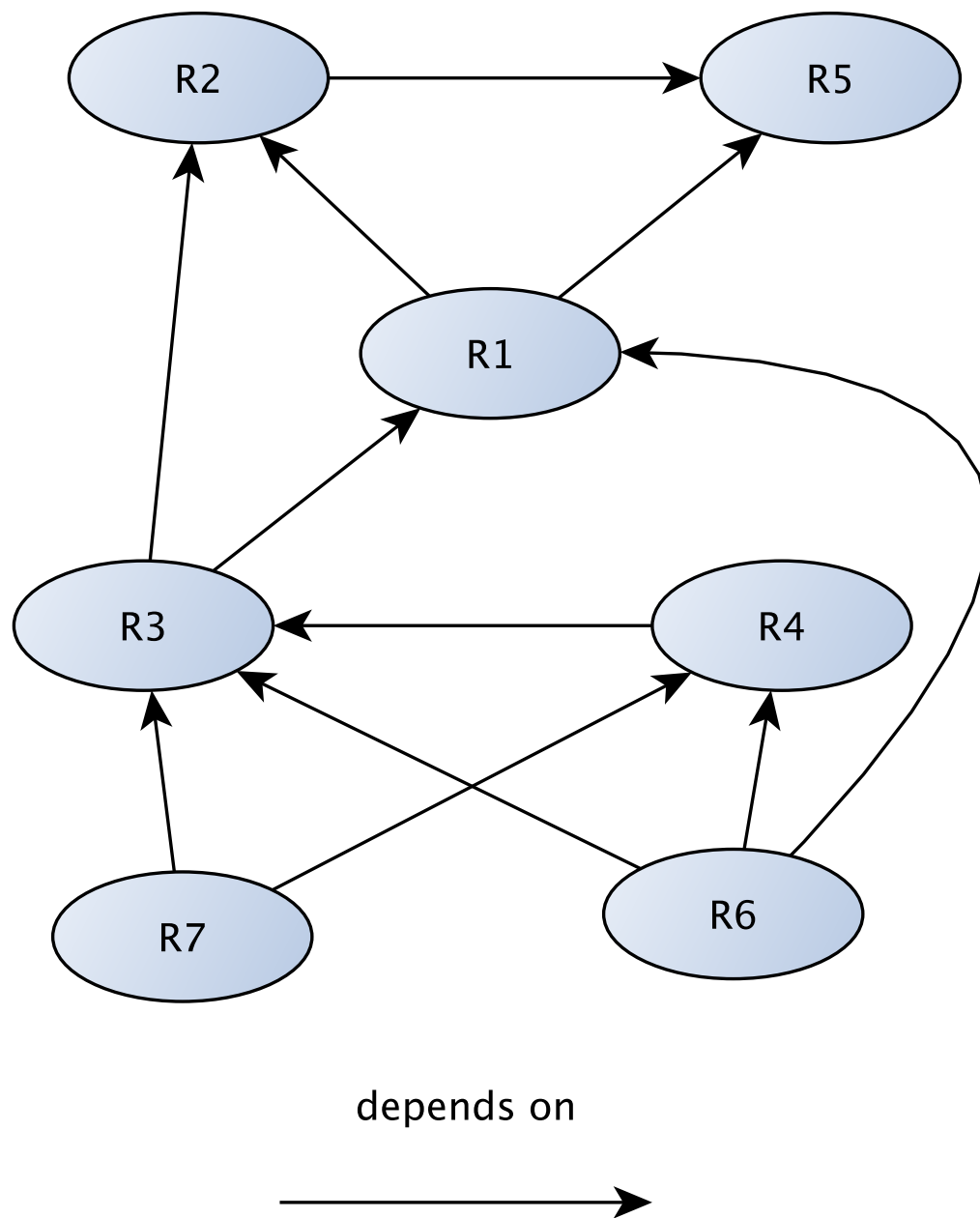


FIGURE 3.6: The dependencies among requirements

The research questions specified in Chapter 1 are also closely related to the solutions to the requirements, as demonstrated in

The rest of this thesis will follow the requirements identified in this Chapter and discuss them one by one towards recommended solutions. Thus R1 to R7 outlines the structure of the rest of the thesis.

TABLE 3.1: Relationships between requirements and research questions

Research Question	Requirements
Q1	R1, R2, R3, R4 and R5
Q2.1	R4 and R7
Q2.2	R6
Q2.3	R4 and R7
Q2.4	R4

Chapter 4

Interlinking Media Fragments Principles

In the previous chapter, several requirements were proposed to bridge media fragments and the semantic Web. This chapter studies the requirements directly related to the Linked Data principles, i.e. R1 to R5, and reveals the essential principles for publishing media fragments to the Linked Open Data Cloud. No matter how different media fragments are, the principles should follow the same general guidelines of Linked Data principles. Applying those four principles to the media fragment domain leads to the following questions.

1. Is there a format-independent syntax to encode different dimensions of a multimedia resource in the URI? (R1)?
2. How can the URIs of media fragments be dereferenced, e.g. is the normal practice of URI dereferencing (Berrueta and Phipps, 2008) still working? (R2)
3. What information needs to be provided as the RDF description of a media fragment URI? (R3)
4. What external links are needed in the RDF description of a media fragment URI, and how can these links be created? (R4)

Publishing media fragments as Linked Data will enable semantic-aware agents on the Web to index, retrieve and query media fragments, and therefore improve the online presence of media fragments. Linked Data is a highly scalable mechanism for interlinking knowledge on the Web, so there are growing research interests in applying Linked Data principles to media fragments (Hausenblas et al., 2009a). To many projects, such as

LinkedTV¹ and MediaMixer², media fragments have been considered as the core function to link multimedia resources to the rest of the Web. Despite all of this research work, there is still no guideline similar to the Linked Data principles that systematically summarises the best practice for media fragment publishing.

Thus, the discussion in this chapter summarises the issues that need to be considered and the best practice that should be implemented when publishing media fragments. Sections 4.1, 4.2 and 4.3 present a discussion about the solutions to R1, R2 and R3. The discussion refers to the use cases of Chapter 3. In Section 4.4, summarises several publishing patterns on applying interlinking media fragments principles to existing web applications as the means for interlinking media fragments. The media fragments publishing patterns extend the Linked Data publishing patterns (Figure 2.3) and mainly build extra layers on top of the existing frameworks (R5).

4.1 Choosing URIs for Media Fragments

This section will look at the problems of choosing URIs to encode different dimensions of media fragments. The main consideration is whether the URI can be successfully dereferenced by existing multimedia systems if they have not built media fragments as part of their functionality (R2 and R5).

The R1 in Section 3.5 indicates the media fragment URI:

- R1.1: must be able to encode different dimensions in different formats (UC1)
- R1.2: must be independent of the multimedia file format (UC7, UC8 and UC9)
- R1.3: syntax should be easy to implement and parse (UC1)

There are many standards targeting different dimensions of media fragments (see Section 2.3). For R1.1, one can theoretically choose any string combination, and it is better to reuse the existing standards if possible, such as MPEG-21 and W3C-MFURI. Some standards are also format independent and easily parsed (R1.2 and R1.3). The problem of choosing URIs for media fragments is actually which mechanism should be used to encode them: hash, slash or query. Whichever URI encoding is chosen, it must be dereferenced correctly (R2) and it must retain backward compatibility with the existing system (R5).

Whether hash, slash or query is chosen to encode media fragments, the media fragments should be delivered by at least both servers. Thus, in the **Player Server**'s HTML page, i.e. user interface, the media fragment URI should be correctly requested and

¹<http://linkedtv.eu>

²<http://mediamixer.eu>

visually highlighted by the multimedia player. Given the same URI, the **Multimedia Host Server** must be able to stream the fragment to the **Player Server**. To do so successfully, the **Player Server** and **Multimedia Host Server** must agree on the URI encoding of temporal, spatial and other dimensions. For example, the “deep linking” function of YouTube allows users to get the URL for a certain time point of the video, and all of the following four URLs are acceptable:

```
http://www.youtube.com/watch?v=Wm15rvkifPc#t=120
http://www.youtube.com/watch?v=Wm15rvkifPc&t=120
http://www.youtube.com/watch?v=Wm15rvkifPc#t=1h9m20s
http://www.youtube.com/watch?v=Wm15rvkifPc&t=1h9m20s
```

LISTING 4.1: Examples of YouTube Deep Linking URIs

When users attach “*t* =” at the back of the landing page URL as a parameter or URL hash, the YouTube player will send the request to the multimedia host for the byte ranges according to the specified time³. On receiving the request, the YouTube multimedia host server will then deliver the video via HTTP Live Streaming. On the landing page, the YouTube player will start playing the video from that time point.

In this example, both YouTube player and the multimedia host server are actually in the same domain, or at least controlled by the same agent. If the services are not further divided into many sub-components, YouTube as a whole is in charge of developing the YouTube player, how the videos are hosted and how the player communicates with the YouTube video host server. Theoretically, as long as the **Player Server** and **Multimedia Host Server** are agreed on the media fragment syntax in the URI, any valid URI could be chosen to represent media fragments and the owners of the application are free to ignore any media fragment dimension. Applications like Facebook (see UC8 in Section 3.3.2) allow users to share videos from different resources, such as YouTube, Dailymotion and Vimeo, but a Flash player will be first delivered from those applications and embedded in an iframe in the page, so actually the **Player Server** agrees with the **Multimedia Host Server** and is still in the same domain.

In many other cases, however, the **Player Server** and the **Multimedia Host Server** are not controlled by the same agent and the choice of URI may affect whether the multimedia resource can be delivered or not. As an example, Synote does not host any video resources. Instead, it uses an HTML5 native player to embed the video specified by a URL and users can make annotations, which are stored on the Synote server. So Synote is a **Player Server**, but not a **Multimedia Host Server**. If Synote wants to replay the following video from the 100th second:

`http://example.org/test.mp4`

there are many syntaxes that could encode the fragment information. For instance:

³In reality, the bytes delivered to the player may not be the exact time as requested, and a couple of more requests will be sent simultaneously for some successor byte packages as cache.

```

http://example.org/test.mp4#time=100
http://example.org/test.mp4?t=100
http://example.org/test.mp4/100/
http://example.org/test.mp4?randomparam=1m40s

```

LISTING 4.2: Example Media Fragment URIs

Ideally, Synote as a **Player Server** would ideally agree the media fragment syntax with the **Multimedia Host Server** *example.org* for the media fragment syntax, as well as how to set up the session for media fragment delivery. On parsing the media fragment URIs, the Synote player and *example.org*. On parsing the media fragment URIs, Synote player can map the encoded start time to a certain byte range in the request just as YouTube does. Or the start time can be directly passed to the host server as a parameter and then mapped to a sequence of bytes and delivered to the Synote player. Unfortunately, unfortunately, neither of these is realistic for applications like Synote. The Synote player is not designed for a specific **Multimedia Host Server**, i.e. it is an HTML native player, so it is impossible to make it compatible with all **Multimedia Host Servers**, which may define different ways of requesting media fragments.

One solution to this problem is for the program to extract the start time from the media fragment URIs on the client side, so that the player still requests for *test.mp4* without any fragment. The multimedia host server in this case does not need to deal with any media fragment parameters in the request and the player can jump to the start time when necessary cache has been downloaded from *example.org*. This solution works very well when Linked Data is not considered (Li et al., 2012), but when the **Multimedia Host Server** needs to publish media fragments as Linked Data, some of the media fragment encodings will cause problems. For those URIs, the questions are whether they are good URIs for Linked Data and whether they are dereferenceable.

Standards such as MPEG-21 and CMML, already used a single URI to identify media fragments. But in the background of choosing URIs for Linked Data use, the mechanisms have to be re-considered and evaluated for their usability in real applications. According to the URI definition in RFC3986 (Berners-Lee et al., 2005) and “Cool URIs for the Semantic Web” (Sauermann and Cyganiak, 2008), there are three valid ways to add fragment information into URIs: URI query, slash namespace, URI hash fragment (hash namespace). (Rodriguez, 2008) pointed out that URI query should be avoided as far as possible in RESTful Web services. Since Linked Data has a RESTful nature, the query string is not widely used to identify a resource in RDF. In addition, the term media fragments literally means part of the multimedia resource, so the URI query, which returns a completely new resource from the server and loses the affiliation between parent and children resources (Troncy et al., 2012), is not very suitable for use as “media fragments”. In this sense, only hash and slash URIs should be considered when choosing URIs encoding for media fragments.

W3C has given a summary of Hash vs. Slash in the semantic Web context⁴:

Advantages of using hash: (1) The information is easy to publish using an editor for RDF files. With slash, the server may need to be set up to do a 303 redirect from the URI for the thing to the URI for the document about it. This involves control of the web server which many people do not have, or have not learned to do. (2) Run time speed: The client looking up the URI just strips of the right part, and performs a single access to get the document about whatever it is. This will in many cases also give information about other related things, with URIs starting with the same document URI. Further fetches will not be necessary at run time.

Advantages of using slash: With hash, one document, whose URI is the bit up to the hash, has to contain information of all the things whose URIs are the same before the hash. For a hand-written ontology or data file this is fine, but for a machine-generated system it could be too large.

In the media fragment context, the slash namespace has several merits. Firstly, each annotation will be able to (but not necessarily) have an individual RDF representation (see Recipe 5 in Berrueta and Phipps (2008)), which is flexible in that necessary HTTP 3XX redirection can be configured on the server side for each annotation. Secondly, the server side can easily obtain the parameters indicating the temporal and spatial dimensions of the media fragment, and return the corresponding byte ranges. However, one big concern of using slash namespace is that, if *example.org* originally only serves *test.mp4* and cannot handle a request for *test.mp4/100/*, a “404 not available” response will be returned when the RDF representation is requested by any client agent. If developers have control of the **Multimedia Host Server**, they can implement the function to handle the request from a semantic Web application. But for applications like Synote, it is not realistic to expect all the **Multimedia Host Servers** to implement that function. The choice of URI hash will not have this problem as the hash fragments will not be passed to the server at all. Thus, **Multimedia Host Server** does not need to add any programme to deal with this information.

The hash URI method is the one adopted by W3C Media Fragment URI (basic) (Troncy et al., 2012) and MPEG-21. Compared with W3C-MFURI, MPEG-21 is more expressive. For example, it defines hash syntax to retrieve moving spatial regions on the video and the offset function can select a range of bytes in the video. However, MPEG-21 is still only applicable to MPEG media (a conflict with R1.2), whose MIMETypes are audio/mpeg, video/mpeg, video/mp4, audio/mp4, and application/mp4. In addition, the expressiveness compromises the uptake of the standard and no MPEG implementation fully realises this fragment addressing scheme⁵ (R1.3). In W3C-MFURI, different

⁴<http://www.w3.org/wiki/HashVsSlash>

⁵<http://www.w3.org/2008/WebVideo/Fragments/wiki/MPEG-21>

dimensions in multimedia are expressed in simple URI fragments and there is no strict limitation on the media format that is applicable to this standard.

Nor is hash URI a perfect solution and several weaknesses have been pointed out in Hausenblas et al. (2009a). The first problem is that the semantics of URI fragments for most multimedia formats are undefined⁶. According to Jacobs and Walsh (2004), new registrations must be done in the IANA URI scheme so that the semantics of a fragment for each media type can be clarified for UAs. An experiment conducted by the Media Fragment Working Group has shown that there is “no single media type in the audio/*, image/*, video/* branches that is defining fragments or fragment semantics along with it”. Therefore, when the hash URI of a media fragment is typed into the address bar of the UAs, the media fragment cannot be understood just as URI fragments in an HTML document, and the manipulation of media fragments (such as parsing and visualisation of media fragments) can only be processed with the assistance of UA plugins or client-side scripts (javascript). To highlight a URI fragment in an HTML document, the UA can scroll to the HTML element identified by the URI fragment, which has been implemented in most UAs. However, there is no universal standard yet to tell the UA how to highlight different dimensions encoded in the URI fragment if the media type is image, audio or video. Currently, the temporal fragment only is supported by a few browsers⁷.

The second problem of hash URI is that, when dereferencing the URI, the RDF description will contain information for all the things about the URI preceding the hash identifier. If the video/audio has many fragments, but the requester only wants to see the triples about a certain media fragment, a large amount of unwanted data will be dereferenced together, i.e. triples about other fragments. It is against best practice that “hash URIs should be preferred for rather small and stable sets of resource”.

From the Linked Data point of view, solutions for choosing URIs are acceptable provided that the media fragment can be universally identified by the URI. However, the encoding of URIs will affect the way the URI could be dereferenced. URI query and slash namespaces would be more convenient for servers to get fragment information and return a suitable media fragment as well as RDF representation. But the **Player Server** could not simply attach a slash or query string at the back of the multimedia file URI unless the **Multimedia Host Server** made it valid (conflict to R5). made it valid (conflict with R5). In this condition, the hash URI more appropriately follows Linked Data principles. For semantic applications using the hash URI, the media fragment is universally identified and since the **Multimedia Host Server** will never receive the string in the URI hash, and will not throw an error for dereferencing the RDF representation of the media fragment URI. For the traditional agent, it is the UAs’ (or the application developers) responsibility to present the media fragment in a sensible way.

⁶<http://www.iana.org/assignments/uri-schemes.html>

⁷<http://www.w3.org/2008/WebVideo/Fragments/wiki/Showcase>

The UA, if it could not understand the semantics of the fragment, will just ignore the hash fragment and no further error will be signalled. So for both agents, the **Player Server** can add media fragments into its system without worrying about whether the **Multimedia Host Server** can handle media fragments or not.

From this discussion, the first principle of interlinking media fragments about choosing URIs can be summarised as:

Use hash URI to encode media fragment information into the URIs. The syntax defined in the W3C Media Fragment URI is recommended.

This principle can successfully satisfy R1, while it is still compatible with R2 and R5 if hash URI is chosen as the encoding mechanism.

4.2 Dereferencing URIs for Media Fragment

This section will discuss how R2, the URI of the actual multimedia file and the URI of the landing page for the multimedia file should be distinguished and both of them dereferenceable, can be satisfied using the recommended W3C-MFURI. There is a dilemma that sometimes the landing page is the only URI that can be used to refer to the multimedia resource, but the RDF representation of a landing page is not equal to the representation of the multimedia resource itself. To find a solution to this problem, it is important to understand that the dereferencing of the landing page URI and the media fragment URI are two separate processes. So it is necessary to examine both dereferencing processes separately and then investigate if they are connected to each other. The discussion in this sub-section assumes that the media fragment URIs follow the syntax defined in W3C-MFURI.

The landing page of a multimedia resource is a normal information resource (HTML page), which provides higher-level annotations about the multimedia resource, such as description, tags and comments. Technically, there should not be any problem in dereferencing content applicable to the landing page URI, i.e. the landing page is the HTML representation of the landing page URI. The **Player Server** needs to provide an RDF representation for the landing page URI following best practice such as using 303 redirect, content negotiation and RDFa mentioned in “Cool URIs” (Sauermann and Cyganiak, 2008). So the main problem is how to dereference the Media Fragment Identifier to get both RDF and video/audio stream data from the **Multimedia Host Server**.

According to the Web architecture (Jacobs and Walsh, 2004), a representation is a stream of bytes in a particular format, such as HTML, PNG and RDF/XML. A single resource can have different representations. This process can be done through content negotiation, where `Accept: application/rdf+xml` or `Accept: text/html` can distinguish

the response as an RDF or HTML document. For the **Multimedia Host Server**, the Media Fragment Identifier actually points to a real file and the content negotiation may happen between requests for audio/video bytes and the RDF representation.

The proposed way of processing the byte stream in Protocol for Media Fragments 1.0 Resolution in HTTP (Troncy et al., 2011) is to map the timeline with the byte stream with the help of “smart enough” UAs and servers, so that only the byte ranges corresponding to the media fragment will be returned (Van Deursen et al., 2010). Compared with requesting an HTML page, the Media Fragment 1.0 Resolution⁸ introduces the use of a Range header in the HTTP request in several cases. When the UA is capable of mapping the time fragment into byte ranges, the UA will include the Range header indicating the byte ranges it wants the server to deliver⁹. For example:

```
GET /test.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: bytes=10293-20334
```

The server then will respond 206 Partial Content with a Content-Range header encoding the real byte ranges that will be delivered and the total bytes of the video file:

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes
Content-Length: 10151
Content-Type: video/ogg
Content-Range: bytes 10283-20434/256124390
Etag: "a2d60-13c9246-21d2467012568"

{binary data here}
```

If the UA cannot map the time fragment to byte ranges, it will rely on the server to choose corresponding to the time fragment. In this case, the UA needs to use the Range header to encode the time fragment. For example, if the UA requests 5 to 12 seconds of the video, the request may look like this:

```
GET /test.ogv HTTP/1.1
Host: www.example.com
Accept: video/*
Range: t:npt=5-12
```

⁸<http://www.w3.org/TR/2011/WD-media-frags-recipes-20111201/>

⁹<http://www.w3.org/TR/media-frags-recipes/#processing-protocol-UA-mapped-new>

The server will do the mapping the return the binary data, including a Content-Range-Mapping header to explain how the requested time fragment is mapped to the byte ranges:

```
HTTP/1.1 206 Partial Content
Accept-Ranges: bytes, t, id
Content-Length: 10151
Content-Type: video/ogg
Content-Range: bytes 10283-20434/256124390
Content-Range-Mapping: {t:npt 4.95-12.33/0.0-234.23}={bytes 10283-20434/256124390}
Etag: "a2d60-13c9246-21d2467012568"

{binary data here}
```

Those two basic examples show how HTTP request and response are used to deliver media fragments. However, most current multimedia servers are not compatible with W3C-MFURI and the proposed Media Fragment Resolution. So, both client UAs and servers need to be upgraded in order to make them compatible with those standards (R5). NinSuna is a server-side implementation of the proposed media fragment resolution, but all the media resources need to be broken down into units which can be mapped to byte ranges (Deursen et al., 2009). Since the first principle of interlinking media fragments suggests the adoption of hash URI to encode media fragment information, the URI will be independent of the server-side compatibility of media fragments. The solutions to dereferencing such URIs should also consider both **Multimedia Host Servers** that are compatible and incompatible with media fragments.

In the HTTP request from a UA in the two examples above, it is clear that the main differences between the two requests are: (1) the content negotiation for a video or audio; (2) a Range header with requested byte ranges or time fragment. The Media Fragment Resolution also defines many other cases of servers' responses involving caching and codec setup. However, they are only related to the delivery of the media fragment, and what Linked Data principles focus on is the dereferencing of the semantic representation and those two differences will not interfere with the dereferencing of the RDF representation of an audio or video file. Even though the **Multimedia Host Server** is not compatible with media fragments, it can still be easily configured with content-negotiation and 303 redirect to make the media fragment URI dereferenceable. For example, a UA that is not able to map time fragments to byte ranges can send this request to the server:

```
GET /test.ogv HTTP/1.1
Host: www.example.com
Accept: application/rdf+xml
```

Range: t:npt=5-12

Based on the content negotiation, the **Multimedia Host Server** can configure a 303 redirect to an RDF file describing *test.ogv* (Figure 4.1(a)) and then requesting the RDF file location from the UA can lead to the following response:

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 2034
Content-Type: application/rdf+xml
Content-Location: http://www.example.com/test.rdf
Vary: accept
```

The Range header in the request will be simply ignored if the server cannot handle it. So technically, on **Multimedia Host Server**, the common practice, such as content negotiation and 303 redirect, is still applicable to dereference media fragment URIs. However, Figure 4.1(a) looks weird considering that both *http://www.example.com/test.ogv* and *http://www.example.com/test.rdf* are the representation of the same information resource *http://www.example.com/test.ogv*. Hausenblas et al. (2009b) argued that using content negotiation here was simply wrong because the video file and the RDF representation cannot convey the same information, i.e. the RDF representation cannot describe everything that is included in the video file. Figure 4.1(a) is not aligned with the best practices of “Cool URIs” (Sauermann and Cyganiak, 2008) either.

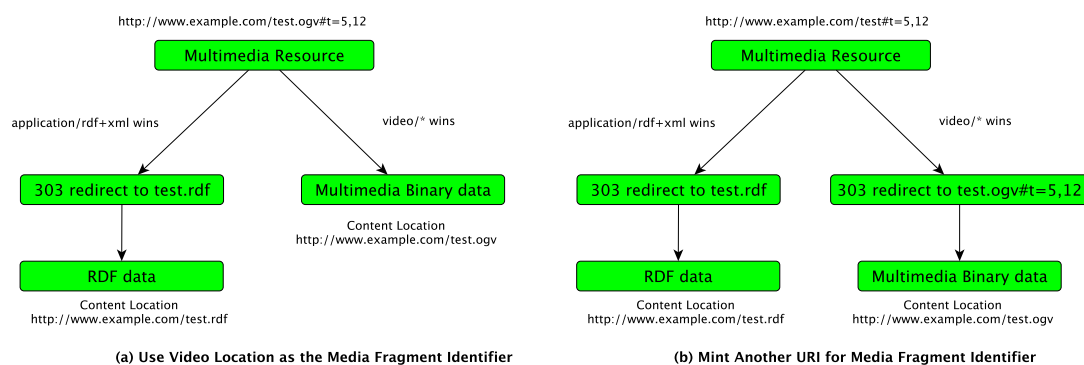


FIGURE 4.1: Use Content Negotiation and 303 Redirect to dereference Media Fragments

So the main problem in Figure 4.1(a) is that when the video/audio resources are watched on the Web, users are usually unaware of the difference between a “video object” and the “location of the video”. The Media Fragment Identifier:

http://www.example.com/test.ogv#t=5,12

is actually an information resource, i.e. a real electronic file accessible by the URL, instead of a “video object”. In Linked Data, there is a common “misunderstanding” between, for example, a URI referring to a webpage about a person and a URI referring to a physical person. Furthermore, a physical person can have more than one webpage about him or herself on the Web. In this sense, the URL of the location of the video is only a representation of a “video object” on the Web and one “video object” theoretically can have many representations on the Web with different codecs, bitrate and resolutions. Heath and Bizer (2011) and Sauermann and Cyganiak (2008) both suggest that the Linked Data application needs to prepare at least three URIs:

- a URI for the real-world object itself. For example, the URI for Yunjia Li as a person at the University of Southampton is:
<http://id.ecs.soton.ac.uk/person/12034>
- a URI for a related information resource that describes the real-world object and has an HTML representation. For example, Yunjia Li’s home page in HTML is:
<http://www.ecs.soton.ac.uk/people/yl2>
- a URI for a related information resource that describes the real-world object and has an RDF/XML representation. For example, the RDF representation of Yunjia Li is:
<http://rdf.ecs.soton.ac.uk/person/12034>

The multimedia file on the Web is not a “real-world” object and neither does it have an HTML representation. However, to avoid the confusion of the “video object” and the “location of the video”, a new URI can be minted for the “video object”, for example <http://www.example.com/test>. Then it is safe to use content negotiation as shown in Figure 4.1(b), where both *application/xml+rdf* and *video/** will be redirected to the proper resource respectively. Such a method has been (partially) implemented in NinSuna (See Listings 1 and 8 in Van Lancker et al. (2013)).

Most of the video sharing platforms stream their videos from the **Multimedia Host Servers** and hide them from direct downloading. However, those video sharing platforms do not keep a static and stable URI for their video files, i.e. the “downloadable URI” either does not exist or changes from time to time. So the landing page URI is the only URI externally naming the “video object”. From this point of view, those applications have already minted a URI for the “video object” and it is distinguished from the video location URL. In contrast to the URI minting in Section 4.2, the new URI is created on the **Player Server** instead of the **Multimedia Host Server**. Because the **Player Server** in this case has no control of the original video/audio resource, it must mint its own URI within its own domains for the “video/audio object”. Since those URIs represent “video objects”, it is reasonable to encode media fragment information in those URIs in order to construct the “Media Fragment Identifier”, which helps the

applications to highlight media fragments when the media fragment is requested (R6). This is also the only way to add media fragment information to the URI of the “video object”, since the video file locations are not publicly available.

One may argue that the redirection for *video/** may cause a problem in that the HTML page on the **Player Server** may not be able to play the multimedia stream as there is an extra 303 redirect to get the multimedia binary data. However, for most of the UAs that implement HTML5, this will not be a problem as HTML5 defines the “fetch” activity¹⁰ for the native video and audio player. Basically, if the fetched resource is an HTTP redirect, the UA will follow the target redirect URL and get the resource from that URL.

In summary, introducing a newly minted URI for the reuse of multimedia resources on the **Player Server** can successfully resolve the confusion of landing page URI and original multimedia file URI. In this case, the **Player Server** does not need to know the URI for the multimedia file on the **Multimedia Host Server**, but at the same time, the **Player Server** can still publish its annotations about media fragments and make sure they are the annotations of the multimedia object instead of the landing page. Even though the “video location” URI is available on the **Multimedia Host Server**, it is still good practice to mint and duplicate a URI for it on the **Player Server** as the “video location” URI is hosted in another domain and it can be changed or even vanish at any time. So the second principle of interlinking media fragments can be described as:

Mint a new URI for a "video/audio object" whenever creating new or reusing existing video/audio resources. Use hash to encode media fragment information into the URIs. Syntax defined in W3C Media Fragment URI is recommended.

4.3 RDF Description for Media Fragments

This section will analyse the R3: how to describe media fragments using RDF:

- R3.1: The detailed value of each dimension
- R3.2: Use domain specific vocabularies to contextualise media fragments if necessary
- R3.3: Annotations need to be linked to the URIs that representing the real multimedia resources instead of the landing pages
- R3.4: Annotations need to be linked to the URIs that representing media fragments instead of the whole multimedia resource

¹⁰<http://dev.w3.org/html5/spec-LC/fetching-resources.html>

- R3.5: The publication of annotations should consider both cases of “Multimedia Host Server” and “Player Server”, making sure both of them can publish their own annotations.

The previous section proposed that whenever a multimedia resource is reused, a new URI should be minted to represent the non-informational “multimedia object”. In this way, R3.1, R3.3 and R3.5 can be easily satisfied. So this section will focus on the resolutions of R3.2 and R3.4.

4.3.1 Choosing Vocabularies to Describe Media Fragments

Many vocabularies exist to describe the metadata of multimedia resources, such as MPEG-7 and IPTC. Usually, such vocabularies define simple descriptions for the multimedia resource, such as duration, title, creator, description, copyright. Those vocabularies differ from each other, so it is not realistic to select a dominant vocabulary that fits all types of multimedia resource. In addition, most of them are not based on RDF, so they are not ready to be published as Linked Data. In order to achieve the mutual understanding among these vocabularies using RDF, W3C-MA defines the core vocabulary for multimedia resources and provides ontology (or vocabulary) mappings to other existing formats. Applications, which serve metadata in MPEG-7 for example, do not need to re-write their metadata format, but use a mapping to publish MPEG-7 metadata in W3C-MA format. As an alternative, much alignment work has also been done in M3O, which offers rich semantic description of media resources. Another reason for us to recommend W3C-MA as the basic ontology to describe multimedia resource is that it clearly defines the notion of media fragments, so that the affiliation can be setup between the multimedia resource and its media fragments easily.

