Towards Using Semantic-Web Technologies for Multi-Modal Knowledge Graph Construction

Matthias Baumgartner
University of Zurich
baumgartner@ifi.uzh.ch

Luca Rossetto
University of Zurich
rossetto@ifi.uzh.ch

Abraham Bernstein
University of Zurich
bernstein@ifi.uzh.ch

ABSTRACT

While a multitude of approaches for extracting semantic information from multimedia documents has emerged in recent years, isolating any form of holistic semantic representation from a larger type of document, such as a movie, is not yet feasible. In this paper we present our approaches used in the first instance of the Deep Video Understanding Challenge, using a combination of several multi-modal detectors and an integration scheme informed by methods from the semantic web context in order to determine the capabilities limitations of currently available methods for the extraction of semantic relations between the characters and locations relevant to the narrative of a movie.

CCS CONCEPTS

• **Information systems** → *Evaluation of retrieval results*; Semantic web description languages; Video search; • **Computing methodologies** → Video summarization.

KEYWORDS

Video Understanding, Knowledge Graphs

ACM Reference Format:
Matthias Baumgartner, Luca Rossetto, and Abraham Bernstein. 2020. Towards Using Semantic-Web Technologies for Multi-Modal Knowledge Graph Construction. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3394171.3416292

1 INTRODUCTION

With the rapid growth in video content, mechanisms to extract its semantic content for purposes of analysis or retrieval become increasingly important. Much progress has been made in recent years in the area of extracting semantic content from the different modalities of individual videos, such as object detection [12], text extraction [18] or scene captioning [16] in still images, pose-tracking [3], and action recognition [9] in video or speech-transcription [4] in audio. More recently, approaches have emerged, which leverage the multi-modality of video for tasks such as audio source separation [17]. As far as such methods are concerned with the semantic content of the video, they have however primarily focused on shorter temporal units, such as individual scenes, rather than entire movies which tell a more intricate story.

The Deep Video Understanding Challenge aims at catalyzing progress in this area by providing both an evaluation dataset as well as a clearly defined task with accompanying evaluation protocols. In this first instance of the challenge, a graph representation describing the characters and places relevant for the story told by a movie, as well as their relations to each other are to be extracted directly from the provided videos. This way, questions about the characters and their relations within any particular story can be transformed into relatively simple graph queries which can be answered in a consistent and uniform way.

In this paper, we describe our approaches which aim at solving the tasks posed in this first iteration of this challenge. We treat this first participation as a baseline study in order to identify the limits of currently available approaches and identify areas where further research is required in order to solve such tasks more effectively. We describe our information extraction efforts in Section 2, while Section 3 outlines the integration of this extracted data as well as the ways queries are executed on them. Section 4 provides some insights into the differences between the provided and the produced graphs before Section 5 then discusses all the limitations of currently available approaches we encountered along the way. Finally, Section 6 concludes and offers some outlook.

2 MULTI-MODAL INFORMATION EXTRACTION

While video contains information in several modalities, the effort required for its extraction might differ greatly across them. For our subsequent information extraction, we limit ourselves to two modalities; static visual—meaning we only look at one frame at a time without considering them in sequence—as well as speech. Other information, such as non-speech components of the audio signal or temporal visual aspects like action recognition, are not considered due to the lack of efficient and robust pre-trained models.

2.1 Provided Data

The challenge uses the recently introduced *High Level Video Understanding* (HLVU) dataset [5] which consists of 10 movies released under creative commons licenses with a total combined duration of 681 minutes. For each of the 10 videos, the dataset also provides cropped key-frames, showing either characters or locations which are relevant for the story told by the movies. An ontology of relevant actions and relations is provided as well. For 6 out of the 10 videos, which serve as the development set, a human generated ground truth knowledge graph is provided in *Trivial Graph Format*, while the remaining 4 serve as a test set. For this test set, a number

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3416292

of queries are provided in a custom XML format, which are to be answered for the challenge.

The graphs provided as part of the development data describe the entities and their relations in a static way, indicating that none of the relationships of the characters in the story changes throughout. We therefore assume that this is also the case in the test set, meaning that a relation which is identified at any point within the video remains valid for its entirety.

2.2 Data Pre-processing

In order to enable the subsequent analysis, some data pre-processing is required. To avoid having to process every frame of the videos for all subsequent analysis, we perform shot-segmentation on the videos using the *cineast*[1] multimedia retrieval engine [13], which in addition to the shot boundaries produces one key-frame per shot.

