



ulm university universität
uulm

Ulm University | 89069 Ulm | Germany

**Faculty for Engineering,
Computer Science and
Psychology**
Institute of Databases
and Information Systems
(DBIS)

Video2Metadata: Semantic Metadata Extraction using Automatic Video Captioning

Bachelor Thesis at Ulm University

Presented by:

Johannes Scherer
johannes.scherer@uni-ulm.de
946323

Supervisor:

Prof. Dr.-Ing. Ansgar Scherp

In Collaboration with:

Dr. Deepayan Bhowmik, University of Stirling

2022

Version April 14, 2022

CONTENTS

List of Figures	5
List of Tables	5
List of Acronyms	5
Abstract	6
1 Introduction	6
2 Related Work	7
2.1 Modelling Semantic Metadata for Videos	7
2.2 Computer Vision Methods	8
2.3 Text Information Extraction	8
2.4 Text Classification	8
3 Semantic Metadata Extraction from Videos	9
3.1 Dense Video Captioning	9
3.2 Event Processing	9
3.3 Language Processing	9
3.4 Entity Extraction	11
3.5 Property Extraction	11
3.6 Relation Extraction	11
3.7 Text Classification	13
4 Experimental Apparatus	13
4.1 Datasets	13
4.2 Procedure	14
4.3 Hyperparamters and Hyperparameter Optimization	14
4.4 Measures and Metrics	15
5 Results	16
5.1 Dense Video Captioning	16
5.2 Entity Extraction	16
5.3 Property Extraction	16
5.4 Relation Extraction	16
5.5 Text Classification	18
6 Discussion	18
6.1 Key Results	18
6.2 Threads to Validity	20
6.3 Generalizability	20
6.4 Future Work	21
7 Conclusion	21
References	21
Supplementary Materials	23
A Tags and Dependency Labels	23
B Extended Results	23
C User's Guide	30

LIST OF FIGURES

1	Proposed framework for automatic semantic metadata extraction from videos.	7
2	Example for semantic metadata extraction with our framework.	10
3	Illustration of the processing of captioned events into text.	10
4	Confusion matrix for the classification method when trained with captioned events generated by MT.	27
5	Confusion matrix for the classification method when trained with captioned events generated by PDVC.	28
6	Confusion matrix for the classification method when trained with captioned events from ActivityNet Captions.	29

LIST OF TABLES

1	Categories of YouTube videos and their frequency in our classification dataset.	14
2	Number of different video-level entities in the ActivityNet Captions dataset per entity frequency threshold.	14
3	Event localization and dense video captioning results for different numbers of generated captioned events.	17
4	Experimental results for video-level entity extraction.	17
5	Experimental results for event-level entity extraction.	17
6	Experimental results for property extraction.	18
7	Experimental results for video-level relation extraction.	18
8	Experimental results for event-level relation extraction.	19
9	Experimental results for the classification of videos.	19
10	Parameter combinations for best F1 score performances of our framework.	19
11	Evaluation of the entity, property, and relation extraction methods on manually annotated videos.	20
12	List of Universal Dependencies part-of-speech Tags.	23
13	List of ClearNLP dependency labels.	24
14	Statistics on most frequently extracted video-level semantic metadata by our framework.	25
15	Event localization and dense video captioning results.	26
16	Dense video captioning results per video category.	26
17	Classification performance per video category when using captioned events generated by MT.	27
18	Classification performance per video category when using captioned events generated by PDVC.	28
19	Classification performance per video category when using captioned events from ActivityNet Captions.	29

LIST OF ACRONYMS

BERT Bidirectional Encoder Representations from Transformers.

CNN Convolutional Neural Network.

DVC Dense Video Captioning.

MLP Multi-layer Perceptron.

MT End-to-End Dense Video Captioning with Masked Transformer.

NP Noun Phrase.

Open IE Open Information Extraction.

PDVC End-to-End Dense Video Captioning with Parallel Decoding.

POS Tag Part-of-Speech Tag.

tIoU Temporal Intersection over Union.

