Weaving the Web(VTT) of Data

Thomas Steiner,[1] Hannes Mühleisen,[2] **Ruben Verborgh**,[3]
Pierre-Antoine Champin,[1] Benoît Encelle,[1] and Yannick Prié[4]

[1] CNRS, Université de Lyon LIRIS, UMR5205 Université Lyon 1, FR
[2] Database Architectures Group CWI, Amsterdam, NL
[3] Multimedia Lab Ghent University – iMinds, Gent, BE
[4] LINA – UMR 6241 CNRS Université de Nantes, Nantes, FR

{tsteiner,pachampin,bencelle}@liris.cnrs.fr
hannes@cwi.nl
ruben.verborgh@ugent.be
yannick.prie@univ-nantes.fr

Contributions

Agenda
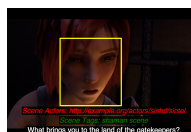
- Large-Scale Common Crawl study of the state of Web video.

- WebVTT conversion to RDF-based Linked Data.

- Online video annotation format and editor.

- Data and code.

Introduction

From <OBJECT> to <video>

- In the "ancient" times of HTML 4.01, the `<OBJECT>` tag was intended for allowing authors to make use of multimedia features like including video.
- To render data types they did not support natively—namely videos— user agents generally ran external applications and depended on plugins like Adobe Flash.

- Today, more and more Web video is powered by the native and well-standardized `<video>` tag that no longer depends on plugins (albeit some video codec and Digital Rights Management issues remain).
- HTML5 video has finally become a first class Web citizen.

Technologies Overview

WebVTT

- Straight-forward textual format for providing subtitles (translated speech), captions (hard-of-hearing), descriptions, chapters, and metadata for videos and audios.

```
WEBVTT


warning
00:01.000 --> 00:04.000
Never drink liquid nitrogen.


00:05.000 --> 00:09.000
It will perforate your stomach.
```

- We are especially interested in kind metadata tracks meant to be used from a scripting context and never directly displayed to the user.

Technologies Overview

JSON-LD

JavaScript Object Notation for Linked Data, allows for adding meaning to object properties by means of data contexts.

```
WEBVTT
cue1
00:00:00.000 --> 00:00:12.000
{
  "@context": "http://champin.net/2014/
        linkedvtt/demonstrator-context.json",
  "tags": ["wind scene", "opening credits"],
  "contributors": ["http://ex.org/sintel"]
}
```
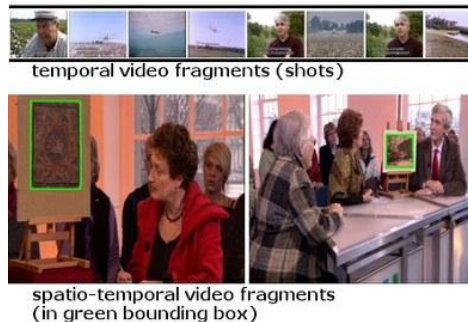
- We embed JSON-LD as payload of metadata text tracks.

Technologies Overview

Media Fragments URI

- Allows for addressing fragments of videos.
  Example: http://www.example.org/video.webm#t=20,30

  Addresses a 10 seconds long media fragment, starting at 20 seconds and ending at 30 seconds.

Source: http://community.mediamixer.eu/images/fragmentcreation/@@images/f85e14d0-ff52-4e47-8c4e-5b6db9001d00.jpeg

Ontology for Media Resources

- Serves to bridge different description methods of media resources and to provide a core set of descriptive properties.

Common Crawl Study

Objectives

- Part of the objectives behind the *Web(VTT) of Data* is *"to create a truly interconnected global network of and between videos containing Linked Data pointers to related content of all sorts, where diverse views are not filtered by the network bubble, but where serendipitously new views can be discovered by taking untrodden Linked Data paths."*

- In order to get there, we have conducted a large-scale study based on the Common Crawl corpus to get a better understanding of the *status quo* of Web video and timed text track deployment.

