TELL ME WHY! AIN'T NOTHIN' BUT A MISTAKE? DESCRIBING MEDIA ITEM DIFFERENCES WITH MEDIA FRAGMENTS URI AND SPEECH SYNTHESIS

Thomas Steiner

Google Germany GmbH
Hamburg, Germany
tomac@google.com

Raphaël Troncy

EURECOM
Sophia Antipolis, France
raphael.troncy@eurecom.fr

ABSTRACT

We have developed a tile-wise histogram-based media item deduplication algorithm with additional high-level semantic matching criteria that is tailored to photos and videos gathered from multiple social networks. In this paper, we investigate whether the Media Fragments URI addressing scheme together with a natural language generation framework realized through a text–to–speech system provides a feasible and practicable way to visually and audially describe the differences between media items of type photo and/or video, so that human-friendly debugging of the deduplication algorithm is made possible. A short screencast illustrating the approach is available online at http://youtu.be/DWqwEnhqTSc.

***Index Terms*—** Media Fragments URI, Media Fragments, Media Items, Deduplication, Social Networks

1. INTRODUCTION

The music band *Backstreet Boys* (BSB) was formed in 1993 in Orlando, FL and is still the best-selling boys band of all time. In 2013, the band will celebrate their 20th anniversary with a new album and a world tour. Reason enough for us to make them titular saint of this paper with their hit song *I Want It That Way* from the album *Millennium*. While the spike of their career was in the late 90s, even today people still actively share,[1] publish[2] and follow the group on *social networks*.

Social networks are at the heart of our research on event summarization, specifically deduplicating *exact-* and *near-duplicate* media items that are sometimes embedded in status messages referred to as *microposts* on multiple social networks. In the context of our research, we define a *media item* as either a photo (image) or a video that was *publicly* shared or published on at least one social network. Figure 1 shows an example where two users of the social networks Facebook and Google+ independently of each other share a *near-duplicate* media item in form of the music video *Everybody* performed by the *Backstreet Boys*. In order to detect, deduplicate, and cluster such occurrences of *exact-* and *near-duplicate* media items, we have implemented a tile-wise histogram-based algorithm with additional high-level semantic matching criteria that was shown to work effectively and efficiently.

During previous experiments on deduplicating event-related media items, we noticed that human raters wanted to know *why*[3] particular media items are clustered as *exact-* or *near-duplicates*. In this paper, we investigate in how far Media Fragments URI [1] combined with speech synthesis provide a feasible way to tell human annotators why media items are clustered. As we deal with media items of type photo and/or video, we make simultaneous use of two types of media fragments dimensions: the temporal dimension and the spatial dimension.

The remainder of this paper is structured as follows. In Section 2, we report on related work on media fragments, digital storytelling, and natural language generation. In Section 3, we describe our requirements on media fragments identifiers. In Section 4, we detail how the media item deduplication algorithm works and show low-level debugging approaches to check *if* or *if not* media items are clustered. In Section 5, we elaborate on how this low-level debug output gets lifted to natural language stories for human raters to understand *why* or *why not* media items are clustered. We evaluate our approach in Section 6. We conclude and give an outlook on future work in Section 7.

[1] BSB shared on Google+: http://bit.ly/backstreet-gplus
[2] BSB published on Facebook: http://bit.ly/backstreet-fb
[3] Tell me why! Ain't nothin' but a mistake?

