Multimed Tools Appl
DOI 10.1007/s11042-016-4275-4

Videolization: knowledge graph based automated video generation from web content

Murat Kalender[1,2] · M. Tolga Eren[2] · Zonghuan Wu[3] · Ozgun Cirakman[2] · Sezer Kutluk[2] · Gunay Gultekin[2] · Emin Erkan Korkmaz[1]

Received: 30 March 2016 / Revised: 20 October 2016 / Accepted: 15 December 2016

Abstract Web content nowadays can also be accessed through new generation of Internet connected TVs. However, these pro ducts failed to change users' behavior when consuming online content. Users still prefer personal computers to access Web content. Certainly, most of the online content is still designed to be accessed by personal computers or mobile devices. In order to overcome the usability problem of Web content consumption on TVs, this paper presents a knowledge graph based video generation system that automatically converts textual Web content into videos using semantic Web and computer graphics based technologies. As a use case, Wikipedia articles are automatically converted into videos. The effectiveness of the proposed system is validated empirically via opinion surveys. Fifty percent of survey users indicated that they found generated videos enjoyable and 42 % of them indicated that they would like to use our system to consume Web content on their TVs.

Keywords Semantic web · Computer graphics · Text-to-video · Entity linking · Knowledge graph · DBpedia

✉ Murat Kalender
murat.kalender@huawei.com

Zonghuan Wu
zonghuanwu@huawei.com

Emin Erkan Korkmaz
ekorkmaz@cse.yeditepe.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Yeditepe University, Istanbul, Turkey

[2] Huawei Technologies, Huawei Turkey R&D Center, Istanbul, Turkey

[3] Huawei Technologies, Software Lab, Santa Clara, CA, USA

Published online: 29 December 2016

1 Introduction

Web content nowadays can also be accessed through the new generation of Internet connected TVs, which include a web browser and a virtual keyboard so that users can browse and search online content using their TV sets. Consuming the Web on giant screens is a promising utility of these TV sets.

Although the average American consumer spends five hours in front of the TV every day,[1] it has been shown that only ten percent of Internet connected TV owners have ever used their built-in browsers.[2] We believe there are two reasons behind this finding: i) Built-in browsers are inferior to their desktop or mobile counterparts. ii) Interaction methods are not as intuitive as users are accustomed to. Because of these issues, users prefer PCs or mobile devices for Web consumption. Furthermore, we argue that this is inherently related to the design philosophy of the Web content.

Many of the most popular websites on the Web such as Facebook and Reddit feature an infinite scrolling style browsing. Additionally more static websites can come in varying sizes. Web content may be very tall or wide. Consequently, it is an undesirable experience to use a TV remote as an input device, causing the user to repeatedly scroll down or across a Web page in order to read it. This study addresses the problem of how to effectively and intuitively consume Web content on Internet connected TVs. We can specify this problem further in selected domains as well; how to turn news websites into news shows, how to turn e-commerce sites into shopping TV broadcasts and how to turn bulletin board systems (BBS) into talk shows or debate-style broadcasts. We believe that with a change in design philosophy, it is possible to deconstruct the Web content and then reconstruct it in a TV friendly format. This would allow the users to surf the Web in a completely different yet satisfying way.

In order to investigate this problem, it is imperative to understand the Web content first. The Web content can be textual, visual or audial. This content is encountered as a part of the user experience on the Web. It is possible to have text, images, sounds, videos and animations on a Web page, however a great portion of the Web content is predominantly composed of text. Thus a workflow to convert textual information into multimedia format is needed. Motivated by sayings like, “A picture is worth a thousand words”, this study introduces *Videolization*, a knowledge graph based video generation system that automatically converts textual Web content into videos using semantic Web and computer graphics based techniques.

The *Videolization* system visualizes text content by utilizing visual representations of extracted entities. One of the main steps of this workflow is entity linking which is the generation of assignments from knowledge graph entities to documents. The visualized elements are displayed and animated using a director-actor analogy. Additionally, the audio component of the video can be produced with a text-to-speech (TTS) system. Combining this audio with visual representations of the most significant entities in the given sentence produces a video segment. Stitching segments to each other with appropriate transitions yields the final video. Figure 1 shows an example of a video generated automatically by analyzing USA Wikipedia article.

[1] http://www.nydailynews.com/life-style/average-american-watches-5-hours-tv-day-article-1.1711954

[2] https://www.npdgroupblog.com/internet-connected-tvs-are-used-to-watch-tv-and-thats-about-all/

Fig. 1 Thumbnails from a video generated automatically by analyzing USA Wikipedia article. The proposed system *videolizes* each sentence as a scene in one of the following formats: entity graph (*Scenes 1 and 4*), entity video (*Scenes 3 and 6*), entity image (*Scene 7*) and text (*Scene 5*)

Although the first and main motivation of the Videolization system is to provide videos for TVs, it can be used on any device with a broadband internet connection and video playing capability such as desktop computers and mobile phones. The passive, filtered and compact way of presenting information provided by the system would be useful on any device.

Through our experiments and evaluations, we show that TV-friendly videos can be generated by semantic analysis of Web content in order to understand its context and to decide how to visualize it in a video. We detail and demonstrate our system through a use case based on Wikipedia[3] articles. Furthermore, we evaluate the user friendliness and effectiveness of the system with a qualitative user study. The survey results show that 42 % of the users prefer using *Videolization* to consume Web content on their TVs and mobile devices. The main contributions of this paper are summarized as follows.

- A novel visualization method is presented to convert Wikipedia articles and other web sources such as news web sites and social media into videos using a director-actor analogy. A Videolization video description language (VVDL) is proposed to describe videos in XML format and provide interdependence between video content and its visualization.
- A simplified version of DBpedia is utilized as a knowledge graph, featuring the most important 100 properties.

2 Related work

This section provides background information about existing studies on entity linking, text visualization and automatic video generation.

[3] http://en.wikipedia.org/

2.1 Entity linking

Generating assignments of knowledge base entities to documents is known as an entity linking process. Several entity linking studies were proposed mostly for the English language, such as TagMe [6], Illinois Wikifier [19] and Wikipedia-miner [16]. These entity linking studies propose a variety of techniques ranging from hand-coded rules to statistical machine learning techniques. The systems usually utilize Natural Language Processing (NLP) methods and knowledge bases (mostly Wikipedia) to detect spots, and perform disambiguation and ranking.

Specifically, AIDA [10] searches for entities using the Stanford NER Tagger and adopts the YAGO2 knowledge base [9] as a catalog of entities, including their semantic distance. Disambiguation comes in three variants: PriorOnly (each mention is bound to its most commonly linked entity in the knowledge base), LocalDisambiguation (each mention is disambiguated independently from others, according to a set of features which describe the mention and the entities), and CocktailParty (YAGO2) which is used to perform a collective disambiguation which aims at maximizing the coherence among the selected annotations, via an iterative graph based approach. AIDA has been designed to deal with English documents of arbitrary length, it offers a publicly available API.

CMNS [13] generates a ranked list of candidate entities for all n-grams in the input text. The list is created through lexical matching and language modeling. The disambiguation is done using a method based on supervised machine learning that takes as its input a set of (short) texts and for each one, a set of human annotations. CMNS is designed to deal with very short texts only (mainly tweets).

CSAW [11] searches the input text for entities extracted from Wikipedia anchors and titles. It uses two scores for each annotation, one local and one global. The local score involves 12 features built upon the terms around the mention and the candidate entities. The global score involves all the other annotations detected for the input text and averages the relatedness among them. This was the first system to formulate the disambiguation process as a quadratic programming optimization problem aiming for a global coherence among all mentions. CSAW has been designed to deal with English documents of arbitrary length, but is quite slow because of the quadratic programming approach.

Illinois Wikifier [19] searches the input text for mentions extracted from Wikipedia anchors and titles, using the Illinois NER system. Disambiguation is formulated as an optimization problem which aims at global coherence among all mentions. It uses a novel relatedness measure between Wikipedia pages based on Normalized Google Distance (NGD) and point-wise mutual information.

DBpedia Spotlight [14] searches the input text for mentions extracted from Wikipedia anchors, titles and redirects. It then associates a set of candidate entities to each mention using the DBpedia Lexicalization data set. Given a spotted mention and a set of candidate entities, the context of both the mention and the candidate entities are cast to a Vector-Space Model (using a BOW approach) and the candidate entity whose context has the highest cosine similarity with the mention context is chosen. Note that no semantic coherence is estimated among the chosen entities.

TagMe [6] searches the input text for mentions defined by the set of Wikipedia page titles, anchors and redirects. Each mention is associated with a set of candidate entities. Disambiguation exploits the structure of the Wikipedia graph, according to the relatedness measure introduced in [27] which takes into account the number of common incoming links

between two pages. TagMe's disambiguation is enriched with a voting scheme in which all possible bindings between mentions and entities are scored and then they express a vote for each binding. A proper mix of heuristics is eventually adopted to select the best annotation for each mention. TagMe is designed to deal with short texts, it offers a publicly available API.