If the RDF description for those media fragments is available, it can be expected that (except for lower level metadata) media fragments are also associated with user-generated content. For example, on YouTube a user can embed a start time or time span in their comments¹¹, which actually indicate that this comment is an annotation of that media fragment. . In order to include such annotations in the RDF description of media fragments, some annotation ontologies need to be applied to model this relationship. Some of those ontologies are general-purpose and some of their vocabularies can be used to annotate multimedia resources, such as Dublin Core¹² and Open Annotation Ontology (Haslhofer et al., 2012). Some of them are designed for a specific domain, such as rNews ontology¹³ and BBC Corenews Ontology¹⁴. The data publisher need to choose or create ontologies to describe this annotation relationship.

¹¹<http://www.wikihow.com/Link-to-a-Certain-Time-in-a-YouTube-Video's-Comment-Box>

¹²<http://dublincore.org/>

¹³<http://dev.iptc.org/rNews-10-The-Audio-Image-Video-Object-Classes>

¹⁴<http://www.bbc.co.uk/ontologies/news/2013-05-01.html>

4.3.2 Describe Media Fragments Annotations

As discussed earlier, there are three sources of media fragment annotations that can be published:

- Case 1: Annotations for video/audio files on the Multimedia Host Server
- Case 2: RDF description for video/audio objects on the Player Server when the referred video/audio source is NOT available
- Case 3: RDF description for video/audio objects on the Player Server when the referred video/audio source is available

Theoretically, the annotations from all three cases can be freely described using appropriate vocabularies. However, for the solution of R2 where new multimedia object URIs are minted for both the **Multimedia Host Server** and the **Player Server**, the references between the two URIs must be retained. As the “Cool URI” guideline suggests:

All the URIs related to a single real-world object resource identifier, RDF document URL, HTML document URL should also be explicitly linked with each other to help information consumers understand their relation.

There are generally two kinds of solution to implement such links. The first one uses the HTTP Link Header¹⁵ as suggested by Hausenblas et al. (2009a). Following the example in Section 4.2, the HTTP response after a 303 redirect will look like:

```
HTTP/1.1 200 OK
Accept-Ranges: bytes
Content-Length: 2034
Content-Type: application/rdf+xml
Content-Location: http://www.example.com/test.rdf
Link: <http://www.example.org/test.ogv>;
rel="http://www.w3.org/2000/01/rdf-schema#seeAlso";
Vary: accept
```

This solution has two problems. First, the purpose of the HTTP Link Header is pointing “an interested client to another resource containing metadata about the requested resource”¹⁶. However, this usage is more like the other way round, i.e. pointing to the original resource given the metadata. Secondly, the HTTP Link Header is not well

¹⁵<http://tools.ietf.org/html/draft-nottingham-http-link-header-06>

¹⁶<http://www.w3.org/wiki/LinkHeader>

adopted by major UAs¹⁷ and it has not been included in the HTTP/1.1 specification¹⁸. So a more appropriate solution is to include the location of the video/audio file in the RDF description. W3C-MA defines *ma:locator* with the purpose of linking the URI of the media resource to the real location of the multimedia resource. Listing 4.3 is an example.

```
<http://www.example.com/test>
  a ma:MediaResource ;
  ma:hasFragment <http://www.example.com/test#t=5,12>;
  ma:locator <http://www.example.com/test.ogv>.
```

LISTING 4.3: Example of using *ma:locator*

For the RDF description of a new URI, *ma:locator* needs to be applied to connect the “video/audio object” with the location of the video/audio resource. However, if reusing the videos in **Case 2**, it can only assume that the landing page URI for the video is the video location URL, since the real location of the video is not stable and for public access. For example, Listing 4.4 is a sample RDF description in Synote if the referred to video is from YouTube (Li et al., 2012):

```
<http://linkeddata.synote.org/resources/36513#t=5,12>
  a ma:MediaFragment ;
  ma:isFragmentOf <http://linkeddata.synote.org/resources/36513>;
  dc:subject "World Wide Web";
  ma:locator <http://www.youtube.com/watch?v=0M6XIICm_qo#t=5,12>.
```

LISTING 4.4: N3 Code Clip to Describe a Media Fragment

It seems also reasonable to use the videos landing page directly as an *ma:Resource* without minting a new URI in this example, and attaching the media fragment after the landing page URI will not cause any problem. However, it is better to keep out of the namespaces that the application does not own (Heath and Bizer, 2011), because the application will lose control of what could be dereferenced. In addition, if later applications like YouTube change the meaning of *#t=5,12* to represent something other than media fragments, or they change the syntax of media fragments, the URIs will not be valid any more. Then it will be very expensive to update the entire media fragment URIs in the YouTube namespace.

In this example, *http://linkeddata.synote.org/resources/36513#t=5,12* is an instance of *ma:MediaFragment* (R3.4) and the media fragment’s *dc:subject* is “World Wide Web”. To our best knowledge, the YouTube landing page is the location of the original video source, so the media fragment string is directly attached after the landing page URI to indicate the corresponding fragment in the original source. The object of *ma:locator* can also be any video files online, such as:

<http://example.org/1.ogv#t=5,12>

¹⁷<https://lists.webkit.org/pipermail/webkit-dev/2010-June/013063.html>

¹⁸<http://www.rfc-editor.org/rfc/rfc2616.txt>

Section 4.1 showed that choosing hash URI to encode media fragment information can make sure that *1.ogv#t=5,12* is dereferenceable even though the host server is not compatible with media fragments (R2 and R5). Otherwise, there will be a broken link in the RDF description.

The vocabularies used in the RDF description for media fragments are dependent on specific domains (R3.2), however two relationships must be retained in the description:

The links between “video/audio URIs” and their affiliated media fragment URIs should be included in the RDF descriptions of media fragments. Using `ma:hasFragment` and `ma:isFragmentOf` in W3C-MA is recommended.

The RDF description of “video/audio object” should include an explicit link between the “video/audio URI” and the location URL of the video/audio resources to the best of your knowledge. Using `ma:locator` in W3C-MA is recommended.

4.4 Media Fragments Interlinking

The fourth principle of Linked Data indicates that links to other URIs should be included. When a video/audio object on the Web has been identified by URI and its descriptions have been made available in RDF, the next step is to investigate what links could be built between this URI and other URIs in the LOD Cloud and how to build these links (R4). More importantly, considering the fact that many multimedia sharing platforms have already made their multimedia and text resources available via means such as Web APIs, the media fragment interlinking process will, preferably, be an extra layer on top of existing systems rather than developing new systems from scratch (R5).

The key procedure to solve this problem is to get the “Structured Data” and “Text” from the multimedia resources, which is the pre-condition to publish Linked Data following different patterns. Depending on how the links are created between multimedia resources at the fine-grained level, Hausenblas et al. (2009b) divides the interlinking process into four categories: manual, collaborative, semi-automatic, and fully automatic interlinking. This sub-section differs from this approach and proposes different ways of interlinking creations in order to reveal how the “Structured Data” and “Text” can be extracted from raw multimedia resources. Through this process, no new interlinking methods are invented, but they are adapted for publishing multimedia and media fragments data.

Figure 4.2 extends Figure 2.3 by adding an extra layer “Multimedia Data Extraction and Alignment” between the “Raw Multimedia Data” and “Data Preparation”. The “Raw Multimedia Data” includes multimedia resources and external annotations. Images, audios and videos are basic multimedia resources and the external annotations are mainly

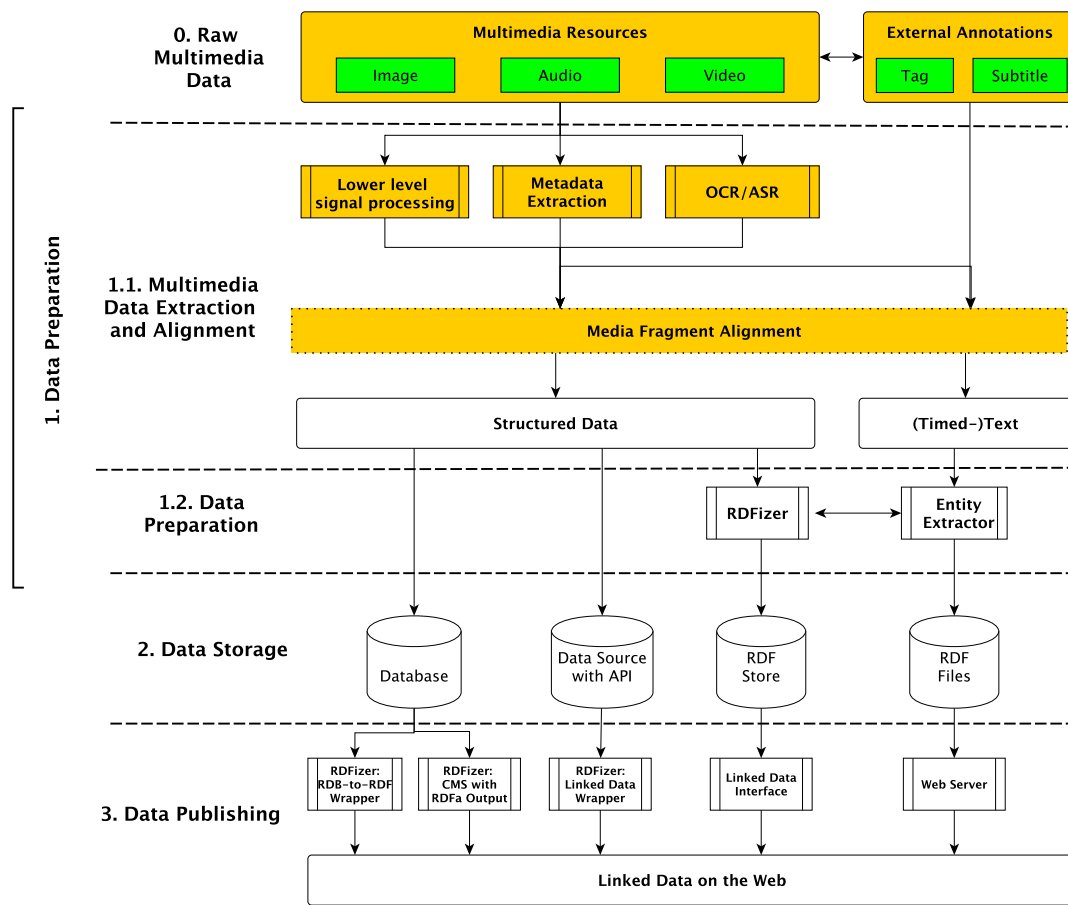


FIGURE 4.2: Media Fragments Interlinking and Publishing Patterns

user-generated multimedia enrichment, such as tags, subtitles¹⁹ and comments. From Figure 4.2, it can be seen that the raw multimedia data is two steps away from the structured data or text that can be RDFized for publication. The detail of multimedia data extraction and media fragment alignment techniques are outside the scope of this thesis, but some examples will be listed below to explain how the data preparation can be implemented.

The first step is to extract desired features or identity data from the raw multimedia resources. Technically, there are at least three ways to implement this function automatically. Lower level signal processing refers to the analysis of colour, contrast or sound waves in video/audio resources and the output the entities that have been detected, such as sky, bird, or plane. A second process is to apply speaker diarisation²⁰ for the sound track in video/audio resources and identify where the different speakers take turns. Thirdly, metadata extraction will use tools, such as FFmpeg²¹, to read

¹⁹Closed captioning can be embedded in the multimedia as a track or multiple tracks. But here, the subtitle means plain text transcript or timed-text generated by users, which is stored externally from the original multimedia resource.

²⁰http://en.wikipedia.org/wiki/Speaker_diarisation

²¹<http://ffmpeg.org/>

the metadata from the raw multimedia files. The metadata includes information like codecs, tracks, streams and bitrate, as well as plain-text metadata like title. Track is one of the dimensions supported by W3C-MFURI, so the metadata extraction can provide information needed to construct media fragment URIs. Optical Character Recognition (OCR) and Automatic Speech Recognition (ASR) can analyse images or sound waves to extract plain-text or transcript from the raw video/audio resources. Even though the results generated from OCR and ASR are usually noisy, they are still a straightforward means to extract plain-text annotations from the continuous multimedia resources.

After multimedia data extraction, the extracted entities and the external annotations need to be aligned with different dimensions of the media fragments. This step is critical to interlink multimedia resources to annotations at a fine-grained level. The implementation of the media fragment alignment is very dependent on the technique. For example, Synote uses IBM Atilla as the ASR engine to obtain a timed-text in XML format, where every word is aligned with a certain time in the audio. Another example is using FFmpeg to generate screenshots from a set of times in the video and use OCR software to extract text from each screenshot. The text is then synchronised with the time when the screenshot was taken. The media fragment alignment for external annotations will rely on whether the source of annotations provides fragment alignment or not. If not, additional techniques will be needed. For example, CMU Sphinx forced alignment²² can be applied to find start and end times of each word in a plain-text transcript. Also emerging are crowd-sourcing ways to create subtitles for YouTube videos, such as Amara²³.

After multimedia data extraction and alignment, developers can choose a route in Linked Data publishing patterns to publishing media fragments and annotations, depending on the individual requirements of the system. The “Structured Data should have media fragment information included and associated with other data. There is no specific format for timed-text either, and some commonly used formats include SRT and WebVTT.

In Figure 4.2, RDFizer, including RDB-to-RDF wrapper and RDFa output, and Entity Extractor are the key components to link “Structured Data” and “(Timed-)Text” into the LOD Cloud. Text can be processed automatically by named entity recognition tools such as DBpedia Spotlight²⁴ and Open Calais²⁵. Triples can then be written to say that the named entity annotates the media fragment which the text corresponds to. Chapter 5 discusses RDFizer in more detail.

The **Player Server**, which has no access to the original multimedia resources, will have difficulty in using any multimedia data extraction method introduced in 4.2. However, such applications as Synote and Amara can still contribute to the media fragment interlinking as they provide external annotations with media fragment alignment. The

²²<http://goo.gl/mI4zI3>

²³<http://www.amara.org/en/>

²⁴<http://dbpedia.org/spotlight>

²⁵<http://www.opencalais.com/>

applications on the **Multimedia Host Server** should take more responsibility and release more features to help media fragment interlinking.

4.5 Summary of the Principles

This chapter discussed the principles of bridging media fragments and Linked Data, addressing the solutions for requirements R1 to R4. Recommendations and good practice was listed as the main body of Interlinking Media Fragments Principles. The core of the principles is to treat each video/audio as a concept or object and mint a new URI whenever the video/audio object is reused. While avoiding using the location of the video/audio files as the stem of media fragment URIs, a triple should be included to describe the relationship between the newly minted URI and the original location of the file. In this way, video resources hosted by different applications can be safely reused on a media fragment level.

Meanwhile, since R1 and R2 depends on R5, the Interlinking Media Fragment Principles also ensure that, even though the existing **Multimedia Host Server** is not compatible with media fragments and has no intention of providing an RDF description of its resources, the media fragments in those resources can still be reused and referred to, and media fragments and annotations may be published using Linked Data. In this way, the principles enable the existing multimedia resources and annotations to be seamlessly integrated with the current Web of Data at media fragment level. To sum up, the Interlinking Media Fragments Principles brought forward in this Chapter contain the following rules:

1. **Choosing URIs and Dereferencing:** Mint a new URI for “video/audio object” whenever creating new or reusing existing video/audio resources. Use hash URI to encode media fragment information into the URIs, so that the URI dereferencing will be compatible with servers which are not media-fragment-ready. Syntax defined in W3C Media Fragment URI is recommended.
2. **RDF Description:** The following two relationships must be contained in the RDF descriptions of media fragments:
 - (a) The links between “video/audio URIs” and their affiliated media fragment URIs should be included in the RDF descriptions of media fragments. Using `ma:hasFragment` and `ma:isFragmentOf` in W3C-MA is recommended.
 - (b) The RDF description of “video/audio object” should include an explicit link between the “video/audio URI” and the location URL of the video/audio resources to the best of your knowledge. Using `ma:locator` in W3C-MA is recommended.

Those principles actually answer what relationship should be included to model media fragment data with RDF. There was no conclusion on the technical implementation of such a model or how the media fragment interlinking can be implemented. The common Linked Data publishing patterns of Figure 2.3 are still applicable to media fragments. The only difference is that the source of the data, either “Structured Data” or “Text” needs to be extracted first from the raw multimedia data and those annotations need to be aligned with media fragments.

The Interlinking Media Fragments Principles are the first step towards the improvement of the online presence of media fragments. In Chapter 5, the Interlinking Media Fragments Principles will be further extended by proposing a full RDF model for media fragments and annotations, and a framework to automate the interlinking process.

Chapter 5

A Framework for Linking Media Fragments into the Linked Data Cloud

The Interlinking Media Fragments Principles were put forward in Chapter 4, which included the principles for choosing URIs, dereferencing, and RDF descriptions for media fragments. Several ways were proposed to extract data from raw multimedia resources and made them ready for publishing as Linked Data. This chapter will take the solutions to requirements R1 to R5 one step further and propose a framework to automate the media fragment interlinking and publishing process, which will be referred to as *media fragment enrichment*. In this sense, R5.4 will be the focus of this chapter.

R5.4 can actually be divided into two main tasks based on the Interlinking Media Fragment Principles: (1) a core model to describe media fragments, media annotations and other necessary information (see R1 to R3) as the basis for interlinking media fragments to other resources. The core model will be domain-independent and could serve as a general purpose model; (2) a framework which can automatically generate the instances of the classes and relationships defined in the core model and enrich media fragments with proper data (see R4 and R5).

This Chapter first presents the Core Model for Media Fragment Enrichment, which is a universal model that describes media fragments and links them to other resources in LOD Cloud. Then Section 5.2 will pick from Figure 4.2 external annotations and entity extractions as the pattern for automatically publishing media fragments, and presents the Media Fragment Enrichment Framework. To demonstrate the use of the core model and enrichment framework, Section 5.3 and Section 5.4 present two examples: YouTube enrichment demonstrates how videos from major online sharing platforms can be enriched automatically and provide more useful information to the end users; while the research perspective shows how interlinked media fragments data opens up a new

method for improving video classification problems.

5.1 RDF Descriptions for Media Fragments

The Interlinking Media Fragments Principles only regulate the following two relationships that should be included in the RDF description: (1) “video/audio objects” and their fragments; (2) “video/audio objects” and the location of the original files. To enrich such a description for media fragments, it is natural to think of interlinking happening in an annotation relationship, i.e. a multimedia resource or media fragment annotates, or is annotated by, an external resource. As Figure 4.2 showed, there are many ways that the annotation relationship can be constructed, such as entity extractions from external annotations, or from OCR for a series of video frames. So it is important in the RDF model to record where the annotation comes from, i.e. provenance data should be added into the media fragment description and interlinking.

This section now introduces the general Core Model for Media Fragment Enrichment, which shows what general information needs to be included to describe media fragments according to the requirement identified in R3. This is followed by an implementation of the Core Model for Media Fragment Enrichment using more specific ontologies.

5.1.1 Core Model for Media Fragment Enrichment

Figure 5.1 shows the Core Model for Media Fragment Enrichment. A node with a dashed border means those relationships are optional in the core model, but they are recommended when entity extractions are applied to multimedia annotations. The model consists of three main components: media resource and fragments description, annotation description, and provenance description. All the components rely on well-known ontologies that have been widely applied in different domains. Following the second Principle in Interlinking Media Fragments, the application should mint its own URIs like `http://example/media/1` for each multimedia resource.

W3C-MA is used to link media fragments with their parent media resource via *ma:isFragmentOf* and *ma:locator* is used to connect the newly minted URI to the media file resource. The video content is annotated with different degrees of granularity using the W3C-MFURI specification for addressing segments of this content (R3.4). Hence, the instances of the *ma:MediaFragment* class are the anchors where entities are attached. The NinSuna Ontology¹ defines a more sophisticated vocabulary for describing the details of temporal and spatial fragments encoded in the W3C-MFURI, compared with W3C-MA (R3.1). In this example, *nsa:temporalStart* and *nsa:temporalEnd* correspond to the start and end time of the media fragments respectively. Theoretically, parsing the media fragment

¹<http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna>

URI can get the detailed information in each dimension encoded in the URI. However, in some cases, it will be better if the value of each dimension is directly available via a SPARQL query instead of parsing the hash fragment using programming language specific libraries. UC1 and UC3 in Chapter 3 are such examples.

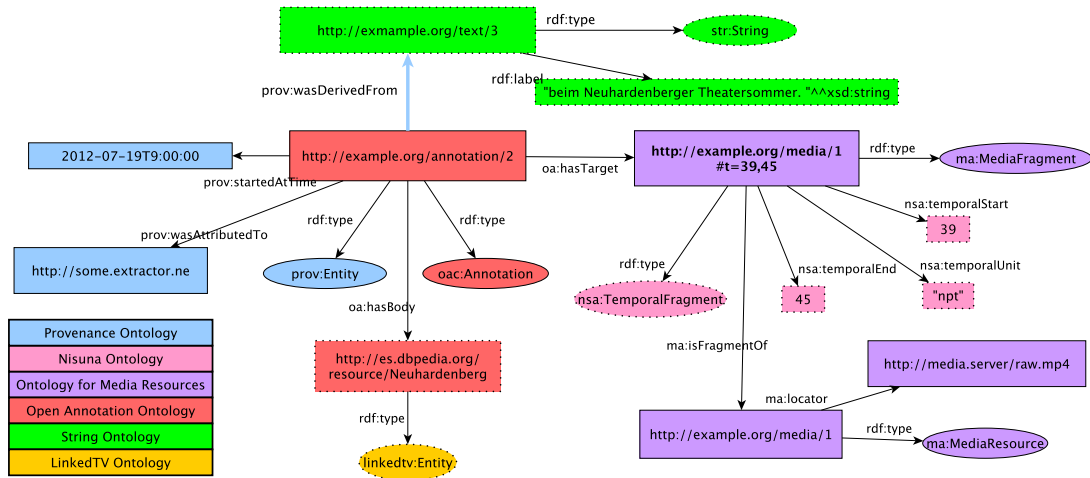


FIGURE 5.1: The graph depicts the MediaFragment serialization and how an Entity and its corresponding text are attached to a MediaFragment through an Annotation.

The core vocabulary to link media fragments with external resources is the Open Annotation Data Model (Haslhofer et al., 2012) (R3.2). The media fragment URI is the target of annotation and the named entity “Neuhardenberg” in DBpedia is the body of the annotation. Using fragment URIs to identify an annotation target or body is a common practice for the Open Annotation Data Model, to make the annotation more precise. The body of the annotation could be any URI or literal depending on the requirements. However, the Open Annotation Data Model recommends assigning a general type to the target and body of the annotation² according to the types defined in Dublin Core Types³. If named entity recognition is used for the interlinking, the annotation target will be a *ma:MediaResource* and the annotation body should be a general type of “Named Entity”. LinkedTV ontology⁴ defines an *Entity* class representing the atomic elements recognised by named entity extractors from a text string.

In general, the object of *oa:hasBody* could be any URI or literal that is able to express the meaning of the annotation relationship. As described in Section 2.2.3, text resources are widely available as the enrichment of the multimedia resource and some of them are aligned with the audio or video timeline. If the description only wants to express the relationship that some plain-text annotates a media fragment, the range of *oa:hasBody* could be a string instead of an instance of *linkedtv:Entity*. However, as the fourth rule of Linked Data Principles states, URIs in other domains should be included so that this RDF description of media fragments is promoted to “5 stars” (Section 2.2.4). So the

²<http://www.openannotation.org/spec/core/core.html#BodyTargetType>

³<http://dublincore.org/documents/dcmi-type-vocabulary/#H7>

⁴<http://semantics.eurecom.fr/linkedtv/>

named entities should be extracted from the plain-text and act as a key node to bridge media fragments with the Linked Data Cloud (R4).

Meanwhile, the provenance of the annotation needs to be provided to identify how the annotation was created; especially which extractor was used to determine the named entity, and the original text the named entity is extracted from (R3.2). The Open Annotation Data Model has defined some vocabulary to attach provenance information to the Annotation, Body and Target⁵. However, the vocabulary has not been finalised yet (2015). So in the core interlinking model, the W3C Provenance Ontology⁶ is adopted to describe the important provenance information.

In this example, the named entity is extracted by `some.extractor.ne` from a text string “beim Neuhardenerberger Theatersommer”. String Ontology⁷ is used to describe the properties of the source string from which the named entities are extracted (or from which the annotation is derived in provenance). In this example, a new URI `http://example.org/text/3` is minted for the string resource. This design allows description of the location or character count in the text string from which the named entity is extracted, and helps developers to highlight the word or phrases on the user interface, see Section 5.2.4 (R6). However, as a pre-condition, the API of the named entity extractor must be able to return the entity position, which is not supported by some extractors (See Table 1 in Rizzo and Troncy (2012)). So this description is not included in the core RDF model.

To sum up, media fragment description, annotations to named entities and provenance are the three main components in the core model for interlinking media fragments. The model can be extended or modified according to the publishing requirements for media fragments, but at least those three components should be included and the Principles of Interlinking Media Fragments should be followed. As has been discussed in Section 4.4, there are many approaches to link media fragments to the Linked Data Cloud (R4). Using named entity recognition can automate this process, but sometimes this approach is not applicable to the cases that the text enrichment of the multimedia resource does not exist or is very limited. In those cases, the *oa:hasBody* No matter what the final implementation of this model is, all the URIs, especially the ones minted by each application, must be dereferenceable (R2).

5.1.2 Example Implementation of Core Model for Media Fragment Enrichment

This subsection shows an implementation of the Core Model for Media Fragment Enrichment from the LinkedTV project. Figure 5.2 demonstrates how a *ma:MediaFragment* instance is linked to a *linkedtv:Entity* named entity. The Core Model for Media Fragment

⁵<http://www.openannotation.org/spec/core/core.html>

⁶<http://www.w3.org/TR/prov-o/>

⁷<http://nlp2rdf.lod2.eu/schema/doc/string/index.html>

Enrichment is further extended by the LinkedTV ontology and the NIF ontology (R3.2). NERD mints a URI in the LinkedTV domain for each named entity recognised by its supported extractors. The NIF ontology together with the W3C Provenance Ontology defines properties that are related to this extraction operation, such as which string the named entity is extracted from and which program extracts it. In the example shown, the entity labelled as *Neuhardenberg* and classified as *nerd:Location* is attached to the media fragment through *oa:annotation*. The media fragment is associated with a subtitle block using the *linkedtv:hasSubtitle* property. Both the entity label and the subtitle block are serialized according to the NIF Specification⁸.

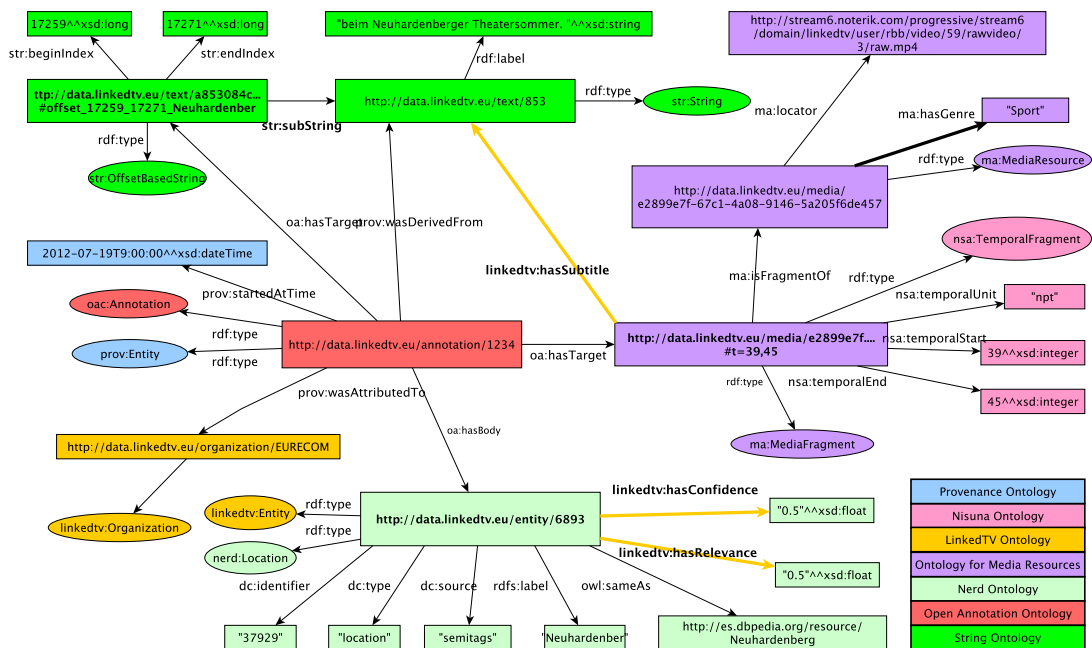


FIGURE 5.2: The graph depicts the MediaFragment serialization and how an Entity and its corresponding Subtitle are attached to a MediaFragment through an Annotation

The Turtle serialization of the example provided in Figure 5.1 follows below. For the code snippets in Listings 5.1, 5.2 and 5.3, the instance URIs for media, text, entity and annotation are automatically created in the LinkedTV domain.

```
<http://data.linkedtv.eu/media/e2899e7f-67c1-4a08-9146-5a205f6de457#t=1563.56,1566.8>
a nsa:TemporalFragment , ma:MediaFragment ;
linkedtv:hasSubtitle <http://data.linkedtv.eu/text/1ca03938-c7ae-4311-a6ed-0540152b651a> ;
nsa:temporalEnd "1563.56"^^xsd:float ;
nsa:temporalStart "1566.8"^^xsd:float ;
nsa:temporalUnit "npt" ;
ma:isFragmentOf <http://data.linkedtv.eu/media/e2899e7f-67c1-4a08-9146-5a205f6de457> .
```

LISTING 5.1: Example RDF Description of Media Fragment Serialised in Turtle

⁸<http://nlp2rdf.lod2.eu/schema/string>

In Listing 5.1, the temporal references are encoded using the NinSuna Ontology⁹. The relationship that a *ma:MediaFragment* belongs to an entire video is modelled with the property *ma:isFragmentOf*. The physical location of the video file online is linked with the newly minted media URI via *ma:locator* following the Interlinking Media Fragment Principles. The entity *Neuhardenber* is further described using Dublin Core¹⁰ and LinkedTV properties in order to specify the entity label, the confidence and relevance scores of the extraction, the name of the extractor used in the process, the entity type and a disambiguation URI for the entity that will generally point to a LOD resource (Listing 5.2). For each entity, an instance of the class *oa:Annotation* is created. This annotation establishes an explicit link between the entity extracted and both the media fragment and its subtitles. The provenance information is also attached by using the Provenance Ontology¹¹ (Listing 5.3).

```
<http://data.linkedtv.eu/entity/9f5f6bc5-fa3a-4de1-b298-2ef364eab29e>
  a nerd:Location , linkedtv:Entity ;
  rdfs:label "Neuhardenber" ;
  linkedtv:hasConfidence "0.5"^^xsd:float;
  linkedtv:hasRelevance "0.5"^^xsd:float ;
  dc:identifier "77929" ;
  dc:source "semitags" ;
  dc:type "location" ;
  owl:sameAs <"http://de.dbpedia.org/resource/Neuhardenberg">.
```

LISTING 5.2: Example RDF Description of Named Entity Serialised in Turtle

```
<http://data.linkedtv.eu/annotation/b85339f5-8b89-4bf9-a049-d663c50e7ae9>
  a oa:Annotation , prov:Entity ;
  oa:hasBody <http://data.linkedtv.eu/entity/9f5f6bc5-fa3a-4de1-b298-2ef364eab29e> ;
  oa:hasTarget <http://data.linkedtv.eu/media/e2899e7f-67c1-4a08-9146-5a205f6de457#t=1563.56,1566.8> ,
    <http://data.linkedtv.eu/text/1ca03938-c7ae-4311-a6ed-0540152b651a#offset_12770_12776_Turkey>,
  prov:startedAtTime "2013-02-08T14:14:39.4Z"^^xsd:dateTime ;
  prov:wasAttributedTo <http://data.linkedtv.eu/organization/EURECOM> ;
  prov:wasDerivedFrom <http://data.linkedtv.eu/text/1ca03938-c7ae-4311-a6ed-0540152b651a> .
```

LISTING 5.3: Example RDF Description of Annotations Serialised in Turtle

5.2 Media Fragment Enriching Framework

The Core Model for Media Fragment Enrichment needs to be supported by a framework that can collect and generate the data needed by the model. This subsection introduces the Media Fragment Enriching Framework that fulfils the media fragments interlinking requirement of R4.