Since no subtitles were provided with the videos, we have to extract the spoken dialog directly from the audio signal. To do this, we first re-sample the audio signal of each video into a 16kHz mono representation which is then passed to a voice activity detector[2] to isolate the sections which contain dialog and discard those without. To transcribe these sections containing speech into text, we use Mozilla's implementation[3] of Deep Speech [7]. The transcript is then stored in a subtitle-like format which retains the temporal range of each extracted speech element. As an alternative method, we uploaded the videos to Microsoft Stream,[4] which automatically produces subtitles for the added videos. These subtitles are then available in the WebVTT format. We further remove all punctuation, special characters, and repeated white spaces from the text.

Lastly, we extend the ontology given in the dataset in three aspects. First, we order relations in a hierarchy, from specific ones at the bottom to generic ones towards the top. For example, the `:friend-of` relation implies that these people `:knows-of` each other. A lower level relationship implies all higher ones in that branch. We deduce this hierarchy from the relationship descriptions. Relations which have no subordinate are instead assigned to an artificial relation `:root`. This ensures that the hierarchy is connected. Second, we define entity subtypes. For example, a `:Person` can be a `:Professional`, or a location can be a `:MedicalFacility`. As with relations, entity types are also ordered in a hierarchy, with the meta-class `:Entity` at the top. We assign subtypes to entities manually from their label and type given in the dataset. Third, we formulate constraints on the subject and object types of each relation. For example, `:doctor-at` is a relation between a `:Professional` and a `:MedicalFacility`. Possible types include entity types from the dataset as well as our own entity subtype definitions. We declare these three extensions in RDFS [2]. Figure 1 shows an examples in the Turtle syntax [1].

2.3 Entity Recognition

2.3.1 From Video. We use different methods for the detection of people and locations from the frames of the videos. In order to detect and identify people, we use an open-source face detection mechanism.[5] Detected faces are then compared against the provided example images in order to identify the visible person. Due to changes in size and orientation of the people on screen, as well as variations of overall image quality throughout the videos, we apply the face detection and identification method densely, i.e., on every frame of the videos. To increase the stability against miss-identification of people, we cluster all detections within the temporal boundaries of a shot spatially and use the most common label for each cluster as a final detection of a person.

```
@prefix : <http://www.example.ai/DVU#> .
@prefix rdfs:
 <http://www.w3.org/2000/01/rdf-schema#> .

# relation hierarchy
:friend-of rdfs:subPropertyOf :knows-of;

# entity subtypes
:Professional rdfs:subClassOf :Person;
:MedicalFacility rdfs:subClassOf :Location;

# relationship constraints
:doctor-at rdfs:range  :Professional .
           rdfs:domain :MedicalFacility;
```

Figure 1: Examples of ontology extensions

For the labelling of locations, we use the last layer of a ResNet152 [15] pre-trained on Imagenet [6] as a feature encoder to extract a semantic representation of both the provided examples as well as all representative key-frames generated during shot-segmentation. The feature representations of the key-frames are then compared against those of the example images. We use a binary nearest neighbor classification scheme for each of the locations in order to assign labels to shots. Since we consider a false negative to be of greater negative impact for the subsequent steps than a false positive, we keep all the location labels in case multiple locations have been detected for a single shot.

2.3.2 From Audio. We build a list of textual surface forms for each entity, consisting of the entity's label, and common abbreviations (e.g. Charlie for Charles). An entity is then detected in text through simple string comparison between any of its surface forms to the transcript (as generated in Section 2.2). We apply the same procedure to all entity types, although concept and location entities rarely exhibit usable labels and therefore remain mostly undetected.

2.4 Relation Estimation

2.4.1 From Video. For the relation estimation process based on visual information, we start with the premise that two people have *a* relation *if and only if* they appear jointly on screen. We can therefore generate the edges of our graph, at least between nodes corresponding to people, solely based on the person-to-shot association determined previously. To predict the labels for these edges, we construct several, admittedly weak predictors, whose output is aggregated in a subsequent step. For this modality, predictions are made based on the appearance of the scene and shared screen time.

[1] https://github.com/vitrivr/cineast
[2] We use the implementation provided by Google for WebRTC
[3] https://github.com/mozilla/DeepSpeech
[4] https://www.microsoftstream.com/
[5] https://github.com/ageitgey/face_recognition

The prediction based on the visual appearance works analogously to the estimation of a shots location described previously. We use all the representative frames of the ground-truth videos where two people with a known relation are visible as positive examples and all others as negative examples for a type of relation and classify all shots of the test videos based on the same ResNet-based features. Due to the large imbalance between positive and negative examples, we again use a nearest neighbor binary classifier. In case there are more than two people in a shot, the relation is predicted for each pair of two people. This results in a large list of predictions which is used in the subsequent aggregation.