Common Crawl Study

Video Statistics

- Analyzed the entire 148 terabytes of crawl data using an Elastic Compute Cloud job whose code was made available as open-source.
- Rather than parse each document as HTML, we have tested them for the regular expression `<video[^>]*>(.*?)</video>`.
- We tested exactly 2,247,615,323 Web pages that had returned a successful HTTP response to the Common Crawl bot.
- The job took five hours on 80 c1.xlarge machines and costed $555.
- On these webpages, we detected exactly 2,963,766 `<video>` tags, resulting in a 1.37 gigabyte raw text file that we have made available.
- This means that on average only ≈0.132% of all Web pages contain HTML5 video (we were not interested in proprietary Flash videos).
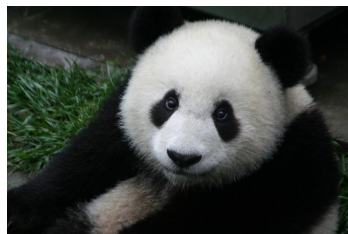
Source http://upload.wikimedia.org/wikipedia/commons/9/90/Giant_Panda_2.JPG

Common Crawl Study

Track Statistics

- From all 2,963,766 `<video>` tags, only 1,456 (≈ 0.049%) had a `<track>` child node, and almost all had only exactly one.
- The overwhelming majority of all `<track>`s are unsurprisingly used for subtitles or captions.
- Almost no chapter usage was detected and neither metadata nor description usage at all.
- Looking at the languages used in the captions and subtitles, these were almost exclusively English and French
- About half of all `<track>` source attributes end with "vtt" or match `/\bvtt\b/gi`, about a quarter end with "srt".

Common Crawl Study

Source Statistics

- The "same" video can be available in several encodings, realized through different `<source>` tags.
- The most common MIME types are `video/mp4` and `video/webm`. It is not uncommon that one video has up to four sources or more.
- Problematic because of identifiers.

Common Crawl Study

Implications for Linked Data

```
<div about="kitten.jpg">
  <img src="kitten.jpg" alt="Cute kitten" />
  <a rel="license" href="http://cc.org/licenses/by-sa/3.0/">
    Creative Commons Attribution Share-Alike 3.0
  </a>
</div>


<div about="kitten.mp4">
  <video>
    <source src="kitten.mp4"/>
    <source src="kitten.webm"/>
  </video>
  <a rel="license" href="http://cc.org/licenses/by-sa/3.0/">
    Creative Commons Attribution Share-Alike 3.0
  </a>
</div>
```

WebVTT conversion to Linked Data

RDF Schema Ontology and WebVTT Conversion

- WebVTT spec defines a semantics for its syntax in terms of how Web browsers should process such tracks via an underlying data model.
- This data model can easily be mapped to RDF-based Linked Data, and thus allowing for many other usage scenarios for this data.
- Propose an RDF Schema ontology conveying WebVTT data model.
- Takes the URL of any WebVTT file, the contents of a raw WebVTT file, or a YouTube URL of any video with closed captions as an input, and applies the conversion from WebVTT to Linked Data on-the-fly.
- **Ontology:**
  http://champin.net/2014/ linkedvtt/onto#
- **LinkedVTT conversion tool code:**
  https://github.com/pchampin/linkedvtt

WebVTT Linked Data interpreter

Interpreter Input

WebVTT IRI	enter IRI
Video IRI	http://ex.org/video
Context IRIs	space-separated list of IRIs
Output format	Ntriple / Nquads ☑ Display in browser
Output type	flat
	Interpret

Examples
- Example 1
- Example 2

Find more
- Get linkedvtt on github

WebVTT conversion to Linked Data

1) Subtitles/Captions: Start with WebVTT

```
WEBVTT


warning
00:01.000 --> 00:04.000
Never drink liquid nitrogen.


00:05.000 --> 00:09.000
It will perforate your stomach.
```