Wikipedia Miner [16] is one of the first approaches proposed to solve the entity-annotation problem. This system is based on a machine learning approach that is trained with links and contents taken from Wikipedia pages. Three features are then used to train a classifier that selects valid annotations discarding irrelevant ones: i) the prior probability that a mention refers to a specific entity, ii) the relatedness to the context from which the entity is extracted, given by the non-ambiguous spotted mentions, and iii) the context quality which takes into account the number of terms involved, the extent that they relate to each other, and how frequency of use as Wikipedia links.

Cornolti et al. [4] propose a benchmarking framework to fairly and fully compare publicly available entity annotation systems. Their experimental results show that TagMe outperforms the other annotators in terms of $F1$ score and run-time duration. TagMe has a Web service to identify and link meaningful short-phrases in an unstructured text to Wikipedia articles.

2.2 Text visualization

Web pages consist of textual and audio-visual content which designate the user experience on websites. Although a Web page may include text, images, sounds, videos and animations, the majority of Web content is predominantly composed of text. Image generation and image retrieval based approaches in the literature mitigate this situation by converting general text to visual representations.

Image generation based approaches create animations using computer graphics technologies for a given text content. Several image generation based studies were proposed mostly for English; namely U-Pav [23], Web2Animation [20], e-Hon [22] and WordsEye [5].

U-Pav [23] is proposed by Tanaka, where the system reads out the entire text in the Web content together with an image animation. The proposed system shows the title and Web content through a ticker and the web page images are animated at the same time. The tickers, animations and TTS output are synchronized.

Web2Animation [20] is a system that analyzes the semantics of recipes on the Web and generates 3D animations for them. The recipe instructions are mapped to a set of animation clips using a semantic analyzer and the clips are displayed.

E-Hon [22] is a system that converts Web content into a storybook with dialogues and animation. It is especially designed to help children to understand Web content through animation. The e-Hon system utilizes semantic tags that are associated with the text on the Web. To transform the Web content into dialogues, the system generates a list of subjects, objects, predicates, and modifiers from the text and connects them in a colloquial style. A subject is treated as a character and a predicate is treated as the action. An object is also treated as a character, and an associated predicate is treated as a passive action. Many characters and actions have been recorded in their database.

WordsEye [5] produces highly realistic 3D scenes by utilizing thousands of predefined 3D polyhedral object models with detailed manual tags and deep semantic representations

of the text. Consequently, WordsEye works best with certain descriptive sentences, e.g., "The cat is 5 feet behind John. The cat is 10 feet tall".

In [8] a semantic model for learning vector representations of words related to images is proposed. After training, the model can be used to generate images for a given concept.

Image retrieval based approaches retrieve images using image search techniques for a given text content. Such studies generally [2, 15, 29] apply NLP techniques to extract important words or phrases, and computer vision techniques are used to find the corresponding images from image databases. Finally, they use computer graphics techniques to render the retrieved images in a picture. With recent advancements in computer vision and NLP, there has been significant work in relating images to their sentence-based semantic descriptions [30]. Socher et al. [21] proposes a model to map sentences and images into a common embedding space to retrieve one from the other. They introduce the dependency tree recursive neural network (DT-RNN) model which uses dependency trees to embed sentences into a vector space. Then these embeddings can be used to retrieve the images described by those sentences. Recently, Tao et al. propose McMil [24] and DR-KISS [25] for labeling images and videos. These labels can also be used to retrieve images and videos for a given text content.

2.3 Automatic video generation

Automatic video generation services have been on the commercial scene for a while, as part of Web 2.0. Commercial services such as Stupeflix,[4] SoMedia,[5] Winston,[6] Wibbitz,[7] Animoto,[8] Magisto,[9] Sezion,[10] Videolicious,[11] and WeVideo[12] provide the users with tools for automatic video generation from the users' images, texts, videos, and musical data. These tools include a storyboard editor, various visual effects, and video editing utilities.

Stupeflix generates videos from the photo albums of users in social media accounts semi automatically. It has an OpenGL based video generation technology. SoMedia is a video production platform that allows users to select animation styles, customize colors and music, choose scenes, and upload content. Using this data provided by the user the system produces videos. Winston, which is a mobile application, produces audiovisual newscasts from social media feeds and news. Wibbitz converts textual content from the Web into videos using artificial intelligence and NLP technologies. Animoto is a web tool for creating slideshows, marketing and mobile videos. Magisto creates videos from the provided videos and images with the selected music, themes and effects. Sezion is a platform for marketers and agencies to create targeted marketing videos for each customer. Videolicious is for video journalism,

[4] https://studio.stupeflix.com

[5] https://www.somedia.net

[6] http://getwinston.com/project/apptour

[7] http://www.wibbitz.com

[8] https://animoto.com

[9] https://www.magisto.com

[10] https://sezion.com

[11] https://videolicious.com

[12] https://www.wevideo.com

providing tools for video editing and adding voice-overs, logos and watermarks. WeVideo is a cloud-based collaborative video creation platform that provides solutions for personal, business, and educational use.

The system presented in this study differs from other automatic video generation services in terms of the intelligent methods utilized. *Videolization* introduces algorithms for intelligently collecting and processing data, and presenting it as a video.

Additionally, some effort has been made to incorporate intelligence into the automatic video production process. For instance, automatic music video generation methods were proposed [3, 18, 28]. These studies incorporate NLP methods in order to process the lyrics, images and video analysis in order to find related images and videos. They also incorporate audio processing techniques for bar and tempo detection or genre classification. However, these studies are limited to a specific domain and they cannot be considered as general systems that can convert arbitrary content into a video.

3 Videolization

A notable difference between the contents of a Web page and a TV program is that the former is a document-based presentation whereas the latter is a time-based continuous information media. This situation creates a difference for the information accessing methods that can be utilized. Conventional "Web browsing" is an active process of accessing information. On the other hand, conventional "TV watching" is a relatively passive way of accessing information. In order to consume Web content effectively on a TV, a media conversion product which enhances the TV watching experience, is needed. This problem of consuming Web content on TV motivated us to develop the Videolization system.

Videolization is a knowledge graph based visual interpretation system that aims to automatically create TV program contents in a video format from Web content and hence provide a passive consumption service for TV users. Via the Videolization product, TV users can watch their favorite Web content instead of having to read it. Figure 2 shows the system architecture of the Videolization system. It has three major modules: Videolization Channels, Repository and Video Generation Pipeline. In the following sections, these modules will be analyzed comprehensively.

3.1 Videolization channels

TV program contents are generated in several categories based on the source and type of the Web content. Each category is presented in a separate TV channel to the TV audience. A few possible channels of a Videolization product are listed below:

- Encyclopedia Channel: Information about entities are presented to the user. For instance, when the user asks for $|Huawei|$ entity, the channel will present a video about Huawei with important facts such as its foundation year, gross profit, etc.
- News Channel: Online news documents are presented to the user. News documents are split into several categories such as politics, sports, etc. allowing users to watch news from different categories.
- Social Network Channel: Posts from the social network platforms are presented to the user. By watching this channel, user can get a brief report about his or her recent social media activities. Figure 3 shows a visualization of a tweet posted by Barack Obama, the 44th president of the United States. The social media post contains an entity (Clean

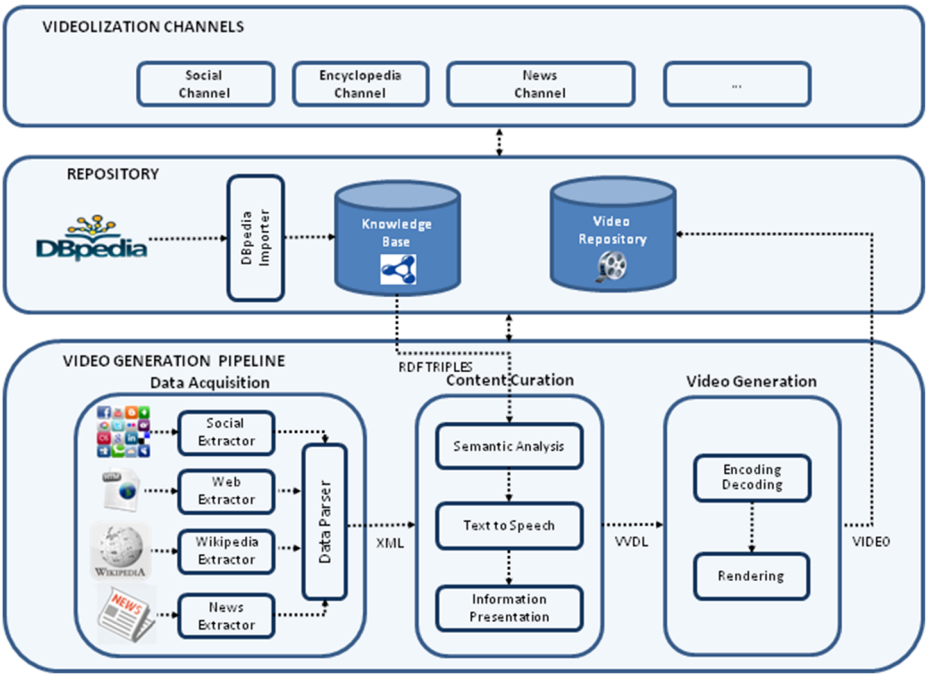


Fig. 2 Architecture of the Videolization System

Water Act) and a link to the Barack Obama website. As shown in the figure, they are also represented visually with their logos.