⁹<http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna>

¹⁰<http://dublincore.org/documents/2012/06/14/dces>

¹¹<http://www.w3.org/TR/prov-o>

Figure 4.2 indicated that structured or text data can be extracted either from the raw multimedia resources or from external annotations. Sometimes, the raw multimedia resources are not accessible by applications like Synote, which reuse the multimedia resources from existing multimedia sharing platforms. With the fast development of video sharing applications, end users have more and more access to higher level text-based content. The titles, descriptions and tags are usually visually available on the landing page for users, and they are also exposed via Web APIs. Subtitles (or transcripts) of the videos are also supported by major video sharing applications, which enriches the video content and improves the accessibility of the videos. So it is necessary to exploit those higher level text-based materials as the way to link media fragments into the massive concepts and entities available in the LOD Cloud. The preliminary input data of the Media Fragment Enriching Framework that will be introduced here will focus on the metadata and timed text from major video sharing platforms.

For any existing multimedia sharing applications, which are not Linked Data driven, it is important to automate the process of interlinking and to build an extra layer on applications (R5). Figure 5.3 shows the modular implementation of the proposed framework. In a nutshell, the framework enables the retrieval of video metadata and timed-text from video sharing platforms, the extraction of named entities from timed text, the modelling of the resulting semantic metadata in RDF, and optionally, it provides a user interface that supports browsing in enriched hypervideos.

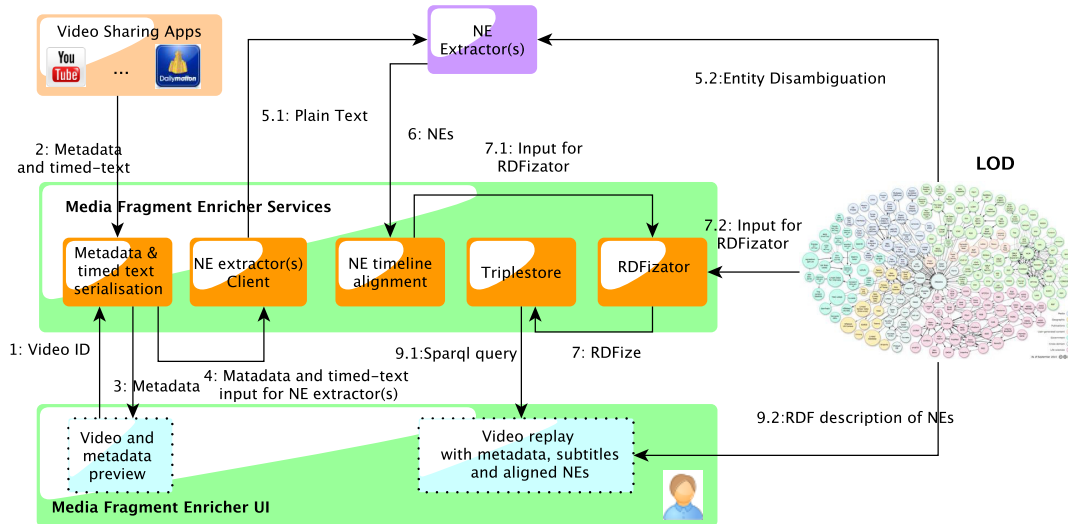


FIGURE 5.3: Architecture of proposed Media Fragment Enriching Framework

The workflow is as follows. First, a viewer or any program provides the ID of the video to the **Metadata and Time Text Serialization** module which retrieves the media resources metadata (title, description, statistics about its popularity) and timed-text. The timed-text could be subtitles in SRT or WebVTT format, or any comments with time alignment. Secondly, the viewer launches the annotation process using a

NE extractor Client that will extract named entities from the metadata and timed-text. The text resource like subtitles usually have already been synchronised with the multimedia. Sometimes, user-generated comments also have a timestamp aligned with the video. In these cases, after the **NE Extractor** extracts named entities from the text, the **NE timeline alignment** module will align the named entities back into the multimedia resource. The named entities extracted from other text-based metadata will then be associated with the whole video.

All these annotations are then serialized in RDF by the **RDFizator** module that creates media fragments for each timed-text block in which entities have been spotted. The output of **RDFizator** is an instance and extension of Core Model for Media Fragment Enrichment. The resulting RDF graph is stored in a **Triplestore** and could be queried via SPARQL endpoint. Finally the video, timed-text and the named entities extracted are pulled together for the viewer that can interact with the content and get additional information coming from the LOD cloud. In the following subsections, further details will be given about these processes.

5.2.1 Metadata and Timed-Text Retrieval

The Media Fragment Enriching Framework does not specify how the metadata and timed-text is retrieved from video platforms. Unless the user is the owner of the videos, the user needs to rely on the APIs of the applications to obtain the metadata. The retrieved metadata is both depicted to the user in the final interface and is further processed by the *Metadata and Time Text Serialization* module. Metadata is important since it contains general information about the video being retrieved such as: title, description, tags, channel, category, duration, language, creation date, publication date, view, comment, favourites, and rankings. Some of the elements are not directly related to the named entity extraction, but will be useful for the video classification tasks covered in Section 5.4.

As each video sharing platform has its own schema to describe the metadata, it is necessary to develop a general model to harmonise the metadata for later data processing and visualisation. Most platforms currently publish the timed-text or subtitles in plain-text or SRT format currently, so it is not necessary to further harmonise the format of timed-text. The metadata schema from YouTube and Dailymotion are mapped to the harmonised metadata model as shown in Table 5.1 Since YouTube and Dailymotion APIs mainly return data in JSON format, the JSON path is used to refer to the properties in each schema.

Listing 5.4 is the JSON format of the harmonised model.

```
{
  "id": string,
  "metadata": {
```

TABLE 5.1: Comparison of Harmonised Model with YouTube, Dailymotion Metadata Schemas

Property Name	YouTube API v3	Dailymotion API
<i>Descriptive metadata</i>		
id	id	id
metadata.title	snippet.title	title
metadata.description	snippet.description	description
metadata.tags	snippet.tags[]	tags
metadata.category	category	channel
metadata.duration	contentDetails.duration	duration
metadata.language	n/a	language
metadata.creationDate	snippet.publishedAt	created_time
metadata.publicationDate	recordingDetails.recordingDate	taken_time
<i>Social media statistics</i>		
statistics.views	statistics.viewCount	views_total
statistics.comments	statistics.commentCount	comments_total
statistics.favorites	statistics.favoriteCount	bookmarks_total
statistics.ratings	statistics.likeCount- statistics.dislikeCount	ratings_total

```

    "title": string,
    "description": string,
    "tags": [
      string
    ]
    "category": {
      "label": string,
      "uri": string,
    }
    "duration": string,
    "language": string,
    "creationDate": datetime,
    "publicationDate": datetime,
  },
  "statistics": {
    "views": unsigned long,
    "comments": unsigned long
    "favorites": unsigned long,
    "ratings": unsigned long,
  }
}

```

LISTING 5.4: JSON Format of the Harmonised Metadata Model

5.2.2 Named Entity Extraction, Timeline Alignment and RDF Generation

Many named entity recognition tools are available online and each tool may provide different extraction results given the same text. The proposed framework does not limit which extractor should be used. The timeline alignment module links the named entities

with the start and end times of the corresponding text block. This function depends on the data structure of the timed-text, as well as the response from the extractor. There is no universal algorithm as to how this could be done. For example, most named entity extractors take a plain-text document as input, but some of them, such as DBpedia Spotlight and Zemanta, also return the character count or offset, where the named entity has been spotted (Rizzo and Troncy, 2012). In this case, if the subtitle is in SRT format, the program can extract the characters from each text block in the SRT file and merge them into a single document as the input to named entity extractors. Then the named entities with character offsets can be mapped back into the text blocks of the SRT file, and aligned with the time span the text blocks correspond to.

The timeline alignment result is serialized in some data structure, which is picked up by the *RDFizator* module that will consider the named entities as temporal anchors for creating the annotated media fragments. The *Triplestore* contains RDF descriptions following the Core Model for Media Fragment Enrichment.

5.2.3 Media Fragment Enricher UI

The Media Fragment Enricher UI visualises the data collected from various components of the Media Fragment Enricher Services. A common practice for visually connecting the named entities with media fragments is by temporally and spatially highlighting the named entities on the video, similar to the idea of an interactive transcript on a landing page, but the UI design depends on the individual application’s requirement. The “Video and metadata preview” in the UI is not required and just gives a preview of what the metadata of the “Video ID” is about. The interactive video player provides an overview of the output of the framework and interacts directly with the end users. An example of the UI implementation is presented in Chapter 6.

5.2.4 Implementation of Media Fragment Enricher Services

Media Fragment Enricher Services is a general-purposed framework, so different detailed implementations can be specified from the following aspects.

- Source of metadata and timed-text: could be any video-sharing platforms (not limited to YouTube, Dailymotion) with their data exposed via some means.
- Extension of Core Model for Media Fragment Enrichment: linking named entities to media fragments via Open Annotation Data Model may not be the best way to model the data relationship in a specific applications. More domain-specific vocabularies should be used in this case as the extension of the core model.
- Interlinking generation:

- Interlinking methods other than named entity extraction, such as manual interlinking, crowdsourcing, etc.
- Named entity extractors: NERD is a recommendation, but any application can be used or a combination of applications to process the named entity recognition, for example Stanford Named Entity Recognizer¹² and Apache Stanbol¹³.
- Media Fragment Enricher UI: there is no guideline for the UI design.

Figure 5.4 shows a modular implementation of the Media Fragment Enricher Services. It retrieves metadata about the video content from YouTube and Dailymotion, generates entities from timed text, and models it using RDF. The sequence of actions is as follows: first, the viewer¹⁴ demands a certain video that is hosted by one of the supported video platforms. The Metadata and Time Text Serialization module retrieves the media resource and its associated metadata (statistics and subtitles). When this information is available, the viewer launches the process of subtitle annotation through named entities (NERD Client). The retrieved entities are used as inputs of the RDFizator module, which creates the media fragments and represents them according to the LinkedTV Ontology. The created RDF graph is stored inside the Triplestore so that the Media Fragment Enricher UI can query it and visualise the results.

The RDFizator will take the output of NERD and write all those pieces of information into a single graph for each named entity. Finally, all the instances are interconnected creating a graph that can be queried through the SPARQL endpoint exposed by the *Triplestore*, which will serve as the input of the Media Fragment Enricher UI.

5.3 Creating Enriched YouTube Media Fragments with NERD using Timed Text

In this section an experiment is undertaken as an example implementation of the Media Fragment Enriching Framework. This example covers all the steps of enriching YouTube videos, including retrieving metadata and subtitles from the YouTube API, extracting named entities from the subtitles using NERD, generating RDF descriptions following the model presented in Section 5.1, and visualising the data with an implementation of Media Fragment Enricher UI. In particular, preliminary data is collected as a proof of concept for the video classifications required in Section 5.4.

In this experiment, NERD is used for extracting named entities from timed-text associated with YouTube videos in order to generate media fragments annotated with resources

¹²<http://nlp.stanford.edu/software/CRF-NER.shtml>

¹³<http://stanbol.apache.org/>

¹⁴The “viewer” here can be a physical user or any automatic programme.

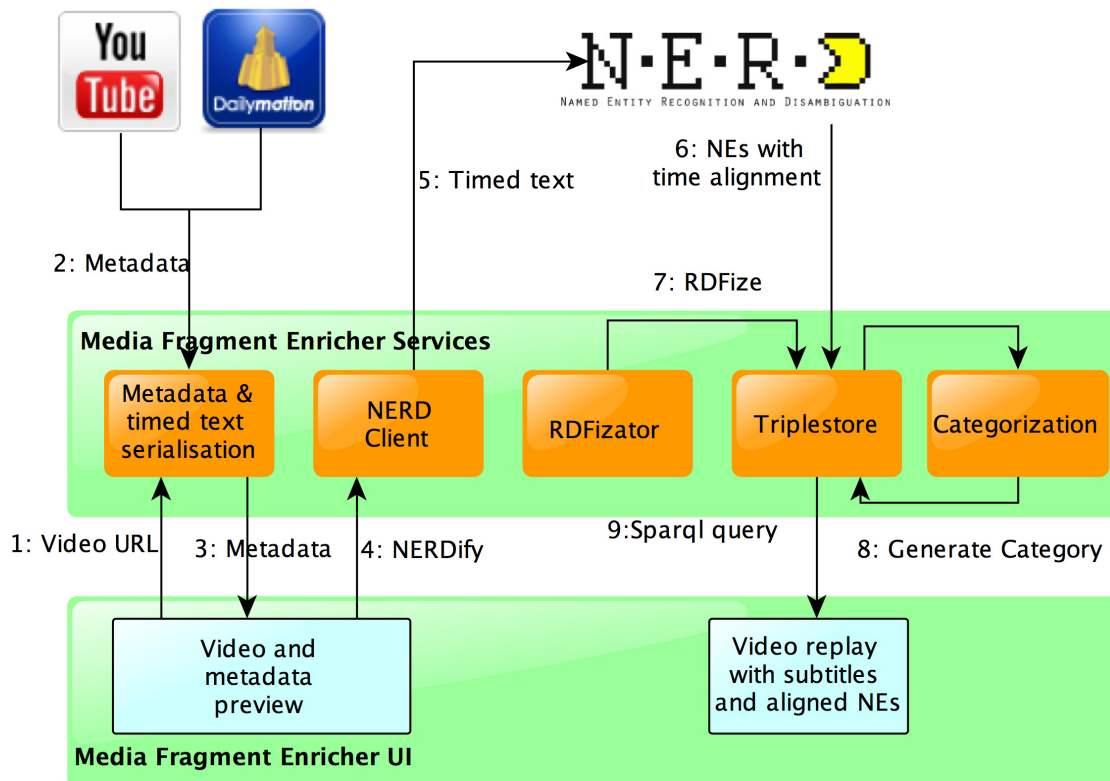


FIGURE 5.4: Implementation of the Media Fragment Enricher Framework using NERD for metadata extraction, name entity recognition and disambiguation

from the LOD Cloud. The combined strategy from NERD is applied here instead of examining each extractor individually, and all the named entities extracted from all the extractors will be matched with the 9 top level classes in the NERD ontology as the output. NERD can take the timed-text document in SRT format and align the named entities with the timeline denoted by *startNPT* and *endNPT* in the extraction results. So the RDF description for each YouTube video fragment identified by the SRT files are constructed following the model, and visualised in the Media Fragment Enricher UI (Figure 6.2). 60 videos¹⁵ with subtitles were collected with their metadata and filtered for 3 different categories: *People and Blogs*, *Sports*, and *Science and Technology*. There are 20 videos in each category. Videos have different durations ranging from 32 to 4505 seconds, and different popularities ranging from 18 to 2,836,535 views (as at 30 July 30 2012). The corpus is available at <http://goo.gl/YhchP>.

classification results are aligned to 8 main types (Event is only supported by OpenCalais in the beta version) plus the general type Thing used as fallback where NERD cannot find a more specific type.

Following the workflow introduced in Figure 5.4, the experiment collects subtitles and performs entity recognition using NERD. All extractors supported by NERD are combined together and the classification results are aligned to 8 main types (*Event* is only

¹⁵The video #16 in the *People and Blogs* category is discarded because its subtitles are written in Romanian. So there are actually 19 videos in that category

TABLE 5.2: Upper part shows the average number of named entities extracted. Lower part shows the average number of entities for the top 9 NERD top categories grouped by video channels.

	People and Blogs	Sports	Science and Technology
n_d	19	20	20
n_w	7,187	21,944	39,661
r_w	378.26	1,097.20	1,983.05
n_e	610	897	1,303
r_e	32.11	44.85	65.15
n_w/n_e	11.78	24.46	30.43
Thing	6.68	15.35	14.75
Person	4.42	9.75	14.55
Function	0.74	7.35	1.15
Organization	3.63	9.20	12.25
Location	3.89	8.05	6.40
Product	3.26	2.60	6.40
Time	3.95	13.80	3.35
Amount	5.47	9.30	6.30
Event	0.05	0.00	0.00

supported by OpenCalais in the beta version) plus the general type *Thing* used as fallback where NERD cannot find a more specific type. used as fallback in the case NERD cannot find a more specific type. The final results for each YouTube video are visualised in an implementation of Media Fragment Enricher UI, which previews the video together with the subtitles and named entities extracted from NERD.

Some previous evaluation has used NERD to extract named entities from three datasets: transcripts of TED talks, news articles from the New York times (collected from 09/10/2011 to 12/10/2011) and 217 abstracts of the WWW2011 conference (Rizzo and Troncy, 2012). The results have revealed that the extraction ratio by extractors varies depending on content of the datasets. Each extractor may be good at extracting entities in some classes (Person, Location, Time, etc.). However, in this experiment, the differences between the extractors are not the main focus. This experiment is more interested in how many named entities can be extracted in each video category and their distributions along the 9 top level NERD classes. To reveal this correlation, the following variables are defined for evaluation: number of documents per category n_d ; total number of words n_w ; number of words per document ratio r_w ; total number of entities n_e ; number of entities per document r_e (Table 5.2). The n_w/n_e means how many words on average contain one named entity that is recognized by NERD. A larger value of this variable generally indicates that, given the same number of words of transcript in each category, this category is expected to extract more entities.

Table 5.2 shows that *People and Blogs* contains fewer entities than *Science and Technology* and *Sports*. Another observation is that *Science and Technology* videos tend to

be more about people and organisations while *Sports* videos more often mention locations, times and amounts. The class distribution for *People and Blogs* videos are more evenly distributed compared with both *Sports* and *Science and Technology*. It is also interesting to see that this type of video can be used to train event detection.

This experiment has shown that the number of named entities varies between the three different video categories. It is possible to use this variable as a feature for video classification. In addition, the features related to media fragments have not been considered, i.e. the entities distribution along the timeline of the video. In order to prove the hypothesis that named entities and media fragments can be used for video classification, this experiment needs to be improved with a larger dataset with more categories, more sophisticated algorithms, and a thorough measurement regime. This experiment is extended in Section 5.4 to show that how named entities and their distribution along the timeline can be used for video classification.

5.4 Video Classification using Media Fragments and Semantic Annotations

Section 5.3 briefly showed that named entities may have correlations with the categorization of the video. This section will explore the possibilities of using the named entities and media fragment data, generated from the Media Fragment Enriching Framework, as the basic features for classifying videos. This is a good example of the use of interlinked media fragments to solve research problems in other areas.

Video subtitles (or timed text) are an ideal textual resource for getting insights of the content. Katsioulis et al. (2007) have applied named entity recognition techniques on video subtitles together with domain ontologies in order to improve video classification. This reference suggested as future development that video segments could be used together with named entities to improve classification results. This approach is described here where an experiment is conducted aiming to best combine named entities and media fragments for providing a video classification framework.

For the experiment, a set of videos were collected from Dailymotion as input for the Media Fragment Enriching Framework and named entities extracted and aligned with media fragments along the timeline. A collection of 805 videos with subtitles from Dailymotion were randomly selected, coming from different channels (or categories). Named entities were extracted from the subtitles using the NERD framework (Rizzo and Troncy, 2012). Based on the video collection, the number of named entities extracted from video subtitles, their type and their appearance in the timeline were exploited as features for classifying videos into different categories. Four basic machine learning algorithms for multiclass classification problems were designed for this purpose: Logistic

Regression (LG), K-Nearest Neighbour (KNN), Naive Bayes (NB), and Support Vector Machine (SVM). For each approach, the overall classification accuracy as well as the precision, recall and F1-score are calculated for each channel, and for each algorithm-experiment pair. The research questions addressed by this experiment are:

Question 1 : is the number of named entities for each NERD type correlated with a channel?

Question 2 : is the total number of named entities across the different temporal groups correlated with the channels?

Question 3 : are the number of named entities for every NERD type and temporal group correlated with the channels?

Question 4 : which machine learning algorithm(s) can best find correlations allowing us to predict the category of a video?

Thus, the contribution of this section is twofold: 1) interesting insights regarding the named entity distribution along the timeline of videos for a subset of Dailymotion channels; 2) a video classification framework using this named entity distribution and other temporal features sampled from media fragments.

Section 5.4.1 describes the dataset and the data model used for the experiment, showing the number, types, and temporal distribution of named entities in the video items investigated. Section 5.4.2 presents the evaluation methodology, Section 5.4.3 discusses the results of this experiment and Section 5.4.4 summarises the findings.

5.4.1 Dataset

A random set of 805 videos with their subtitles (in SRT format) were obtained from Dailymotion. Using the site API, the video metadata was also collected including the channel the video belongs to and the video duration. The complete dataset was processed using the framework described in Figure 5.4, where the named entities are automatically extracted from the video subtitles and aligned with the corresponding media fragments according to a start time and end time provided by NERD. Even though the original language for the videos varied, including English, French and Russian, all the subtitles were written in English with some special characters in different languages. The duration of the videos ranged from 17 to 7654 seconds. 9 different channels were covered in the video collection and the distribution of videos per channel was found to be: fun (96), technology (44), sport (163), news (66), creation (55), lifestyle (194), shortfilms (81), music (42) and other (64). The videos were associated with channels according to the video owner, and was therefore potentially incorrect. A unique id was assigned to each channel. The

TABLE 5.3: Video and Video Metadata Distribution for the Different Channels (*ne* stands for named entity)

channel	id	video	named entities (ne)	length in secs	ne/videos
fun	1	96	1026	30220s	10.67
technology(tech)	2	44	4071	24201s	92.57
sport	3	163	2794	35940s	17.14
news	4	66	4921	28419s	74.58
creation(creat)	5	55	1966	24283s	35.75
lifestyle(life)	6	194	6996	62490s	36.09
shortfilms(films)	7	81	16806	231657s	207.64
music	8	42	1617	17432s	38.52
other	9	64	4279	29775s	66.88
Total	-	805	44476	484417s	55.28

TABLE 5.4: Number of Named Entities grouped by Type for Each Channel

	Thing	Amount	Animal	Event	Func	Loc	Org	Person	Product	Time
fun	274	106	0	4	11	103	125	182	151	70
tech	1514	689	92	5	66	269	233	571	358	274
sport	618	544	2	20	55	362	197	462	184	350
news	1018	810	3	8	138	827	554	789	374	400
creat	581	274	11	4	60	194	132	379	189	142
life	2175	2010	5	6	107	328	550	867	589	359
films	1511	1729	14	63	492	1369	1705	7532	1233	1158
music	337	201	2	3	45	206	163	403	136	121
other	933	686	49	11	126	604	371	791	381	327
total	8961	7049	178	124	1100	4262	4030	11976	3595	3201

number of videos per channel and the total number of named entities extracted per channel are shown in 5.3. The NERD framework enables grouping the named entities according to 10 top-level types, namely Thing, Amount, Animal, Event, Function, Location, Organization, Person, Product and Time. The type ‘Animal’ has been added here, compared with the 9 major types in Chapter 5. Table 5.4 shows the number of named entities for each of these types for each video channel. Most of the named entities belong to Thing, Amount and Person, while Animal and Event are much less extracted. Furthermore, *shortfilms* has a large number of named entities of type Person and Function, and more than a third of the named entities in Product and Time. It is interesting to notice that most of the named entities of type Animal are extracted from the *tech* channel.

As explained above, each named entity extracted from the video subtitles is aligned with a media fragment, where the start and end time of the media fragment corresponds to the subtitle block time boundaries. The named entities are further grouped in different types in Table 5.4 by the temporal position this named entity is aligned with. As the duration of the videos varies, the temporal positions need to be normalised instead of using the absolute value so that videos with different durations can be compared with

one another. For this purpose, the variable tp (named temporal position) is defined as $0 \leq tp = \frac{st+et}{2 \times dur} \leq 1$, where st and et are the start time and end time of the media fragment the named entity is aligned with, and dur is the duration of the video. When grouping the named entities according to their tp , each video is equally divided into N fragments, so every named entity is included in the fragment its tp falls into. A named entity with temporal position tp belongs to group n if $1 \leq n = tp \times N \leq N$. Figure 5.5 demonstrates the tp distribution of different types of named entity for each channel. For all the plots in Figure 5.5, the x axis is the temporal position of the named entities in the videos and the y axis is the number of named entities in the temporal segments. Different colours in the figure represent the different NERD types.

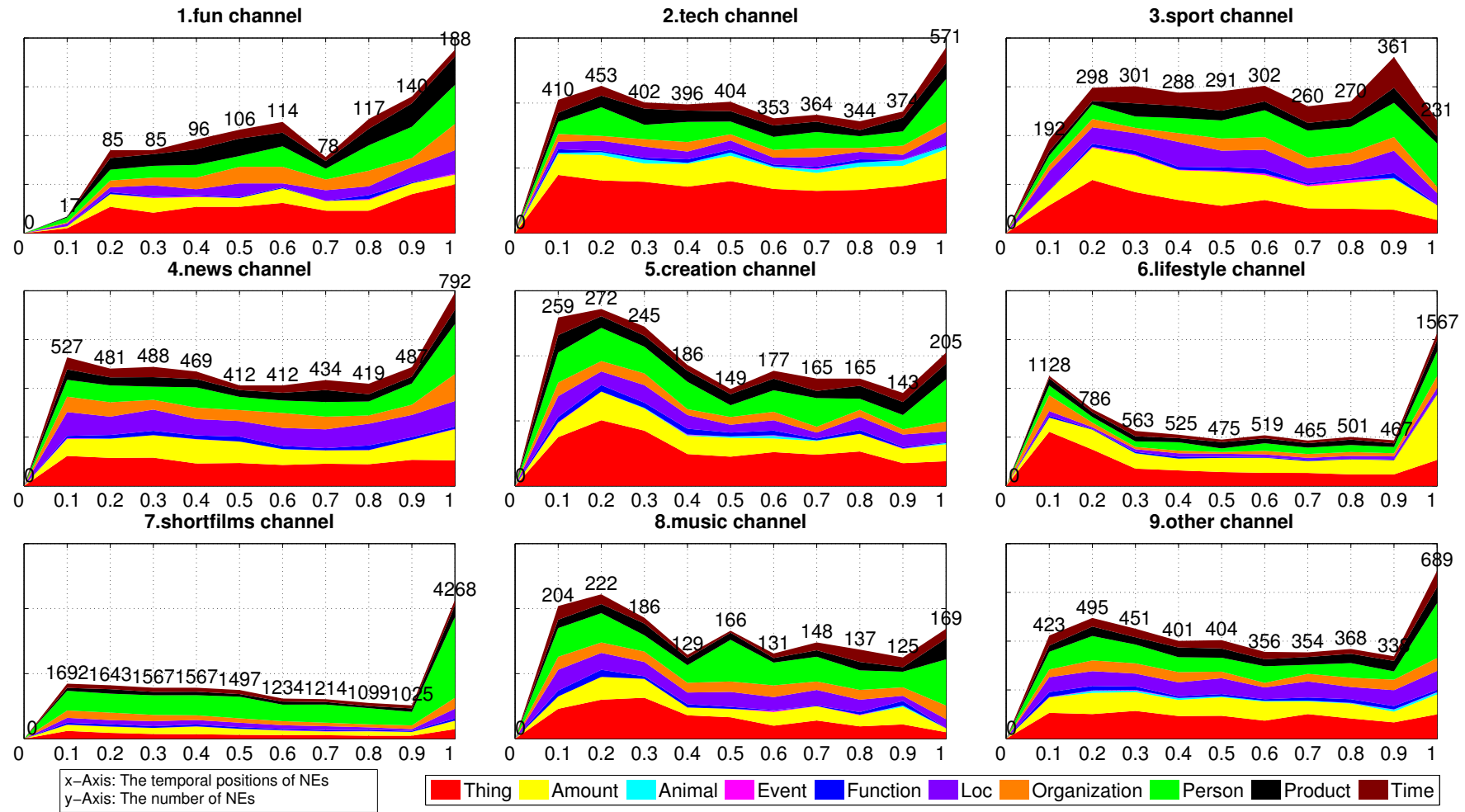


FIGURE 5.5: Distribution of named entities extracted from subtitles for each channel and the summary of their temporal position in the videos

It is obvious in Figure 5.5 that, for some channels, a large number of named entities are aligned with the end of the videos. For *shortfilms*, 4268 named entities are extracted when $tp \in (0.9, 1]$, and a large proportion of them is of type Person. The *lifestyle* channel has spikes both at the beginning and the end. . The fun channel has very few named entities at the beginning and the named entities in *tech*, *news* and other channels have a relatively even distribution when $tp \leq 0.9$ but the numbers are slightly higher at the end. Different from other channels, *sport* has a low number at both the beginning and the end, while it is difficult to see a pattern for *creation* and *music*. If we suppose that the named entities extracted are all correct (which is generally not true, but follows the same trend), these patterns imply some important information that could be useful for video classification and retrieval based on the temporal features.

5.4.2 Methodology

A multiclass classification experiment is conducted for each research question defined in the beginning of this chapter: *Exp1*, *Exp2* and *Exp3*, each of which categorizes the video dataset into the 9 different Dailymotion channels using different features and algorithms. The channel information retrieved from the Dailymotion API is considered as the labelled (ground truth) data.

As shown in Table 5.3, an *id* is assigned to each channel c , and $c \in \{1, 2 \dots 9\}$. In *Exp1*, we use the number of named entities per each NERD type t as the features. As there are 10 NERD types, each observation is a feature vector $\vec{x} = [x_1, x_2 \dots x_{10}]$ and $|\vec{x}| = 10$, where x_t represents the total number of named entities for each NERD type t in a given video. For *Exp2*, the named entities are weighted with their temporal position values tp and grouped into N groups. The feature vector in *Exp2* is $\vec{x} = [x_1, x_2, x_3 \dots x_n]$, where $1 \leq n \leq N$. The choice of N may affect the prediction results: at the beginning, a relatively large number is chosen $N = 20$ and then N is gradually decreased to see how the results change. *Exp3* is a combination of *Exp1* and *Exp2* and the temporal distribution of named entities for each NERD type are used as features. Consequently, there are $10 \times N$ features in *Exp3* and $\vec{x} = [x_{1,1}, x_{1,2} \dots x_{t,n}]$. When $N = 20$, $|\vec{x}| = 200$. *Exp1* is a subset of *Exp3* where $N = 1$.