Video2Metadata: Semantic Metadata Extraction using Automatic Video Captioning

Johannes Scherer

johannes.scherer@uni-ulm.de

Ulm University

Germany

ABSTRACT

Incorporating metadata is essential to enable users and programs to efficiently search, organize, and re-use data. Semantic metadata is data that describes the meaning of data or its content, thus providing context for interpretation for both, machines and humans. To make the best use of data, in particular multimedia data, it needs to be tagged with relevant semantic information. However, annotation of multimedia data by humans is time-consuming and costly, while reliable automatic generation of semantic metadata is a major challenge. To this end, we propose an end-to-end framework that automatically generates semantic metadata from videos of four different types: entities, their properties, relations between entities, and the video category. Our framework combines an automatic video captioning model, which generates semantic information for a video, with natural language processing methods that further process the output of the video captioning model. We evaluate the performance of our framework for each metadata type, while varying the amount of information the video captioning model provides. Furthermore, we compare the quality of semantic metadata extracted by our framework when it uses different video captioning models.

CCS CONCEPTS

- Computing methodologies → Computer vision tasks; Information extraction.

KEYWORDS

semantic metadata, automatic, dense video captioning

1 INTRODUCTION

With the already existing and constantly rising amount of multimedia data available in digital format, it is desirable to have a standardized description of this data in the form of metadata. Programs and its users use metadata to efficiently search, arrange, and repurpose digital multimedia like image, video, and audio data [13]. One can distinguish between *low-level* and *high-level* metadata. For videos, low-level metadata describes characteristics like color, texture, or shape, while high-level metadata allows descriptions on a conceptual level. This can be the digital rights, content summaries, or related content [5]. The term *semantic metadata* refers to data that describes the "meaning" of data. That means, it describes the content of the data to which it refers, and therefore is high-level. For videos, this can be the occurring objects or a summary of what is seen in the video.

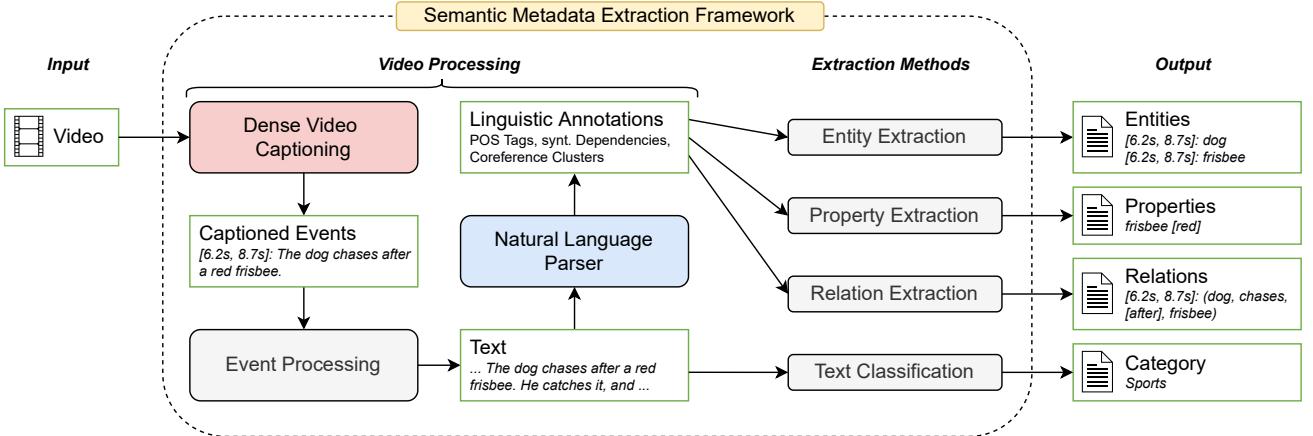
Semantic metadata is utilized in many ways. It provides means for annotating multimedia content with textual information, which enables various applications such as search and retrieval in databases [2]. In contrast, low-level metadata is in most cases not the data that users are interested in. The crucial difference between low-level and semantic or high-level metadata is that low-level metadata can be acquired automatically without user interaction, whereas high-level metadata typically needs to be

annotated manually by the user [5]. However, annotation of multimedia with semantic metadata by humans is time-consuming and costly. Automatic extraction methods exist for different types of high-level metadata, but these methods usually have high error rates and therefore manual correction of the user is still required [26, 34]. Thus, in contrast to the value of semantic metadata, especially when it can be generated automatically, the reliable automatic generation of semantic metadata is still a major challenge. For each semantic metadata type, one could use a different computer vision method to generate the data. For example, video object detection could be used to detect entities in a video, while video visual relation tagging methods find instances of relations between depicted entities. However, when using multiple methods, all of them need to be trained separately and the training is, especially for videos, computationally expensive.