As a use case, the Encyclopedia Channel is developed to present Wikipedia articles in video format in this study. It is a Web application that allows the users to search Wikipedia article titles and watch the generated videos. The user interface of the Encyclopedia Channel is developed through considering human computer interaction (HCI) studies [7] and design principles for usability. Figure 4 shows a screenshot of the Encyclopedia Channel.

3.2 Repository

The Repository module stores generated video files and utilizes a knowledge graph. MongoDB,[13] a NoSQL database, is used for this purpose. In the context of this module, we also carried out the generation of a simplified version of DBpedia knowledge graph. DBpedia contains metadata about entities. However, most of the information consists of details or intermediate information not suitable for presentation in generated videos. Thus, we decided to identify the most significant properties of entities and filter out the rest. For example, for a company entity, its founder, foundation year, and industry are the most informative properties and they are worth presenting in a video.

In order to achieve the simplified version of DBpedia, we decided to manually determine the significant properties, since the DBpedia schema barely changes and there are few

[13] https://www.mongodb.com/

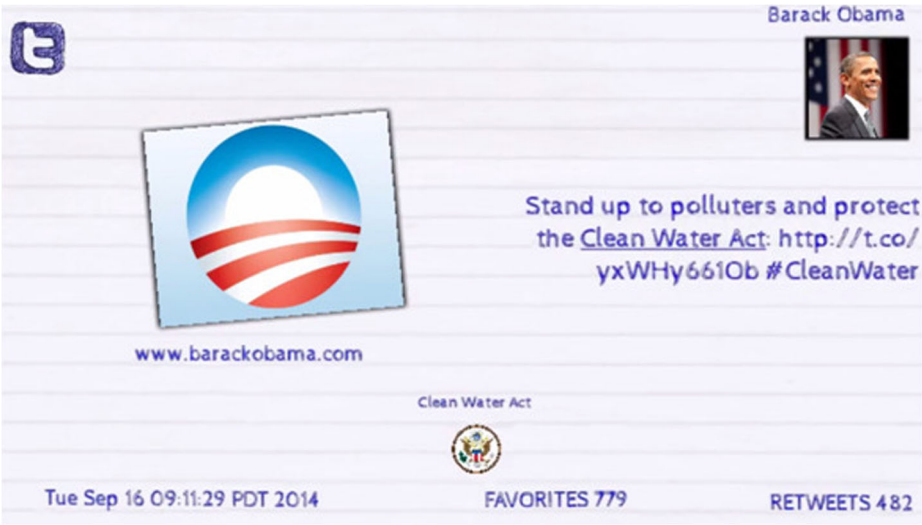


Fig. 3 A sample visualization of a social media post

defined property types. Firstly, we created a histogram of DBpedia properties which produced a list of 1367 distinct properties. The most frequent and informative 100 properties are manually selected (listed in Table 1).

Video Generation Pipeline Video Generation Pipeline is the core module that provides the main functionality of the system. This module transforms acquired Web content such as social feeds, Web documents, news RSS feeds, etc. into video format using semantic Web and computer graphics technologies. Figure 5 shows an activity diagram of the Video

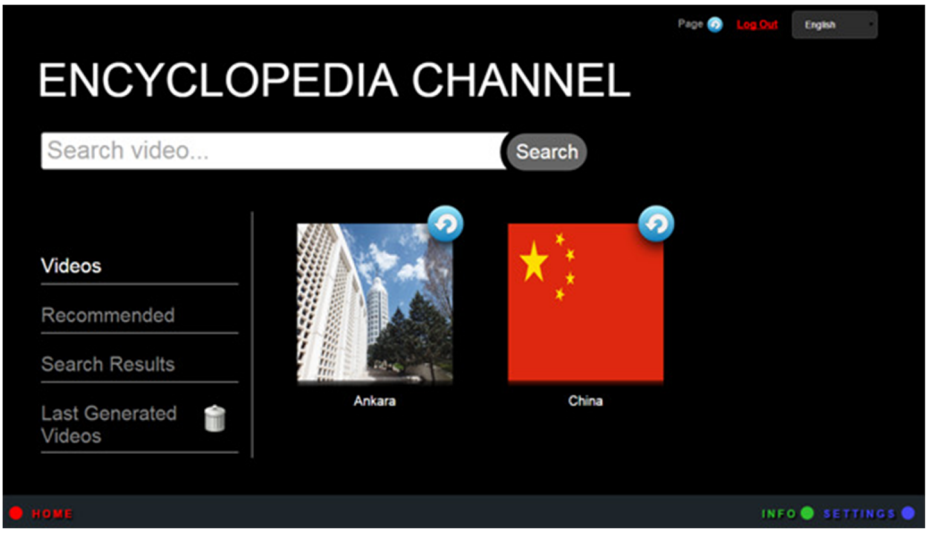


Fig. 4 A screenshot of the Encyclopedia Channel

Table 1 List of curated 100 DBpedia properties used in this study

Team	City	Maximum Elevation (μ)	Ship Beam (μ)
Founder	Film Director	Religion	Architectural Style
Type	Language	Minimum Elevation (μ)	Year
Description	Home Town	Motto	Location City
Birth Place	Alma Mater	Located In Area	Series
Birth Year	Nationality	Department	Creator
Country	Founding Year	Ground	Number Of Episodes
Genre	Population Density	Headquarter	Origin
Death Year	Weight (g)	Builder	Number Of Pages
Location	Length (μ)	Opening Year	Number Of Employees
Family	Party	Known For	Coached Team
Starring	Publisher	Cinematography	Former Name
Population Total	Birth Name	Field	Broadcast Area
Occupation	Owner	Combatant	Affiliation
Death Place	Music Composer	Network	Programme Format
Class	Musical Artist	Strength	Family
Elevation (μ)	Author	Key Person	Route Start
Runtime (s)	Computing Platform	Military Command	Route End
Producer	Album	Band Member	Manufacturer
Position	Product	Literary Genre	River Mouth
Release Date	Commander	Spouse	Engine
Area Total (m^2)	Industry	Developer	Batting Side
Height (μ)	Order In Office	Number Of Students	Draft Year
Writer	Year Of Construction	County	Country
Performer	League	Parent	Population As Of

Generation Pipeline. This module is composed of two sub-modules: Content Curation and Video Generation. In the following sections, these modules are analyzed comprehensively.

3.3 Content curation

The Content Curation sub-module analyzes the collected Web content and firstly determines what to present to the individual TV audience. Then the module decides how to present the selected content. Our methodology for the content curation has three main processing steps: data acquisition, semantic analysis and information presentation.

Data Acquisition This deals with the Web data extraction and retrieval challenges. Various kinds (social media, news, product) and types (xml, html, image, video) of Web resources are collected in order to be presented in the Videolization channels. For this study, we realized the Encyclopedia Channel by parsing a dump of Wikipedia,[14] which is provided in wikitext format.