2.4.2 From Audio. We extract the text of each shot, and the text between two entity mentions within a narrow time interval from the transcript. On these segments we perform sentiment analysis [8] and compute the mean word2vec embedding [10] of their words.[6]

From these features we train a binary classifier for each relation. We use a logistic regression due to its low number of parameters as the number of training samples is limited. As positive examples we use the segments in which the detected entities from Section 2.3 exhibit the relation in question. The negative examples consist of all other text segments.

3 INFORMATION INTEGRATION AND QUERY PROCESSING

3.1 Information integration

The integration of detected entities and estimated relations follows the hierarchy of relations defined in Section 2.2. We descend this hierarchy from the root, at each relation deciding whether to settle or to continue exploring its most plausible sub-relation.

Specifically, we train a multi-class classifier for each node in the relation hierarchy to decide between the node's own relation and any of its sub-relations. For the former we use all samples of the respective relation, for the latter we include samples of any relation in that branch. At this stage we combine the different modalities by constructing a feature vector from the concatenation of the predictions made by the visual and textual relationship estimators from Section 2.4. Predictions are normalized to the unit interval for each mode separately. To that feature vector we further append the frequency of detections of entity pairs with the relation in question.

To infer edges of a knowledge graph, we first build pairs of entities that were detected in a shot or within a time window in the transcript, then estimate the relation of each pair. For this, we evaluate the classifiers from the root downwards. At each node, we determine the admissible relations, using the constraints defined in Section 2.2. From these relations, we follow the one with the highest predicted likelihood. The procedure repeats until a leaf node has been reached. This is the case if a relation has no more sub-relations, or if a node predicts its own relation. In the case where the procedure stops at the root, we discard the entity pair.

3.2 Query execution

Due to the relatively small sizes of the generated knowledge graphs, we opted not to use any triple store or other graph database – which also saved us the effort of translating the queries from the provided custom XML format to SPARQL [11] – but rather implement the query parsing and execution as a dedicated application.

Table 1: Occurrences of relations within the training set in comparison to the generated graphs

relation	provided	generated
acquaintance of	10	0
ambivalent of	4	0
antagonist of	2	0
attended by	0	239
attends	1	239
bullies	1	0
controlled by	1	0
doctor at	1	0
engages with	7	0
ex-partner	1	0
extended family of	1	0
friend of	25	524
in relationship with	2	0
influences	4	0
is liked by	0	11
knows of	4	0
lives at	0	85
manages	1	0
mentor of	1	0
owns	2	0
parent of	15	0
patient at	2	0
patient of	2	0
religious leader at	1	0
residence of	23	85
responsible for	2	0
sibling of	6	0
socializes at	10	0
spouse of	4	0
studies at	3	0
supervisor of	2	0
teacher at	1	0
teacher of	3	0
works at	2	0
would like to know	1	11

4 INSIGHTS

When comparing the provided graphs with the generated ones, there are some discernible differences. Table 1 shows the number of times a particular relation is present in either set, aggregating over all graphs and omitting relations which were not present in any graph, despite their existence being specified. One can see that the generated graphs have a lower diversity of relation types but have a generally higher number of instances. This can also be seen in Table 2 which shows statistics for all the individual graphs. Not only do the generated graphs have a lower number of different

[6] We use the pre-trained word embeddings from https://code.google.com/archive/p/word2vec/

Table 2: Comparison of several properties of the provided graphs in contrast with the generated ones. The density is computed as the ratio between the edge count over the number of possible edges in a fully connected graph with the same node count.

movie	number of nodes	number of relations	number of edges	edges per node	density
honey	12	6	31	2.58	23.5%
huckleberryFinn	20	19	36	1.80	9.5%
nuclearFamily	6	5	11	1.83	36.7%
spiritual contact	13	13	23	1.77	14.7%
superHero	12	14	25	2.08	18.9%
valkaama	13	6	19	1.46	12.2%
avg. provided	12.67	10.50	24.17	1.92	19.2%
shooter	10	5	88	8.80	97.8%
sophie	22	5	312	14.18	67.5%
theBigSomething	11	3	92	8.36	83.6%
timeExpired	35	7	702	20.06	59.0%
avg. generated	19.5	5	298.5	12.85	77.0%

relations but they also have a higher number of relations per node and are closer to the maximum possible density of a fully-connected graph. This discrepancy in density and diversity of relations can be attributed to the relative sparsity of example relations which makes it difficult to learn a prior probability of a relation. Small perturbations can lead to a relation being under-predicted with respect to the provided graphs, resulting in no or very few predictions, or to over-predict, increasing the graph density.