For question 4, four basic classification algorithms are applied to each experiment: Logistic Regression (LG), K-Nearest Neighbour (KNN), Naive Bayes (NB) and Support Vector Machine (SVM). The mathematical details of each algorithm are out of the scope of this thesis. Instead, an explanation is provided on how the algorithms are used in the experiments. First, as they are supervised algorithms, the dataset needs to be divided into a training set and a test set. To make the full use of the dataset and reduce the overfitting problem of each algorithm, 10-fold cross validation is applied in each experiment. The 805 videos are divided into 10 equal-sized groups and in each fold, 9 groups are the training set and 1 group is the test set. In this way, each video in the dataset

appears only once in the test set. Then, when applying different algorithms to each fold, the results can be generically defined as:

$$\hat{\mathbf{R}} = \text{predict}(\mathbf{X}^e, \mathbf{Y}^e, \mathbf{X}^r, \mathbf{Y}^r, \text{params}) \quad (5.1)$$

\mathbf{X}^r is a $m_r \times |\vec{\mathbf{x}}|$ matrix of the training data, where m_r is the number of all training videos in the 9 groups. \mathbf{Y}^r is a $1 \times m_r$ matrix of grouping variables, where each entry in \mathbf{Y}^r is the labelled channel id c . Similarly, \mathbf{X}^e is a matrix of testing data. Both \mathbf{Y}^e and $\hat{\mathbf{R}}$ are $1 \times m_e$ matrix and m_e is the number of videos in the test set. Each entry in \mathbf{Y}^e is the labelled channel id c for the videos in the test set, while each entry in $\hat{\mathbf{R}}$ is the predicted channel \hat{c} given the feature vector $\vec{\mathbf{x}}$. The actual definition of the *predict* function in Equation 5.1 changes according to the different algorithms used. *params* is a set of parameters that we use to tune each algorithm so that the best results can be obtained.

LG is a statistical machine learning algorithm that uses exponentiation to convert linear predictors to probabilities. For the experiments, the multinomial LG was adopted and the linear predictors are defined as:

$$g(\vec{\mathbf{x}}) = \ln \frac{\pi(\vec{\mathbf{x}})}{1 - \pi(\vec{\mathbf{x}})} = \beta_0 + \sum_{m=1}^{|\vec{\mathbf{x}}|} \beta_m x_m \quad (5.2)$$

The result for each video using the LG classifier is a vector $\vec{\mathbf{r}} = [r_1, r_2 \cdots r_c]$, where r_c is the probability that this video belongs to channel c . In this case, $|\vec{\mathbf{r}}| = 9$ and $c = \text{col}(r_c)$. In the experiment, the channel which has the largest possibility is selected as the final prediction result, i.e.:

$$\hat{c} = \text{col}(\max_{c=1}^9(r_c)) \quad (5.3)$$

To reduce the overfitting problem, the L2-Regularization (Ng, 2004) is applied to the logistic regression and empirically, $\lambda = 0.0001$ was selected as the best bias-variance trade-off. NB is also a statistical machine learning algorithm, but it has many choices for modelling the data distribution. The multivariate multinomial distribution was chosen as it best fits the problem. Similar to LG, Equation 5.3 was applied to get the prediction result \hat{c} in NB.

KNN is an instance based algorithm and the main tuning parameter is the choice of k . As there is still a lack of principled ways to choose the variable k (Kotsiantis, 2007), a series of tests are run with $k = 1, 2, 3 \cdots 40$ in order to find the best k value. The best results were obtained when $k \in [18, 22]$ and $k = 20$ is chosen for all our experiments. Euclidean Distance is selected as the method to calculate the distance between two instances. Unlike other algorithms, SVM cannot be directly applied to multiclass classification problems, so the LIBSVM¹⁶ was used to implement a 1-vs-1 SVM algorithm and the linear kernels $K(x, y) = (x \cdot y + 1)^P$ was chosen as the kernel function for all the

¹⁶<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>

experiments. Finally, to measure the accuracy of each experiment and algorithm, the precision P , recall R and F1-score $F1$ for each channel c are defined as Equations 5.4, 5.5 and 5.6 respectively.

$$P_c = \frac{\sum_{f=1}^{10} |\hat{R}_f(c) \cap Y_f^e(c)|}{\sum_{f=1}^{10} |\hat{R}_f(c)|} \quad (5.4)$$

$$R_c = \frac{\sum_{f=1}^{10} |\hat{R}_f(c) \cap Y_f^e(c)|}{\sum_{f=1}^{10} |Y_f^e(c)|} \quad (5.5)$$

$$F1_c = 2 \times \frac{P_c \times R_c}{P_c + R_c} \quad (5.6)$$

$\hat{R}_f(c)$ is the set of videos that have been predicted as belonging to channel c in the f th fold of cross validation, while $Y_f^e(c)$ is the set of videos that have been labelled in channel c . Therefore, $|\hat{R}_f(c) \cap Y_f^e(c)|$ is the number of videos that are correctly categorized in channel c in a cross validation fold. There is the possibility that $\sum_{f=1}^{10} |\hat{R}_f(c)| = 0$ if no video has been categorized in the channel c . In this case, the value of P_c is NaN . The dataset has videos in all channels, so $\sum_{f=1}^{10} |Y_f^e(c)| \neq 0$. To evaluate the overall accuracy of the algorithm in each experiment on the entire dataset, acc is calculated as:

$$acc = \frac{\sum_{c=1}^9 \sum_{f=1}^{10} |\hat{R}_f(c) \cap Y_f^e(c)|}{805} \quad (5.7)$$

The overall accuracy is the total number of videos that have been correctly classified divided by the total number of the videos since each video appears exactly once in the test set.

5.4.3 Experiments and Discussion

This section analyses the results obtained from each experiment in order to see which set of features and which algorithm(s) perform best for the automatic video classification task, depending on the channels.

5.4.3.1 Overall Accuracy

Figure 5.6 shows the overall accuracy for each experiment. Three grouping numbers $N = 5, 10, 20$ were tried and in most of the experiments, $N = 20$ outperformed the other numbers. So the experimental results presented below for Exp2 and Exp3 were obtained when $N = 20$. The most accurate was KNN-Exp1 (46.58%) and the worst was LG-Exp3 (33.54%). Generally speaking, there were no major differences in accuracy between the algorithms using different sets of features. The features chosen in Exp1 performed better than the other two feature sets using LG and KNN. For NB, acc for Exp1 and Exp3 are close and they are both better than Exp2. SVM-Exp3 outperformed Exp1 and Exp2 and it was also the most accurate for Exp3 compared to the other algorithms.

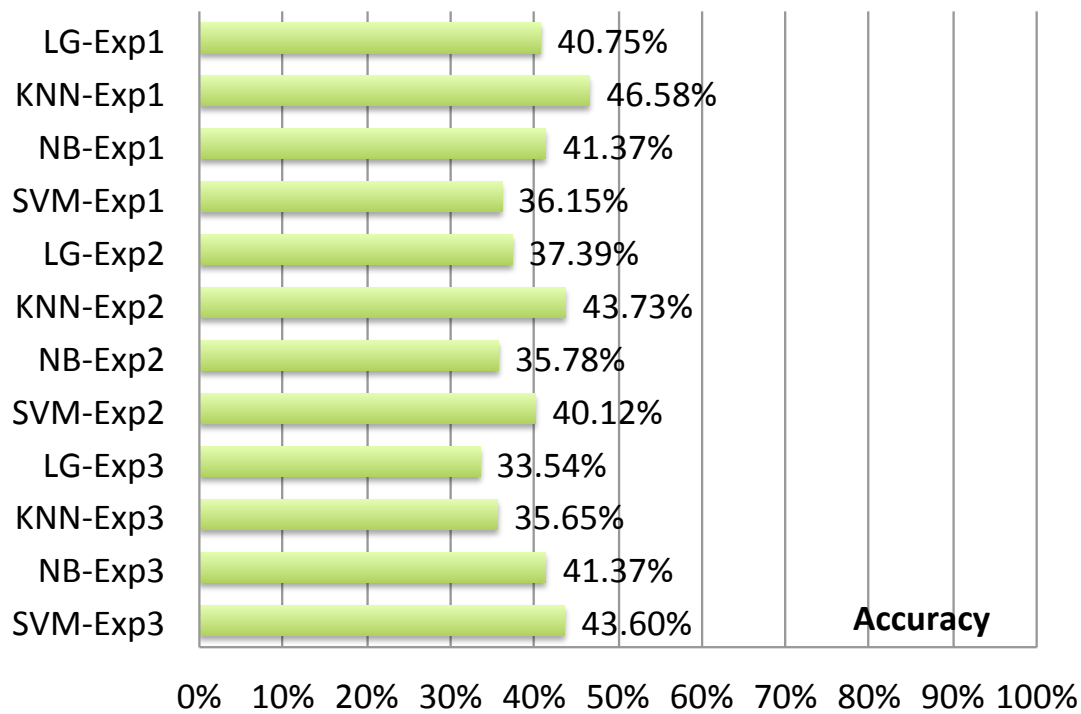


FIGURE 5.6: Accuracy Comparison for each Algorithm-Experiment Pair

No conclusion could be drawn regarding which algorithm and feature set combination performed best (Figure 5.6). However, if the feature set includes the breakdown of named entities based on NERD types (Exp1 and Exp3), the accuracy is quite likely to be better than the ones using only temporal positions (Exp2). From this point of view, it is possible to infer that the number of named entities and their type is an indicator to be taken into account for improving the video classification algorithm, assuming of course that there are a sufficient number of named entities detected for each NERD type.

5.4.3.2 Breakdown scores per Channel

Tables 5.5, 5.6, 5.7 and 5.8 show the scores for each channel and experiment by precision, recall and $F1$. The three largest numbers for each measurement are highlighted in bold. If $F1$ is used as a general measure of accuracy, *sport*, *life* and *shortfilms* usually achieve the best accuracy in the different experiments and using different regression algorithms, while the $F1$ of *news*, *creation*, *music* and *other* are usually below 20%. This behaviour makes sense since the number of samples available for that first set of channels was bigger than for the second group, therefore the training phase of the classification performed better.

Using LG with $\lambda = 0.0001$, *lifestyle* and *shortfilms* consistently gained high accuracy in all three experiments. All P , R and $F1$ scores are high for *shortfilms* in Exp1,

TABLE 5.5: Precision (P), Recall (R) and F-measure (F1) on different channels for the experiments using Logistic Regressions (%), $\lambda = 0.0001$.

	Exp1			Exp2			Exp3		
Ch.	P	R	F1	P	R	F1	P	R	F1
fun	28.87	29.17	29.02	35.71	31.25	33.33	18.87	20.83	19.8
tech	33.33	15.91	21.54	24	13.64	17.39	17.54	22.73	19.8
sport	35.69	71.17	47.54	32.18	68.71	43.84	38.82	36.2	37.46
news	32.26	15.15	20.62	30.77	12.12	17.39	15.39	18.18	16.67
creat	8.33	1.82	2.99	5.26	1.82	2.7	7.02	7.27	7.14
life	49.78	58.25	53.68	50.23	56.19	53.04	57.9	56.7	57.29
films	73.13	60.49	66.22	66.67	41.98	51.52	54.76	56.79	55.76
music	NaN	0	0	5.88	2.38	3.39	10	11.91	10.87
other	16	6.25	8.99	0	0	0	12.9	6.25	8.42

TABLE 5.6: Precision (P), Recall (R) and F-measure (F1) on Various Channels for the Experiments using K-Nearest Neighbour (%), $k = 20$.

	*Exp1			Exp2			Exp3		
Ch.	P	R	F1	P	R	F1	P	*R	F1
fun	23.91	22.92	23.4	47.69	32.29	38.51	21.05	20.83	20.94
tech	45	20.46	28.13	37.93	25	30.14	42.86	6.82	11.76
sport	50	66.87	57.22	42.48	58.9	49.36	29.86	26.38	28.01
news	54.17	19.7	28.89	18.75	9.09	12.24	42.86	4.55	8.22
creat	28.57	18.18	22.22	6.25	1.82	2.82	33.33	1.82	3.45
life	48.01	74.74	58.47	44.04	81.96	57.3	34.36	86.08	49.12
films	72.29	74.07	73.17	86	53.09	65.65	80.65	61.73	69.93
music	20	2.38	4.26	20	2.38	4.26	NaN	0	0
other	23.08	9.38	13.33	19.05	6.25	9.41	0	0	0

TABLE 5.7: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Naive Bayes (%).

	Exp1			Exp2			Exp3		
Ch.	P	R	F1	P	R	F1	P	R	F1
fun	31.82	29.17	30.43	18.75	12.5	15	22.68	22.92	22.8
tech	40.74	25	30.99	30.77	9.09	14.04	28.26	29.55	28.89
sport	44.87	42.95	43.89	32.89	60.74	42.67	47.4	55.83	51.27
news	29.83	25.76	27.64	38.46	15.15	21.74	33.33	25.76	29.06
creat	26.32	9.09	13.51	9.09	5.45	6.82	13.51	9.09	10.87
life	44.06	72.68	54.86	46.28	60.83	52.56	52.56	63.4	57.48
films	55.77	71.61	62.7	62.9	48.15	54.55	61.18	64.2	62.65
music	12	7.14	8.96	3.7	2.38	2.9	19.36	14.29	16.44
other	0	0	0	8.33	3.13	4.55	12.5	6.25	8.33

TABLE 5.8: Precision (P), recall (R) and F-measure (F1) on various channels for the experiments using Support Vector Machine (%).

	Exp1			Exp2			*Exp3		
Ch.	P	*R	F1	P	R	F1	P	*R	F1
fun	33.33	8.33	13.33	45.71	16.67	24.43	52.63	20.83	29.85
tech	NaN	0	0	NaN	0	0	26.92	15.91	20
sport	50	62.58	55.59	43.48	61.35	50.89	34.78	88.34	49.91
news	50	4.55	8.33	0	0	0	25.81	12.12	16.49
creat	26.67	7.27	11.43	37.5	10.91	16.9	0	0	0
life	31.37	87.63	46.2	49.1	70.1	57.75	66.47	57.22	61.5
films	36.36	4.94	8.7	26.75	80.25	40.12	49.17	72.84	58.71
music	NaN	0	0	NaN	0	0	NaN	0	0
other	0	0	0	NaN	0	0	40	3.13	5.8

but the *F1* for *creation*, *music* and *other* is very low ($\leq 10\%$). When the temporal distribution of media fragments is considered (Exp3), the *F1* for *sport* and *shortfilms* is lower than Exp1, but *creation* and *music* are improved. For KNN, the *P*, *R* and *F1* are all above 70% for *shortfilms* in Exp1, which is the overall best score. Compared with Exp1, Exp2 and Exp3 obtained worst results for nearly all channels. In Exp3, no instances were correctly recognised in *music* and *other*. Using NB, *F1* for *sport*, *lifestyle* and *shortfilms* were good in both Exp1 and Exp3. SVM performed better when dealing with multi-dimensional data, so the best result for SVM was in Exp3 where 200 features were used for the classification. SVM also generated the best *F1* for *lifestyle* (61.5%) compared to all the other algorithms. But unlike other algorithms, the accuracy for the *shortfilms* channel in SVM-Exp1 was very low, while *sport* and *lifestyle* were still high. This is due to the fact that SVM relies more on the size of the samples when the size of the features are small.

Algorithms such as LG and SVM require a large training dataset to achieve better classification results (Kotsiantis, 2007). Hence, channels with a large sample size (*sport* and *lifestyle*) are more likely to obtain high accuracy in most of the algorithms. However, even though the sample size of *shortfilms* was not big, the *NEs/Videos* value in Table 5.3 is much larger than for the others channels. This is because the average length of the video in the *shortfilms* channel is longer than the videos in other channels and more named entities can be extracted from their subtitles. Considering the use of media fragments in this experiment, the characteristics of temporal and NERD type distribution of named entities for *shortfilms* are also exceptional: a large number of named entities are associated with the end of the videos and most of them are Person. So considering those two factors and the sample size of *shortfilms*, it is possible to understand why the accuracy of this channel was higher for most of the experiments. Sample size is still the key factor for SVM regression in this context.

Some algorithms achieved a very high recall score but pretty low precision in some experiments. For example, the *R* of *lifestyle* in KNN-Exp3 (Table 5.6) is very high

(86.08%), but the P is very low (34.36%). Taking a deeper look at the content of the datasets, it is possible to see that there were 194 videos in *lifestyle*, but in the results of the classification, many more instances were marked as belonging to this channel (486 in total). However, 319 of them have been wrongly categorized. Similar situations occurred in the *lifestyle* channel in SVM-Exp1 and in the *sport* channel in SVM-Exp3 (Table 5.8). In some channels, the classification accuracy is very low when not enough instances are predicted to belong to this channel. These two phenomena, together with Table 5.3 and Figure 5.5, demonstrate that channels with very clear entity distribution patterns or with large sample size (e.g. *sport*, *lifestyle* and *shortfilms*) will tend to have high R but low P . So it is safe to conclude that classification can be improved with a larger sample size but also by investigating which features will be the most influential for each algorithm.

5.4.4 Summary of the Video Classification

This section discussed a new way for classifying video based on the Core Model for Media Fragment Enrichment and Media Fragment Enriching Framework. It demonstrated that, with the help of interlinked media fragments, a new research area could be explored using the annotations linked to media fragments. The results obtained from the three experiments indicate that the method implemented is very promising in the context of the classification of online videos.

Summarizing the experiment results, positive answers can be achieved for the first three research questions described at the beginning of this chapter.

Question 1 : is the number of named entities for each NERD type correlated with a channel?

Question 2 : is the total number of named entities across the different temporal groups correlated with the channels?

Question 3 : are the number of named entities for every NERD type and temporal group correlated with the channels?

It is also clear that there is no dominant algorithm, which outperforms the others for the three experiments in terms of the overall accuracy given the dataset used. When using named entities and media fragment features together, SVM obtains the best overall result. Among the individual channels, *sport*, *lifestyle* and *shortfilms* have the highest prediction accuracy. obtained when using the number of named entities in each NERD type and the KNN ($k = 20$) algorithm to predict videos in the *shortfilms* channel. Apart from the type of regression algorithms used, three different factors are observed that affect the prediction accuracy in each channel: the sample size, the average number

of named entities in each NERD type per video, and the temporal position distribution of the entities within the media item. Among those three factors, the sample size can be increased when collecting more data, but the other two may follow some distributions for each channel, which requires further investigation.

As the dataset used in this experiment has not been used in other similar research, it is difficult to compare the algorithms in this experiment with other classification methods. Therefore, the most important future work to extend this experiment is to apply the methodology to some widely-used video archive with subtitles. Then a valid baseline can be created to see whether the proposed methodology can achieve better classification results. Meanwhile, it is also interesting to determine the classification results for videos on other platforms and to see if the patterns observed are similar in YouTube and in Vimeo compared with Dailymotion. By contrast, this experiment can help to study how accurate human classification is with respect to automatic classification based on the video metadata. The proposed new features regarding named entities and media fragments can be combined with other features, either low-level or high-level, to improve video classification and retrieval.

5.5 Summary

This Chapter focused on the design and implementation for R3 RDF description of media fragments and R4 media fragments interlinking. Section 5.1 designed an RDF model revealed what exact information should be included in the core model to connect media fragments to the Linked Data Cloud. This is another step forward from Chapter 4 in the publishing of media fragments as Linked Data. The Core Model for Media Fragments Enrichment was established using well-known vocabularies, such as W3C-MA, NinSuna Ontology, the Open Annotation Model, and W3C Provenance Ontology. In this model, the media fragments are linked to URIs in the Linked Data Cloud via the Open Annotation Model. In the example shown in this chapter, the annotation body is a named entity, but it could be any URI that semantically annotates the media fragments. The importance of the Core Model for Media Fragment is that it enables media fragment discovery through querying annotations, and it improves the online presence of media fragments for semantic aware agents.

The Media Fragment Enriching Framework is an implementation of R4, which automates the process of producing instances of the core model of interlinking media fragments, so that media fragments' annotations can be automatically created and published in massive quantities. The basic idea of the framework is to retrieve text metadata and timed-text via Web APIs exposed by online video sharing platforms, extract named entities and produce instances of the Core Model for Media Fragment Enrichment with the timeline alignment information of the video. The framework is designed for both

developers who have control of the raw multimedia resources and for developers who want to reuse the data from existing online video sharing applications, such as YouTube and Dailymotion. This framework can have many implementations depending on the requirement of a specific application and what extractors are used in the implementation. For example, the developers can use methods other than Web API for the acquisition of “metadata and timed-text” (see the second step in Figure 5.3).

When R1 to R4 are satisfied, it is safe to say that media fragments are published into the LOD Cloud following the Linked Data principles. Then further reasoning (UC1 in Section 3.1.2), queries (UC3 in Section 3.1.3), and other curation services can be built on top of such datasets. More importantly, the implementations introduced in this chapter can be seamlessly integrated with the current video/audio sharing platforms in an automatic manner, so that massive quantities of media fragments can be created and integrated into the Web of data without manual interference (R5).

To demonstrate the use of interlinked media fragments, two implementations were given in Section 5.3 and 5.4. The enrichment of YouTube demonstrates how the Media Fragment Enriching Framework could be applied to the YouTube platform and the further enriched data is re-published into the LOD Cloud. The video classification tasks for Dailymotion videos indicate the enrichment information, including the named entities and time alignment of those entities, could be used in video classifications together with machine learning techniques. The results have shown that this methodology is very promising even though further research is still needed in this area.

None of the implementations introduced in this chapter are domain specific, and rely only on named entity extractions to link media fragments to the LOD Cloud. There are many use cases to which media fragments should be applied with domain specific vocabularies, and the linkage can be constructed by means other than named entities. For example, UC4, UC5 and UC6 described in Section 3.2 are obviously different from the case of annotating “general” YouTube videos. The UK Parliamentary Debate videos in UC4 focus totally on the debate itself, i.e. “who speaks what and when”. So the basic model to describe the debate is obviously “event-driven”. Each speech in the debate is an event and it is important to model such relationships in the RDF description of the media fragments. Meanwhile, the “event” will be the central glue of different pieces of data instead of media fragments and annotations. Further discussion on the implementation of the framework can be found in Section 8.2.1.

Chapter 6

Visualisation of Media Fragments and Semantic Annotations

Previous chapters have introduced implementations of R1 to R5. This Chapter focuses on R6, the visualisation of media fragments and annotations. Recall:

R6 when a media fragment URI is opened in the browser, the media fragment should be visualised properly, including:

- R6.1: highlighting the dimensions encoded in a media fragment URI by the media fragment player (UC1)
- R6.2: highlighting directly the media fragment and annotations, if any, when opening the URI in the browser (UC1)

In the Media Fragment Enriching Framework (Figure 5.3), those functions should be implemented in the Media Fragment Enricher UI. If R6.1 and R6.2 are examined separately, two components need to be addressed: the video player that can highlight media fragments in both temporal and spatial dimensions and an interactive panel to display annotations with named entities. This chapter will cover the design and evaluation of both components.

6.1 Synote Media Fragment Player

This section introduces the Synote Media Fragment Player (smfplayer), which is a client-side implementation of W3C-MFURI. The smfplayer was developed to visualise both temporal and spatial dimensions of a media fragment. It is evaluated on different devices and platforms to check its compatibility.

6.1.1 Motivation

The development of the Synote Media Fragment Player¹ comes from the idea of highlighting a media fragment and synchronously displaying video/audio with annotations in Synote. The original video player in Synote presented in Li et al. (2009)

was designed to be compatible with various browsers on different platforms when the specification of a native HTML5 player was not widely applied in major browsers and Adobes Flash Player still dominated at that time. One limitation of the Flash Player is that it only supports some video codecs. Videos containers like Windows Media Video (WMV) and Audio Video Interleaved (AVI) will not be able to be decoded by Flash Player. Synote wanted to play as many formats of video/audio as possible without designing its own player. So considering this limitation, Synote designed a program to automatically select the best “embeddable native player” according to the browsers, platforms and the container of the video/audio file. For example, Windows Media Player² is not available on Mac OS and Linux, so Synote will choose VLC player instead, if available. Synote also normalised the APIs provided by different embedded players and defined universal control APIs for javascript. Those APIs include controls such as play, pause, rewind, set current position, etc. This video player was the predecessor of the smfplayer

As the native video/audio player defined in HTML5 was adopted by more and more browsers, especially those on mobile devices, the embedded native players, such as Windows Media Player, Quicktime³ and VLC were gradually replaced by HTML5, plus Flash Player as a “fallback” method for video/audio files not yet supported by HTML5. Such examples include JWPlayer⁴ and Flowplayer⁵. However, as different browsers adopted different codecs for videos, there was still no video container that could be played in native players across all the major browsers without further assistance from other programs. Meanwhile, there emerged a “fallforward” way to solve this problem, a term coined by MediaElement.js⁶. Instead of offering separate Flash players to older browsers, MediaElement.js builds Flash and Silverlight plugins that mimic the HTML5 MediaElement API⁷. In this way, developers can always use `<video>` and `<audio>` tags defined in HTML5 even on old browsers that do not support HTML5 and MediaElement.js will select the best way (native, Flash or Silverlight player) to play the video/audio. There are still gaps since some devices or browsers will not support a certain container format and the file cannot be played whichever player is chosen. For example, iOS does not support Flash video (flv) any longer⁸ and WMV file is not supported by either iOS

¹<http://smfplayer.synote.org/smfplayer/>

²<http://windows.microsoft.com/en-GB/windows/windows-media-player>

³<http://www.apple.com/uk/quicktime>

⁴<http://www.jwplayer.com/>

⁵<http://flowplayer.org/>

⁶<http://mediaelementjs.com/>

⁷<https://developer.mozilla.org/en/docs/Web/API/HTMLMediaElement>

⁸http://en.wikipedia.org/wiki/Apple_and_Adobe_Flash_controversy

and Android. A more detailed discussion about this problem can be found in (Wald et al., 2013). Regardless of those problems created by the browser vendors as they follow different specifications, MediaElement.js is by far the best solution for Synote, where many video/audio formats need to be played. In a later version of Synote, the original video player was replaced by MediaElement.js (See Figure 6.2). Another advantage of MediaElement.js is that it supports chromeless player from YouTube and Dailymotion⁹

With this in mind, the development task then becomes designing a client-side implementation compliance with W3C-MFURI. W3C-MFURI is intended for user-agents, so to be more precise, the smfplayer is “polyfill” as it uses javascript to implement the media fragment highlighting function, which should be implemented by user-agents natively. The smfplayer is not a plugin of a user-agent, so it cannot interfere with the user-agent interactions with the **Multimedia Host Server** as discussed in Section 4.2. However, the polyfill can help to highlight the temporal and spatial fragments to make it “look like” the client-server interaction has been successfully processed.

6.1.2 Implementation and Evaluation of the smfplayer

The smfplayer is a jQuery¹⁰ plugin that highlights both temporal and spatial fragments of online video/audio files and videos from video sharing platforms, such as YouTube and Dailymotion. The smfplayer wraps up MediaElement.js and builds an extra layer for parsing¹¹, highlighting and controlling the replay of media fragments. Figure 6.1 is a screenshot of the smfplayer highlighting temporal and spatial fragments of a YouTube video. In this example, the URL of the YouTube video is:

`http://www.youtube.com/watch?v=Ug6XAw6hzaw`

and two dimensions need to be highlighted, which are temporal fragment from 1:30 to 1:33 when Tim Berners-Lee is interviewed, and spatial fragment “xywh=40,220,200,50” for the player with 480x320 resolution. According to the first principle of Interlinking Media Fragments, attaching the hash fragment directly to the end of this URL will generate a Media Fragment URI:

`http://www.youtube.com/watch?v=Ug6XAw6hzaw#t=00:01:30,00:01:33&xywh=40,220,200,50`

To visualise the temporal fragment, the smfplayer will start playing the video from 1:30 and pause at 1:33 after the video is loaded, and the spatial fragment is highlighted by

⁹MediaElement.js does not support Dailymotion videos in the master releases yet. However, another fork of the player has implemented this function: <https://github.com/johndyer/mediaelement/pull/663>

¹⁰<http://jquery.com/>

¹¹The parsing function is built on Thomas Steiner’s Media Fragments URI parser <https://github.com/tomayac/Media-Fragments-URI/>

a rectangle overlaid above the player. Since the overlay does not apply to the player in full-screen mode, the spatial highlighting will not act properly if the video is played in this mode. This problem becomes more serious when the smfplayer is used on smaller screens such as mobile phones, where the video plays at full-screen by default. As a jQuery plugin, initialising the smfplayer is very simple. See Listing 6.1 as an example.

```
<div id='player_div'></div>
<script type='text/javascript'>
  var uriStr = 'http://path/to/file.mp4#t=11,19&xywh=120,120,120,120';
  $(document).ready(function(){
    var mfuri = uriStr;
    var player = $('#player_div').smfplayer(
      /*smfplayer options*/
      {
        mfURI:mfuri,
        autoStart:true,
        width:480,
        height:320
      }
    );
  });
</script>
```

LISTING 6.1: Example code of initialising smfplayer

To reveal the compatibility of the smfplayer on different browsers and platforms, tests were conducted to play an MP4 (H.264) file and see whether both temporal and spatial media fragments can be successfully highlighted. Table 6.1 shows the test results with an explanation of why a test did not fully pass. The smfplayer passed most of the tests using desktop browsers. The only problem (issue 2 in the table) happened in Safari for the MP4 file, where the position of the start time in the smfplayer was not “exactly” the start time defined in the media fragment URI. Actually, other browsers had a similar issue, but the errors they committed are under one second, so were not noticeable. The time-seeking in Safari can sometimes be two seconds before (or after) the actual start time requested.



FIGURE 6.1: Screenshot of Synote Media Fragment Player

TABLE 6.1: Compatibility of Synote Media Fragment Player with different browsers and devices (Test file container MP4, video codec H.264).

Browsers	IE	Firefox	Chrome	Safari	Opera	iPad	Android Tablet	iPhone	Android Phone
Version	9+	19.0.2	25.0.x	6.0.3	12.14	6.1.x1	3.2.1	6.1.x1	2.3.51
Device/OS	Windows 7 x64	Mac OS 10.7.5	Mac OS 10.7.5	Mac OS 10.7.5	Mac OS 10.7.5	iPad3 ¹	Samsung Galaxy Tab 7.0 Plus ¹	iPhone 3GS ¹	HTC Sensation XL ¹
File	Pass	Pass	Pass	Partial ²	Pass	Partial ³	Failed ⁷	Partial ⁶	Partial ⁶
YouTube	Pass	Pass	Pass	Pass	Pass	Failed ⁴	Pass	Partial ⁶	Partial ⁸
Dailymotion	Pass	Pass	Pass	Pass	Pass	Failed ⁵	Pass	Failed ⁵	Partial ⁸

1. Native browsers, i.e. Safari for iOS and Chrome for Android, were used for the test.
2. There are issues when start playing.
3. The player would not jump to the start time when playing started, but paused at the end time.
4. The player had problems connecting to YouTube, with error message: Unable to post message to <http://www.youtube.com>. Recipient has origin <http://yourdomain>.
5. Could not load the player.
6. Temporal fragment played correctly, but the spatial highlight could not be displayed due to full screen mode.
7. Could not correctly load metadata.
8. Temporal fragment played correctly, but the spatial highlight could not be displayed even though the video was not played in full-screen mode.

On tablets and mobile phones, the smfplayer failed or partially failed in most of the tests. Issue 3 results from the operation of seeking start time of the media fragment being sent before the media is successfully loaded, so the player will just start playing from the very beginning. Issues 4, 5, 6, 7 and 8 are mainly due to incorrect loading or initialising the MediaElement.js player, so the detailed reason for those issues are out of the scope of this thesis. In the next section, the smfplayer will be applied as part of the Media Fragment Enricher UI.