[14] https://dumps.wikimedia.org/enwiki/

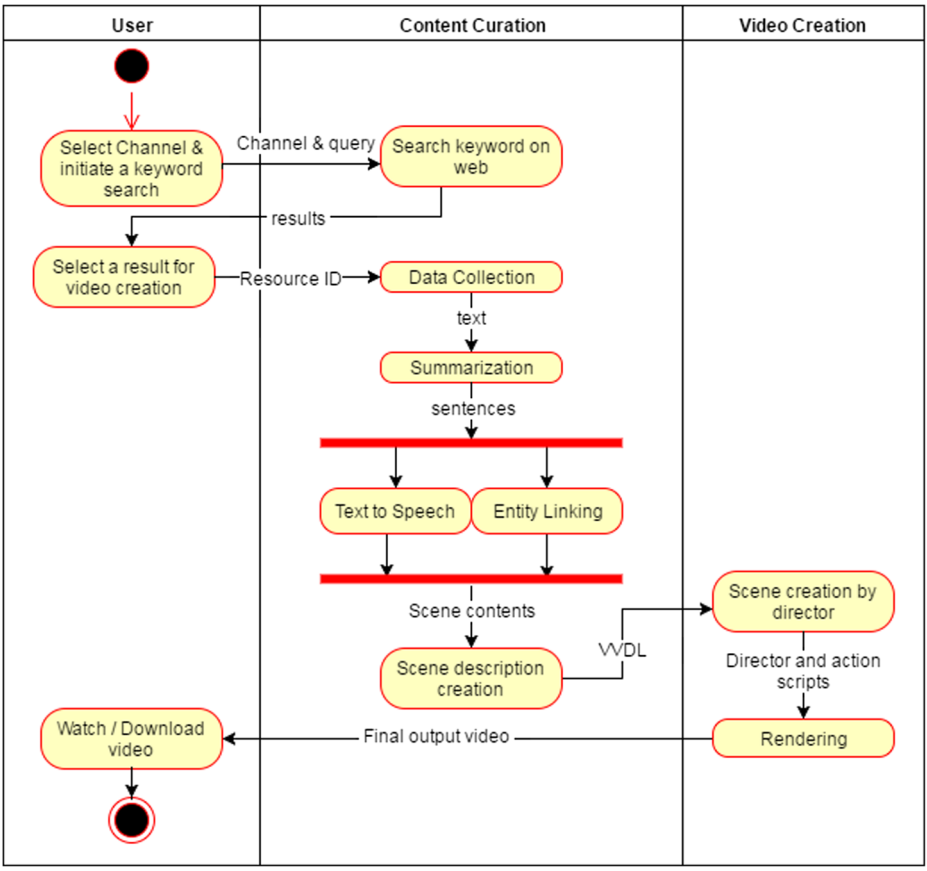


Fig. 5 Activity diagram for a video generation flow

Semantic Analysis Acquired Web content is first semantically analyzed in order to understand the context by performing semantic annotation (entity linking). In this study, we used TagMe for entity annotation. It provides a list of entities for a given text content with their significance values. Then we retrieve the properties of these entities from the Videolization Knowledge Graph (a simplified version of DBpedia) to enrich the video content.

Text to Speech Generation A TTS generator module is required to read out the narrative text content in a video. In order to find a high quality and feasible TTS solution we evaluated several popular and publicly available TTS solutions such as MaryTTS,[15] Vocalware,[16] etc. We observed that the voice quality of Vocalware is better than other TTS solutions, so we utilized Vocalware's commercial TTS solution for this purpose.

[15] http://mary.dfki.de/

[16] https://www.vocalware.com

3.3.1 Information presentation

A template based approach is utilized to render the analyzed text content into video format. For each of the Videolization system predefined channels - Encyclopedia, News and Social Network - the system has a corresponding template. The templates include semantic and functional rules. For example, in the Encyclopedia Channel firstly the summarization function is utilized to create a more concise representation which will still retain the most important sentences. The videolization system utilizes an extraction based automatic text summarization sub-module [17]. After summarization, each sentence is *videolized* separately by forming audio and video components. For the audio component, the TTS is generated and optional background music is added. For the visual element, semantic analysis is performed and Information Presentation module determines a visualization type for each sentence of the input text. Formally let S be the current input sentence to the system, let $E = \{e_1, e_2, ...e_n\}$ be the set of entities extracted from S and let O_{e_i} be the number of occurrences of entity e_i in the input text. Then $\forall e_i$, $O_{e_i} \geq \beta$, where β is a manually set threshold, $f(e_i)$ is determined,

$$f : E \rightarrow \{EntityGraph, EntityVideo, EntityImage, Text\}. \quad (1)$$

The f function selects the visualization type for the sentence by using the procedure described in Algorithm 1. The most time consuming step in Information Presentation is the selection of the representing entity in the input text. Other steps require linear time in terms of the input text length. In this study, we used TagMe for entity linking and its runtime complexity is $O(d_{in} \times (n \times s)^2)$ where n is the number of anchors detected in input text T, s is the average number of senses potentially associated with each anchor, and d_{in} is the average in-degree of the corresponding Wikipedia page. If k sentences exist in T, the total complexity of Information Presentation process is $O(k \times (d_{in} \times (n \times s)^2))$.

3.3.2 Scene types

For all scene types we have two general rules related to entity selection:

i The entity must occur in the whole document at least twice.
ii If there are multiple candidates, the one with the highest *TagMe* significance weight is chosen.

We enforce these rules in order to select more coherent entities in the document. Additional rules exist for each scene type and they are listed as follows:

Entity Graph A sentence can be *videolized* as an entity graph scene if it has an entity that fulfills the following conditions:

i The candidate entity must have at least two properties from Table 1.
ii It must not have been presented as an entity graph in the previous sentences.

The call *has_entity_graph* in Algorithm 1 refers to these conditions. The visual component of the video is created using the selected entity's key properties and its image. Figure 6 shows an entity graph scene generated from the sentence "As Russia mobilised in support of Serbia, Germany invaded neutral Belgium and Luxembourg before moving towards France, leading the United Kingdom to declare war on Germany."

Algorithm 1 The algorithm for a rule based scene selection for an input document.

```
summary = summarization_service(document);
while sentence in summary do
    sentence.scene = Text;
    tts = tts_service(sentence);
    if tts.length < min_scene_duration then
        sentence.scene = IgnoredSentence;
        next sentence;
    end
    entities = semantic_linking_service(sentence);
    sort entities by importance;
    for entity in entities do
        count = document.count(entity);
        if used(entity) or count < min_occurrence then
            next entity
        else if has entity graph(entity) then
            sentence.scene = EntityGraph;
            break;
        else if has entity video(entity) then
            sentence.scene = EntityVideo;
            break;
        else
            sentence.scene = EntityImage;
            break;
        end
    end
end
```

Entity Video Similar to the entity graph scenes, the most significant entity of a sentence is selected for visualization purposes. If the selected entity does not have any significant property value, the system may show a video that represents the entity, if the following conditions are fulfilled:

i The candidate entity does not fulfill the conditions of entity graph.
ii It must not have been presented as an entity video in the previous sentences.

Fig. 6 A sample scene of type entity graph

Fig. 7 A sample scene of type entity video

iii A video related to the entity can be found in available repositories.

The call *has_entity_video* in Algorithm 1 refers to these conditions. We utilize the Shutterstock[17] website as a video repository. Along with entity text, its type and document title are also used as search terms to handle disambiguation (e.g. Apple: company or fruit) and context relevancy problems.

Additionally we filter the search results based on video duration. The minimum video duration is determined based on the duration of the TTS audio file. The maximum video duration is fixed for 60 seconds. When the duration of the video found is longer than the TTS duration, we apply a greedy algorithm to resolve audio-video synchronization issues. As long as total audio duration is shorter than the video duration, the following sentences in the text are concatenated to the current scene and their TTS are read out over the video. If all the remaining sentences are consumed and the video is still not finished, the video is cut with a transition effect. Figure 7 shows an example entity video scene generated from the sentence "It was one of the deadliest conflicts in history, and paved the way for major political changes, including revolutions in many of the nations involved".

Entity Image A sentence can be *videolized* as an entity image scene if it has an entity that fulfills the following conditions:

i The candidate entity does not fulfill the conditions of entity graph and video.
ii It must not have been presented as an entity image in the previous sentences.

The call *has_entity_image* in Algorithm 1 refers to these conditions. We utilize images from the entity's Wikipedia page and the results from a Google image search. Wikipedia articles generally have at least one associated image and this representation is preferred by the Videolization system. If the entity's Wikipedia page does not have an image, we perform a search using entity text, its type and document title. The top result for this search query is used as the entity image. Figure 8 shows an example scene of type entity image that is generated from the sentence "Darth Vader, born Anakin Skywalker, is a fictional character in the Star Wars universe".

[17] https://http://www.shutterstock.com/

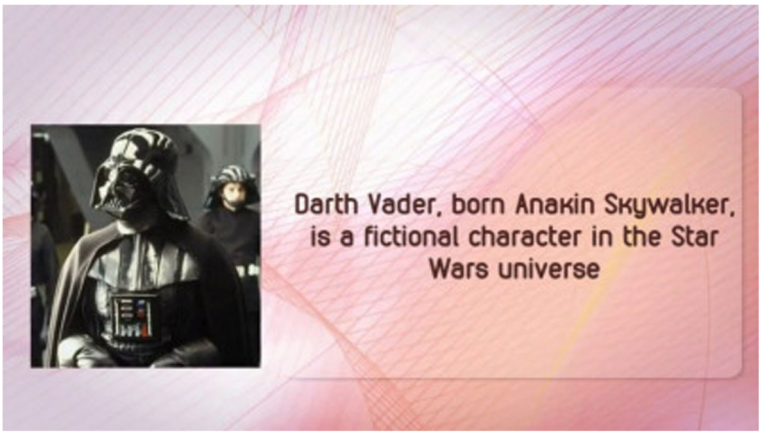


Fig. 8 A sample scene of type entity image

Text This scene type can be considered as a fallback visualization option. It is the least preferred option and it is only utilized when conditions of the other scene types are not fulfilled. Generally, this alternative comes into play when the sentence has no entities. The visual component of the video is created by simply depicting the sentence text in a TV friendly way. Figure 9 shows an example text scene that is generated from the sentence "More than 70 million military personnel, including 60 million Europeans, were mobilised in one of the largest wars in history.".