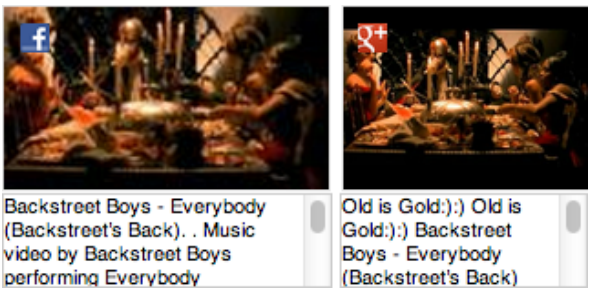

Fig. 1: *Near-duplicate* music video *Everybody* by the *Backstreet Boys* shared independently on Facebook and Google+

2. RELATED WORK

Media Fragments. There are many online video hosting platforms that have some sort of media fragments support. In the following, we present two representative ones. The video hosting platform YouTube[4] allows for deep-linking into videos via a proprietary URL parameter `t`, whose value has to match the regular expression `\d+m\d+s` (for minutes and seconds), as documented in [2]. Dailymotion[5] has similar URL parameters `start` and `end`, whose values have to match the regular expression `\d+` (for seconds). The CSS Backgrounds and Borders Module Level 3 specification [3] defines the `background-size` property that can be used to crop media items visually and thus create the illusion of a spatial media fragment when combined with a wrapping element. Media Fragments URI [1] specifies a syntax for constructing media fragments URIs and explains how to handle them when used over the HTTP protocol [4]. The syntax is based on the specification of particular name-value pairs that can be used in URI query strings and URI fragment identifiers to restrict a media resource to a certain fragment. The temporal and spatial dimensions are currently supported in the basic version of Media Fragments URIs. Combinations of dimensions are also possible.

Digital Storytelling. Pizzi and Cavazza report in [5] on the development of an authoring technology on top of an interactive storytelling system that originated as a debugging[6] tool for a planning system. Alexander and Levine define in [6] the term *Web 2.0 storytelling*, where people create *microcontent*—small chunks of content, with each chunk conveying a primary idea—that gets combined with social media to form coherent stories. We use Media Fragments URIs to help human annotators to understand the results of an algorithm by converting dry software debugging data to digital stories.

Natural Language Generation. Natural language generation is the natural language processing task of generating natural language from a machine representation system. This field is covered in great detail by Reiter and Dale in [7]. They divide the task into three stages: document planning, microplanning, and realization. *Document planning* determines the content and structure of a document. *Microplanning* decides which words, syntactic structures, *etc.* are used to communicate the chosen content and structure. *Realization* maps the abstract representations used by microplanning into text.

3. MEDIA FRAGMENTS REQUIREMENTS

In the context of our research on media item deduplication and clustering, media fragments identifiers need to be capable of expressing the following concepts.

i Given a rectangular media item with the dimensions $width \times height$, express that in turn rectangular tiles of smaller dimensions are part of the original media item.

ii Given detected faces at the granularity level of bounding rectangles, express that these bounding rectangles are within the dimensions of the original media item and that each bounding rectangle contains a face.

iii Requirements *i* and *ii* need to be fulfilled for both types of media items, photos and videos; in case of the latter, video subsegments of any length—including video still frames—need to be supported.

Media Fragments URI [1] as described in the basic version of the specification supports all three requirements. The *temporal dimension* is denoted by the parameter name `t` and specified as an interval with a begin time and an end time. Either one or both parameters may be omitted, with the begin time defaulting to 0 seconds and the end time defaulting to the duration of the source media item. The interval is half-open: the begin time is considered part of the interval, whereas the end time is considered to be the first time point that is not part of the interval. If only a single value is present, it corresponds to the begin time, except for when it is preceded by a comma, which indicates the end time. The temporal dimension is specified as Normal Play Time (NPT, [8]).