6.2 Visualise Media Fragments with Annotations

The second component in the Media Fragment Enrichment UI is an interactive panel to visualise annotations together with media fragments. As the annotations are aligned with media fragments, the interaction links between the smfplayer and the annotation panel needs to be set up properly, in order that both media fragments and related annotations can be highlighted correctly when the video is playing. With this in mind, the layout of the UI follows the design of Synote Player, where video/audio, transcript and user-generated annotations are displayed in a synchronised manner (Li et al., 2011b).

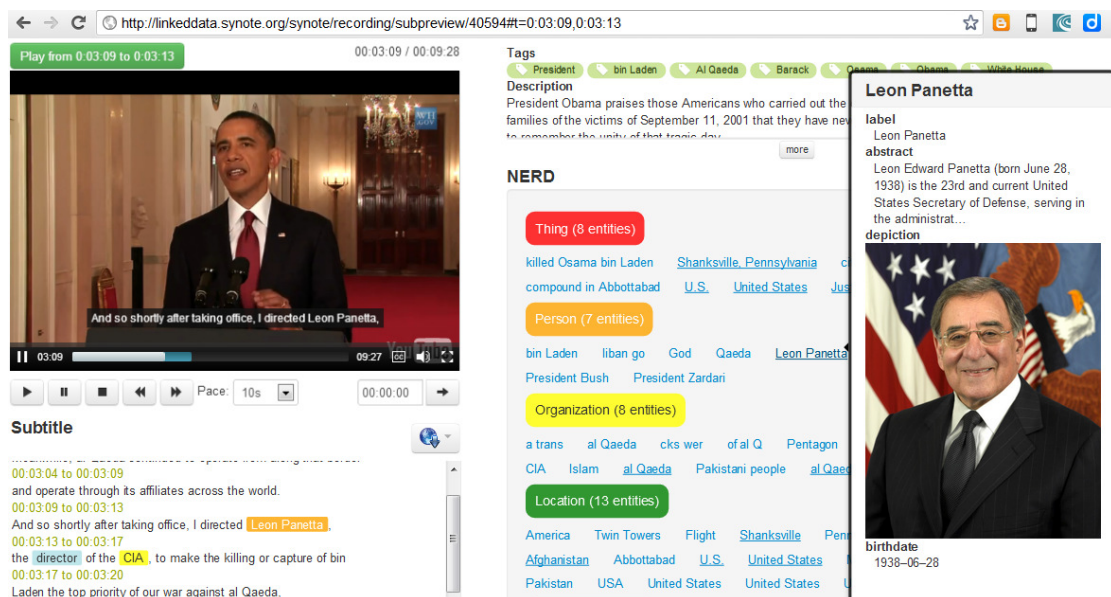


FIGURE 6.2: Screenshot of Media Fragment Enricher

Figure 6.2 is the screenshot of a preview page. The right column displays the named entities found, grouped according to the 9 main NERD categories. The YouTube video is included in the left column using the smfplayer together with the interactive subtitles. The named entities are highlighted in different colours according to their categories. If a media fragment is used in the preview page URI, the video starts playing from the media fragment start time and stops playing when the end time is reached. When clicking on a named entity, the video jumps to the media fragment that corresponds to

the subtitle block from where the named entity has been extracted. If a named entity has been disambiguated with a DBpedia URI, the entity is underlined. In addition, when the entity is hovered over, a pop-up window shows additional information such as the generic label, abstract and depiction properties. For named entities of type Person, the birth date is displayed while latitude and longitude information are given for Location.

6.3 Summary

The visualisation of media fragments and annotations (R6) has been identified as a challenge and many development efforts, as well as research, have been devoted towards its solution. For this reason, this chapter reported the design of an open source and reusable media fragment player, Synote Media Fragment Player, which is a browser polyfill to highlight temporal and spatial media fragments. The player is an important part of media fragment and named entities visualisation in the Media Fragment Enrichment UI, which was covered in Section 6.2. The most important feature of the Media Fragment Enrichment UI is the alignment and interaction between the media player and annotations, so that corresponding annotations can be highlighted while playing the video.

The visualisation of media fragments is an ongoing work and a lot of effort is still needed to work towards a native implementation instead of a browser polyfill. However, the smfplayer is still a good solution to benefit end users for the application of media fragments.

Chapter 7

Automatic Media Fragments Indexing For Search Engines

An important feature of media fragment semantics is the indexing of the media fragments through the annotations exposed by the Linked Data format. Requirements R1 to R6 have been resolved in the previous chapters. This chapter will discuss requirement R7 regarding how the solutions of R1 to R6 can be applied to obtain better indexing of media fragment content and providing a user friendly experience to search multimedia resources at a fine-grained level.

The example of Media Fragment Enriching Framework showed that, with the help of Web API and named entity recognition tools, media fragments can be automatically linked to the named entities in the LOD Cloud, and the RDF description of the media fragments can be published as Linked Data. A semantic-aware agent can therefore query such a dataset and obtain the RDF descriptions of a certain media fragment. Meanwhile, with the development of techniques for embedding semantic markups in (X)HTML, such as RDFa and Microdata, Web search engines that are based on crawling the HTML pages and returning pages based on user-provided keywords, start to “understand” the meaning of the keywords and can match the keywords to the intention of the query. Web pages with embedded semantic markups can thus obtain better ranking in the search results. Schema.org also defines *VideoObject* as the primary object for embedding a structured description in the web pages, mainly to serve videos. From this point of view, embedding semantic descriptions using RDFa or schema.org into the landing page of a video will help the indexing of the video.

Even though major search engines can deserialize the structured video description from web pages, indexing and searching multimedia resources at a fine-grained level is still far away from us. Some applications, like Synote (Li et al., 2011b), have built-in functions to index annotations associated with media fragments and make them searchable by

keywords, but most search engines, such as Google, Yahoo! and Bing¹, cannot return links to media fragments as search results. Many online video sharing applications like YouTube have deep-linking to a certain temporal fragment of the video and the URI is publicly available. So theoretically, search engines can crawl the webpage that the media fragment URI points to, together with the semantic markups embedded in the page. They should also be able to match the keywords supplied by users and return the media fragment URI as the search result. However, this is currently not the case.

The problem lies in the convention of using the landing page as the viewpoint of a video. The real situation is that multimedia resources are seldom displayed alone on the webpage. Usually, a multimedia player is embedded in the landing page together with all metadata and annotations. So even though applications like YouTube can provide a deep-linking to a certain point of the video, retrieving that media fragment URI will result in the same landing page. In the following YouTube example, the two URIs are pointing to the same web page with the same title, description, comments, related videos and other content on the page:

```
http://www.youtube.com/watch?v=Wm15rvkifPc  
http://www.youtube.com/watch?v=Wm15rvkifPc#t=120
```

There is nothing wrong with this behaviour, but this convention makes media fragment indexing really difficult. Even though some annotations are semantically about a certain media fragment, they are buried together with annotations about other media fragments or with the whole video on the same page. Sometimes, the annotations are associated with a certain time point of the video and javascript is used to control the interactivity between those annotations and the video. For example, users can make a timed comment for a YouTube video and clicking on the time will trigger the YouTube Player to jump to that time point. But again, this operation does not lead to another page referred to by a different media fragment URI. Search engines thus will not be able to distinguish which annotation is related to which media fragment and the keyword search will still lead to the landing page with all the annotations.

To resolve this issue, Section 7.1 below introduces the Media Fragments Indexing Framework to enable major search engines, such as Google and Bing, to index media fragments using related annotations, so that they can be found by keyword search. This model follows the Interlinking Media Fragments Principles listed in Chapter 4 and uses W3C-MFURI syntax. Section 7.2 and 7.3 demonstrates an expanded framework with annotation data collected from social media such as Twitter. Section 7.2 describes a survey to determine which video sharing applications have functions similar to the deep-linking of YouTube. The question is: Are most of the videos that users view and share on social

¹<http://www.bing.com>

media “media fragment ready”? Based on the survey results, Section 7.3 runs an experiment to collect the Tweets with media fragment URIs and extract that information using the Media Fragments Indexing Framework so that the media fragment URIs can be indexed by Google.

7.1 Media Fragments Indexing Framework

This section introduces the Media Fragments Indexing Framework, which will help developers of video sharing applications to optimise the media fragment presence for major search engines. The section starts with analysis of the problems and reveals the key points required to solve them. An algorithm is then designed using Google’s Ajax crawling infrastructure to index the media fragments. The implementation of the framework is evaluated using Google as the target search engine where media fragments are supposed to be indexed. Other evaluations are also carried out using Microdata validation tools to show that the Microdata (schema.org) embedded in the Web pages are valid.

7.1.1 Problem Analysis

The online presence of media fragments is currently very poor. This is partly because many applications do not provide media fragments at all. The tags, descriptions and other forms of annotations are on the whole at multimedia level. Even though some applications provide deep-linkings in video and synchronised annotations, they are loaded together with the whole multimedia resource on the same logical page. This is reasonable as it provides an interactive experience for users. For example, TED Talks² and YouTube’s interactive transcript³ allow users to click on the transcript block and the media player embedded on the page will start playing from that time point.

This user-friendly function is not search-engine-friendly for two reasons. First, most search engines only fetch pages as a direct response from the server. So any dynamically generated content on the client-side is ignored. YouTube loads interactive transcript via Ajax, so the transcript will not be indexed by Google from the replay page. TED does not have this problem because the transcript is generated by a server-side script. Secondly, as media fragments and annotations share the same HTML page with the whole multimedia resource, search engines cannot find a unique logical page for each media fragment. When a keyword is searched, the search engine can only return the whole page and users will lose track of which media fragment is related to the keyword. This also causes accessibility problems for devices with low bandwidth, because much traffic is wasted on downloading unnecessary data.

²<http://www.ted.com/talks>

³<http://goo.gl/tinMj>

One solution to these two problems is slicing the whole page into different pages according to media fragments, but the interactive experience will be lost in that users cannot watch different media fragments on the same page. It is not good practice either to ask users to go through different Web pages in order to watch the entire video. The acceptable solution for both multimedia applications and end users must satisfy the following criteria expanding R5 and R6:

- Keyword search should only return the documents directly related to the media fragments, i.e. all the content in the document should annotate this media fragment. It is better if the media fragment could be highlighted in the search results. (R6)
- The interactive experience should be kept. But when users click the link in the search result, the media fragment corresponding to the link should be highlighted. (R6)
- Few changes to the server are required (R5)

7.1.2 Implementation

Google has developed a framework for crawling Ajax applications (Figure 7.1). If the “hashbang” token (“#!”) is included in the original URL⁴, Google crawler will know that this page contains Ajax content. The crawler will then request “ugly URL”. On receiving this “ugly URL” request, the server can return the snapshot page representing the page after the dynamic information has been fully generated by javascript. The content in the snapshot page will be indexed in the original “pretty URL”.

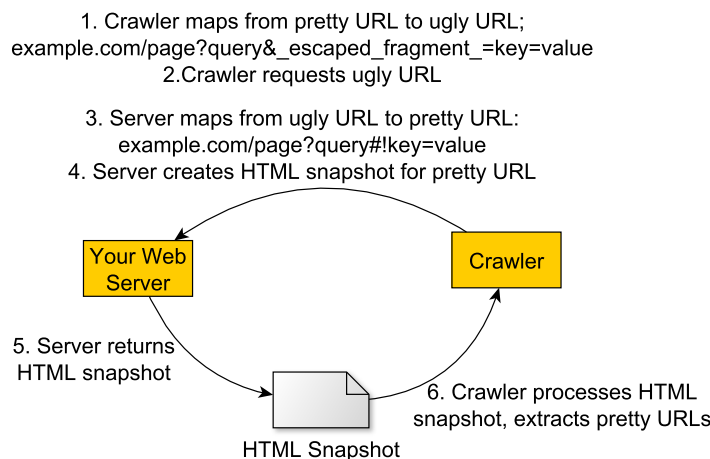


FIGURE 7.1: Google Ajax Crawler

³<http://goo.gl/dPc81>

⁴The HTTP protocol and domain names will be ignored in the URLs following in this chapter.

In this framework, the crawler does not care how the server generates the snapshot page, so, to make use of this framework, developers can keep two sets of pages: the original landing pages and the snapshot pages. Developers do not need to discard the existing landing pages, but manage to cut one landing page into many snapshot pages based on the time span defined in the “_escaped_fragment_” parameter in the “ugly URL”. Each snapshot page contains metadata and annotations only related to the requested URL. The hash in W3C-MFURI syntax, however, is replaced by hashbang as “#!” is the token required by the crawler. When the requested URL contains the “_escaped_fragment_”, the server returns the snapshot page. If not, the normal landing page will be returned.

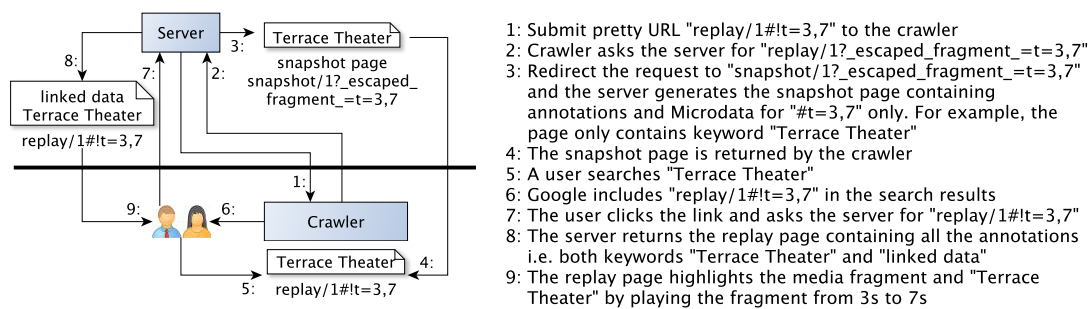


FIGURE 7.2: Model to improve media fragment presence based on Google Ajax Crawler

Figure 7.2 explains how this model could be used to index media fragments. As a pre-condition, the time information must be available for the developers so that the W3C-MFURI can be constructed. The returned page in step 4 only contains keywords related to fragment “t=3,7”. In the Google index, the “pretty media fragment URLs” are associated with the snapshot page. So what Google actually indexes is the URL of the replay page with hashbang and W3C-MFURI syntax attached. Step 8 still returns the whole page, but in step 9, the fragment will be passed to the URL representing the real location or the service which delivers the multimedia file. For example, if the request URL is *example/replay/1#!t=3,7*, the fragment “#t=3,7” will be attached at the back of *example2/1.ogv*, which is the video embedded in the replay page. Hashbang is not a valid syntax in W3C-MFURI specification, so developers need to parse the information in the hashbang URL before attaching the fragment to the URL of the actual multimedia file. Then the embedded player needs to play the fragment from 3s to 7s controlled by javascript and the corresponding annotations are highlighted straight away. In this case, step 6 will return the URL of the media fragment instead of the replay page. This design not only makes sure media fragments are indexed precisely with the keywords related to them, but also preserves the existing user interface and the interactive experience.

On the server side, changes need to be made. The first is the program to detect the “_escaped_fragment_” parameter in the requested URL from Google, and redirect it to the snapshot generation program, which is the second program to be added to the server. The snapshot page does not need to be user-friendly, but the metadata and annotations related to the media fragment should be presented in a well-structured

manner. Microdata (Hickson, 2012) defined in schema.org⁵ are also embedded into the page. Each media fragment is defined as a “VideoObject”, and the tag annotations are defined as “keywords” and other properties in schema.org.

The third component developers need to write is the program to highlight the corresponding media fragment when the page is opened. For this demonstration, the smf-player is used to fulfil the visual output of the temporal dimension. Developers also need to include the newly created hashbang URLs in the video sitemaps so that they could be easily found by Google⁶.

7.1.3 Evaluation and Discussion

The Media Fragment Indexing Framework demonstrated in Figure 7.2 is implemented in Synote, where some media fragments and annotations are pre-defined in the database. Sitemaps containing URIs like *replay/1#!t=3,7* have been submitted to Google for indexing. To enable a quick evaluation of what Googlebot fetches, several URIs were submitted to Google Web Master Tools⁷. The result shows that on fetching this URL:

<http://linkeddata.synote.org/synote/recording/replay/36513#!t=00:00:01,00:00:14>

The content in the snapshot page is only related to fragment “*#!t=00:00:01,00:00:14*”. The Microdata embedded in the snapshot page can be successfully recognised by Live Microdata⁸ and Linter Structured Data⁹. If the keyword “Terrace Theater”, for example, is searched in Google, the snapshot page can be successfully found in the search results instead of the whole replay page. When users click the link in search results, the Synote Player page in Figure 7.3 will be opened and the button in the top-left corner indicates that a media fragment is requested. On opening the page, the video will start playing from 1s to 14s and the related annotation on the right column will be highlighted.

As another example for evaluation, one can search the sentence “*All kinds of conceptual things, they have names now that start with HTTP*” in Google. This sentence is included in the transcript of the video resources for both TED Talks and Synote. The first result in Figure 7.4 is from TED Talks. Clicking on the link will open the replay page of the talk, but users still need to manually find the sentence in the interactive transcript. So even though the sentence logically annotates part of the video, the search engine still associates it with the whole document, i.e. the landing page. The second result in Figure 7.4 indicates that this sentence in Synote is directly related to the fragment defined in the replay page. On clicking this search result, the video embedded in the

⁵<http://schema.org>

⁶<https://support.google.com/webmasters/answer/80472?hl=en>

⁷<http://www.google.com/webmasters/tools/>

⁸<http://foolip.org/microdatajs/live/>

⁹<http://linter.structured-data.org/>

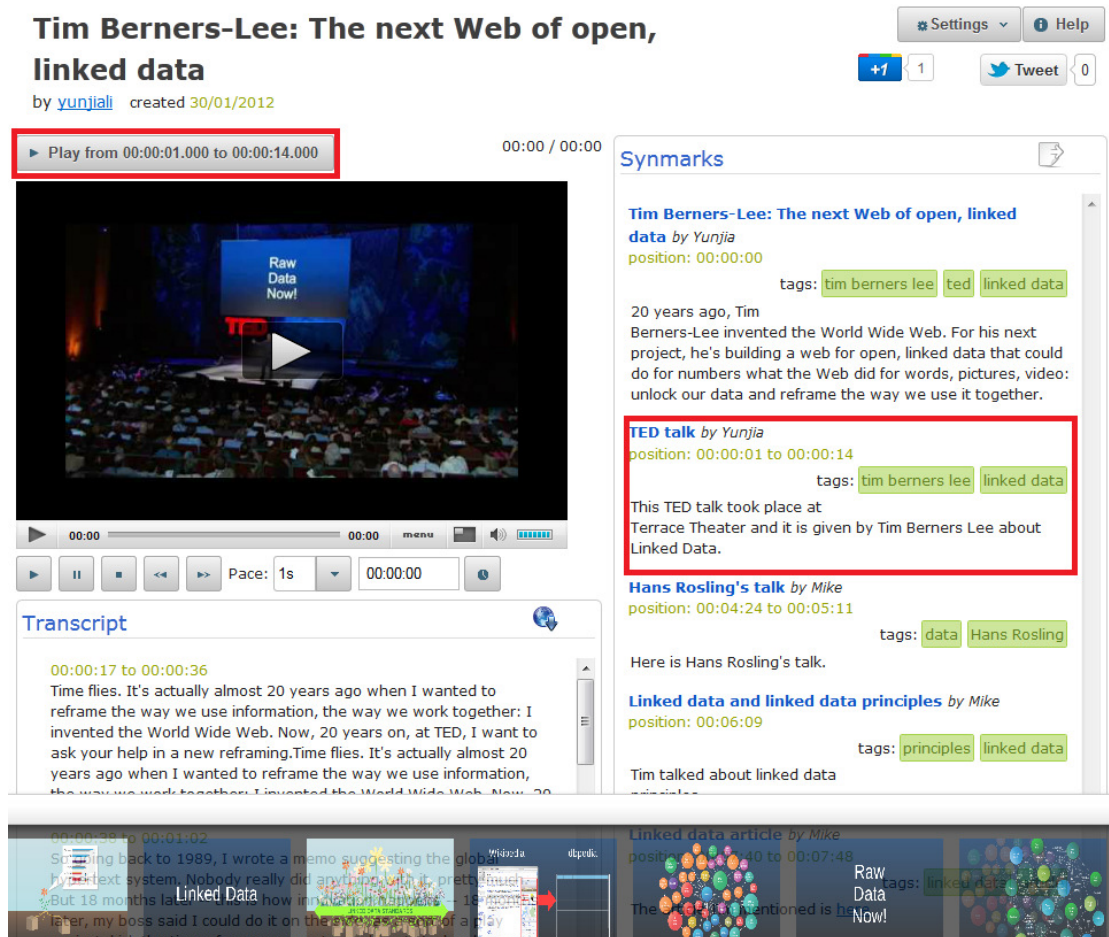


FIGURE 7.3: Screenshot of Synote replay page

replay page will start playing from the start time of this media fragment. The evaluation is available as a screencast online¹⁰.

The main idea of this model is to generate a set of search-engine-friendly snapshot pages on the fly according to the time span defined in the fragment of the requested URL. For applications heavily reliant on Ajax or Flash, it is useful to keep two sets of pages for both rich user interaction and search engine optimisation (SEO). Considering the Web Content Accessibility Guidelines¹¹, it is also good practice to provide such snapshot pages because the interactive feature is not usually accessible to screen readers and keyboard users since it depends much on javascript.

This solution follows the Interlinking Media Fragments Principles in that the landing pages that already exist are not the physical location of the video files, so it is safe to attach the media fragment encoding as part of the hashbang URL (instead of hash URI defined in W3C-MFURI), and the original landing page will not have any problem in dereferencing the HTML representation of the URL. The RDF descriptions of

¹⁰<http://goo.gl/4zl1V>

¹¹<http://www.w3.org/TR/WCAG20/>

[Tim Berners-Lee on the next Web | Video on TED.com](http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html)
www.ted.com/talks/tim_berners_lee_on_the_next_web.html
All kinds of conceptual things, they have names now that start with HTTP.
 Second rule, if I take one of these HTTP names and I look it up and I do the web thing ...
 You've visited this page 6 times. Last visit: 22/03/12

<?xml version="1.0" encoding="UTF-8"?> <html> <head> <title> Tim ...
linkeddata.synote.org/synote/recording/replay/36513#t=00:06...
All kinds of conceptual things, they have names now that start with HTTP.
 > </div> <div class="transcript mediaObject" itemscope="itemscope" ...

FIGURE 7.4: Search results comparison between TED Talks and Synote

the *ma:isFragmentOf* and *ma:locator* relationships suggested by the principles can be embedded in the snapshot page using RDFa. Schema.org defines *contentUrl* in the *MediaObject*, which can be used as a substitute for *ma:locator*. However, schema.org so far has not defined any vocabulary for *ma:isFragmentOf*. The property with the closest meaning is *isBasedOnUrl* in *CreativeWork*.

This model relies on hashbang URLs, which has been widely used in Facebook and Twitter to allow the indexing of Ajax content. But hashbang is claimed not to be good practice of URL design¹² and it should be replaced by HTML5 PushState¹³ in applications heavily based on Ajax. Twitter has announced that it would (actually already has) remove all hashbang URLs¹⁴. In order to use Google Ajax Application Crawler, the Media Fragment Indexing Framework needs to adopt the hashbang URLs, but obviously the Ajax crawler was not originally designed for this purpose. Another limitation of this model is that it only works for search engines which have such an Ajax crawler. The general solution to media fragment indexing might be by embedding Rich Snippet (Steiner et al., 2010), such as Microdata and RDFa (Adida and Birbeck, 2008), into the page so that search engines can highlight them in the search results. Major search engines are currently still far from reaching an agreement on the vocabularies to describe media fragments.

The implementation of the Media Fragment Indexing Framework introduced above has a major flaw: it is not very scalable. On the application level, users need to manually create a recording in Synote, make annotations and link them to media fragments. In other words, unless Synote has millions of users, who want to duplicate YouTube or Dailymotion videos in Synote and manually make annotations, the Media Fragment Indexing Framework as implemented can only index a limited number of videos shared in Synote. The Media Fragment Indexing Framework itself is very scalable as the Ajax crawler is designed for crawling Ajax content on the whole Web, but on the application level, more users or contributors need to generate media fragment annotations, so that the Media Fragment Indexing Framework can have enough input data.

¹²<http://goo.gl/xfvWT>

¹³<http://diveintohtml5.info/history.html>

¹⁴<http://storify.com/timhaines/hashbang-conversation>

It is a common practice now to share video on social media applications, such as Twitter. As some video sharing applications expose the deep-linking into the video as URLs and allow users to share them on Twitter, it is a straight-forward thought that a Tweet message containing media fragment URI(s) can be treated as an annotation to the media fragment(s). Thus, if the Tweets can be monitored and filtered by whether media fragments are contained in the message, those Tweets can be automatically uploaded as the input of the Media Fragment Indexing Framework. In this way, massive numbers of videos can be indexed at the media fragment level. The following two sections present the rationale of a Twitter Media Fragment Indexer and some preliminary evaluation.

7.2 A Survey of Media Fragment Implementation on Video Sharing Platforms

Theoretically, any URL shared on Twitter could be a media fragment URI, so ideally the indexer should be able to monitor all the Tweets and check whether a media fragment URI is included in the message. However, this method is not realistic. First, the Tweet stream is too large to be monitored in real-time. According to a report¹⁵, Twitter had more than 200 million users who send over 400 million Tweets daily. In other words, the indexer would need to check over four thousand Tweets a second to decide whether a Tweet contains a media fragment URI. While not impossible as Twitter provides APIs to access the full public stream of Tweets¹⁶ (also known as “firehose” API), collecting every Tweet would need extensive computing power and network bandwidth. There are also services that provide archived historical Tweets so that the indexer can work offline, but those services are very costly.

The second problem is that even though every Tweet message is monitored, it is still difficult to automatically decide which URL in the message is a media fragment URI. As discussed in the literature review, the implementation of W3C-MFURI syntax is very limited but many online video sharing applications have implemented the notion of media fragments or deep-linking into their systems. So it is not expected that the standard W3C-MFURI parser can recognise the media fragment URI in other syntax. In addition, some URLs may use syntax similar to W3C-MFURI that have nothing to do with media fragments (false positive cases), for example:

`http://www.example.org/1234#t=23`

Unless the HTML page is retrieved and examined manually, it is hard to decide whether this URI is about a media fragment, but `#t=23` could be viewed as a valid W3C-MFURI

¹⁵<http://goo.gl/WfnCAA>

¹⁶<https://dev.twitter.com/docs/api/1.1/get/statuses/firehose>

encoding of a media fragment. To prove this assumption, a program was developed using the Twitter firehose API to examine every Tweet message within one minute; the program uses the Media Fragment URI Parser¹⁷ recommended by the Media Fragment Working Group to decide whether a URL in the Tweet is a valid media fragment URI. Over four thousand Tweets with URIs were examined of which around 50 were recognised as media fragment URIs. However, after manual inspection, only 2 of them were real media fragment URIs shared from YouTube (See Appendix B for details). Even though other URIs also encode *t=xx* or *track=xx* as URI hash or query, they are obviously not media fragment URIs. This program shows that by parsing the URLs shared on Twitter only, the false positive rate of the general monitor is too high. So the Twitter Media Fragment Indexer needs to apply other methodologies that are cost-efficient and less error-prone.

W3C-MFURI was published as a W3C recommendation in September 2012. Three years later, most of the videos users watch online are hosted on major video sharing platforms. So it is reasonable to expect that most of the media fragments shared in Twitter are coming from those websites. If the indexer can filter the Tweets containing URIs from those domains and parse the URIs according to the media fragment syntax defined in each website, less computing power will be needed and the parsing results will be more accurate. Of course, some of the media fragments could be missing because the domains which host the videos are not monitored by the indexer. So, as the first step to use Twitter as the data source of media fragment indexing, a survey needs to be conducted to find out which online video sharing platforms have actually implemented or partially implemented the notion of media fragment, i.e. are they “media fragment ready”, so that their media fragments can be shared via Twitter? It is also interesting to know roughly how many videos, or what percentage of videos online, have the potential to be annotated at a media fragment level. The rest of this section will design a survey for this purpose and analyses the survey results.

7.2.1 Methodology

The first step in the methodology is to decide which video sharing application(s) need to be investigated. The Wikipedia page “List of video hosting services”¹⁸, lists the major video hosting websites. This experiment slightly modifies the list by adding a couple of well-known applications, such as TED.com¹⁹ and videlectures.net²⁰, and removing the ones that are not public video sharing websites or sharing adult videos. Finally, 59 websites were investigated as shown in Table 7.1. Some of the websites mainly serve videos other than English and some of them have access restrictions, such as Hulu²¹,

¹⁷<https://github.com/tomayac/Media-Fragments-URI/>

¹⁸ Accessed Oct, 2013, http://en.wikipedia.org/wiki/List_of_video_hosting_services

¹⁹<http://ted.com>

²⁰<http://videlectures.net>

²¹<http://www.hulu.com>

where the main content can only be accessible in the U.S.A.

It is not realistic to contact the owners or developers of those websites to find out whether media fragment is supported or partially-supported by those websites. Instead, the following steps are performed to see whether media fragment sharing is available on the user interface:

1. Open a desktop browser. Google Chrome was used for this investigation.
2. Go to the landing page of a random video. Login if necessary.
3. Right click on the player, and see if there is any option similar to “Get video url at the current time”.
4. On the landing page, find out whether there is a social sharing button (including the buttons that emerge after pausing the video player) allowing the video to be shared at a certain time point.
5. Look for buttons on the landing page, inside the player or after player has paused, that indicate a user can highlight a certain spatial area of the video display.
6. Go to Twitter and search whether any video fragment has been shared recently.
7. If none of the above steps leads to any clue about media fragments, draw the conclusion that this video hosting website does not support media fragments, at least does not support W3C-MFURI.

The experiment then obtained the page views per day for the website from WebsiteOutlook²². Finally, it determined how many out of the 59 websites have partially- or fully-implemented media fragments and the page views per day, for all 59 websites.

There are some obvious limitations to this methodology. First, as an uploader of the video the methodology did not investigate whether part of the video can be annotated and saved as a URI, even though the URI may not be public. Secondly, it was not possible in a short period of time to find out how many videos are hosted by each website, and it might be more helpful to answer “how many videos are exposed at the media fragment level and could be shared via Twitter” compared with page views per day. Thirdly, WebsiteOutlook claims that the page views per day is not guaranteed accurate and page views of a website does not equal the viewing of the videos either, especially for websites like Facebook. So other reports reflecting the video views were referred to, to get a rough idea of video views for those websites. If no valid page view data was found, the page views per day for that application was marked as zero. Finally, some of the media fragment functions could have been missed by the investigation, especially when there was a language barrier and access restrictions. Despite these flaws

²²<http://www.websiteoutlook.com/>

in the methodology, as a preliminary study, they did not critically affect the result. So the experiment largely reflected the media fragment implementations for major video sharing platforms.

7.2.2 Survey Results

Table 7.1 shows the investigation results at October 2013, where the temporal and spatial columns stand for the implementation of W3C-MFURI syntax dimensions. “Y” fully implemented following the W3C-MFURI, “P” stands for partially implemented, “N” stands for not implemented, and “U” stands for unknown.