WebVTT conversion to Linked Data

2) Subtitles/Captions: Convert WebVTT cues to RDF nodes

WEBVTT

<#id=warning>

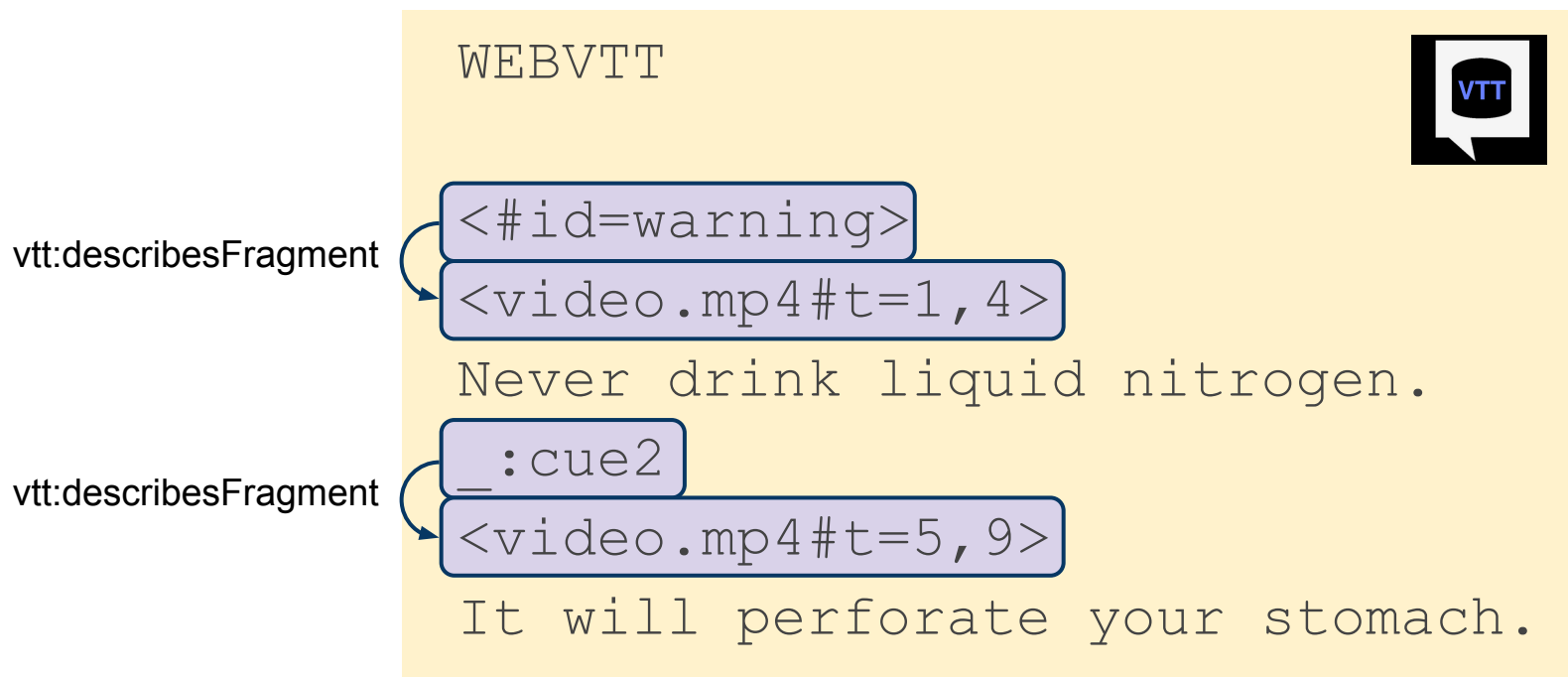00:01.000 --> 00:04.000

Never drink liquid nitrogen.

_:cue2

00:05.000 --> 00:09.000

It will perforate your stomach.