The *spatial dimension* selects an area of pixels from media items. In the current version of the specification, only rectangular selections are supported. Rectangles can be specified as pixel coordinates or percentages. Rectangle selection is denoted by the parameter name `xywh`. The value is either `pixel:` or `percent:` (defaulting to `pixel:`) and four comma-separated integers. The integers denote x, y, $width$, and $height$ respectively, with $x = 0$ and $y = 0$ being the top left corner of the media item. If `percent:` is used, x and $width$ are interpreted as a percentage of the width of

```
@base <http://example.org/> .
@prefix ma: <http://www.w3.org/ns/ma-ont> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix db: <http://dbpedia.org/resource/> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix col: <http://purl.org/colors/rgb/> .

<video> a ma:MediaResource .
<video#t=,10&xywh=0,0,30,40> a ma:MediaFragment ;
                             foaf:depicts db:Face .
<video#t=,10&xywh=0,0,10,10> a ma:MediaFragment ;
                             dbo:colour col:f00 .
```

Listing 1: Description of two 10 sec long media fragments: *(i)* a tile of dimensions 30×40 pixels starting at pixel coordinates $(0, 0)$ that contains a face; and *(ii)* a tile of dimensions 10×10 pixels starting at pixel coordinates $(0, 0)$ of red color

[4]YouTube: `http://www.youtube.com/`

[5]Dailymotion: `http://www.dailymotion.com/`

[6]Pizzi and Cavazza use the term *debugging* in the non-IT sense: to check for redundancy, dead-ends, consistency, *etc.* in authored stories

the original media item, while y and $height$ are interpreted as a percentage of the original height. While (at time of writing) the temporal dimension is implemented natively in common Web browsers, this is not the case for the spatial dimension.

The intent of the Ontology for Media Resources [9] by Lee *et al.* is to bridge different description methods of media resources and to provide a core set of descriptive properties. Combined with Media Fragments URI, this allows for making statements about media items and fragments thereof. An example in RDF Turtle syntax [10] is given in Listing 1.

4. MEDIA ITEM DEDUPLICATION ALGORITHM

4.1. Algorithm Description

The deduplication algorithm described in this paper belongs to the family of tile-wise histogram-based clustering algorithms. As an additional semantic feature, the algorithm considers detected faces. It is capable of deduplicating media items of type video and/or photo. In the case of video, frames at camera shot boundaries are used. A camera shot in video production and filmmaking is a series of frames that runs for an uninterrupted period of time. For media items to be clustered, the following clustering conditions have to be fulfilled.

Cond. 1 Out of m tiles of a media item with n tiles ($m \leq n$), the average color of at most *tiles_threshold* tiles may differ not more than *similarity_threshold* from their counterpart tiles.

Cond. 2 The numbers f_1 and f_2 of detected faces in both media items have to be the same. We note that the algorithm does not *recognize* faces, but only *detects* them.

Cond. 3 If the average colors of a tile and its counterpart tile are within the black–and–white tolerance *bw_tolerance*, these tiles are not considered and *tiles_threshold* is decreased accordingly (we talk about *effective_tiles_threshold* in Section 4.2).

The black–and–white tolerance *bw_tolerance* avoids media items to be clustered when the particular tiles are too dark (*e.g.*, for the video borders in Figure 1) or too bright (*e.g.*, for screenshots of Web pages or applications, which frequently appear on social networks). In order to illustrate the way the algorithm deduplicates media items, Figure 2 shows a debug view of the algorithm for the two clustered media items related to the previous example around the *Backstreet Boys* music video. Independent of the actual media items' aspect ratios, the tile-wise comparison always happens based on a potentially squeezed square aspect ratio version.

4.2. Debugging the Algorithm

In this section, we consider the following three debug scenarios that occurred most frequently during our previous experiments with human raters. They correspond to situations

(a) From Facebook user (b) From Google+ user

Fig. 2: Debug view of the media item deduplication algorithm: since no faces are detected in Figure 1, the clustering is based on tile similarity; black tiles are not considered due to the chosen black–and–white tolerance)

where, given a set of deduplicated and clustered media items, a human annotator wanted to understand the specific details leading to the decisions taken by the algorithm that they were unsure about or had decided on differently.

Clustering Consent. Two or more media items are clustered by the algorithm and the human rater agrees. The human rater wants to understand why they were clustered.