Table 7.1: Media Fragment Compatibility on Video Hosting Services (Oct. 2013)

Name	temporal	spatial	Views/Day	Notes
56.com	P	N	7,142,857	Chinese
Archive.org	N	N	5,978,260	
AfreecaTV	U	U	91,674	Korean
Blip.tv	N	N	214,174	
BlogTV	N	N	33,475	Now at www.younow.com
Break.com	N	N	804,681	
Buzznet	N	N	120,733	
Comedy.com	N	N	3,520	
Crackle	N	N	344,611	
DaCast	N	N	2,897	
Dailymotion	P	N	11,702,127	
EngageMedia	N	N	3,426	
ExpoTV	N	N	34,042	
Facebook	N	N	18,600,000	video since 2007. Estimated based on: http://marketingland.com/facebook-558-million-video-views-36251
Funnyordie.com	N	N	506,678	
Funshion	N	N	601,300	Chinese, number of page views are got from http://www.websitelooker.net/
Fotki	N	N	139,611	
GodTube	N	N	68,909	formerly Tangle.com
Hulu	P	N	3,142,857	
Lafango	N	N	11,620	

LeTV	N	N	3,459,119	Chinese
Livleak	N	N	1,929,824	
Mail.ru	N	N	0	Page views unknown as the video section is a subset of the whole website, and there is no relevant data about video views online.
Mefedia	N	N	173,803	
Metacafe	N	N	1,127,049	
Mevio	N	N	52,276	
Mobento	N	N	1,014	Focus on video search
Myspace	N	N	0	No data about the video views.
MyVideo	N	N	553,319	
MUZU.TV	N	N	13,801	
Nico Nico Douga	N	N	7,746,478	Japanese
Openfilm	N	N	8,810	
Photobucket	N	N	0	Mainly sharing photos (5 263 157 views per day). Not sure about video views.
RuTube	N	N	601,750	Russian
Sapo Videos	N	N	0	Portuguese. Sapo.pt is a portal website, not sure about the video views per day.
SchoolTube	N	N	9,893	
ScienceStage	N	N	10,314	
Sevenload	N	N	7,291	
SmugMug	N	N	542,138	
Tape.tv	U	U	41,323	Area restriction, only music videos
TED.com	N	N	722,733	
Trilulilu	N	N	68,293	Romanian
Tudou	P	N	6,010,928	Chinese
Vbox7	P	N	319,303	Bulgarian
Veoh	N	N	359,359	
Vevo	N	N	685,358	
Viddler	N	N	122,073	
Videojug	N	N	86,901	
Videolog	N	N	79,687	Portuguese
videlectures.net	N	N	6,697	
Vidoosh	N	N	3,786	
Viki.com	N	N	203,628	Allow timed comments

Vimeo	P	N	19,680,000	
Vuze	N	N	54,380	
Wildscreen.tv	N	N	987	
Wistia	N	N	153,331	
Yahoo! Video	N	N	10,000,000	video since 2008. Estimated based on: http://marketingland.com/facebook-558-million-video-views-36251
Youku	P	N	15,277,777	Chinese
YouTube	P	P	366,666,666	

In November 2013 YouTube introduced a new feature which allowed users to “tag” a spatial area at a temporal point. Each tag that a user creates has a URL pointing to a landing page hosted via Clickberry²³ and this link can be shared via Facebook and Twitter. Here is an example of the tag URI:

<https://clickberry.tv/video/6dafe30e-dcb8-44b8-8190-32be8249a297>

But it has no temporal or spatial syntax encoded in the URI and there is no affiliation between the fragment and the original video resource. Thus Table 7.1, indicates that YouTube has only partially implemented the spatial fragment.

In Table 7.1, only 9 out of the 59 websites (15%) partially implemented the notion of media fragment and there is only one implementation for spatial fragment. Table 7.2 lists some example URIs with the media fragment encoded. Generally speaking, most of them use URI query to encode the temporal fragment and only YouTube and Vimeo use URI hash. Only video in Hulu can encode both start and end time in the URIs. All of them adopt seconds as the basic unit to represent temporal scale, and YouTube also allows the *ddhddmdds* string format.

7.2.3 Discussion

Only 15% of the websites in the investigation partially implemented media fragments. However, considering the use case that end users want to share videos at a media fragment level, increased viewing of the websites indicates a greater chance that they can be shared. So the question becomes “if you are watching a video online now, what is the possibility that this video can be shared at a media fragment level?”. This question is more important than the number of websites that implemented media fragments,

²³<http://clickberry.tv>

TABLE 7.2: The Supported Media Fragment Syntax in Different Video Hosting Services (Oct. 2013)

Name	Example url
56.com	http://www.56.com/u92/v_OTgwMTk4NDk.html#st=737
Dailymotion	http://www.dailymotion.com/video/xjwusq&start=120 http://www.dailymotion.com/video/xjwusq?start=120
Hulu	http://www.hulu.com/embed.html?eid=sepr2dtbsyn7idlhbuzlbw&et=135&st=13
Vbox7	http://vbox7.com/play:cc7d3fc2?start=10
Viddler	http://www.viddler.com/v/bb2a72e9?offset=12.083
Vimeo	http://vimeo.com/812027#t=214 http://vimeo.com/812027?t=214
Tudou	http://www.tudou.com/listplay/H9hyQbAj4NM/2tzZHTtq4GA.html?lvt=30
Youku	http://v.youku.com/v_show/id_XNjE2OTQ0MTI4.html?firsttime=147
YouTube	http://www.youtube.com/watch?v=Wm15rvkifPc#t=120 http://www.youtube.com/watch?v=Wm15rvkifPc?t=120 http://www.youtube.com/watch?v=Wm15rvkifPc&t=1h9m20s http://www.youtube.com/watch?v=Wm15rvkifPc#t=1h9m20s

because the answer to this question can reflect, to some degree, how many videos can be indexed by search engines using the mechanism that was introduced in Section 7.1, and how many of them are ready to be annotated by named entities at a media fragment level. From the page views per day in Table 7.1, the proportion of page views for websites implementing media fragments can be roughly calculated (Figure 7.5).

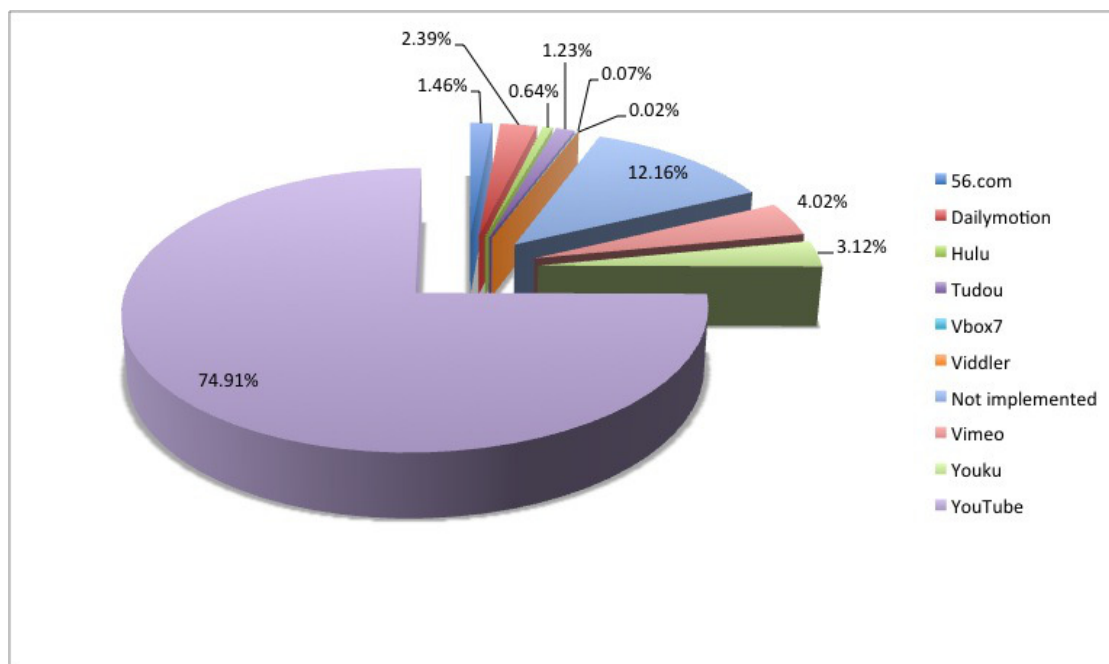


FIGURE 7.5: Percentage of Page Views for Websites Implementing Media Fragments Out of the Total Page Views for the 59 Websites Investigated

YouTube, as the largest video sharing platforms online, holds around 75% of the daily page views among all the websites investigated and only 12% of the views of the videos are not accessible in media fragment level. So from a Linked Data point of view, it is possible to construct valid media fragment URIs (even though they may not conform to W3C-MFURI) for most of the videos that are watched online and at least the HTML representation, i.e. the landing page, of the video could be dereferenced directly. When an individual URI is associated with a media fragment with a dereferenceable HTML representation, it is potentially indexable by major search engines. If those applications also embed semantic descriptions about the media fragment in the landing page using RDFa or schema.org, the semantic-aware search engines, such as Sindice²⁴ and Google with Rich Snippet parser, can further extract an RDF description about the media fragment and link it to the LOD Cloud. Even though the survey results reveal that most videos are “media fragment ready” and the media fragment indexing is not far away, the public videos on YouTube, Vimeo and Dailymotion are not searchable at a media fragment level via search engines even though they are able to name media fragments with URIs.

According to Figure 7.5, only the Tweets containing URLs from those 9 websites are possibly valid and dereferenceable media fragment URIs, because only those 9 websites have already implemented the notion of media fragments. Of course, as the Interlinking Media Fragments Principles (Chapter 4) suggest, the hash fragment can be attached at the back of any landing page URL and indicate a certain media fragment, and the media fragments from other video sharing applications can be also shared on Twitter. But this practice has very limited adoption in Twitter and thus could be simply ignored. As Twitter is still banned in China (Jan, 2015), those websites mainly based in China are unlikely to have their videos shared via Twitter. So the indexer ignores those websites: 56.com, Tudou and Youku. Hulu.com also has access restrictions from outside the U.S.A., so the indexer also ignores Hulu.com. Finally, YouTube, Dailymotion, Vbox7, Vimeo and Viddler are the video sharing applications that were selected for monitoring. The next Section will introduce the architecture of the Twitter Media Fragment Indexer and show some preliminary evaluation results.

7.3 Indexing Media Fragments Using Social Media

The survey revealed which applications the indexer needs to monitor Twitter. The main tasks of the Twitter Media Fragment Indexer are to collect the Tweet messages, extract URLs that encode media fragment information embedded in the message, and using the Media Fragment Indexing Framework to publish Tweet messages as annotations to the media fragment. Even though the indexer was limited in scope to monitoring Tweets

²⁴<http://sindice.com/>

of only five websites, the number of Tweets was still considerably large. As a proof-of-concept, the indexer only collected enough Tweets and media fragments to demonstrate the system. The following subsections detail the workflow of the indexer and discuss the evaluation results.

7.3.1 Workflow of Twitter Media Fragment Indexer

Figure 7.6 shows the workflow of the Twitter Media Fragment Indexer. Generally, there are three stages: Twitter Media Fragment Crawling, Data Preparation and Media Fragment Indexing.

The first part (Process 1.1) of the indexer is crawling the data from the Twitter Stream API with “youtube, dailymotion, vimeo, vbox7, viddler” as the keywords to the “track” parameter in the Twitter API 1.1. According to the Twitter status filter API²⁵, this filter phrase will return all the Tweets that match any of the keywords specified in the phrase regardless of its case. The matching works not only for the text attribute of the Tweet, but also the hashtag, users mentioned (@user) and the URLs in the expanded format, which by-passes the problem that shortened URLs usually will not match any of those keywords. When Tweets are returned by the Twitter Stream API, Process 1.1 will also filter out the Tweets that do not contain any URLs, because the indexer is looking for URLs with media fragment information encoded.

In Process 1.2, a program was developed to parse the URLs contained in the Tweets, based on the URL patterns observed in. The new parser is an extension of the Media Fragment URI Parser mentioned above. Basically, the new parser takes both the domain names and the query or hash string in the URLs into account, so that URLs from domains other than the five predefined values (youtube.com, dailymotion.com, vimeo.com, viddler.com and vbox7.com) will not be accepted as media fragment URIs, even though URLs follow the media fragment encoding syntax. The final process in the crawler is to save all the Tweets and the parsed media fragment information for the next stage.

After the required Tweets and media fragments have been saved locally, the Data Preparation stage will group the Tweets and media fragments by videos (Process 2.1) and collect necessary metadata for the content that will be displayed on the landing page and the snapshot page, which will be fetched by the Google Ajax crawler. The landing page provided by the indexer in Media Fragment Indexing stage will be similar to Figure 7.3 in Section 7.1.3. One video can be shared by many Tweets, and each Tweet may refer to a different time point in the video. So at the data preparation stage, the Tweets need to be grouped by video, so that Process 3.1 can assign a URI for each landing page, where different media fragments in the video can attach themselves at the back of the landing page URI using hashbang and the W3C-MFURI syntax.

²⁵<https://dev.twitter.com/docs/api/1/post/statuses/filter>

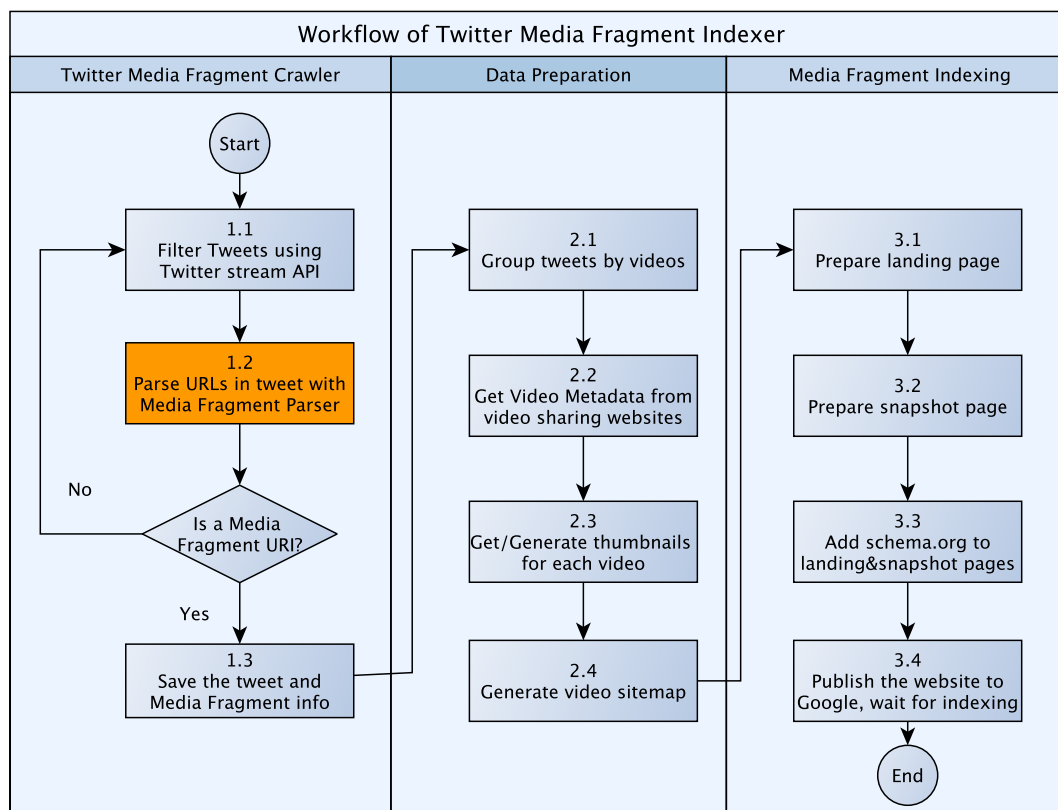


FIGURE 7.6: The Workflow of Twitter Media Fragment Indexer

This demonstration expects to create thousands of media fragments, so in order to allow the media fragments to be automatically indexed by Google, instead of submitting individual URLs for indexing, it is recommended that a video sitemap is placed at the root of the application. Following the guidelines of preparing a video sitemap for Google²⁶, some metadata about the video need to be retrieved from their original publishers (YouTube, Dailymotion, Vimeo, etc), which include the titles and descriptions (Process 2.2) and the thumbnails of the videos (Process 2.3). The thumbnails are downloaded, optimised and cached locally for search engine indexing. When all the information required for the video sitemap is ready, the sitemap.xml file will be generated containing every distinct media fragment URI crawled from Tweets. In some cases, especially in retweet messages, two or more Tweets may contain the URL about the same video at the same time point and all those Tweets will be annotating the same media fragment URI.

In Stage 3, Media Fragment Indexing applies the Media Fragment Indexing Framework to the metadata, media fragments and Tweets data that have been collected so far. Two sets of pages are prepared: a landing page for users to view the video and Tweets, and a snapshot page for the Google Ajax crawler to get the HTML content related to a certain media fragment. Process 3.3 also embeds some simple Microdata into the landing and snapshot pages as optimisation for the search engine. The final step is to publish the

²⁶<https://support.google.com/webmasters/answer/80472?hl=en>

website and submit the sitemap to Google, and wait for the snapshot pages to be crawled and indexed.

7.3.2 Implementation and Evaluation of Twitter Media Fragment Indexer

The Twitter Media Fragment Indexer is implemented in Node.js²⁷ using MongoDB²⁸ as the backend database. Because the Tweets crawled from the Twitter Stream API are serialised in JSON format, the NoSQL database can best store and manipulate such data without spending too much time on designing the database schema. The demonstration website is available online²⁹ and any user can search the website content within Google using:

YOUR KEYWORDS `site:twitter-mediafragment-monitor.herokuapp.com`

and examine the indexed media fragments.

In the experiment, the crawling program examined around 50 hours of non-stop Twitter stream (from 12:00:00 GMT, 22 December 2013 to 14:00:00 GMT, 24 December 2013) with the filter phrase “youtube, dailymotion, vimeo, vbox7, viddler”. During those 50 hours, the indexer examined 5,779,858 Tweets, of which 5,269,742 Tweets included one or more URLs. A media fragment URI parser was developed for detecting the media fragments encoded in those URLs³⁰. In total, there were 5,483,668 URLs processed by Process 1.2, out of which 32,796 URLs were valid media fragment URIs while 32,754 Tweets contained valid media fragment URIs. So roughly, only 0.6% of the video URLs were shared from those websites via Twitter and encoded media fragment information. Table 7.3 shows the number of the media fragment URIs shared in each website monitored. YouTube took nearly all the media fragment URIs shared on Twitter, while the indexer did not observe any media fragments shared from VBox7 and Viddler.

In the experiment, the crawling programme examined around 50 hours of non-stop Twitter stream (from 12:00:00 GMT, 22nd Dec, 2013 to 14:00:00 GMT, 24 Dec, 2013) with the filter phrase “youtube, dailymotion, vimeo, vbox7, viddler”. During those 50 hours, the indexer examined 5,779,858 Tweets, in which 5,269,742 Tweets include one or more URLs. A media fragment URI parser has been developed for detecting the media fragments encoded in those URLs³¹. In total, there were 5,483,668 URLs processed by Process 1.2 in Figure 7.6, out of which 32,796 URLs are valid media fragment URIs and 32,754 Tweets contain valid media fragment URIs. So roughly, only 0.6% of the video

²⁷<http://nodejs.org>

²⁸<http://www.mongodb.org/>

²⁹<http://twitter-mediafragment-indexer.herokuapp.com>

³⁰<https://github.com/yunjiali/Media-Fragments-URI-Loose>

³¹<https://github.com/yunjiali/Media-Fragments-URI-Loose>

TABLE 7.3: Number of Media Fragment URIs shared in each Website

Website	Number of Media Fragment URIs	Percentage %
YouTube	32,666	99.604
Dailymotion	101	0.308
Vbox7	0	0
Viddler	0	0
Vimeo	29	0.088

URLs shared from those websites via Twitter and encode media fragment information. Table 7.3 shows the breakdown number of the media fragment URIs shared in each website that is monitored. YouTube takes nearly all the media fragment URIs shared on Twitter, while the indexer did not observe any media fragments shared from Vbox7 and Viddler.

In the Data Preparation stage, the grouping of Tweets by video (Process 2.1) resulted in 13,088 videos in total, which means at least one media fragment in those videos has been shared via Tweets. The number of videos was far fewer than the number of total Tweets with media fragment URIs for two main reasons. First, many Tweets shared the same popular video, including the retweets. For example, ten Tweets annotated the following YouTube video and three of them were retweets:

<http://twitter-mediafragment-indexer.herokuapp.com/v/280qnrHpuc8>

The second reason involves Process 2.2. Tweets are publicly available in all countries which have access to Twitter; however, on YouTube and Dailymotion, some videos have country and region level access control. As this experiment was conducted in the UK, some of the video shared by Twitter was not accessible in the UK. 104 videos fell into this access control. The indexer was not be able to get the metadata of those videos and they were thus ignored in the final video collection. In Process 2.3, 13,066 thumbnail pictures were retrieved from those websites and 22 thumbnail pictures were missing because the original pictures for some videos were not available. Finally, in Process 2.4, 17,854 video entries with media fragment URIs were generated in the sitemap. Even though 32,796 media fragment URIs were collected during Process 1.3, some of them referred to the same video and same time point as previously described. The sitemap should avoid duplicated URLs, even though they are shared by different Tweets. All the URLs included in the sitemap were newly minted within the “twitter-mediafragment-indexer.herokuapp.com” domain and the “content loc” attribute in the video sitemap was used to link the landing page to the original URL of the video in YouTube, Dailymotion or Vimeo.

For the Media Fragment Indexing stage, a very simple landing page and snapshot page were designed as a proof-of-concept. The landing page (Figure 7.7) displays the title

description and Tweets related to this video. The smfplayer is still used to highlight the temporal fragment. The snapshot page (Figure 7.8) only includes the title and the text from the Tweet corresponding to the temporal fragment encoded as the value of the “_escaped_fragment_” query parameter. VideoObject defined in schema.org is embedded in both pages. As schema.org has no definition of media fragment object, the title, description and thumbnail of the video are still treated as the same attributes for the media fragment in the snapshot page.

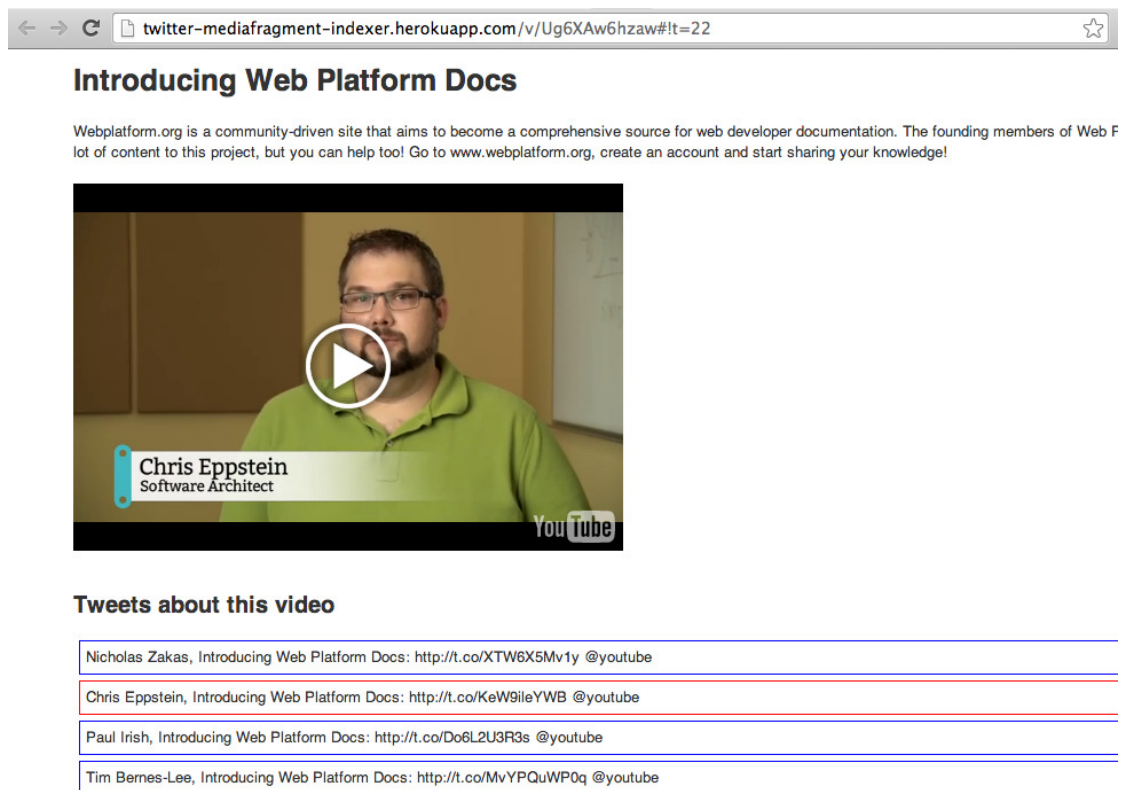


FIGURE 7.7: The Landing Page in Twitter Media Fragment Indexer

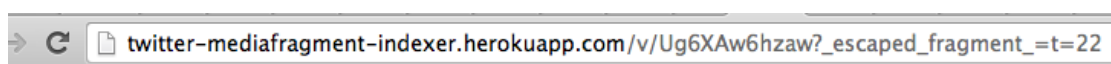


FIGURE 7.8: The Snapshot Page in Twitter Media Fragment Indexer

At the time of writing, 17,479 URLs in the sitemap out of the total of 17,854 URLs (around 97.9%) have already been indexed by Google. For the sake of evaluation, four Tweets were deliberately created during the experiment, all of which annotate the “In-

roducing We Platform Docs” video on YouTube³². The content of the four Tweets are in Listing 7.1.

```
Nicholas Zakas, Introducing Web Platform Docs: http://t.co/XTW6X5Mv1y @youtube
Chris Eppstein, Introducing Web Platform Docs: http://t.co/KeW9ileYWB @youtube
Paul Irish, Introducing Web Platform Docs: http://t.co/Do6L2U3R3s @youtube
Tim Bernes-Lee, Introducing Web Platform Docs: http://t.co/MvYPQuWP0q @youtube
```

LISTING 7.1: Test Tweets for Twitter Media Fragment Indexer

The URLs in the Tweets are media fragment URIs pointing to the times that the people indicated in the Tweets start to talk in the video. For example, the second Tweet in Listing 7.1 shows that Chris Eppstein is interviewed at the 22nd second in the video. For evaluation, those Tweets were collected from the crawler and the media fragment URIs included in the sitemap, which was submitted to Google. After those URIs were indexed by Google, searching in Google for one of the names in the Tweets returns the “ugly URL” with hashbang. Clicking on that URL opens the landing page and the smfplayer starts playing the video from the start time represented by the media fragment URI in the corresponding Tweet. Figure 7.9 shows that searching “Chris Eppstein” returned the following URL:

<http://twitter-mediafragment-indexer.herokuapp.com/v/Ug6XAw6hzaw#!t=22>

Opening this link will lead to the landing page in Figure 7.7 and the smfplayer will jump to the 22nd second of the video, while the Tweet containing keywords “Chris Eppstein” is automatically highlighted. So the whole workflow suggested by the Twitter Media Fragment Indexer can successfully use Twitter as a social annotation platform for media fragments indexing in Google.

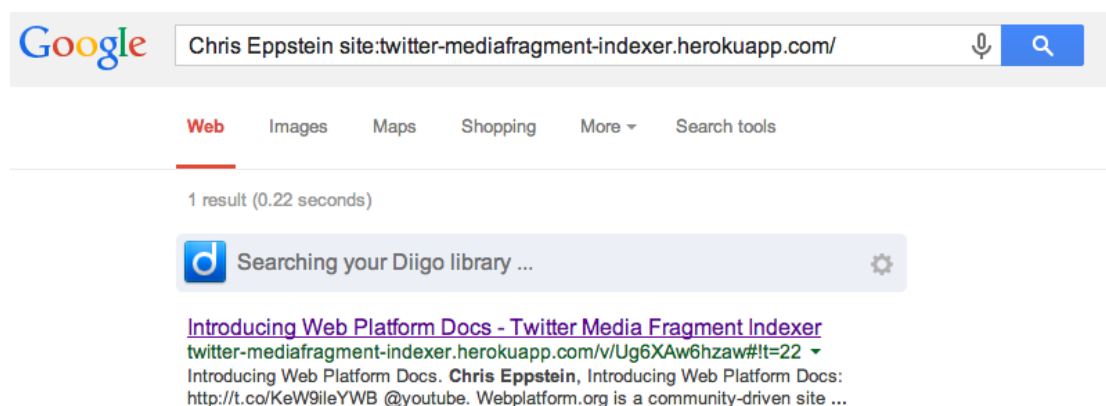


FIGURE 7.9: Searching Media Fragment URIs provided by Twitter Media Fragment Indexer

One problem of this experiment is that a lot of redundant search results will be generated because they all contain the same keywords. For example, the search for “Introducing

³²<http://www.youtube.com/watch?v=Ug6XAw6hzaw>

Web Platform Docs” in Listing 7.1 will return all the media fragments including the landing page itself. Because there no clear definition of what the “title” and “description” should be for a media fragment, all the media fragments share the same title and description in the sitemap and schema.org markups, which is the same as the original video. So searching any keywords included in the video title and description will also return the URLs for media fragments. Title and description cannot be ignored in the sitemap, nor in the schema.org markups, so the best solution for the future is to define some special markups in sitemaps and schema.org to address media fragments.

7.4 Summary

The concept of media fragment is important on the Web as it describes the content inside a multimedia resource. However, major search engines currently are not able to index media fragments using their related annotations. The key problem is that media fragments and annotations do not have their own page on the Web and they share the same landing page as the parent resource. This chapter introduced the Media Fragment Indexing Framework as the solution (R7). This framework uses Googles and Bings crawling infrastructure for Ajax applications. The basic idea behind the framework is that each media fragment with its related annotations will have a URI, where media fragments are encoded following W3C-MFURI, and an individual snapshot page, which could be indexed by the crawler. A demonstration was implemented on top of the Synote system and evaluated on Googles Ajax crawling infrastructure. The demonstration implemented the temporal dimension for video/audio resources, but the concept could be easily extended to spatial and other dimensions defined in W3C-MFURI. Initial evaluation has shown that the snapshot pages were successfully fetched by Googlebot and it is expected that more media fragments will be indexed using this method, so that the search for multimedia resources would be more efficient. Users could play the media fragment directly from the links provided by the Google search results.

One issue of the Media Fragment Indexing Framework is that users need to make annotations linked to media fragments within a specific application, which limits the sources of input of data. In order to expand the Media Fragment Indexing Framework to a larger scale, it was proposed that social media, such as Twitter, be used to acquire more annotations linked to media fragments. A survey was conducted to determine which video platforms allowed users to share the deep-linking of videos on social media. The survey found out that most of the major video sharing platforms allow users to do this at a certain time point of the video, so it would be possible to monitor the sharing activities from social media and collect them as annotations as the media fragments. Based on this survey, the Twitter Media Fragment Indexer was designed to collect Tweets as annotations of media fragments and used the Media Fragment Indexing Framework to make the annotations shared on social media indexable in Google. The experiment showed

that, after 50 hours of monitoring, YouTube was the most important resource for media fragment sharing on Twitter. 17,479 media fragment URIs were created automatically from Twitter monitoring and more than 97% were successfully indexed by Google.