From this motivation, we propose an end-to-end framework that generates semantic metadata from videos of not only one, but multiple types. Depending on the video application, there are various semantic metadata types of interest. We focus on four different of those types, namely the depicted *entities* and their *properties*, the observable *relations* between entities and the video *category*. Additionally, we consider semantic metadata on different levels, namely event-level, where temporal information is relevant, and video-level. Our framework combines several methods from the fields of computer vision and Natural Language Processing (NLP) (see Figure 1). For an input video, a Dense Video Captioning (DVC) model generates a set of natural language sentences for multiple temporally localized video events, thus providing a richly annotated description of video semantics. We process the captioned events into text to make them accessible for different NLP methods. Text classification is used to determine the category of a video using this text, while the extraction methods for the semantic metadata types entities, properties, and relations rely heavily on the linguistic annotations produced by a language parser.

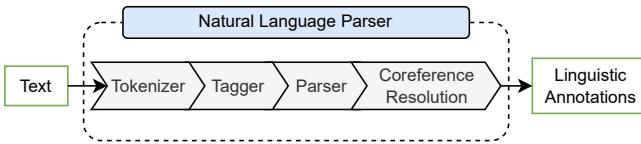
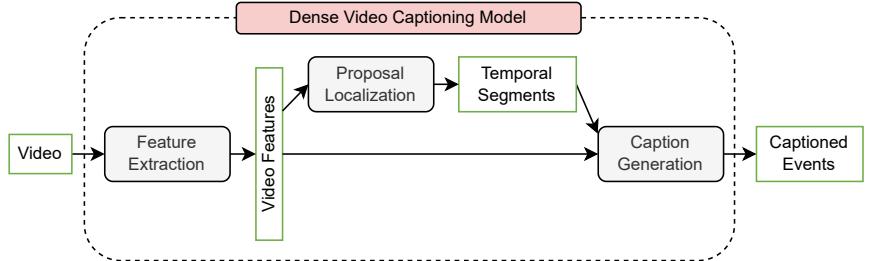
Our work includes the following contributions:

- We present a framework for the automatic extraction of semantic metadata of various types from videos. Our framework combines an automatic video captioning model with several natural language processing methods.
- We provide methods that determine entities, the properties of entities, relations between entities, and the video category using either directly the output of the video captioning model, or corresponding linguistic annotations produced by a language parser.
- We evaluate 1) the capability of our framework to extract semantic metadata, 2) if and how the quality of extracted metadata is affected by the amount of information that is generated by the video captioning model, and 3) whether there are correlations between the performance of the video captioning model and the quality of semantic metadata that is extracted from the textual information it generates.



(a) Our framework combines several computer vision and natural language processing methods. A dense video captioning model captures video semantics, which are processed into text. Furthermore, linguistic annotations are generated by a language parser. For each of the four semantic metadata types, a specific extraction method is used that exploits the processed video semantics.

(b) Dense video captioning models generate natural language sentences for multiple temporally localized video events, i.e., captioned events. Such models often consist of three main components: the video encoder extracts visual features, the proposal decoder generates event proposals, and the captioning decoder generates a caption for each event proposal. Adapted from [40].



(c) In a pipe-lined manner, language parser take in raw text and run a series of natural language annotators on the text. The semantic metadata extraction methods of our framework rely on the part-of-speech tags, syntactic dependencies, and coreference clusters generated by the language parser¹.

Figure 1: Our proposed framework for automatic extraction of semantic metadata from videos (a) and its two main components, a dense video captioning model (b) and a natural language parser (c).

2 RELATED WORK

Visual information of videos can be regarded at several abstraction levels, from low-level to semantic aspects of content. Video data models organize visual information to enable effective retrieval in video databases at different levels of abstractions [7, 14]. We present different video data models that include semantic information and show what types of semantic metadata they consider (Section 2.1). In Section 2.2, we present different computer vision methods, which in a way specialize in the extraction of a single type of semantic metadata. Our extraction methods for the semantic metadata types entities, properties, and relations rely on the linguistic annotations produced by the language parser. In Section 2.3, we present methods from the task of open information extraction that use a similar methodology. Finally, we present related work on text classification (Section 2.4).

2.1 Modelling Semantic Metadata for Videos

Jain and Hampapur [16] presented a data model for videos based on a study of the applications of video, the nature of video retrieval requests, and the features of video. They argue that, unlike text, video data is opaque to computers. Therefore, to enable

content-based access of video, a video database needs to possess some representation, i.e., metadata in addition to the video. In their data model, the basic representational unit for videos are temporal segments. The authors distinguish between different metadata types. Meta features capture content-independent features of a video like the date of production. For content-dependent features, they distinguish between image features (based on a single frame) and video features (based on a time interval). Note that not all content-dependent features are semantic (e.g., image brightness). Among others, the semantic metadata types that their data model considers are name, color, location and structure of objects, spacial activities, subjective properties like intentions and emotions, and the video category.