Clustering Dissent. Two or more media items are clustered by the algorithm, but the human rater thinks that they should not have been clustered. The human rater wants to understand why they were incorrectly clustered.

Non-Clustering Dissent. Two or more media items are not clustered by the algorithm, but the human rater thinks that they should have been clustered. The human rater wants to understand why they were not clustered.

In order to provide answers to these human raters' information needs, different levels of the algorithm's internals have to be debugged. Is the *tiles_threshold* (*i.e.*, the number of tiles that may differ) too high or too low? Complementary to this, is the *similarity_threshold* (*i.e.*, the maximum amount two tiles may differ) too high or too low (**Cond. 1**)? Are the number of detected faces f_1 and f_2 the same? Are all faces correctly detected, or should the face matching condition be temporarily disregarded, *e.g.*, with too tiny media items, where faces fail to be detected (**Cond. 2**)? If the media items to be compared have very dark and/or very bright parts, is the *bw_tolerance* too high or too low (**Cond. 3**)?

4.3. Low-Level Debug Output

As a consequence of the previous subsection, the low-level debug output must include the currently selected *tiles_threshold* and *similarity_threshold* and how many tiles with the present algorithm settings currently fulfill **Cond. 1**. In addition to that, the debug output has to contain the number of detected faces f_1 and f_2 in each media item, *i.e.*, whether **Cond. 2** is fulfilled, as well as the number of not considered

tiles (according to *bw_tolerance*), which implies fulfillment of **Cond. 3** and potentially impacts **Cond. 1** in form of the *effective_tiles_threshold*. For instance, the low-level debug output for the media items from the running example of the *Backstreet Boys* media items for the music video *Everybody* reads as follows.

```
- Similarity threshold: 15 (Cond. 1)
- Tiles threshold: 67 (Cond. 1)
- Similar tiles: 52 (Cond. 1)
- Faces left: 0. Faces right: 0 (Cond. 2)
- BW tolerance: 1 (Cond. 3)
- Not considered tiles: 22 (Cond. 3)
- Effective tiles threshold: 45 (Cond. 3)
```

While this low-level debug output is sufficient to respond to the polar question (yes–no question) whether media items are clustered at all or not, it does not help with the non-polar *why* question (the linguistic term for this type of questions is *wh–question*). In the following section, we show how this low-level debug output can be lifted.

5. FROM DEBUG OUTPUT TO STORY

In order for human raters to get answers to the question on *why* media items are clustered, we need to lift the low-level debug output to a high-level natural language story for the previously defined debug scenarios **Clustering Consent**, **Clustering Dissent**, and **Non-Clustering Dissent**. This results in a natural language generation task, whose three stages according to Reiter's and Dale's architecture [7] will be detailed in the following subsections.

5.1. Generating Natural Language

5.1.1. Document Planning

In our context, the document is a set of low-level debug data as illustrated in Section 4.3. The natural language generation task is thus manageable. We need to convey the currently selected *tiles_threshold* and *similarity_threshold*, the number of detected faces f_1 and f_2 in each media item, and the number of tiles not considered given the *bw_tolerance* parameter.

5.1.2. Microplanning

The microplanning task is driven by the debug scenarios described previously. Initially, we need to decide on a matching condition aspect of the algorithm that will be first highlighted. Typically, this will be the overall tiles statistics. Afterwards, we need to elaborate on secondary matching conditions such as detected faces and black–and–white tolerance. The grammatical number (plural or singular) needs to be taken into account when statements about tile(s) or face(s) are planned. Some values, *e.g.*, the percentage of matching tiles, are calculated. The microplanner needs to decide when exactness (*e.g.*, *"99% of all tiles"*) and when approximation of calculated values (*e.g.*, *"roughly 50%"*) better suits the human evaluators' information needs. Neutral non-judgmental statements (*e.g.*, *"45 tiles"*) and biased judgmental statements (*e.g.*, *"not a single one [tile]"*) need to be carefully balanced. Finally, in the interest of a more naturally sounding phrase composition, the microplanner needs to be aware of contrasting juxtaposition (*e.g.*, *"Both the left and the right media item contain one detected face." vs. "The left media item contains no detected faces, while the right media item contains one detected face."*).