While Twitter was used as the data source for media fragment annotations, the methodology could be applied to other social media resources. For example, a program could crawl through subtitles or timed-text from YouTube and Dailymotion APIs and chunk the video accordingly for media fragment indexing. This chapter mainly focused on using the Media Fragment Indexing Framework for videos in the temporal fragment dimension. Nevertheless, the framework could easily be applied to images and spatial dimensions. Flickr and Facebook both allow users to tag people within a spatial area on the photo, and the spatial fragment could be indexed by Google using the proposed framework.

Chapter 8

Conclusions and Future Work

This chapter summarises the previous chapters and reflects on the major results and contributions of this research. Based on the current research results, it outlines the future research directions.

8.1 Conclusions

Current multimedia applications in Web 2.0 have generated large repositories for multimedia resources and annotations. But each application is "a silo, walled off from the others" (Berners-Lee, 2010). This problem becomes more serious for applications serving and sharing multimedia resources on the Web. While the metadata such as title, description and tags, have made whole multimedia resources searchable, the internal content of multimedia, i.e. the media fragments, are still locked inside the container of the multimedia files. The multimedia resources are not yet able to be bookmarked, linked, navigated, indexed and shared at a fine-grained media fragments level. As an analogy, a text book would be difficult to read if there is no content table and page numbers. So there is an urgent need to expose and link media fragments to other resources on the Web.

Linked Data has provided a way forward to expose, index and search media fragments and annotations on a large scale, which used to be isolated in different applications. This work identified the basic research problem and requirements of publishing media fragments using Linked Data principles in order to improve the online presence of media fragment. This was further extended as a framework to automate the process of interlinking media fragments to the LOD Cloud. The main contributions of this research are:

1. **Interlinking Media Fragments Principles** (Chapter 4): the principles specify

the minimum requirements to choose media fragment URIs and to make them dereferenceable, and what should be included in their RDF description.

2. **Core Model for Media Fragment Enrichment** (Chapter 5): the core model reuses well-known vocabularies to describe media fragments and set up connections between media fragments and concepts in the LOD Cloud.
3. **Media Fragment Enriching Framework** (Chapter 5): the framework uses named entity extractors to automate the interlinking of media fragments to the LOD Cloud for video/audio resources on the Web.
4. **Synote Media Fragment Player** and **Media Fragment Enricher UI** (Chapter 6): they are part of the components implemented in the Media Fragment Enriching Framework and they are examples of the visualisation of media fragments together with named entities.
5. **Media Fragment Indexing Framework** and media fragment indexing using Twitter (Chapter 7): the indexing framework allows search engines such as Google to index media fragments with minimum change to the multimedia host server. The idea of using Twitter as the source for media fragments annotations makes the indexing framework scalable on the Web.
6. **Applying media fragments and semantic annotations in video classification** (Section 5.4): media fragments and named entities are introduced as new features for video classification.

The literature review in Chapter 2 showed that both multimedia annotations and Linked Data are mature research areas with many previous research results. Media fragments have also been addressed in multimedia and hypermedia research. Bringing the semantic Web to multimedia resources on the web, especially using the semantic Web to present structured metadata, have been well developed. However, few of them have systematically analysed the problems of applying Linked Data principles into media fragments and listed the problems and solutions. Researchers and developers of multimedia platforms should also be interested in the best practice and recommendation implementations regarding enhancing the current platform with Linked Data. Unfortunately, the research in this area is very limited.

Through the case studies in Chapter 3, seven major requirements were revealed, which outlined the problems that this work has specifically tackled. The key requirement is that the existing situation of current multimedia platforms needs to be taken into account when applying Tim Berners-Lee's four Linked Data principles. In detail, there are two different types of server that are involved in delivering and presenting the multimedia content and annotations: "Multimedia Host Server" and "Player Server". It is important to know that the landing page of the video is different from the real video object, and

the annotations linked to the media fragments could come from different resources. Meanwhile, the solutions to publication of media fragments and annotations must be aligned with current platforms, because it is not practical to totally ignore current systems. This research focused on using real data instead of generating experimental data within ideal situations where everything is under control.

Chapter 4 examined the relationship between the Linked Data principles and the current situation of media fragments on the Web, especially regarding the solutions for R1 to R4. The research in this chapter recognised that the key idea of Interlinking Media Fragments was to treat each video/audio as a concept or object and mint a new URI whenever the video/audio object is reused, which also differentiates the landing page of the multimedia resource from the multimedia file itself. While avoiding using the location of the video/audio file as the stem of media fragment URI, a triple should include the relationship between the newly minted URI and the original location of the file. In this way, video resources hosted in different applications can be safely reused at the media fragment level. These solutions only cover the minimum requirements for media fragment dereferencing and RDF description, which makes sure that even though the **Multimedia Host Server** is not compatible with media fragments and it has no intention of serving RDF descriptions of its resources, the media fragments in those resources can still be reused or referred to, and media fragments and annotations published that follow the Linked Data principles (R5).

The Interlinking Media Fragments Principles only define the minimum recommendations or best practice for publishing media fragments, but more details still need to be specified for any real multimedia applications to achieve the implementation of R3 and R4. In order that different systems sharing media fragments can interact with each other, Chapter 5 proposes a Core Model for Media Fragment Enrichment, which reuses W3C-MFURI, W3C-MA, NinSuna Ontology, Open Annotation Core Model and Provenance Ontology, to model the media fragments and their relationships with resources in the LOD Cloud. As most multimedia sharing platforms expose text-based annotations to the media fragments, a Media Fragment Enriching Framework was designed to automatically generate the instances of the core model. The framework takes the timed-text as the input from multimedia sharing platforms and uses extractors to obtain named entities from the timed-text. The named entities are further linked (with the Open Annotation Core Model) to the media fragments corresponding to the timed-text from which the named entities were extracted. Actually, there are many means of interlinking as introduced in (Hausenblas et al., 2009b) and Figure 4.2 in Section 4.4. However, for online video sharing platforms, which are different from storing millions of videos on the local hard disk, the lower-level signal analysis will be costly and impossible for most researchers, so using user-generated text-based data exposed via a Web API for the media fragment interlinking will be ideal.

Chapter 6 proposed further solutions for R6 media fragment visualisation by designing

an implementation of the Media Fragment Enrichment UI. The smfplayer is part of the Media Fragment Enricher UI in the Media Fragment Enriching Framework. A demonstration was developed where extracted named entities are displayed with the YouTube video and its subtitle in a synchronised manner.

Chapter 7 showed that the difficulty of indexing media fragments for search engines lies in the convention that the multimedia resource and its annotations are aggregated on the same landing page, where users can watch and interact with the video. The Media Fragment Indexing Framework was proposed as the solution to this problem. First, a snapshot page needs to be prepared for each media fragment. Then the server needs to be configured to satisfy the requirements of the Google Ajax Crawler by attaching the hashbang string encoding the media fragment, at the end of the original landing page URL and redirecting the alternative request from the crawler agent with “_escaped_fragment_” query to the snapshot page. The media fragment encoding only differs from the W3C-MFURI syntax in that hashbang is used instead of hash in the URL, because this is a requirement of the Google Ajax Crawler. On opening the landing page, the media fragment encoded in the hashbang URL will be parsed and highlighted immediately. This solution fully follows the Interlinking Media Fragment Principles, and RDF descriptions of the video can be embedded using RDFa or schema.org. Following this solution, users can still use a keyword search in Google, and clicking on the returned links will lead to the landing page, where the video will start playing from where the media fragment encoded in the landing page URL indicates.

To scale-up this solution with more input data, Chapter 7 also proposed using Twitter as a crowdsourcing annotation medium for online videos. In this proposal, each Tweet containing deep-linking URLs of the video is treated as an annotation to the media fragment, because the deep-linking URLs are usually the landing page URL from video sharing applications, such as YouTube and Dailymotion, plus a start time indicating the position where the embedded player in the landing page will start playing when opening the page. An experiment was conducted of monitoring the Tweet stream for five different video sharing applications for 50 hours. 17,854 media fragment URLs were generated and submitted to Google. Using the Media Fragment Indexing Framework, around 97.9% of these were successfully indexed by Google. Searching for content in the Tweets can lead to the specific media fragment being highlighted on the landing page, which shows that the proposed framework can greatly improve the media fragment indexing on search engines, and that social media can be used as a crowdsourcing way of media fragment indexing.

The research contributions of this work will be of value to future research into multimedia sharing platforms and the publishing of media fragments and annotations into the Linked Data Cloud, as the major concerns of such systems have been defined here and solutions proposed. Furthermore, this research investigated the use of media fragments in different areas, such as searching, visualisation and video classifications.

8.2 Future Work

Linking media fragments to the LOD Cloud opens the doors to many research areas. The remainder of this section lists several future research areas based on the results of this research.

8.2.1 Media Fragment Semantics in Debate Analysis

Use cases in Section 3.2 have shown the potential for applying media fragments to domain-specific problems and helping researchers in those areas with their research questions. At the early stage of this research, a demonstration was developed as a proof-of-concept to link UK Parliamentary Debate to various resources in the LOD Cloud (See Figures 8.2 and 8.1). This demonstration first collects the parliamentary debate videos and transcripts from the official Hansard¹ report and TheyWorkForYou². Then using both user manually annotated data, such as linking the debate transcript to an external report (Part2 in Figure 8.2) or to a related video on YouTube (Part3 in Figure 8.2), and DBpedia spotlight (Part1 in Figure 8.2) as the named entity recognition tool to create the interlinking between the debate transcript and the resources on the Web. Then all the annotations are visualised in a landing page (Figure 8.1). The screencast of the demonstration is available on YouTube³, but is no longer accessible online due to a browser compatibility problem.

There is some published research on bringing semantics to debate content. One example is PoliMedia (Juric et al., 2013), which uncovered the links between debate content and the media, such as newspapers, library archives, etc. PoliMedia is based on the Dutch Parliament, but the idea could easily be expanded to the UK Parliamentary Debate, which is also well archived since the early 20th century. In addition, the idea can be expanded to link debate to resources from multimedia archives (such as the BBC World Services Archive), social media (such as YouTube) and TV programmes. These multimedia resources could be linked back to a certain part of the debate video via the transcript of the debate, so that users can directly watch the specific point in the debate video at which the keyword or some topics are discussed. The Media Fragment Indexing Framework could also help in this case to make a certain part of the debate video searchable.

Together with natural language processing techniques, this tool would be very useful for researchers in politics analysing the Parliamentary debate. It would also attract more people to follow the parliamentary debate and thus improve the transparency of

¹<http://www.parliament.uk/business/publications/hansard/>

²<http://www.theyworkforyou.com>

³http://www.youtube.com/watch?v=KkWI-DHLD_M


Keith Vaz Labour, from Leicester East

<http://reference.data.gov.uk/id/mp/leicester-east/keith-vaz>


Will the right hon. Lady join me in commending the work of Jan Berry, who was appointed by the previous Government but completed her report under the present Government, and her recommendations to reduce police bureaucracy? Will the right hon. Lady give the House an undertaking that that work will continue, and that Jan Berry or someone like her will continue to monitor the reduction in the bureaucracy that is hampering the police in doing their job?

[Hide Media Fragment URI](#)

http://www.twofourdigital.net/UKParliament/Archive/0000014318_wmv.aspx#t=12590,12615

 **play this fragment**

More...



Time >> 12596.513s

Speaker Annotations DBpedia

Speaker Information from data.gov.uk

predicate	object
type	Person
description	Keith Vaz represents the constituency Leicester East
partyMemberOf	Labour
uriSet	Dataset of Ministers in the House of Commons
uriSet	Dataset of MPs
seeAlso	1118281
seeAlso	leicester_east
name	Keith Vaz
hasMembership	seat
hasMembership	6
holdsSeat	Leicester East
lastName	Vaz

Theresa May Conservative, from Maidenhead

<http://reference.data.gov.uk/id/mp/maidenhead/theresa-may>

I am happy to take up the point made by the right hon. Gentleman. Jan Berry did a very good job in looking at police bureaucracy. Obviously, she had considerable experience which enabled her to do that. I can reassure the right hon. Gentleman that the work will continue. We are already taking forward further work in a number of ways to examine the bureaucracy surrounding policing so that we can take further steps to reduce the amount of bureaucracy that the police have to deal with.

With a strong democratic mandate from the ballot box, police and crime commissioners will hold their chief constable to account for cutting crime. They will have the power to appoint and dismiss chief constables if they do not believe they are performing effectively. If the public do not believe that their police and crime commissioner is performing effectively, the commissioner will face the ultimate sanction of rejection at that same ballot box. Importantly, police and crime commissioners will set the annual budget for their force and will determine the local precept-the local contribution to policing costs.

Police authorities are not properly accountable for how public money is used, so they do not drive value for money in their forces. The democratic mandate of police and crime

FIGURE 8.1: Screenshot of UK Parliament Demonstration


Speaker Annotations **DBpedia** 1

"I am very pleased to hear the [right hon. Gentleman](#) [echoing](#) the very words that I have used to the [Association of Chief Police Officers](#) conference and other conferences when I have been speaking about the [key](#) aim of the [police](#), which is indeed to cut crime."

Speaker Annotations **DBpedia** 2

Reducing Bureaucracy in Policing


subject	10121326000115
object	http://www.policessupers.com/uploads/news/reducing-bureaucracy-policing.pdf




Speaker Annotations **DBpedia** 3

Theresa May repeat her point in ACPO conference

subject	10121326000118
object	http://youtu.be/l5roolqivEY?t=1m9s



Theresa May: Don't chase targ



0:00 / 1:21 YouTube

FIGURE 8.2: User generated annotations in UK Parliament Demonstration

government. For future research in this area, media fragments could be applied to answer the following questions:

1. Are a politicians statements contradictory at different times? The statements could be linked to the media fragments as evidence.
2. What are the key statements or blocks of debate that directly affect policy-making?
3. Which MP has made longest speeches in the whole year's (or month's) debating?
4. How long does each topic (finance, crime, foreign affairs, etc.) cover in the whole year's debate?
5. What is the changing sentiment of a given MP during the whole year, for which there is video evidence?

There are actually more questions that could be answered if the time-specific feature of the media fragments could be extracted from the Parliamentary debate.

8.2.2 Extension of Media Fragment Indexing Framework

In Chapter 7, initial evaluation showed that the snapshot pages were successfully fetched by Googlebot and more media fragments are expected to be indexed using this method, so that the search for multimedia resources would become more efficient. Theoretically, the implementation of the Media Fragment Enriching Framework in Chapter 6 could be applied to the backend server and the RDF description of media fragments could be created with the named entities extracted from the annotations as instances of the Core Model for Media Fragment Enrichment. The RDF descriptions could then be embedded in the snapshot pages using RDFa or Microdata, and deserialised by the Google search engine. In this way, media fragments could be massively indexed and searched by traditional search engines.

Currently, there is no clear definition for media fragments as objects in the embedded semantic markups, such as schema.org. This leads to the difficulty of embedding the RDF descriptions of media fragments, such as the Core Model for Media Fragment Enrichment proposed in Section 5.1. The properties defined in schema.org for a video object could be used, but it is still not clear what the title and description of media fragments should be. The use of the Google Ajax Crawler is a work around rather than standardised practice. If the media fragments cannot be separated from the landing page because of the convention of watching videos online, it is better for a crawler to be specifically designed for media fragment indexing. A standard design is preferred, by which named entity extractions could be applied to Tweets, and then the media fragments shared from YouTube and Dailymotion could be further linked to named entities in the LOD Cloud and embedded in the landing page.

In the future, an extension to the Media Fragment Indexing Framework will be needed to (1) extract named entities using the Media Fragment Enrichment Framework; (2)

standardise the embedding and visualisation of RDF descriptions of media fragments on the landing page using Rich Snippets; (3) evaluate the user-friendliness and efficiency of such landing pages compared with the traditional video landing pages.

In summary, the future research work to extend the Media Fragment Indexing Framework could encompass the following aspects:

1. Design a vocabulary to describe media fragments and annotations, and integrate it with schema.org.
2. Design a crawler and algorithm to crawl media fragments hosted in different multimedia sharing platforms. The crawler must consider the separation of landing pages and video files.
3. Design a UI to display media fragment searching results and evaluate it against traditional search results.

8.2.3 Visualisation of Media Fragments and Annotations On Second Screen

Wikipedia defines “second screen” as⁴:

A second screen refers to the use of a computing device (commonly a mobile device, such as a tablet or smartphone) to provide an enhanced viewing experience for content on another device, such as a television. In particular, the term commonly refers to the use of such devices to provide interactive features during “linear” content, such as a television program, served within a special app.

There are cases where users want to view relevant content of the program they are watching on TV. Hildebrand and Hardman (2013) introduces a use case of exploring relevant information about an artist discussed in a TV programme about antique objects *Tussen Kunst en Kitsch* on a second screen. The application on the second screen will display enrichment information related to the antique object or the artist displayed on the main screen. The enrichment information in this use case is actually named entities extracted from subtitles using the same mechanism proposed in the Media Fragment Enriching Framework. The linking between named entities and media fragments provides the possibility of showing the named entities information on a second screen when playing the TV programme on the main screen. A similar idea has also been introduced in Milicic et al. (2013), where a video fragment of a TV programme can be “grabbed” by a user and relevant concepts extracted from NERD are displayed on the second screen.

⁴http://en.wikipedia.org/wiki/Second_screen

The use of a second screen for media fragments on the Web, which is a “lean-forward medium”, will be different from its usage as an extension of TV, which is a lean-back medium. Romero et al. (2013) identifies four different changes between roles of second screen for TV and Web-enriched broadcast video. Web is more interactive than TV regarding the end users, so the second screen example in Milicic et al. (2013) could be further extended by the users making annotations about media fragments on the second screen, and the annotations pushed to the main screen and highlighted. For example, when watching the video, users take a screenshot of the current video frame and highlight a certain area of that picture on their tablets. Then the spatial area in the video on the main screen would also be highlighted.

In summary, the future research work on media fragments and second screening can encompass the following aspects:

1. Carry out use case studies on second screening applications and clarify the general requirements
2. Design an architecture for second screening applications, especially the models for data exchange, and make sure the media fragments are synchronised among different screens.
3. Design user interactions and evaluation.

8.2.4 Video Classification with Media Fragments

As the dataset used in Section 5.4 has not been reused in other similar research, it is difficult to compare the classification results with other classification methods. Google has released a dataset with 100 000 feature vectors extracted from public YouTube videos⁵ and some machine learning research has been conducted based on this video set (Madani et al., 2013).

All the videos in the dataset are well-labelled by one of the 30 classes, and textual features are provided together with visual and auditory features. So the algorithms specified in Section 5.4 could be verified using this dataset and compared with other classification results based on the same dataset. In addition, the algorithm could be further expanded with the combination of lower-level visual and auditory features.

8.2.5 Formal Descriptions of Interlinking Media Fragments Principles

Chapter 4 explained the Interlinking Media Fragments Principles, which defined the minimum requirement for publishing media fragments as Linked Data and interlinking

⁵<http://googleresearch.blogspot.co.uk/2013/11/released-data-set-features-extracted.html>

them to the LOD cloud. However, it is better to offer a formal mathematical description of the principles, which will be disambiguated and more rigorous for everyone to understand. Such a formal description has been bootstrapped and is displayed in Appendix C. Even though the description is still in a preliminary stage and not included in the main work, it may show potential directions for future research.

Appendix A

Namespaces Used in the Thesis

```
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix nsa:     <http://multimedialab.elis.ugent.be/organon/ontologies/ninsuna
#> .
@prefix dbpedia-owl: <http://dbpedia.org/ontology/> .
@prefix prov:     <http://www.w3.org/ns/prov#> .
@prefix foaf:     <http://xmlns.com/foaf/0.1/> .
@prefix linkedtv: <http://data.linkedtv.eu/ontology/> .
@prefix nerd:     <http://nerd.eurecom.fr/ontology#> .
@prefix rdfs:     <http://www.w3.org/2000/01/rdf-schema#> .
@prefix oa:       <http://www.w3.org/ns/oa#> .
@prefix str:      <http://nlp2rdf.lod2.eu/schema/string/> .
@prefix ma:       <http://www.w3.org/ns/ma-ont#> .
@prefix xsd:      <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:    <http://www.w3.org/2002/07/owl#> .
@prefix rdf:      <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
```

LISTING A.1: Namespaces Used in this Thesis

Appendix B

Results of Twitter Media Fragment Monitor Using Twitter Firehose API

This program used Twitter Firehose API to examine all the Tweets sent within one minute. If a Tweet does not contain any URL, it will be ignored. Otherwise, the program will use the Media Fragment URI¹ recommended by W3C Media Fragment Working Group to parse the URL(s) contained in the Tweet. If the parser can successfully return the fragment information encoded in the URL hash or query, then the program will classify it as a media fragment URI. After the one-minute monitoring, the author manually opened all the URLs recognised as media fragment URIs in Web browsers and examined whether the Web page or file actually represented a media fragment. The results are shown in the following table.

¹<https://github.com/tomayac/Media-Fragments-URI/>

TABLE B.1: Twitter Media Fragment Monitor Using Twitter Firehose API

Total Tweets examined	4356
Media Fragment URLs recognised by the parser	49
Real Media Fragments URLs	2
False positive rate	95.9%

The false positive rate is nearly 96%, which proves that parsing URLs in the Tweets from all the domains are error-prone and not efficient.

Appendix C

The Formal Definition of Media Fragment Publishing

This Appendix will give some initial work on a formal definition about the term "media fragment has been published as Linked Data". Section C.1 will firstly go through the mathematical model to represent RDF tripiaset. Most of current models representing RDF graph are used to model RDF entailment or analyse query and semantic reasoning, they could not be applied to Linked Data principles directly. So the RDF graph needs to be extended to the Web domain in order to model the functions defined in the four Linked Data principles. The mathematical model can be treated as the formal descriptions of the Interlinking Media Fragment Principles in Chapter 4.

C.1 RDF Documents as Labelled Directed Multigraph

According to the RDF syntax (Klyne and Carroll, 2004; Gutierrez et al., 2004), we can give definitions of following sets and functions used to describe RDF document (Table C.1). It is commonly accepted that RDF documents can be viewed as graphs, where resources and properties are treated as nodes and arcs.

TABLE C.1: Definition of sets and functions for the formal description of linked data

\mathbb{U}	set of URI references
$\mathbb{B} = \{_a, _b, _c, \dots\}$	set of Blank Nodes
\mathbb{L}	set of Literals, including plain literals and typed literals
$\mathbb{V} = \mathbb{U} \cup \mathbb{B} \cup \mathbb{L}$	set of Vocabulary; \mathbb{U}, \mathbb{B} and \mathbb{L} are pairwise disjoint
$\mathcal{U}(\mathbb{V}), \mathcal{B}(\mathbb{V}), \mathcal{L}(\mathbb{V})$	the set of URIs, Blank Nodes and Literal in \mathbb{V} respectively
$sub(\mathbb{V}), pred(\mathbb{V}), obj(\mathbb{V})$	all the elements appear in \mathbb{V} as subject, predicate and object respectively

Definition C.1. (RDF triple and triplesset) An RDF triple is denoted by $(s, p, o) \in (\mathbb{U} \cup \mathbb{B}) \times \mathbb{U} \times (\mathbb{U} \cup \mathbb{B} \cup \mathbb{L})$, in which s is the subject, p the predicate and o the object. The RDF triplesset \mathbf{T} is a subset of $(\mathbb{U} \cup \mathbb{B}) \times \mathbb{U} \times (\mathbb{U} \cup \mathbb{B} \cup \mathbb{L})$.

We denote by $\mathbb{V}(\mathbf{T})$ all the terms or vocabularies that has been used in tripleset \mathbf{T} and by $ent(\mathbf{T})$ all the subjects and objects (entities) in $\mathbb{V}(\mathbf{T})$. Then we have $\mathbb{V}(\mathbf{T}) = \{v \in \mathbb{V} \mid \exists(s, p, o) \in \mathbf{T}, v \in sub(\mathbb{V}) \text{ or } v \in pred(\mathbb{V}) \text{ or } v \in obj(\mathbb{V})\}$ and $ent(\mathbf{T}) = \{e \in \mathbb{V} \mid \exists(s, p, o) \in \mathbf{T}, e \in sub(\mathbb{V}) \text{ or } e \in obj(\mathbb{V})\}$.

There are many representations for RDF graphs, such as labelled directed (multi)graphs (LDMG) (Gutierrez et al., 2004), bipartite graphs (Hayes and Gutierrez, 2004) and labelled directed hypergraphs (Morales, 2007). LDMG gives a straight forward representation of RDF tripleset (Baget, 2005). Even though there are some flaws (Dau, 2006; Morales, 2007) for the LDMG representation, none of them will affect this research. So we adopt the LDMG representation for RDF tripleset \mathbf{T} and define the labelled, directed multigraph as:

Definition C.2. (Labelled Directed Multigraph Representation for RDF tripleset) Let \mathbf{T} be the RDF tripleset and \mathbb{V} be the terms used by \mathbf{T} . The LDMG representation of tripleset \mathbf{T} over the set of terms \mathbb{V} is \mathbf{G} . Each entity $e \in ent(\mathbf{T})$ represents a distinct node $n(e)$ in \mathbf{G} and each triple $t = \langle s, p, o \rangle \in \mathbf{T}$ represents a distinct arc $a(t)$. Then \mathbf{G} is a 4-tuple $\mathbf{G} = \langle N, A, \gamma, \epsilon \rangle$, where N is a finite set of nodes and $N = \{n(e) \mid e \in ent(\mathbf{T})\}$, A is a finite set of arcs and $A = \{a(t) \mid t \in \mathbf{T}\}$, $\gamma : A \rightarrow N \times N$ maps each arc to a pair of nodes as $\gamma(a(t)) = \langle n(s), n(o) \rangle$ given $s \in sub(\mathbb{V})$, $o \in obj(\mathbb{V})$ and $t \in \mathbf{T}$, and $\epsilon : N \cup A \rightarrow \mathbb{V}$ maps the nodes and arcs back to a term in \mathbb{V} as $\epsilon(n(e)) = e$ and $\epsilon(a(t)) = p$ given $p \in pred(\mathbb{V})$.

Definition (C.2) explains how a LDMG representation \mathbf{G} could be built from RDF tripleset \mathbf{T} . We can use $\mathbf{G} = \mathcal{M}(\mathbf{T})$ to represent this process. More detailed explanations about $\mathcal{M}(\mathbf{T})$ could be found at (Baget, 2005).

C.2 A Mathematics Description for Linked Data Principles

The definitions given in Section C.1 are based on abstract and conceptual domain, but the real application environment of linked data is under Web infrastructure, which consists of documents, links, etc. In order to formally describe the processes defined in linked data principles, such as dereferencing and interlinking, we need to find out the associations between conceptual graph domain and Web domain. The first and obvious connection is the URIs defined in \mathbf{G} . According to the Architecture of World Wide Web (Jacobs and Walsh, 2004), a URI addresses either a information resource or non-information on the Web. So we can define $\mathbb{U}^I \subseteq \mathbb{U}$ and $\mathbb{U}^N \subseteq \mathbb{U}$ as the URI sets for

information and non-information resources respectively. \mathbb{U}^I and \mathbb{U}^N are disjoint.

Another association is between RDF document on the Web and the RDF graph \mathbf{G} . \mathbf{Doc} is the set for documents on the Web and $\mathbf{R} \subseteq \mathbf{Doc}$ is the document containing RDF triples. We assume \mathbf{T}_r , which is the subset of \mathbf{T} , as the tripleset represented in document r . i.e. $g(r) = \mathbf{T}_r$, where $r \in \mathbf{R}$. As Definition (C.2) indicates, for each $r \in \mathbf{R}$, there is a mapping $\mathcal{M}(g(r)) = \mathbf{H}$ where \mathbf{H} is the subgraph of \mathbf{G} .

As a summary, every node defined in RDF graph \mathbf{G} indicates a URI in Web domain and every RDF document r defined Web domain has a corresponding RDF graph \mathbf{H} . Given these two associations, we can map the sets and functions in Web domain to RDF graph domain.

The second rule of linked data requires all the URIs be HTTP URIs. So we can define \mathbb{U}_H as the set for HTTP URIs and \mathbf{G}_H is the RDF graph for terms \mathbb{V} where $\mathcal{U}(\mathbb{V}) \subseteq \mathbb{U}_H$. The information URI and non-information URI are defined as \mathbb{U}_H^I and \mathbb{U}_H^N respectively.

The dereferenceable HTTP URIs can be divided into two categories in linked data (Sauer-mann and Cyganiak, 2008). If the URI indicates a information resource, it directly returns the RDF document with "2XX" status code. If the URI indicates a non-information resource, there are two solutions. One is attaching hash information based on the information document, such as <http://example.org/about#alice>. "#alice" indicates a person in the real world, while <http://example.org/about> is a document on the Web. Another solution is using "3XX" to redirect the non-information URI to a real document on the Web. We use $resolve(u_H^N) = u_H^I$ to indicate the generation of the information URI u_H^I from a non-information URI u_H^N in both hash and "3XX" redirection situations. Other than the two categories, if dereferencing HTTP URI returns "4XX" or "5XX" status code, the nature of the resource is unknown (as defined in httpRang-14¹). In this situation, we say $u_H \in \mathbb{U}_H^K$, because we cannot definitely say $u_H \notin \mathbb{U}_H^I$ or $u_H \notin \mathbb{U}_H^N$.

Definition C.3. (Dataset is dereferenceable) The dereferencing of HTTP URI can be defined as a partial function $deref : \mathbb{U}_H \rightharpoonup \mathbf{H}$, where:

$$deref(u_H) = \begin{cases} deref(u_H) & \text{for } u_H \in \mathbb{U}_H^I \\ deref(resolve(u_H)) & \text{for } u_H \in \mathbb{U}_H^N \\ undefined & \text{for } u_H \in \mathbb{U}_H^K \end{cases}$$

So if $u_H \in \mathbb{U}_H^K$, the URI is not dereferenceable. If $deref(u_H)$ is a total function, i.e. for any $n \in N$ in \mathbf{G}_H , there is $deref(\epsilon(n)) = \mathbf{H}$ and $\mathbf{H} \subseteq \mathbf{G}_H$, we say the tripleset \mathbf{T}_H represented by RDF graph \mathbf{G}_H is dereferenceable.

One thing needs to emphasised is that \mathbf{H} could be an empty graph, which means the dereferencing returns a document with no RDF triples. This situation is contradicted

¹<http://lists.w3.org/Archives/Public/www-tag/2005Jun/0039.html>

to the third rule of linked data, but does not violate the rules defined in httpRang-14 as the word "dereferencing" is not limited to RDF representation. So in linked data, $\mathbf{H} = \emptyset$ can only be treated as a bad practice when publishing data, which indicates the server does not provide useful description for the URI and the traverse of the RDF graph reaches the end.