WebVTT conversion to Linked Data

3) Subtitles/Captions: Convert fragments to Media Fragment URIs and link

WEBVTT

vtt:describesFragment

```
<#id=warning>
<video.mp4#t=1,4>
Never drink liquid nitrogen.
```

vtt:describesFragment

```
_:cue2
<video.mp4#t=5,9>
It will perforate your stomach.
```

WebVTT conversion to Linked Data

4) Subtitles/Captions: Convert payload to literal and link
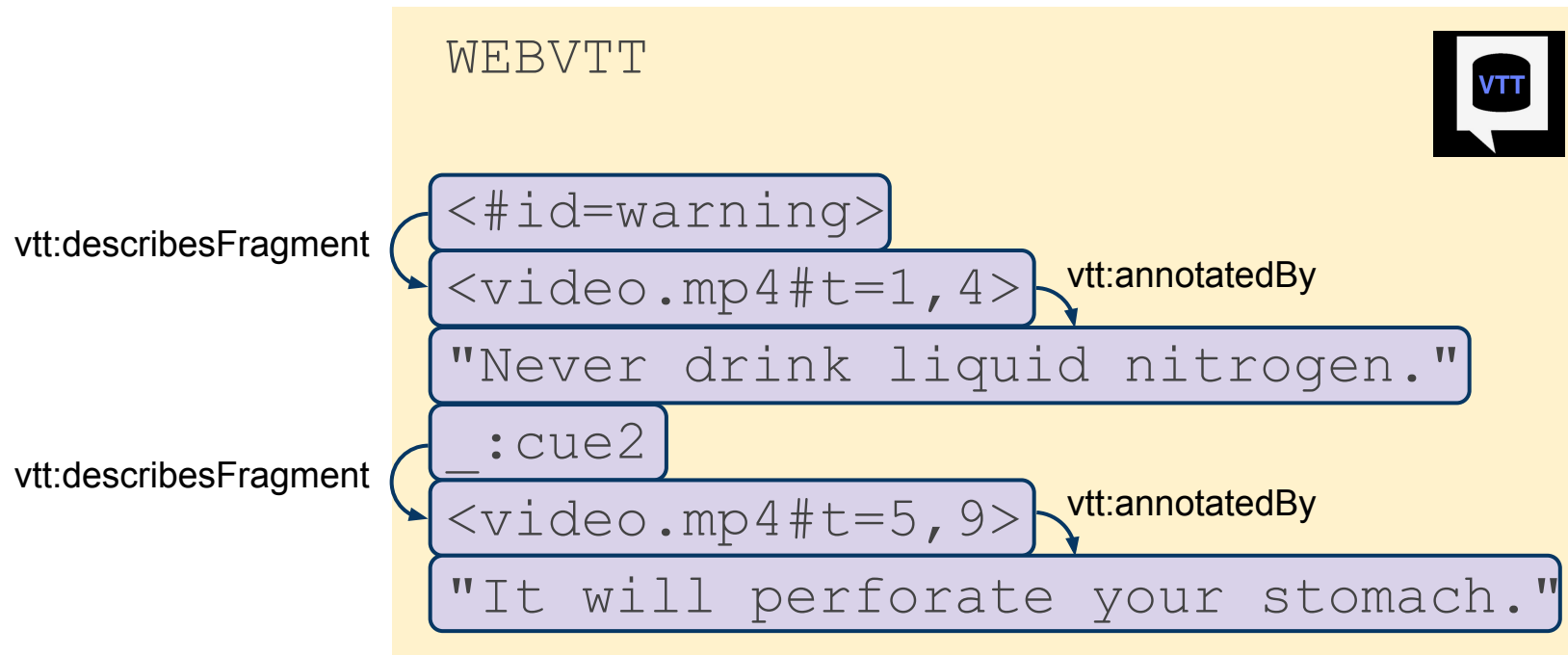
WebVTT conversion to Linked Data