5.1.3. Realization

We show examples of sentences that are actually generated for the three different debug scenarios (Quotes 1–3). For the sake of completeness, we provide one additional example (Quote 4) for the debug scenario **Non-Clustering Consent**. The running example of the *Backstreet Boys* media items for the music video *Everybody* is represented by Quote 1.

Clustering Consent (Quote 1). *"The two media items are near-duplicates. Out of overall 100 tiles, 52 from the minimum required 45 tiles were similar enough to be clustered. This corresponds to 52 percent of all tiles. However, 22 tiles were not considered, as they are either too bright or too dark, which is a common source of clustering issues. Neither the left, nor the right media item contain detected faces."*

Clustering Dissent (Quote 2). *"The two media items are near-duplicates. Out of overall 100 tiles, 41 from the minimum required 41 tiles were similar enough to be clustered. This corresponds to 41 percent of all tiles. However, 26 tiles were not considered, as they are either too bright or too dark, which is a common source of clustering issues. Neither the left, nor the right media item contain detected faces."*

Non-Clustering Dissent (Quote 3). *"The two media items are different. Out of overall 100 tiles, only 8 from the minimum required 67 tiles were similar enough to be clustered. This corresponds to 8 percent of all tiles. The left media item contains 2 detected faces, while the right media item contains 1 detected face."*

(Non-Clustering Consent) (Quote 4). *"The two media items are different. Out of overall 100 tiles, not a single one was similar enough to be clustered. Neither the left, nor the right media item contain detected faces."*

5.2. Technical Implementation

5.2.1. Text–to–Speech

The generated texts are converted to speech using a text–to–speech system. We use the eSpeak [11] speech synthesizer that was originally developed by Jonathan Duddington

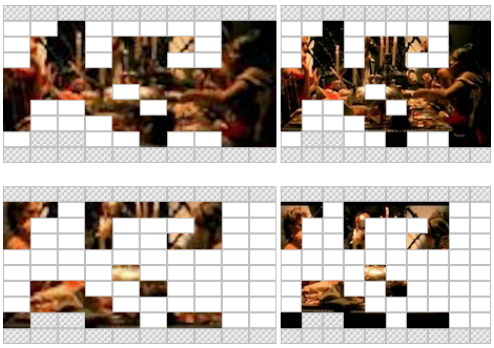

Fig. 3: Similar (upper row) and different (lower row) corresponding tile pairs for the media items from a Facebook (left column) and a Google+ user (right column); checkered tiles are not considered due to the black–and–white tolerance

in a JavaScript port made available by Alon Zakai.[7] This speech synthesizer uses the formant synthesis method, which allows many languages to be provided in a small size. Rather than using human speech samples at runtime, the synthesized speech output is created using additive synthesis and an acoustic model, where parameters such as fundamental frequency, voicing, and noise levels are varied over time to create a waveform of artificial speech. The speech is clear and can be used at high speeds. However, it is not as natural or smooth as larger synthesizers that are based on speech recordings.

5.2.2. *Visual Media Fragments Highlighting*

We treat and address each tile of a media item as a spatial media fragment. Figure 3 shows a grid of similar, different, and not considered tiles from the *Backstreet Boys* media items for the *Everybody* music video. While the speech synthesizer reads the generated text, the corresponding tiles (*e.g.*, the matching tiles or the because of the black–and–white tolerance not considered tiles) are visually highlighted to support the human evaluators' understanding, as can be seen in Figure 4 and in the screencast. Spatial Media Fragments URIs are currently not implemented in any common Web browser [12]. In order to nonetheless support spatial media fragments, we use a so-called JavaScript polyfill for Media Fragments URI made available by Thomas Steiner.[8] In Web development, a polyfill is downloadable code that provides facilities by emulating potential future features or APIs that are not built-in to a Web browser [13]. Steiner's polyfill—in contrast to an additional earlier spatial Media Fragments URI polyfill implementation [12] by Fabrice Weinberg—supports more browsers and both image *and* video.