The fourth rule of linked data implies that the published tripliset (or dataset) should be linked to triplesets in other domains. In other words, the newly published dataset should include HTTP URIs from the linked data cloud and there should be links between those URIs and newly published URIs in the dataset. So we can formally define the publishing of datasets as:

Definition C.4. (Interlinking Datasets into Linked Data Cloud) Let $\mathbf{T}_H^C \neq \emptyset$ be the complete set of all the triples published in the linked data cloud before the new dataset $\mathbf{T}'_H \not\subseteq \mathbf{T}_H^C$ is added into linked data cloud and $\mathbf{G}_H^C = \langle N^C, A^C, \gamma^C, \epsilon^C \rangle$ be the RDF graph representation of \mathbf{T}_H^C . Let $\mathbb{U}'_H = \mathcal{U}(\mathbb{V}(\mathbf{T}'_H))$ and $\mathbb{U}_H = \mathcal{U}(\mathbb{V}(\mathbf{T}_H^C))$ be the HTTP URIs in \mathbf{T}'_H and \mathbf{T}_H^C respectively. The new dataset \mathbf{T}'_H represented by $\mathbf{G}'_H = \langle N', A', \gamma', \epsilon' \rangle$ is successfully interlinked in original linked data cloud iff \mathbf{T}'_H is dereferenceable and $\mathbb{U}'_H \cap \mathbb{U}_H \neq \emptyset$ and $\exists a \in A'$ where $\gamma'(a) = \langle n_i, n_j \rangle$ ($i \neq j$) and either $n_i \in \mathbb{U}'_H \cap \mathbb{U}_H$ or $n_j \in \mathbb{U}'_H \cap \mathbb{U}_H$, not both. The RDF graph $\mathbf{G}_H^{C'} = \langle N^{C'}, A^{C'}, \gamma^{C'}, \epsilon^{C'} \rangle$ after publishing \mathbf{T}'_H into linked data cloud could be built as follows:

1. Let $n'_j = n_i^C$, where $n'_j \in N'$ and $n_i^C \in N^C$, if $\epsilon'(n'_j) = \epsilon^C(n_i^C)$
2. Based on the first step, let $a'_m = a_k^C$, where $a'_m \in A'$ and $a_k^C \in A^C$, if $\epsilon'(a'_m) = \epsilon^C(a_k^C)$
3. Based on the previous two steps, let $N^{C'} = N^C \cup N'$, $A^{C'} = A^C \cup A'$ and $\mathbf{T}_H^{C'} = \mathbf{T}^C \cup \mathbf{T}'$
4. $\gamma^{C'} : A^{C'} \rightarrow N^{C'} \times N^{C'}$
5. $\epsilon^{C'} : N^{C'} \cup A^{C'} \rightarrow \mathbb{V}(\mathbf{T}_H^{C'})$

$\mathbb{U}_H \cap \mathbb{U}_H \neq \emptyset$ in the definition means that some URIs used in the new dataset should already exist in the linked data cloud. These URIs are linked with other URIs in new dataset in \mathbf{G}'_H through triples. " $\exists a \in A'$ where $\gamma'(a) = \langle n_i, n_j \rangle$ ($i \neq j$) and either $n_i \in \mathbb{U}'_H \cap \mathbb{U}_H$ or $n_j \in \mathbb{U}'_H \cap \mathbb{U}_H$, not both" makes sure that in a newly added link, if one end is in \mathbf{G}_H^C , the other end should be in \mathbf{G}'_H .

Figure C.1 is an example of Definition (C.4). \mathbf{G}^C is the complete set presenting all triples, where $N^C = \{u_1, u_2, u_3, u_4, u_5, u_6\}$ and $A^C = a_1, a_2, a_3, a_4, a_5, a_6, a_7$. \mathbf{G}' , where $N' = \{u'_1, u'_2, u'_3, u'_4, u'_6\}$ and $A' = a'_1, a'_2, a'_3, a'_4, a'_5, a'_6$, is the tripliset that is going to be interlinked with \mathbf{G}^C . We have $u_1 = u'_1$, $u_2 = u'_2$ and $u_6 = u'_6$. In the interlinking, u'_1

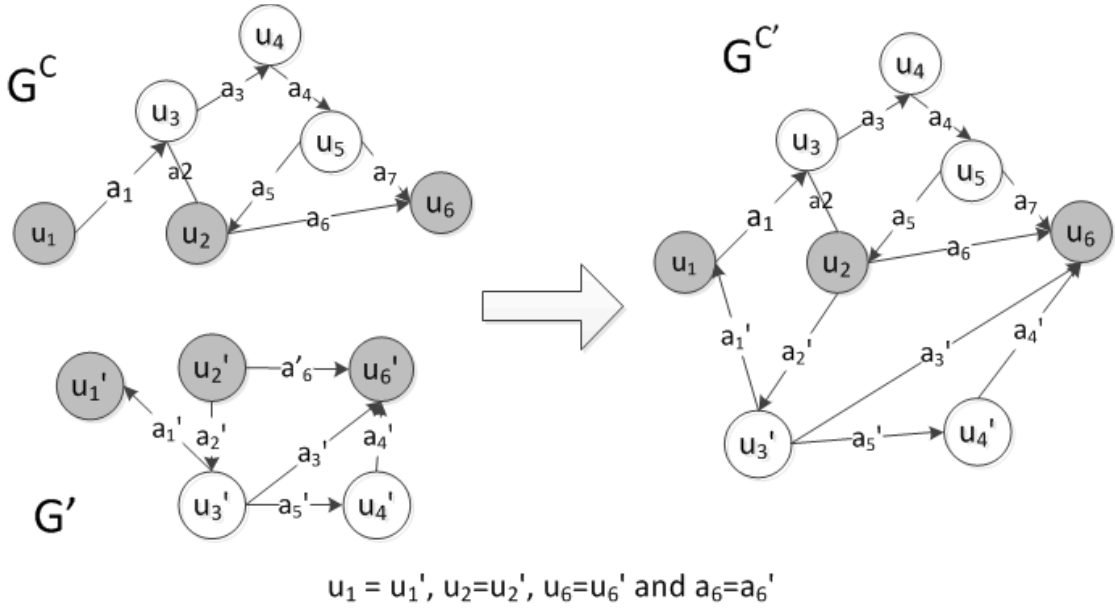


FIGURE C.1: An example of interlinking datasets into the linked data cloud

is merged with u_1 , the same as u_2' and u_6' . $a_6' = a_6$ indicates that both ends of a_6' are already in G^C and $\epsilon'(a_6') \neq \epsilon^C(a_6)$. So in the result graph $G^{C'}$, a_6' is merged with a_6 . For a_1', a_2', a_3' and a_4' , one end of them is in $N^C \cap N'$, so they should be included in $G^{C'}$ as the "link to other URIs" in the fourth rule of linked data principles. One thing which should be noticed is that there might be possibility that a_6 and a_6' share the same start and end nodes, but $\epsilon'(a_6') \neq \epsilon^C(a_6)$, i.e. using different predicates to link the same pair of nodes. In this case, a_6 and a_6' should not be merged together.

The formal definitions given in this section shows an initial attempt to bring the mathematical model of RDF graph and Web infrastructure together. In the real linked data world, the situation is much more complex. We have not considered SPARQL endpoint as a means of dereferencing and dataset query. In addition, the blank node, literal and situations of empty graph are also ignored. A more detailed formal description of linked data will be left for the future work of this research.

C.3 Formalisation of Publishing Media Fragments as Linked Data

According to Tim Berners-Lee's 5 star publishing process (Berners-Lee, 2006), at 4 start level, the data has already been published as on the Web with unique identifier. For many multimedia applications online, it is not necessary (or even possible) for the developers of those applications to deliberately link their resources to other datasets. For example, an application only hosting multimedia resources can publish all the resources and media fragments as URIs and wait for other users link annotations to the media

fragments. So the first initial step of publishing media fragments should be making them available online and dereferenceable.

Definition C.5. (Publishing Media Fragments as Linked Data) Let \mathbb{U}_M be the HTTP URIs for media fragments and \mathbf{T}_M , where $\mathbb{U}_M \subseteq \text{ent}(\mathbf{T}_M)$, be the new dataset which is going to be published to the linked data cloud \mathbf{T}_H^C . We say that media fragments have been successfully published as linked data if following conditions are satisfied:

1. $\mathbb{U}_M \neq \emptyset$
2. $\forall u \in \text{ent}(\mathbf{T}_M) \setminus \text{ent}(\mathbf{T}_H^C), u$ is dereferenceable

Media fragment URIs must be provided in the dataset and must be dereferenceable, as well as other HTTP URIs in the dataset. But if \mathbf{T}_M also link to datasets in other domain, we do not require the URIs from other domains (or namespaces) are dereferenceable. This is because we have no control of URIs in other domains. This situation needs to be considered carefully if the multimedia application wants to link its annotations to media fragments in other domains.

For example, we cannot expect that the dereferencing of this media fragment URI *http://other-domain.org/test.mp4#t=3,7* in other domain returns an RDF document. So it is not a good practice to directly create triples like *my:annotation ex:annotates <http://other-domain.org/test.mp4#t=3,7>*. To solve this problem, one way is to mint a non-information media fragment URI in your own namespace (Heath and Bizer, 2011), such as *http://my-domain.org/test#t=3,7*. If the request asks for RDF representation, we can return the RDF document saved in local domain. If the request asks for the original representation, we redirect the request to the real location of the file, which is at *other-domain.org*. Otherwise, we can state the equivalence between *http://my-domain.org/test#t=3,7* and *http://other-domain.org/test.mp4#t=3,7* using *rdf:seeAlso*, *owl:sameAs* or other vocabularies.

In this situation, using hash URIs is better than slash URIs as discussed in Section 4. If the fragment information is encoded in slash URIs pointing to a resource in other namespaces, for example *http://other-domain.org/test.mp4/3,7*, "other-domain.org" may return "4XX" not found error, which makes the media fragment URI dereferenceable. The hash URIs are safe as the information behind hash will not be passed to the server. The dereferencing of *http://other-domain.org/test.mp4#t=3,7* can at least return a document with no RDF triples, which will not violate the rules of "dereferenceable".

Bibliography

- Ben Adida and Mark Birbeck. RDFa Primer, October 2008.
<http://www.w3.org/TR/xhtml-rdfa-primer/>.
- James F Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983. ISSN 00010782.
- Richard Arndt, Raphaël Troncy, Steffen Staab, Lynda Hardman, and Miroslav Vacura. COMM: Designing a Well-Founded Multimedia Ontology for the Web. In *International Semantic Web Conference*, volume 4825 of *Lecture Notes in Computer Science*, pages 30–43. Springer Berlin Heidelberg, 2007. ISBN 9783540762973.
- Lora Aroyo, Lyndon Nixon, and Libby Miller. Notube: the television experience enhanced by online social and semantic data. In *Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference on*, pages 269–273. IEEE, 2011.
- Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. DBpedia: A Nucleus for a Web of Open Data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudré-Mauroux, editors, *The Semantic Web*, volume 4825 of *LNCS*, pages 722–735. Springer, 2007. ISBN 9783540762973.
- Sören Auer, Sebastian Dietzold, Jens Lehmann, Sebastian Hellmann, and David Aumueller. Triplify: light-weight linked data publication from relational databases. *International World Wide Web Conference*, pages 621–630, 2009.
- Jean-Francois Baget. Rdf entailment as a graph homomorphism. In Yolanda Gil, Enrico Motta, V. Benjamins, and Mark Musen, editors, *The Semantic Web ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, pages 82–96. Springer Berlin / Heidelberg, 2005. ISBN 978-3-540-29754-3. 10.1007/11574620_9.
- Luisa Bentivogli, Pamela Forner, Bernardo Magnini, and Emanuele Pianta. Revising the wordnet domains hierarchy: semantics, coverage and balancing. In *Workshop on Multilingual Linguistic Resources*, 2004.
- Tim Berners-Lee. Linked Data - Design Issues, 2006.

- Tim Berners-Lee. Long Live the Web: A Call for Continued Open Standards and Neutrality. *Scientific American*, 303(6):1–8, 2010. ISSN 00368733.
- Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3logic: A logical framework for the world wide web. *Theory and Practice of Logic Programming*, 8(3):249–269, 2008.
- Tim Berners-Lee, Roy T Fielding, and L Masinter. RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax, 2005.
- Tim Berners-Lee, J Hendler, and O Lassila. The Semantic Web. *Scientific American*, 284(5):34–43, 2001. ISSN 00368733.
- Diego Berrueta and Jon Phipps. Best Practice Recipes for Publishing RDF Vocabularies, 2008.
- Chris Bizer, Richard Cyganiak, and Tom Heath. How to Publish Linked Data on the Web, 2007.
- Christian Bizer and Richard Cyganiak. D2R Server - Publishing Relational Databases on the Semantic Web. In *Proceedings of the 5th International Semantic Web Conference*. Springer, 2006.
- Stefano Bocconi, Frank Nack, and Lynda Hardman. Automatic generation of matter-of-opinion video documentaries. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(2):139–150, 2008.
- Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- Jan Bormans and Keith Hill. MPEG-21 Overview v.5, 2002.
- Damian Borth, Jörn Hees, Markus Koch, Adrian Ulges, Christian Schulze, Thomas Breuel, and Roberto Paredes. Tubefiler: an automatic web video categorizer. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 1111–1112. ACM, 2009.
- Niels Olof Bouvin and René Schade. Integrating temporal media and open hypermedia on the world wide web. *Computer Networks*, 31(11):1453–1465, 1999.
- Darin Brezeale and Diane J Cook. Automatic video classification: A survey of the literature. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 38(3):416–430, 2008.
- Dan Brickley and Ramanathan V Guha. RDF Vocabulary Description Language 1.0: RDF Schema, 2004.

- M Cecelia Buchanan and Polle T Zellweger. Specifying temporal behavior in hypermedia documents. In *Proceedings of the ACM conference on Hypertext*, pages 262–271. ACM, 1992.
- Simon Buckingham Shum, David De Roure, Marc Eisenstadt, Nigel Shadbolt, and Austin Tate. CoAKTinG: Collaborative Advanced Knowledge Technologies in the Grid. *Proc Second Workshop on Advanced Collaborative Environments Eleventh IEEE Int Symp on High Performance Distributed Computing HPDC11*, pages 24–26, 2002.
- Vennevar Bush. As we may think. *The Atlantic Monthly*, 176(1):101–108, 1945.
- Gianluca Correndo, Manuel Salvadores, Ian Millard, and Nigel Shadbolt. Linked Time-lines: Temporal Representation and Management in Linked Data. *1st International Workshop on Consuming Linked Data COLD 2010 Shanghai*, 2010.
- Bin Cui, Ce Zhang, and Gao Cong. Content-enriched classifier for web video classification. In *33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 619–626, 2010.
- Frithjof Dau. Rdf as graph-based, diagrammatic logic. In Floriana Esposito, Zbigniew Ras, Donato Malerba, and Giovanni Semeraro, editors, *Foundations of Intelligent Systems*, volume 4203 of *Lecture Notes in Computer Science*, pages 332–337. Springer Berlin / Heidelberg, 2006. ISBN 978-3-540-45764-0. 10.1007/11875604_38.
- Hugh Davis, Wendy Hall, Ian Heath, Gary Hill, and Rob Wilkins. Towards an integrated information environment with open hypermedia systems. *Environment*, pages 181–190, 1992.
- James R Davis and Daniel P Huttenlocher. Shared annotation for cooperative learning. In *The first international conference on Computer support for collaborative learning*, pages 84–88. L. Erlbaum Associates Inc., 1995.
- Davy Van Deursen, Wim Van Lancker, Wesley De Neve, Tom Paridaens, Erik Mannens, and Rik Van De Walle. NinSuna: a fully integrated platform for format-independent multimedia content adaptation and delivery using Semantic Web technologies. *Multimedia Tools and Applications*, 46(2):371–398, 2009. ISSN 13807501.
- Douglas C Engelbart. *A conceptual framework for the augmentation of mans intellect*, volume 1, pages 1–29. Spartan Books, 1963.
- Orri Erling and Ivan Mikhailov. RDF Support in the Virtuoso DBMS. *Proceedings of the 1st Conference on Social Semantic Web CSSW*, 221 (ISBN 978-3-88579-207-9): 59–68, 2007.
- Ivan Ermilov, Sören Auer, and Claus Stadler. Crowd-sourcing the large-scale semantic mapping of tabular data. *Proceeding of the ACM Web Science http://svn.aksu.org/papers/2013/WebSci_CSV2RDF/public.pdf*, 2013.

- Katja Filippova and Keith B Hall. Improved video categorization from text metadata and user comments. In *34th International ACM SIGIR Conference on Research and development in Information Retrieval*, pages 835–842, 2011.
- Hugh Glaser, Ian C Millard, and Afraz Jaffri. RKBExplorer . com : A Knowledge Driven Infrastructure for Linked Data Providers. *Proceedings of the 5th European Semantic Web Conference ESWC 2008*, pages 797–801, 2008.
- Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- Nicola Guarino. Formal ontology and information systems. *Proceedings of FOIS98*, 46 (June):3–15, 1998.
- Claudio Gutierrez, Carlos Hurtado, and Alberto O Mendelzon. Foundations of semantic web databases. *Proceedings of the twentythird ACM SIGMODSIGACTSIGART symposium on Principles of database systems PODS 04*, 77(3):95, 2004.
- Wolfgang Halb, Yves Raimond, and Michael Hausenblas. Building linked data for both humans and machines. *WWW 2008 Workshop Linked Data on the Web LDOW2008 Beijing China*, 2007.
- Lynda Hardman, Dick CA Bulterman, and Guido Van Rossum. The amsterdam hypermedia model: adding time and context to the dexter model. *Communications of the ACM*, 37(2):50–62, 1994.
- Lynda Hardman, Željko Obrenović, Frank Nack, Brigitte Kerhervé, and Kurt Piersol. Canonical processes of semantically annotated media production. *Multimedia Systems*, 14(6):327–340, 2008.
- Steve Harris, Nick Lamb, and Nigel Shadbolt. 4store : The design and implementation of a clustered rdf store. *Time*, pages 81–96, 2009.
- Bernhard Haslhofer, Wolfgang Jochum, Ross King, Christian Sadilek, and Karin Schellner. The LEMO annotation framework: weaving multimedia annotations with the web. *International Journal on Digital Libraries*, 10(1):15–32, 2009. ISSN 14325012.
- Bernhard Haslhofer, Robert Sanderson, Rainer Simon, and Herbert Van de Sompel. Open annotations on multimedia web resources. *Multimedia Tools and Applications*, pages 1–21, 2012.
- Michael Hausenblas. Linked Data Applications. *First Community Draft DERI*, (July), 2009.
- Michael Hausenblas, Raphaël Troncy, Tobias Bürger, and Yves Raimond. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. *WWW 2009 Workshop Linked Data on the Web LDOW2009*, 2009a.

- Michael Hausenblas, Raphaël Troncy, Tobias Bürger, and Yves Raimond. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. *WWW 2009 Workshop Linked Data on the Web LDOW2009*, 2009b.
- Jonathan Hayes and Claudio Gutierrez. Bipartite graphs as intermediate model for rdf. *The Semantic WebISWC 2004*, (1030810):47–61, 2004.
- Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*, volume 1. Morgan & Claypool, 2011.
- Sebastian Hellmann, Jens Lehmann, and Sören Auer. Linked-data aware uri schemes for referencing text fragments. In *Knowledge Engineering and Knowledge Management*, pages 175–184. Springer, 2012.
- Jim Hendler. Web 3.0 Emerging. *Computer*, 42(1):111–113, 2009. ISSN 00189162.
- Ian Hickson. HTML Microdata, February 2012. <http://dev.w3.org/html5/md/>.
- Michiel Hildebrand and Lynda Hardman. Using explicit discourse rules to guide video enrichment. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 461–464. International World Wide Web Conferences Steering Committee, 2013.
- Chunneng Huang, Tianjun Fu, and Hsinchun Chen. Text-based video content classification for online video-sharing sites. *Journal of the American Society for Information Science and Technology*, 61(5):891–906, 2010.
- Ian Jacobs and Norman Walsh. Architecture of the World Wide Web. *arq Architectural Research Quarterly*, 2(01):1–56, 2004.
- Michael O Jewell, K Faith Lawrence, Mischa M Tuffield, Adam Prugel-Bennett, David E Millard, Mark S Nixon, Nigel R Shadbolt, et al. Ontomedia: An ontology for the representation of heterogeneous media. In *In Proceeding of SIGIR workshop on Mutlimedia Information Retrieval*. ACM SIGIR, 2005.
- Damir Juric, Laura Hollink, and Geert-Jan Houben. Discovering links between political debates and media. *ICWE 2013*, 2013.
- José Kahan, M-R Koivunen, Eric Prud’Hommeaux, and Ralph R Swick. Annotea: an open rdf infrastructure for shared web annotations. *Computer Networks*, 39(5):589–608, 2002.
- Polyxeni Katsiouli, Vassileios Tsetsos, and Stathes Hadjiefthymiades. Semantic video classification based on subtitles and domain terminologies. In *Workshop on Knowledge Acquisition from Multimedia Content (SAMT’07)*, 2007.
- Graham Klyne and Jeremy J Carroll. Resource description framework (rdf): Concepts and abstract syntax. *Structure*, 10(February):1–20, 2004.

- Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media Meets Semantic Web How the BBC Uses DBpedia and Linked Data to Make Connections. In Lora Aroyo, Paolo Traverso, Fabio Ciravegna, Philipp Cimiano, Tom Heath, Eero Hyvönen, Riichiro Mizoguchi, Eyal Oren, Marta Sabou, and Elena Simperl, editors, *The Semantic Web Research and Applications*, volume 5554 of *Lecture Notes in Computer Science*, pages 723–737. Springer, 2009. ISBN 9783642021206.
- Soriris Kotsiantis. Supervised machine learning: A review of classification techniques. *Frontiers in Artificial Intelligence and Applications*, 160:3, 2007.
- Dave Lambert and Hong Qing Yu. Linked Data Based Video Annotation and Browsing for Distance Learning. In *The 2nd International Workshop on Semantic Web Applications in Higher Education (SemHE’10)*, November 2010.
- Avraham Leff and James T Rayfield. Web-application development using the model/view/controller design pattern. *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*, 0:118–127, 2001.
- Yunjia Li, Giuseppe Rizzo, José Luis Redondo García, Raphaël Troncy, Mike Wald, and Gary Wills. Enriching media fragments with named entities for video classification. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 469–476. International World Wide Web Conferences Steering Committee, 2013.
- Yunjia Li, Giuseppe Rizzo, Raphaël Troncy, Mike Wald, and Gary Wills. Creating enriched YouTube media fragments with NERD using timed-text. *11th International Semantic Web Conference, Demo Session*, 2012.
- Yunjia Li, Mike Wald, Shakeel Khoja, Gary Wills, David Millard, Jiri Kajaba, Priyanaka Singh, and Lester Gilbert. Synote: enhancing multimedia e-learning with synchronised annotation. In *Proceedings of the first ACM international workshop on Multimedia technologies for distance learning*, pages 9–18. ACM, 2009.
- Yunjia Li, Mike Wald, Tope Omitola, Nigel Shadbolt, and Gary Wills. Synote: Weaving Media Fragments and Linked Data. In *5th International Workshop on Linked Data on the Web (LDOW’12)*, 2012.
- Yunjia Li, Mike Wald, and Gary Wills. Interlinking multimedia annotations. In *ACM WebSci’11*, pages 1–4, June 2011a. WebSci Conference 2011.
- Yunjia Li, Mike Wald, Gary Wills, Shakeel Khoja, David Millard, Jiri Kajaba, Priyanka Singh, and Lester Gilbert. Synote: development of a web-based tool for synchronized annotations. *New Review of Hypermedia and Multimedia*, 17(3):295–312, 2011b.
- Omid Madani, Manfred Georg, and David A. Ross. On using nearly-independent feature families for high precision and confidence. *Machine Learning*, 92:457–477, 2013. published online 30 May 2013.

- Erik Mannens, Davy Van Deursen, Raphal Troncy, Silvia Pfeiffer, Conrad Parker, Yves Lafon, Jack Jansen, Michael Hausenblas, and Rik Van de Walle. A uri-based approach for addressing fragments of media resources on the web. *Multimedia Tools and Applications*, pages 1–25. ISSN 1380-7501. 10.1007/s11042-010-0683-z.
- Frank Manola and Eric Miller. RDF Primer, 2004.
- Jose M. Martinez. Standards-mpeg-7 overview of mpeg-7 description tools, part 2. *MultiMedia, IEEE*, 9(3):83–93, 2002.
- Jose M Martinez. MPEG-7 Overview, 2004.
- Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
- Vuk Milicic, Giuseppe Rizzo, José Luis Redondo García, and Raphaël Troncy. Grab your favorite video fragment: Interact with a Kinect and discover enriched hypervideo. Number EURECOM+4058, Como, ITALY, 06 2013.
- Amadis Antonio Martinez Morales. A directed hypergraph model for rdf. In Elena Paslaru Bontas Simperl, Jrg Diederich, and Guus Schreiber, editors, *Proceedings of the KWEPSY 2007 Knowledge Web PhD Symposium 2007, Innsbruck, Austria, June 6, 2007*, volume 275 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.
- Andrew Y Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *21st International Conference on Machine learning*, 2004.
- Juan Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. *Computer Vision–ECCV 2010*, pages 392–405, 2010.
- Lyndon Nixon. The importance of linked media to the future web: lime 2013 keynote talk – a proposal for the linked media research agenda. In *Proceedings of the 22nd international conference on World Wide Web companion*, WWW ’13 Companion, pages 455–456, Republic and Canton of Geneva, Switzerland, 2013a. International World Wide Web Conferences Steering Committee. ISBN 978-1-4503-2038-2.
- Lyndon Nixon. Linked services infrastructure: a single entry point for online media related to any linked data concept. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 7–10. International World Wide Web Conferences Steering Committee, 2013b.
- Lyndon Nixon, Matthias Bauer, Cristian Bara, Thomas Kurz, and John Pereira. Connectme: Semantic tools for enriching online video with web content. In *I-SEMANTICS (Posters & Demos)*, pages 55–62, 2012.

- Ryuichi Ogawa, Hiroaki Harada, and Asao Kaneko. Scenario-based hypermedia: A model and a system. In *ECHT*, volume 90, pages 38–51, 1990.
- Silvia Pfeiffer, Conrad D Parker, and Andre T Pang. Specifying Time Intervals in URI Queries and Fragments of Time-Based Web Resources, 2005.
- Eric Prud’hommeaux and Andy Seaborne. SPARQL Query Language for RDF, 2008.
- Yves Raimond, Michael Smethurst, Andrew McParland, and Christopher Lowis. Using the past to explain the present: interlinking current affairs with archives via the semantic web. In *The Semantic Web–ISWC 2013*, pages 146–161. Springer, 2013.
- Giuseppe Rizzo and Raphaël Troncy. NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. In *13th Conference of the European Chapter of the Association for computational Linguistics (EACL’12)*, 2012.
- Alex Rodriguez. Restful web services: The basics. *IBM Developer Works*, pages 1–12, 2008.
- Lilia Pérez Romero, Myriam C Traub, HR Leyssen, and Lynda Hardman. Second screen interactions for automatically web-enriched broadcast video. In *Submitted to: ACM SIGCHI Conference on Human Factors in Computing Systems (CHI 2013) Exploring And Enhancing the User Experience for Television Workshop, Paris, France*, 2013.
- Carsten Saathoff and Ansgar Scherp. Unlocking the semantics of multimedia presentations in the web with the multimedia metadata ontology. *Proceedings of the 19th international conference on World wide web WWW 10*, page 831, 2010.
- Felix Sasaki, Tobias Bürger, Joakim Söderberg, Véronique Malaisé, Florian Stegmaier, and WonSuk Lee. Ontology for media resource 1.0, February 2012. <http://www.w3.org/TR/mediaont-10/>.
- Leo Sauermann and Richard Cyganiak. Cool URIs for the Semantic Web. *W3C Note*, 49(681):1–15, 2008.
- Ronald Schroeter, Jane Hunter, Jonathon Guerin, Imran Khan, and Michael Henderson. A synchronous multimedia annotation system for secure collaboratories. In *e-Science and Grid Computing, 2006. e-Science’06. Second IEEE International Conference on*, pages 41–41. IEEE, 2006.
- Nigel Shadbolt, Tim Berners-Lee, and Wendy Hall. The Semantic Web Revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.
- Nigel Shadbolt and Wendy Hall. Here comes Everything: Preparing for the Linked Web of Data. 2010.
- Nigel Shadbolt, Kieron O’Hara, Manuel Salvadores, and Harith Alani. egovernment, 2011. DOI 10.1007/978-3-540-92913-0_20.

- Vincent Simonet. Classifying youtube channels: a practical system. In *Proceedings of the 22nd international conference on World Wide Web companion*, pages 1295–1304. International World Wide Web Conferences Steering Committee, 2013.
- Michael Smith, Deborah L McGuinness, Raphael Volz, and Chris Welty. OWL Web Ontology Language Guide, 2004.
- Thomas Steiner. SemWebVid - Making Video a First Class Semantic Web Citizen and a First Class Web Bourgeois. In *Proceedings of the 9th International Semantic Web Conferene ISWC 2010*, pages 3–6. Springer, 2010.
- Thomas Steiner, Raphaël Troncy, and Michael Hausenblas. How Google is using Linked Data Today and Vision For Tomorrow. In Sören Auer, Stefan Decker, and Manfred Hauswirth, editors, *Linked Data in the Future Internet 2010*, Ghent,Belgium, 2010.
- Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.
- Raphaël Troncy, Erik Mannens, Silvia Pfeiffer, and Davy Van Deursen. Protocol for media fragments 1.0 resolution in http, December 2011. <http://www.w3.org/TR/media-frags/>.
- Raphaël Troncy, Erik Mannens, Silvia Pfeiffer, and Davy Van Deursen. Media fragments URI 1.0 (basic), March 2012. <http://www.w3.org/TR/media-frags/>.
- Giovanni Tummarello, Renaud Delbru, and Eyal Oren. Sindice.com: Weaving the open linked data. In Karl Aberer, Key-Sun Choi, Natasha Noy, Dean Allemang, Kyung-Il Lee, Lyndon Nixon, Jennifer Golbeck, Peter Mika, Diana Maynard, Riichiro Mizoguchi, Guus Schreiber, and Philippe Cudr-Mauroux, editors, *The Semantic Web*, volume 4825 of *Lecture Notes in Computer Science*, pages 552–565. Springer Berlin / Heidelberg, 2007. ISBN 978-3-540-76297-3. 10.1007/978-3-540-76298-0_40.
- Davy Van Deursen, Raphaël Troncy, Erik Mannens, Silvia Pfeiffer, Yves Lafon, and Rik Van De Walle. Implementing the media fragments URI specification. *Proceedings of the 19th international conference on World wide web WWW 10*, page 1361, 2010.
- Davy Van Deursen, Wim Van Lancker, Erik Mannens, and Rik Van de Walle. Experiencing standardized media fragment annotations within html5. *Multimedia Tools and Applications*, pages 1–20, 2012.
- Wim Van Lancker, Davy Van Deursen, Ruben Verborgh, and Rik Van de Walle. Semantic media decision taking using n3logic. *Multimedia Tools and Applications*, pages 1–20, 2013.
- Jörg Waitelonis and Harald Sack. Augmenting video search with linked open data. In *Proceedings of International Conference on Semantic Systems (i-semantics), September 2-4, 2009, Graz, Austria*. Verlag der TU Graz, Austria, 2009.

Mike Wald, Yunjia Li, EA Draffan, and Wei Jing. Synote mobile html5 responsive design video annotation application. *UACEE International Journal of Advances in Computer Science and its Applications (IJCSIA)*, 3(2):207–211, 2013.

Zhiyong Wang, Genliang Guan, Yu Qiu, Li Zhuo, and Dagan Feng. Semantic context based refinement for news video annotation. *Multimedia Tools and Applications*, pages 1–21, 2012.

John R Zhang, Yang Song, and Thomas Leung. Improving video classification via youtube video co-watch data. In *Workshop on Social and behavioural networked media access*, 2011.