5) Subtitles/Captions: Resulting RDF graph (flat)
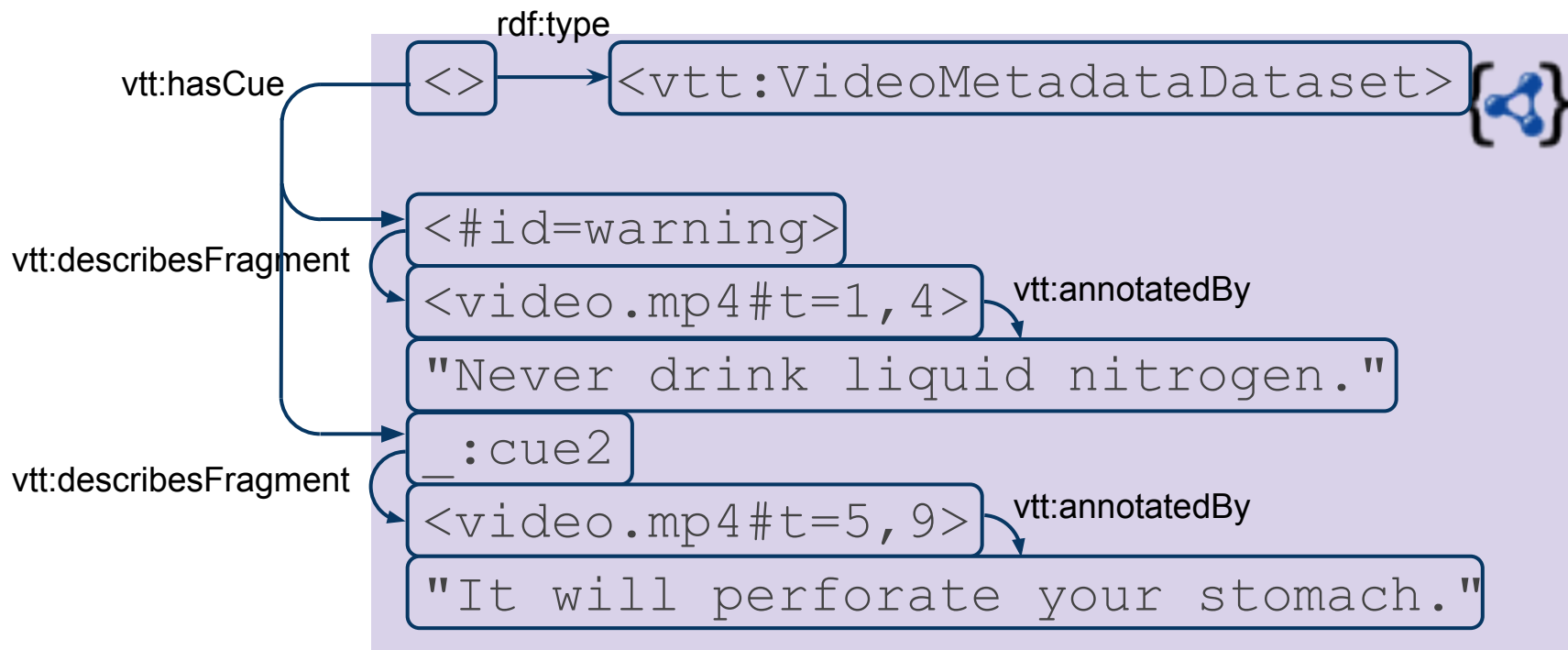
WebVTT conversion to Linked Data

1) Metadata: Special treatment for JSON-LD payloads

```
00:00:00.000 --> 00:00:12.000
{
   "@context": "http://champin.net/2014/
        linkedvtt/demonstrator-context.json",
   "tags": ["wind scene", "opening credits"],
   "contributors": ["http://ex.org/sintel"]
}
```