3.3.3 VVDL output

The final output of the Content Curation module is an XML/JSON file, more specifically; a *"Videolization video description language"* (VVDL) file that defines the content of a generated video. A VVDL file consists of a number of *scene* elements and related configuration such as background music, video language, unique identifier, title, duration, etc. A scene is

Fig. 9 A sample scene of type text

```
[{
"Videolization": {
  "id": unique_id,
  "title": the Title of Wikipedia page,
  "language": "en",
  "channel": "Encyclopedia",
  "duration": video duration in seconds,
  "Scenes":[{
    "t": "Entity Graph",
    "start":scene start sec., "end":scene end sec.,
    "audio": [ TTS information ],
    "text": [{
      "t": "title", "c": Entity (Subject) },{
      "t": "eg", "c": Predicate_1:Object_1 },{
      "t": "eg", "c": Predicate_2:Object_2 },{
      }]
    },{
    "type":"Entity Image",
    "start":scene start sec., "end":scene end sec.,
    "audio": [ TTS information ],
    "image": /path/to/image
    "text": [{
      "t": "sentence", "c": the sentence itself }
      ],
    }]
  }
}]
```

Listing 1 An example of Videolization video description language file for the Encyclopedia channel is listed. This example has two scenes; an entity graph scene (9–16) with three fields and an entity image scene (18–24) with one image. For some fields abbreviated versions are used; "t" stands for "type"; "c" stands for "content" and "eg" stands for "entity graph". Italic text is explanatory information

built up using audio, image, video and text items which are the building blocks of a VVDL document.

Each scene item has a unique identifier used by the video generation module to match the related files to these scene items. The scene item has also a type keyword to identify whether it is an image, video, audio, or text. An audio item can be used to store a text-to-speech audio, music, or sound effects, whereas image and video items may be used for various media formats. Listing 1 shows an example of a VVDL file.

VVDL does not store any information about visual effects, animations, etc. Instead, it only contains the information to present. The way this knowledge is presented is up to the selected template of the video generation module. This allows VVDL to provide interdependence between content curation and video generation modules. The same content could be visualized differently using different visual templates. Alternatively, it is possible to generate videos using a third party tool such as Adobe After Effects [18] from the VVDL file by using a specialized template and a parser for the VVDL. Figure 10 presents two visualizations of same VVDL file.

There are other video description languages such as Stupeflix (SXML)[19] and MPEG-7.[20] SXML is an XML based language that is used to create videos using the commercial video generation service, namely Stupeflix. It is vector-based, 4D, and fully human readable. Every object in an SXML document can be animated, effects and transitions can also be

[18] http://www.adobe.com/products/aftereffects.html

[19] https://stupeflix-sxml.readthedocs.io/en/latest/index.html

[20] http://mpeg.chiariglione.org/standards/mpeg-7

Figure 10 Sample scenes from a video produced through Adobe After Effects using a template and a parser for the VVDL format. The generalized nature of VVDL format allows content curation output to be used in third party applications

defined in an SXML file. MPEG-7 is a multimedia content description standard that stores metadata in XML format, and it can be used for a broad range of applications. MPEG-7 is a standard and an excellent choice for describing multimedia content description, mainly because it is very flexible and comprehensive. However, this increases the complexity of the language [1]. Generic concepts, the high number of different descriptors and description schemes and their flexible inner structure makes it more difficult to work on. In contrast, VVDL is a much simpler description language due to the template-based video generation approach utilized in Videolization. VVDL does not contain any animations, effects, or time-line information. It is more like a data structure that holds various information like scene objects or actors in the video. This provides more flexibility to the system since the same VVDL file can be used to generate different videos by using different visual templates.

As future work, VVDL could adopt MPEG-7 features and MPEG-7 standards could also be supported by Videolization in order to benefit from MPEG-7 advantages such as interoperability, comprehensiveness and flexibility.

3.4 Video generation

The video generation module is responsible for generating the final video using the VVDL file and the multimedia content provided by the Content Curation module. It visualizes and enriches the provided content through multimedia encoding and decoding, scene creation and rendering phases.

3.4.1 Multimedia encoding and decoding

Multimedia content gathered by the Content Curation module can be in various formats and each gathered multimedia file should be decoded to access the raw information in the file. In order to make the storage and transmission processes more efficient, raw video data should be encoded in a compact yet high quality format. In order to decode audio-visual Web content and encode the output video content, an audio and video processing tool is required. In this study, we used Libav[21] libraries for this purpose as it supports many of the common image formats and audio/video codecs such as BMP, JPEG, PNG, AAC, MP3, MPEG-4 and H.264.

[21] https://libav.org/

3.4.2 Scene creation

Videolization of the content collected by the Content Curation module is realized by an actor-director approach. At each time-step, a high level scene manager renders all actors which are on the stage. All received visual, audio and text contents are regarded as actors and a director script assigns roles to the actors by parsing and interpreting the VVDL file generated. Considering the available actors in a scene, the director script decides on a scene template from a set of available visual templates, and maps the actors to the corresponding placeholders in the scene template. Scene templates contain a visual description of a scene and they include action and effect scripts attached to each actor. When the director script selects a scene template and assigns actors to the selected template, each actor plays its role according to the action and effects script in the template. Action scripts define the entrance and exit times of individual actors as well as their corresponding actions at each execution time step. Effect scripts define visual artistic effects that will be applied to each actor at each time step. In order to produce natural and visually pleasant videos to the user, the director script optimizes the video by adaptive positioning of scene elements. On the other side, the content is enriched with audio-visual transition effects and animations. After the scene creation phase, an intermediate video timeline object that holds all the actors and animations is created and sent to the video rendering module.

Adaptive positioning includes selection of the most suitable template for the collected visual data and adapting the size, position and scale of the visual items on the scene. It is mainly carried out to optimize the visual differences that arise from the various aspect ratios, lengths of image and text elements in the scene. First, a base template is designed for each channel, which defines the regions in which each actor can be placed according to their type. Depending on the design of the base template these regions can intersect with each other as shown in Fig. 11. Visual actors such as images, videos or texts are ideally placed inside the designated regions. When regions intersect, actors are prioritized depending on the media types, dimensions and aspect ratio. The actor with the highest priority uses the whole area of its designated region and the actors with lower priorities use the remaining areas of their corresponding regions (Fig. 11). The sizes and positions of the actors are calculated in a relative manner, thus changes in the video resolution or aspect ratio do not affect the outcome of the adaptive positioning process. Positions, dimensions and aspect ratio of the scene items are all updated based on the properties of the video.

Audio-visual enrichment is a template driven task which includes enriching the video scenes with supplementary visual items and effects in order to provide a more complete and pleasant experience to the viewer. It includes additional decoration of the scene using the layout information created by the adaptive positioning phase. Specific animations are also applied to the screen actors for instance upon entrance, exit and etc.

3.4.3 Rendering

After the scene creation process, a video timeline object which holds all actors and their corresponding action scripts is obtained. In order to transform this timeline object into a video, a rendering system is required. Rendering is the process of generating an image or video from a description that contains objects in a strictly defined language or data structure.

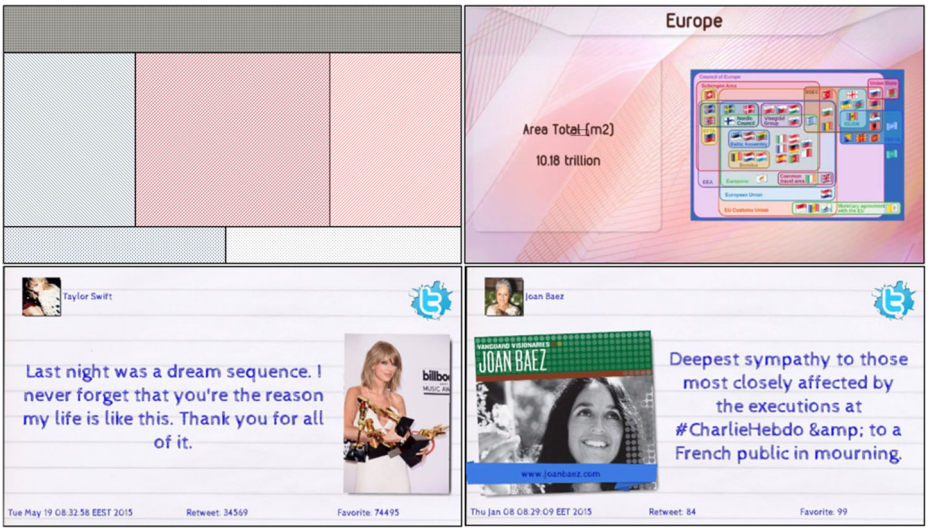


Figure 11 Examples for adaptive positioning. *Upper left*: An example design template. *Upper right & bottom*: Screenshots from videos scenes obtained by adaptive positioning of screen actors according to the content

Geometry, viewpoint, texture, lighting and shading information is given as a description of the virtual scene. The timeline object and its attached action scripts complement the required information. For rendering purposes, we used OpenGL[22] which is the leading open standard library that is used widely for rendering tasks. Using OpenGL, graphical scene actors such as images, video clips and texts are placed and visualized on mesh structures as textures. Animation and effect scripts which are attached to the scene actors are also written using OpenGL shaders. Using OpenGL and LibAV together, visualized output scene frames are directly fed to the video encoding module in order to generate the output video as the final product of the video generation pipeline. Algorithmic steps for the rendering process are provided in Algorithm 2.