[7] Speak.js: `https://github.com/kripken/speak.js`
[8] xywh.js: `https://github.com/tomayac/xywh.js`

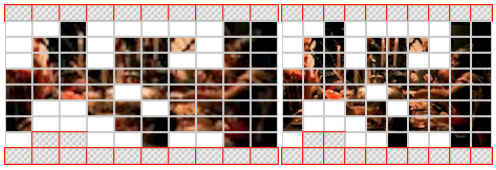

Fig. 4: The highlighted checkered tiles are not considered, as the text–to–speech system explains: *"However, 22 tiles were not considered, as they are either too bright or too dark, which is a common source of clustering issues."*

6. EVALUATION

For the evaluation of natural language generating systems, there are three basic techniques. First, the *task-based* or *extrinsic* evaluation, where the generated text is given to a person who evaluates how well it helps with performing a given task [14]. Second, there are *automatic metrics* such as BLEU [15], where the generated text is compared to texts written by people based on the same input data. Finally, there are *human ratings*, where the generated text is given to a person who is asked to rate the quality and usefulness of the text.

For our evaluation, we chose the third approach of human ratings, as we do not evaluate the natural language generating system in isolation, but in *combination with a visual representation* that makes use of spatial Media Fragments URIs (Figure 3 and Figure 4). Evaluating subjective data like the quality and usefulness of an auto-generated textual, visual, and audial explanation of the results of a deduplication algorithm is a challenging task. For different users, there may be different emphases. A common subjective evaluation technique is the Mean Opinion Score (MOS, [16]), used for decades in telephony networks to obtain the human user's view of the quality of a network. Recently, MOS has also found wider usage in the multimedia community. Therefore, we conducted a set of standard subjective tests, where users rate the perceived quality of test samples with scores from 1 (worst) to 5 (best). The actual MOS is then the arithmetic mean of all individual scores. In the context of this research, we have conducted MOS test sessions with five external human raters. We generated artificially modified deduplicated media item sets around media items about the *Backstreet Boys* that were shared on social networks during the time of writing. These media item sets were curated by yet another independent two external persons, assisted by a previously developed software system that implements the deduplication algorithm described in this paper. We asked the two persons to provoke dissent and consent clustering situations for the five human raters, *i.e.*, obviously correct clustering (**Clustering Consent**), obviously incorrect clustering (**Clustering Dissent**), and obviously incorrect non-clustering (**Non-Clustering Dissent**). We then asked the five human raters to have the system automatically explain the algorithm results to them as described in Section 5. The raters gave MOS scores ranging from 2 to 5, with the overall

average values as follows: **Clustering Consent**: 4.3, **Clustering Dissent**: 3.3, and **Non-Clustering Dissent**: 4.1. The human raters appreciated the parallel explanation approach, where the visual and the audial parts synchronously described what the algorithm was doing. They uttered that the not considered tiles (due to the black–and–white tolerance) as well as erroneously not detected faces were sources of error in the algorithm that they easily understood thanks to the human language description. They sometimes wished for more diversification in the generated texts. Without exception, they liked the system and encouraged future development.

7. CONCLUSIONS AND FUTURE WORK

Concluding, we have successfully demonstrated the feasibility of making the task of debugging a complex algorithm more human-friendly by means of a combined visual and audial approach. We have used Media Fragments URI together with a natural language generation framework realized through a speech synthesizer to visually and audially describe media item differences. Our contribution also includes a polyfill implementation of spatial Media Fragments URIs.