WebVTT conversion to Linked Data

2) Metadata: Convert JSON-LD keys & values to predicates & objects

```
<video.mp4#t=0,12>
{

    "@context": "http://champin.net/2014/
             linkedvtt/demonstrator-context.json",
    "tags": ["wind scene", "opening credits"],
    "contributors": [<http://ex.org/sintel>]

}
```

WebVTT conversion to Linked Data

3) Metadata: Resulting RDF graph per cue

```
<video.mp4#t=0,12>
```
ctag:label

ctag:label

"wind scene"   "opening credits"

`<http://ex.org/sintel>`

ma:hasContributor
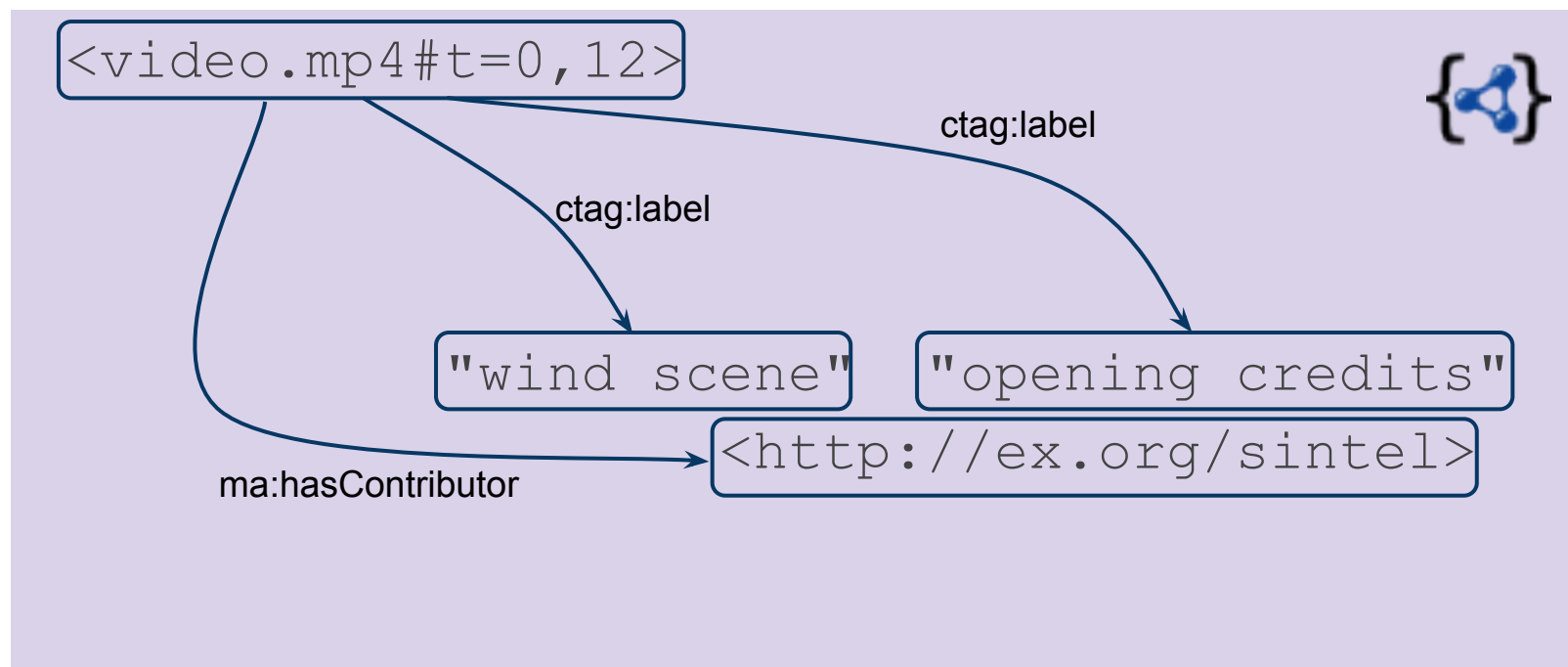
WebVTT Editor

Implementation of the Annotation Model

- We have implemented the annotation model in form of a video editor prototype available at http://tomayac.com/postdoc/webvtt-demonstrator/.
- Semantic Annotation Types:
  - **Plain Text Tags** Annotations of type `tags` allow for adding plain text tags to a media fragment. They are interpreted as Common Tag format ctag:label.
  - **Semantic Tags** Annotations of type `semanticTags` allow for adding semantic tags to a media fragment.
  - **Contributors** The `contributors` annotation type allows for denoting the contributors in a media fragment, like its actors.
  - **Summary** The `summary` annotation type allows for summarizing a media fragment.

Implementation of the Annotation Model

- Presentation-Oriented Annotation Types:
  - **Visual Effect** Annotations of type `visualEffect` allow for applying visual effects in the syntax of Cascading Style Sheets (CSS) to a media fragment.
  - **Audial Effect** The `audialEffect` annotation type allows for applying audial effects to a media fragment.
  - **Playback Rate** The `playbackRate` annotation type allows for specifying the effective playback rate of a media fragment.
  - **HTML Overlay** Via the `htmlOverlay` annotation type, overlays in freeform HTML code can be added to a media fragment.

WebVTT Editor

https://github.com/tomayac/postdoc/tree/master/demos/webvtt-demonstrator

WebVTT Editor

Conclusions

- **How many videos are there?**
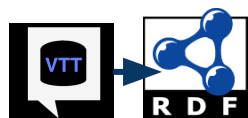