Algorithm 2 Video rendering using a VVDL file.

```
Function video = video_generation(vvdl_file) begin
    while scene in vvdl_file do
        scene_duration = scene.audio.tts_duration + additional_time;
        scene_template = get_scene_template(scene.channel,scene.type);
        scene_description =
        director(scene_duration,scene_template,scene.audio.TTS,scene.text,scene.image);

        video_description.add(scene_description);
    end
    video = video_renderer(video_description);
end
```

[22] https://www.opengl.org/

4 Experiment and user study

We analyze our system from two different aspects; i) we have evaluated visual quality and effectiveness through a qualitative user study and ii) we have measured performance and run-time characteristics through a simulated experiment.

Visualization of Web content has subjective characteristics, hence we have evaluated the system through a user study; the effectiveness of the proposed system is validated empirically through an opinion survey (questionnaire). A survey with 10 questions was prepared for this purpose. We requested 26 participants (14 male, 9 female, and 3 N/A) to use our system and respond to the survey questions. The average participant age was 30.25 with a standard deviation of 4.39. Participants are engineers, teachers, an academician, a lecturer, a credit controller, and graduate students with at least a Bachelor's degree. The participants were given basic training to familiarize them with the system and the video generation process. Then each participant was asked to watch three short videos, namely *Darth Vader*, *United States*, and *World War I*, generated by the system. After watching the videos, they were asked to evaluate the system by answering the questionnaire.

The first six questions assess specific features of the system such as quality of video effects and TTS. The participants were asked to make a judgment for each question using a scale of 1 to 4 (4 being the best) with the option of not giving an answer. One question was about choosing the best video from a set of three videos. Two yes/no questions are then utilized to assess the impression the system has made on the participants. A comment section was also given to the participants in order to get their feedback.

The results of this study are visualized in Figs. 12 and 13. Quality wise the best performing aspect was "visual quality" with $\mu = 3.27, std = 0.60$. For all questions, the performance was above average. The worst performer was "audio-visual synchronization" ($\mu = 2.96, std = 0.89$) which is open to improvement.

Among the three encyclopedic videos: *United States*, *Darth Vader*, and *World War I*, 39% of the participants chose *Darth Vader*, 28 % chose *United States*, and 17 % chose *World War I* as the best video, while the remaining 17 % did not make a choice. The reason for

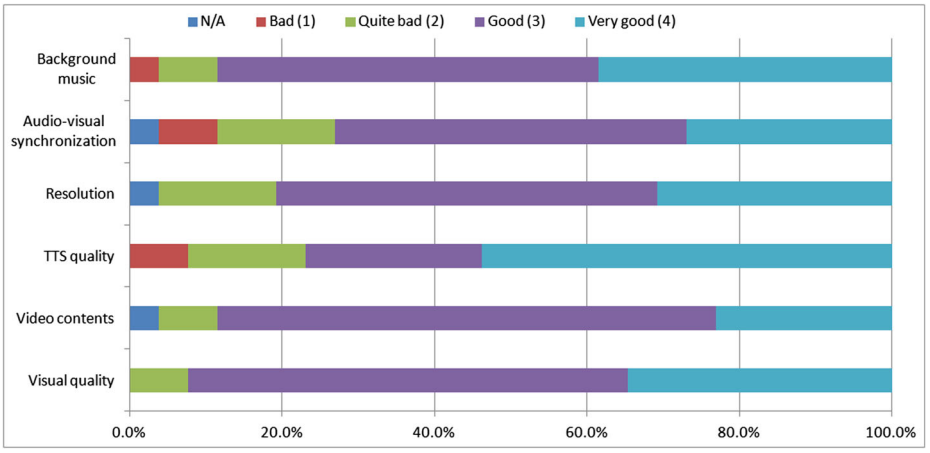


Figure 12 Opinion survey results for the first six questions are shown in a stacked histogram. For these questions the participants were asked to evaluate the quality of the respective aspect using a scale of 1 to 4 with the option of not answering (N/A)

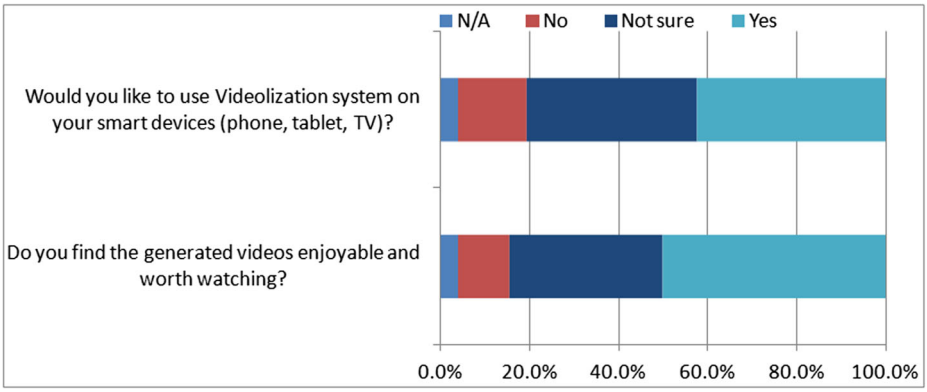


Figure 13 Opinion survey results for the last two questions are shown in a stacked histogram. The participants were asked yes/no questions related to the general effectiveness and appeal of the Videolization system

these results might be that *Darth Vader* is probably a more fun and interesting topic for the users.

Fifty percent of participants found the generated videos enjoyable. Although over 83 % of survey users are satisfied with the specific features of the system, only 42 % of them indicated that they would like to use our system to consume Web content on their TVs and smart devices. Fifteen percent of them would not like to use the system at all while 38 % are not sure. These last two questions evaluate the system in terms of its general performance. The low performance scores in the general evaluation compared to the evaluation of the system features could be explained by users' conservative behaviour to new products.

Runtime efficiency is another key performance indicator of video generation systems. The amount of Web data increases daily and a video generation system must be scalable in order to process it. In order to measure the runtime efficiency and scalability of our system, we created videos for the most popular Wikipedia articles.[23] The experiments resulted in 100 videos generated in a workstation of specification: Intel Xeon E5 CPU @ 2.40 GHz and 16 GB of RAM. Figure 14 shows the performance results.

We observed that on average, a one minute video could be generated in approximately 140 seconds, which shows that our system can generate videos from Wikipedia articles almost in real-time. To further investigate our runtime efficiency we have analyzed the time consumption of our main modules, namely *Content Curation* and *Video Generation*. Although our Content Curation module is faster than realtime, the video generation task is around %70 slower. Moreover, as can be seen in Fig. 14, the performance of video generation has a linear relation with respect to output video length. The longer the output video, the more closer to real-time video generation becomes. Content curation has a less direct relation to output video length as it depends on the type and amount of content that is acquired. However a basic trend-line analysis shows that in general content curation has a similar relation to video length as well.