Future work will focus on generalizing the approach. Many algorithms need expert evaluators for fine-tuning their settings, simply because the debug output is unaccessible to untrained raters. With this work, we have contributed a promising proof of concept, which, through the involvement of non-domain expert raters, has helped us greatly improve our deduplication algorithm's default parameters.

8. ACKNOWLEDGMENTS

This research was partially supported by the European Union's 7th Framework Programme via the projects I-SEARCH (GA 248296) and LinkedTV (GA 287911).

9. REFERENCES

[1] R. Troncy, E. Mannens, S. Pfeiffer, D. Van Deursen, M. Hausenblas, P. Jägenstedt, J. Jansen, Y. Lafon, C. Parker, and T. Steiner, "Media Fragments URI 1.0 (basic)," Recommendation, W3C, 2012, http://www.w3.org/TR/media-frags/.

[2] The YouTube Team, "Link To The Best Parts In Your Videos," 2008, http://youtube-global.blogspot.com/-2008/10/link-to-best-parts-in-your-videos.html.

[3] B. Bos, E. J. Etemad, and B. Kemper, "CSS Backgrounds and Borders Module Level 3," Candidate Recommendation, W3C, 2012, http://www.w3.org/TR/css3-background/.

[4] R. T. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, IETF, 1999, http://www.ietf.org/rfc/rfc2616.txt.

[5] D. Pizzi and M. Cavazza, "From Debugging to Authoring: Adapting Productivity Tools to Narrative Content Description," in *1st Joint International Conference on Interactive Digital Storytelling (ICIDS'08)*, Erfurt, Germany, 2008, pp. 285–296.

[6] B. Alexander and A. Levine, "Web 2.0 Storytelling: Emergence of a New Genre," EDUCAUSE *Review*, vol. 43, no. 6, pp. 40–56, 2008.

[7] E. Reiter and R. Dale, *Building Natural Language Generation Systems*, Studies in Natural Language Processing. Cambridge University Press, 2000.

[8] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol RTSP," RFC 2326, IETF, 1998, http://www.ietf.org/rfc/rfc2326.txt.

[9] W. Lee, W. Bailer, T. Bürger, P.-A. Champin, J.-P. Evain, V. Malaisé, T. Michel, F. Sasaki, J. Söderberg, F. Stegmaier, and J. Strassner, "Ontology for Media Resources 1.0," Recommendation, W3C, 2012, http://www.w3.org/TR/mediaont-10/.

[10] E. Prud'hommeaux, G. Carothers, D. Beckett, and T. Berners-Lee, "Turtle – Terse RDF Triple Language," Candidate Recommendation, W3C, 2013, http://www.w3.org/TR/turtle/.

[11] J. Duddington, "eSpeak Text to Speech," 2012, http://espeak.sourceforge.net/.

[12] F. Weinberg, "Media Fragments URI–Spatial Dimension," 2013, http://css-tricks.com/-media-fragments-uri-spatial-dimension.

[13] R. Sharp, "What is a Polyfill?," 2010, http://remysharp.com/2010/10/08/what-is-a-polyfill/.

[14] F. Portet, E. Reiter, A. Gatt, J. Hunter, S. Sripada, Y. Freer, and C. Sykes, "Automatic Generation of Textual Summaries from Neonatal Intensive Care Data," *Artificial Intelligence*, vol. 173, no. 7–8, pp. 789–816, 2009.

[15] K. Papineni, S. Roukos, T. Ward, and W.J. Zhu, "BLEU: A Method for Automatic Evaluation of Machine Translation," in *40th Annual Meeting on Association for Computational Linguistics (ACL'02)*, Philadelphia, Pennsylvania, 2002, pp. 311–318.

[16] International Telecommunication Union, Telecommunication Standardization Sector, "ITU-T Recommendation P.800: Methods for Subjective Determination of Transmission Quality," 1998, http://bit.ly/Mean-Opinion-Score.