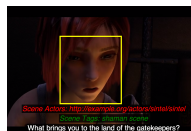 Large-Scale Common Crawl study of the state of Web video.

- **How can we deal with the videos that are there?**

  WebVTT conversion to RDF-based Linked Data.

- **How can we facilitate the generation of more video metadata?**

  Online video annotation format and editor.

- **Do you believe us?**

  Data and code.

Thank you

Resources

- ==These slides: http://bit.ly/weaving-webvtt==.

- **Common Crawl Study and Data:** https://github.com/tomayac/postdoc/tree/master/demos/warczenschwein.

- **LinkedVTT Conversion Tool:** https://github.com/pchampin/linkedvtt.

- **WebVTT Editor:** https://github.com/tomayac/postdoc/tree/master/demos/webvtt-demonstrator.

Backup Slides

- **Common Crawl Details**
- **Nested WebVTT lifting**
- **WebVTT metadata headers**

Common Crawl Study

Technical Details

- Crawl raw data is stored in the Web ARChive format (WARC), an evolution of the previously used Archive File Format (ARC).
- Each crawl run is hierarchically organized in segments directories that contain the WARC files with HTTP requests and responses for fetches.
- While the Common Crawl corpus gets bigger with each crawl run, it obviously does not represent the "whole Web".
- Common Crawl apparently decides on the to-be-included pages based on an implementation of the PageRank algorithm, albeit the inclusion strategy is unknown—despite the foundation's focus on transparency.

Common Crawl

Common Crawl Study

About Common Crawl

- The Common Crawl Foundation is a non-profit organization founded in 2008 by Gil Elbaz.
- Its objective is to democratize access to Web information by producing and maintaining an open repository of Web crawl data that is universally accessible and analyzable.
- All Common Crawl data is stored on Amazon Simple Storage Service (Amazon S3) and accessible to anyone via Amazon Elastic Compute Cloud (Amazon EC2).
- The, at time of writing, latest dataset was collected at the end of 2013, contains approximately 2.3 billion Web pages and is 148 terabyte in size.