We also analyzed scene type distributions in the 100 generated videos. Figure 15 shows the numbers for each scene type. We observed that most of the scenes are entity based. A small portion of the scenes are text based. Although we prioritize videos over images,

[23] https://goo.gl/c759nv

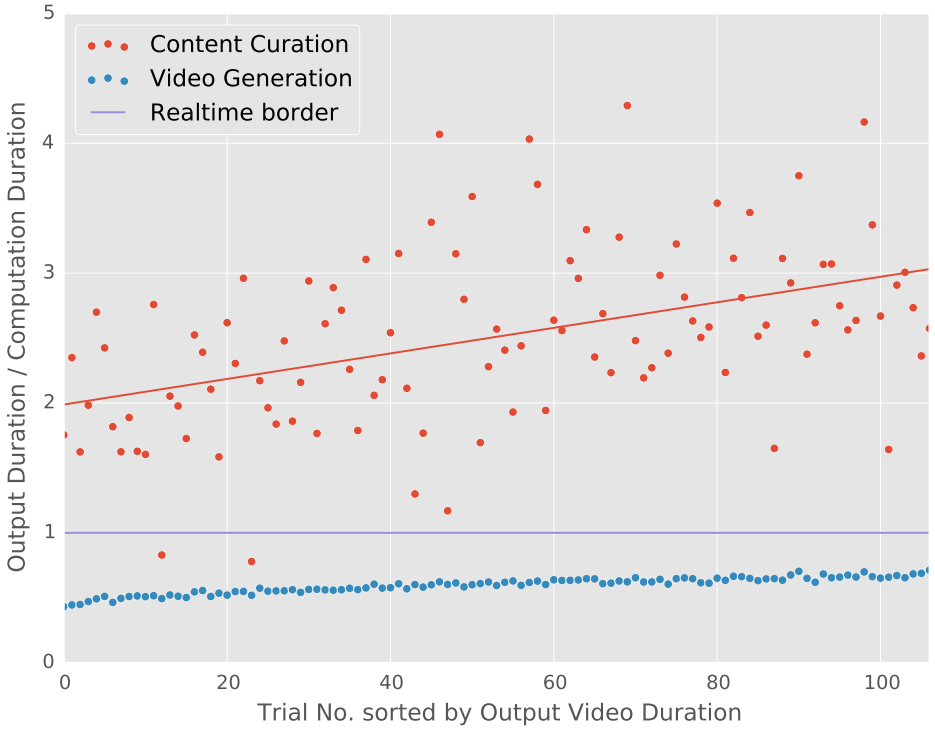


Figure 14 Content creation and Video generation phases are compared with respect to output video duration. A value of 1.0 denotes real-time computation, any value lower than this is slower than real-time and vice versa. Trial instances are sorted with respect to output video duration

there are ten times more entity image representations than entity video representations. We attribute this result to an inadequate size of video repository. Sample videos produced by the *Videolization* system are publicly available.[24]

For the evaluation of the proposed system, a comparison with the state of the art methods and systems [5, 20, 22, 23] would be useful. However, the performance evaluations are not provided for the systems proposed in the literature. On the other side, state of the art systems and Videolization differ from each other in terms of the features utilized. Some significant features such as fully automatic video generation and visualization of a given text do not exist in other systems proposed. For instance, Stupeflix generates a video based on a given content such as images. It does not have the capability of converting text to video. Another example is Web2TV [23] which can generate videos again only by the given content. Any content enrichment is not carried out by the system. Videolization can generate videos from any text by creating an enriched content based on the entities extracted from the given text and the whole process is automatic. It might be possible to evaluate and compare these systems in terms of user satisfaction via comparative user studies. However, the systems are either commercial services or academic works without public access. Hence, it has not been possible to conduct such a user evaluation in order to compare Videolization and other systems.

[24] https://goo.gl/K18mw9

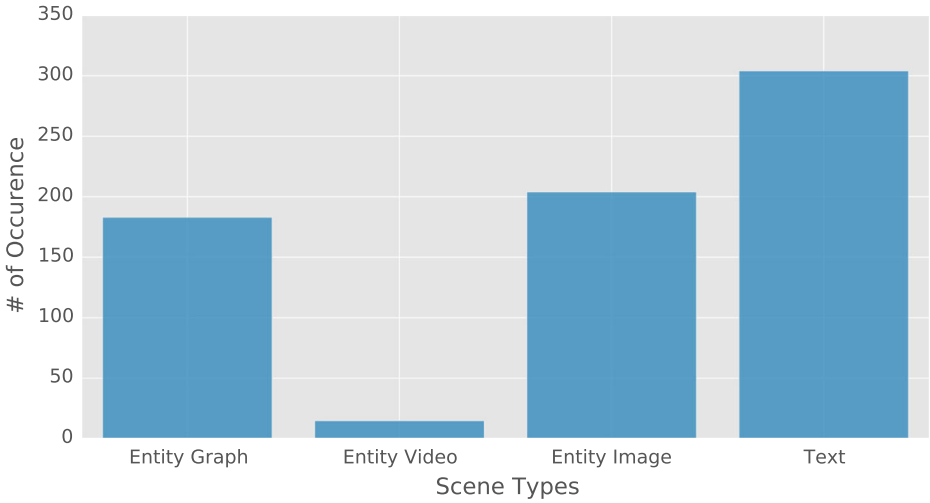


Figure 15 Visualization technique counts for the experiment are shown

5 Discussion

In our experiments we have shown that the *Videolization* system is capable of providing videos in near real-time. Applying concurrent processing or using the big data approach could easily improve the system to real-time performance. The *Videolization* system was designed to allow such improvements, since the sentences can be processed in parallel with minimal changes to the scene type rules. Hence, inter-sentence dependency could be removed easily. Then the *videolized* sentences could be stitched together to output the final video.

The video and image selection process utilized to represent a given entity is also open to improvement. Currently, we optimize this process by using the *entity text*, *type*, and *document title* in the search query in order to acquire a related image or video. An image retrieval based approach [12, 24] over an annotated video or image library can provide more suitable visual representations. The retrieval process can also be utilized to judge the quality of selected images and videos.

In the *Videolization* work-flow, only one entity presentation is allowed per scene. This design choice was made to represent the most salient information. However, in some cases multiple entities are desirable. For instance consider the following sentence:

[... It has 21 R&D institutes in countries including China, the United States, Canada, the United Kingdom, Pakistan, France, Belgium, Germany, Colombia, Sweden, Ireland, India, Russia, and Turkey ...].

None of the entities in this sentence should have more weight compared to the others. An additional scene type that allows multiple entities to be represented can be added to the system. For instance, country flags could be used for this example.

Quantized information can be found in many sentences with numeric values like ratios, counts and monetary values. Although these values are not technically entities, they are suitable for TV friendly visualization through specialized charts, scales and clip art style graphics. An NLP based approach [26] that can convert for instance the following sentence;

[... it currently serves 45 of the world's 50 largest telecoms ...]

into a an animated pie chart style visualization requires further investigation.

In the current Videolization system the Entity Graph is always preferred over other scene types. This is a design choice to maximize the amount of information presented to the user. Depending on the application, it is possible to devise other orderings between scene types. An alternative approach for scene type selection is to utilize a global optimization that can maximize the number of entities visualized and that can minimize the number of text representation types.

Although the participants are from a variety of educational and professional backgrounds, the carried out user study for evaluation of Videolization system still has a certain degree of bias. Because all users have at least a Bachelor's degree, and they were reached through social platforms. That is why all of them are familiar with computers, internet, and smart devices. Future evaluations of the system will be based on a larger group of users that have a more varied educational background and technology familiarity in order to decrease the bias introduced by the test group.

6 Conclusion

This study presents a knowledge graph based video generation system that automatically converts textual Web content into videos using semantic Web and computer graphics based technologies. As a use case, Wikipedia articles are automatically converted into videos. Visualization of text content is the key challenge in the proposed system. To address this problem, a template based visualization algorithm is proposed, which leverages DBpedia as a knowledge graph and TagMe as an entity linking system. The effectiveness of the proposed system is validated empirically through opinion surveys. Evaluations show that the proposed system has a satisfactory video generation performance. Forty two percent of survey users indicated that they would like to use our system to consume Web content on their TVs and smart devices, and 50 % of them find the generated videos enjoyable.

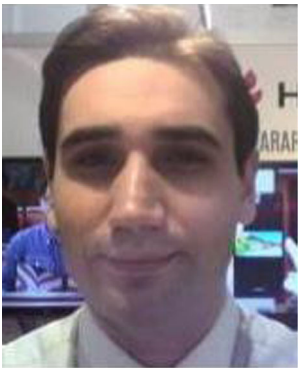
Our research differs from the previous studies in its novel knowledge graph based visualization method used to convert Web content into videos. A simplified version of DBpedia is utilized as a knowledge graph, featuring 100 most important properties. Videolization video description language is proposed to describe videos in XML format and provide interdependence between video content and its visualization.

In conclusion, this paper has demonstrated the potential and promise of the knowledge graphs for the text visualization task. As future work, the text visualization algorithm can be improved in several directions. An image retrieval based approach over an annotated video or image library can provide more suitable visual representations for a given entity. Moreover, a sentence could be visualized better with the presentation of all entities in the sentence, instead of using only the most significant entity. According to the opinions of the participants, improvements can be made by choosing coherent background music to accompany the video content and by providing more interesting and more detailed video contents. More interesting videos could be generated using an abstraction based automatic summarization system rather than the current extraction based one. Then, a summary that is closer to what a human might generate could be obtained. Music properties such as tempo, mood, and genre might be used in order to select more suitable background music. Moreover, a music database might be built up with metadata for choosing music related to the video content.