5 LIMITATIONS AND OPEN CHALLENGES

During the process of trying to solve the tasks of this challenge, we identified several limitations of current approaches as well as some open challenges, which we want so summarize in this section, in order to provide a basis for future activities.

5.1 Training Data

An issue that affects most learning-based systems is the need for a large quantity of training data. In the context of this challenge, many samples would be needed for every entity, entity type, and relation to learn their characteristics. In a video setting an additional challenge is the different domains of movies, which hinders the transferability of relations from one movie to another, and reduces the number of samples per relation. This problem could potentially be overcome by annotating a sufficiently large and diverse video dataset such as [14] with graphs using a consistent ontology.

5.2 Transcription quality

Manual spot-checks of the speech transcription, generated with the workflow described in Section 2.2 revealed that in many instances, the speech-to-text system produced text which was phonetically close to what was said in the video but had an entirely different semantic content. This might be due to the fact that the speech-to-text system was trained on data containing a clean voice signal with little to no background noise or changes in volume, which is not the case for an unfiltered audio signal taken from a movie. The quality of the transcripts generated by the online video platform from Microsoft was, while still not perfect, substantially higher.

5.3 Data Standards

While there are many ways to store a knowledge graph, the set of open standards by the World Wide Web Consortium (W3C[7]) has been shown to be particularly effective for this purpose.

The Resource Description Framework (RDF[8]) provides a means to define knowledge graphs in an intuitive manner. With RDF Schema,[9] constraints on relations can be devised and there exist tools to automatically check the consistency of a graph. Queries against such structures can be formulated in SPARQL query language.[10] As these standards have been implemented in numerous frameworks loading and storing data as well as exchanging data between different parties is greatly facilitated.

Furthermore, there exist public ontologies built on these standards. For example, the FOAF ontology[11] offers definitions classes and relations about people. Since such ontologies are incorporated into public knowledge graphs such as Wikidata,[12] these data sources could be used as background knowledge.

6 CONCLUSION AND OUTLOOK

In this paper, we presented our approaches towards solving the first instance of the Deep Video Understanding Challenge using various weak detectors, aggregated into graphs using some methods common in the semantic web context. This first baseline study identified several limitations of currently available methods and datasets. Based on these first experiences, we see the potential for an increased use of methods and technologies from the semantic web context in the extraction of more intricate semantic information from multimedia documents. Making progress along this path will however require more consistent ontologies as well as a larger amount of annotated training data.

[7] https://www.w3.org/
[8] https://www.w3.org/RDF/
[9] https://www.w3.org/TR/rdf-schema/
[10] https://www.w3.org/TR/sparql11-query/
[11] http://xmlns.com/foaf/spec/
[12] https://www.wikidata.org/

REFERENCES

[1] David Beckett. 2008. Turtle-terse RDF triple language. *http://www. ilrt. bris. ac. uk/discovery/2004/01/turtle/* (2008).

[2] Dan Brickley, Ramanathan V Guha, and Andrew Layman. 1999. Resource description framework (RDF) schema specification. (1999).

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[4] Ronan Collobert, Christian Puhrsch, and Gabriel Synnaeve. 2016. Wav2letter: an end-to-end convnet-based speech recognition system. *arXiv preprint arXiv:1609.03193* (2016).

[5] Keith Curtis, George Awad, Shahzad Rajput, and Ian Soboroff. 2020. HLVU: A New Challenge to Test Deep Understanding of Movies the Way Humans do. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*. 355–361.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[7] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567* (2014).

[8] Clayton J Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Eighth international AAAI conference on weblogs and social media*.

[9] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 1725–1732.

[10] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[11] Eric Prud'hommeaux and Andy Seaborne. 2008. *SPARQL Query Language for RDF*. W3C Recommendation. W3C. http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[12] Joseph Redmon and Ali Farhadi. 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767* (2018).

[13] Luca Rossetto, Ivan Giangreco, and Heiko Schuldt. 2014. Cineast: a multi-feature sketch-based video retrieval engine. In *2014 IEEE International Symposium on Multimedia*. IEEE, 18–23.

[14] Luca Rossetto, Heiko Schuldt, George Awad, and Asad A Butt. 2019. V3C–a research video collection. In *International Conference on Multimedia Modeling*. Springer, 349–360.

[15] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.

[16] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*. 2048–2057.

[17] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. 2018. The sound of pixels. In *Proceedings of the European conference on computer vision (ECCV)*. 570–586.

[18] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. 2017. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 5551–5560.