Common Crawl

WebVTT conversion to Linked Data

How it works 4/4 (nested)
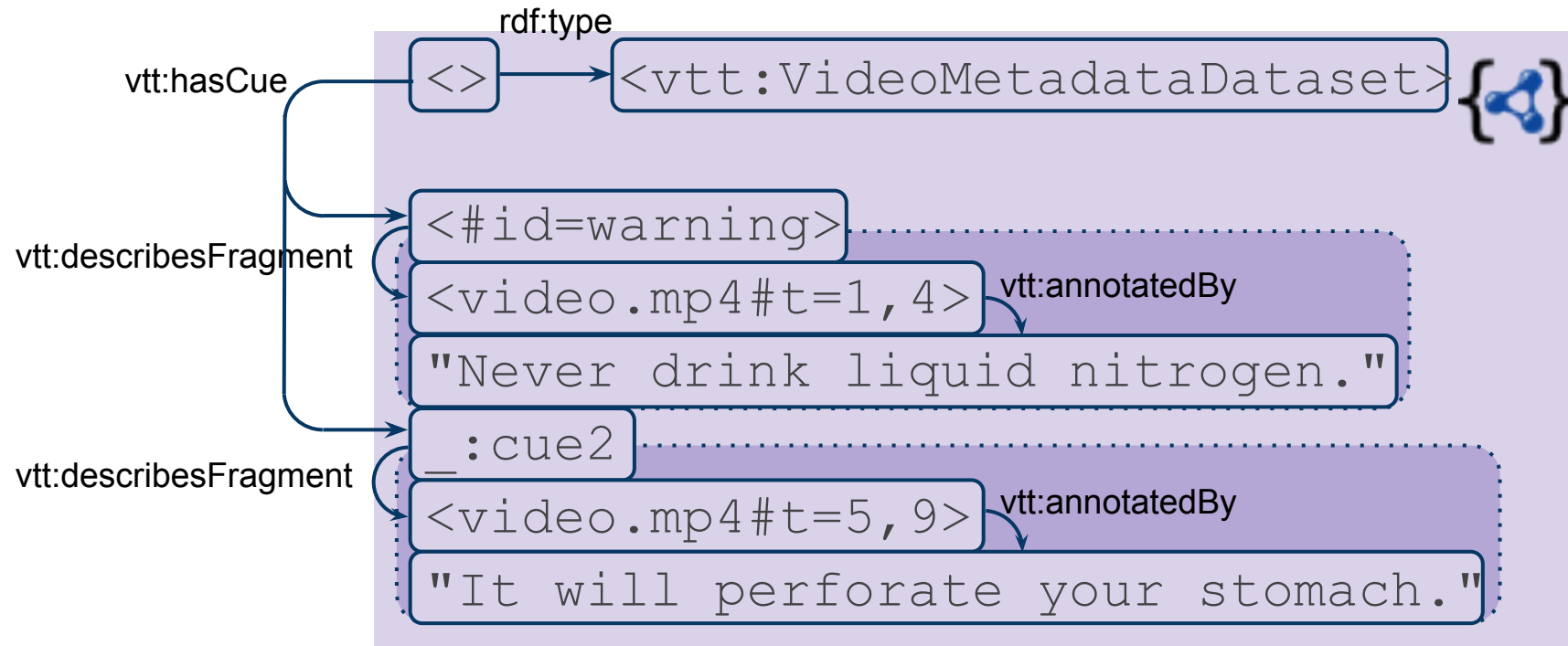
Proposed WebVTT Metadata Headers

- **@base** Sets the base URI used for resolving relative URIs. This applies to any relative URIs that would be found in the JSON-LD descriptions, but also to generate URIs for cues based on their identifiers. It defaults to the URI of the WebVTT file.
- **@context** This key can be used multiple times; each value is the URI of a JSON-LD context that should be used to interpret the JSON payloads in the WebVTT file.
- **@video** Sets the URI for the video for generating media fragment URIs. If not present, the video URI must be provided externally, *e.g.*, via the `<video src>` attribute of the video containing the WebVTT track.