References

1. Bailer W, Schallauer P (2006) Detailed audiovisual profile: enabling interoperability between mpeg-7 based systems. In: 2006 12th International Multi-Media Modelling Conference, pp 8. doi:10.1109/MMMC.2006.1651323
2. Borman A, Mihalcea R, Tarau P (2005) Picnet: Augmenting semantic resources with pictorial representations. In: AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors, AAAI, pp 1–7
3. Cai R, Zhang L, Jing F, Lai W, Ma WY (2007) Automated music video generation using web image resource. In: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, IEEE, vol 2, pp II–737
4. Cornolti M, Ferragina P, Ciaramita M (2013) A framework for benchmarking entity-annotation systems
5. Coyne B, Sproat R (2001) Wordseye: An automatic text-to-scene conversion system. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, ACM, New York, NY, USA, SIGGRAPH '01, pp 487–496
6. Ferragina P, Scaiella U (2010) Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, ACM, New York, NY, USA, CIKM '10, pp 1625–1628
7. Hansen V (2006) Interactive television design – designing for interactive television v 1.0 bbci & interactive tv programmes. BBC
8. Heath D, Ventura D (2016) Creating images by learning image semantics using vector space models. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11961
9. Hoffart J, Suchanek FM, Berberich K, Lewis-Kelham E, de Melo G, Weikum G (2011a) Yago2: Exploring and querying world knowledge in time, space, context, and many languages. In: Proceedings of the 20th International Conference Companion on World Wide Web, ACM, New York, NY, USA, WWW '11, pp 229–232. doi:10.1145/1963192.1963296
10. Hoffart J, Yosef MA, Bordino I, Fürstenau H, Pinkal M, Spaniol M, Taneva B, Thater S, Weikum G (2011b) Robust disambiguation of named entities in text. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP '11, pp 782–792
11. Kulkarni S, Singh A, Ramakrishnan G, Chakrabarti S (2009) Collective annotation of wikipedia entities in web text. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, KDD '09, pp 457–466
12. Liu Y, Zhang D, Lu G, Ma WY (2007) A survey of content-based image retrieval with high-level semantics. Pattern Recognition 40(1):262 – 282. doi:10.1016/j.patcog.2006.04.045. http://www.sciencedirect.com/science/article/pii/S0031320306002184
13. Meij E, Weerkamp W, de Rijke M (2012) Adding semantics to microblog posts. In: Proceedings of the Fifth ACM International Conference on Web Search and Data Mining, ACM, New York, NY, USA, WSDM '12, pp 563–572
14. Mendes PN, Jakob M, García-Silva A, Bizer C (2011) Dbpedia spotlight: Shedding light on the web of documents. In: Proceedings of the 7th International Conference on Semantic Systems, ACM, New York, NY, USA, I-Semantics '11, pp 1–8
15. Mihalcea R, Leong CW (2008) Toward communicating simple sentences using pictorial representations. Machine Translation 22(3):153–173
16. Milne D, Witten IH (2008) Learning to link with wikipedia. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, ACM, New York, NY, USA, CIKM '08, pp 509–518
17. Nenkova A, McKeown K (2012) blubberdiblubb A survey of text summarization techniques. In: Aggarwal CC, Zhai C, blubberdiblubb (eds) Mining Text Data, Springer, pp 43?76
18. Ohya H, Morishima S (2012) Automatic music video creation system by reusing existing contents in video-sharing service based on hmm
19. Ratinov L, Roth D, Downey D, Anderson M (2011) Local and global algorithms for disambiguation to wikipedia. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, Association for Computational Linguistics, Stroudsburg, PA, USA, HLT '11, pp 1375–1384
20. Shim H, Kang B, Kwag K (2009) Web2animation - automatic generation of 3d animation from the web text. In: Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 01, IEEE Computer Society, Washington, DC, USA, WI-IAT '09, pp 596–601

21. Socher R, Karpathy A, Le QV, Manning CD, Ng AY (2014) Grounded compositional semantics for finding and describing images with sentences. TACL 2:207–218
22. Sumi K, Tanaka K (2005) Transforming web contents into a storybook with dialogues and animations. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, ACM, New York, NY, USA, WWW '05, pp 1076–1077
23. Tanaka K (2007) Research on fusion of the web and tv broadcasting. In: Proceedings of the Second International Conference on Informatics Research for Development of Knowledge Society Infrastructure, IEEE Computer Society, Washington, DC, USA, ICKS '07, pp 129–136
24. Tao D, Cheng J, Gao X, Li X, Deng C (2016a) Robust sparse coding for mobile image labeling on the cloud. IEEE Transactions on Circuits and Systems for Video Technology PP(99):1–1. doi:10.1109/TCSVT.2016.2539778
25. Tao D, Guo Y, Song M, Li Y, Yu Z, Tang YY (2016b) Person re-identification by dual-regularized kiss metric learning. IEEE Transactions on Image Processing 25(6):2726–2738. doi:10.1109/TIP.2016.2553446
26. UzZaman N, Bigham JP, Allen JF (2011) Multimodal summarization of complex sentences. In: Proceedings of the 16th International Conference on Intelligent User Interfaces, ACM, New York, NY, USA, IUI '11, pp 43–52. doi:10.1145/1943403.1943412
27. Witten IH, Milne D, 2008 An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy, AAAI Press, Chicago, USA, pp 25?30
28. Wu X, Xu B, Qiao Y, Tang X (2012) Automatic music video generation: cross matching of music and image. In: Proceedings of the 20th ACM international conference on Multimedia, ACM, pp 1381–1382
29. Zhu X, Goldberg AB, Eldawy M, Dyer CR, Strock B (2007) A text-to-picture synthesis system for augmenting communication. In: Proceedings of the 22Nd National Conference on Artificial Intelligence - Volume 2, AAAI Press, AAAI'07, pp 1590–1595
30. Zitnick CL, Parikh D, Vanderwende L (2013) Learning the visual interpretation of sentences. In: ICCV, IEEE, pp 1681–1688

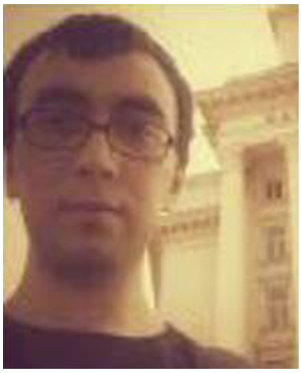


Murat Kalender received his Ph.D. degree from Yeditepe University on Computer Engineering in 2016. His research interests include semantic web, natural language processing, and machine learning. He is also a research engineer at Huawei Turkey R&D Center since 2013.

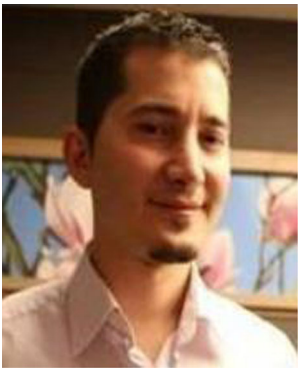
Mustafa Tolga Eren was born in Ankara, Turkey in 1984. He received his BSc and PhD degrees from Sabanci University on Computer Science and Engineering in 2006 and 2013, respectively. His interests include Real-time Visualization, Big Data Analytics, Data Visualization, Interactive Projection Mapping and Augmented Reality. He is currently employed at Huawei R&D, Turkey, as a Research Engineer.

Zonghuan Wu received his Ph.D. degree from Binghamton University on Computer Science in 2003. His interests include Semantic Web, Web Search, and Innovative Web Applications. He is currently employed as senior technology manager at Software Lab, Huawei Technologies in USA.

Ozgun Cirakman is currently a Ph.D. candidate at Istanbul Technical University, Electronics & Communication Engineering Department and joined TI department of Huawei Turkey R&D Center in 2014. Currently he is working on Videolization project alongside developing new project ideas. His research interests include statistical pattern recognition, machine learning and content based multimedia retrieval.



Sezer Kutluk received his BSc degree in Electrical-Electronics Engineering from Istanbul University, and MSc degree in Biomedical Engineering from Istanbul Technical University. He is currently a PhD candidate at Istanbul University, Electrical-Electronics Engineering Department, and he has been working as a Research Engineer at Huawei Turkey R&D Center since 2014. His research interests include machine learning and signal processing.



Gunay Gultekin is currently a Ph.D. student at Istanbul Kemerburgaz University, Electrical and Computer Engineering Department and joined TI department of Huawei Turkey R&D Center in 2010. He had worked different kinds of the projects so far. Currently, he is working on Videolization project.

Emin Erkan Korkmaz is a faculty member at Department of Computer Engineering, Yeditepe University. He received the B.S. degree in Computer Engineering from the Bilkent University, Ankara, in 1994, the M.S. and Ph.D. Degrees in Computer Engineering from the Middle East Technical University, Ankara in 1997 and 2003, respectively. He stayed as a Post-Doctorate researcher in the Department of Computer Science at University of Calgary, Alberta between 2003 and 2004. His research interests include evolutionary computation, machine learning and natural language processing.