Due to the explosive growth in the number of online videos, Algur et al. [3] stress the importance of descriptive metadata to enable efficient query-based video retrieval. They state that, for this purpose, the proper genre or category identification of a video is essential. Their proposed data model gives much more importance to descriptive metadata, for which they distinguish between three types. Among others, the video content summary is classified as general descriptive metadata. Event specific metadata enables users to extract specific video parts or events from videos. For an event, this includes its starting and end time, a

¹Figure adapted from <https://spacy.io/usage/processing-pipelines>.

descriptive event summary, and the event category (e.g., discussion or fight). Lastly, object specific metadata stores information about objects like its caption, motion, shape, and color.

The data model of Hacid et al. [14] for content-based video indexing and retrieval considers two types of information: 1) entities that are described with attributes and modelled with relationships among them in the domain of a video sequence, and 2) video frames which contain these entities. Their model explicitly links objects with relations to specify semantics of video data. They argue that, since information about video content is only true at a given time, temporal management of video information is required. For this, they discuss different approaches of how temporal segments are annotated to elements of interest. For example, each element can be associated with a single temporal descriptor. However, they extend this approach by defining temporal cohesion, which allows a set of non-overlapping time segments to be associated with a description. This allows the handling of all occurrences of an entity in a video document with a single object.

The presented video data models offer similar fields for the representation of semantic metadata, most importantly the video category, and entities or objects and their properties like color, location, and shape. Somewhat more specifically, the model of Jain and Hampapur considers subjective properties like emotions. The model of Algur et al. provides fields for video-level and event-level sentence descriptions of content, while the model of Hacid et al. model allows to link entities with relations. It is common to limit the validity of information to a certain interval.

2.2 Computer Vision Methods

For each of the semantic metadata types mentioned in the previous section, one could think of a computer vision method to extract only a certain type. For example, video object detection involves object recognition, that means, identifying objects of different classes, and object tracking, i.e., determining the position and size of an object in subsequent frames [17]. Therefore, an object detection model could be used to determine the entities of a video and the information about when these are visible.

Introduced by Shang et al. [36], video visual relation tagging aims to detect relations between objects in videos. Here, relations are annotated to the whole video without the requirement of object localization. A relation is denoted by a triplet $\langle \text{subject}, \text{predicate}, \text{object} \rangle$, where the predicate may be a transitive or intransitive verb, comparative, or spatial predicate. With visual relation tagging, semantic metadata about the entities presented in a video and how they are related to each other can be generated. According to the authors, this effectively supports various visual relation-based applications such as video retrieval and visual question answering.

Further methods that could be used for the extraction of video semantics include video classification for determining the category of a video [25], and emotion recognition, which aims to classify videos into basic emotions [43]. However, as mentioned above, it is not efficient to use one computer vision method at a time for the extraction of only one semantic metadata type.

Automatic video description involves understanding and detection of different types of information like background scene, humans, objects, human actions, and events like human-object interactions [1]. In such a way, automatic video description can be seen as a task that unites the mentioned computer vision tasks

like object detection, visual relation tagging, and emotion recognition. Dense Video Captioning (DVC), as first introduced by Krishna et al. [19], generate *captioned events*, which not only involves the localization of multiple, potentially overlapping events in time, but also the generation of a natural language sentence description for each event. Because of the rich information DVC models provide, we utilize such model in our framework. We present two DVC models in more detail later on (Section 3.1).

2.3 Text Information Extraction

In our framework, semantic metadata is extracted from the captioned events generated by a DVC model. This includes the analysis of the events' textual descriptions, for which methods from Open Information Extraction (Open IE) can be employed. Open IE is the task of generating a structured representation of the information extracted from a natural language text in the form of relational triples. A triple $\langle \text{arg1}, \text{rel}, \text{arg2} \rangle$ consists of a set of argument phrases and a phrase denoting a semantic relation between them [30]. Existing Open IE approaches make use of a set of patterns, which are either hand-crafted rules or automatically learned from labeled training data. Furthermore, both methodologies can be divided into two subcategories: approaches that use shallow syntactic analysis and approaches that utilize dependency parsing [33].

Fader et al. [11] proposed REVERB, which makes use of hand-crafted extraction rules. They restrict syntactic analysis to part-of-speech tagging and noun phrase chunking, resulting in an efficient extraction for high-confidence propositions. Relations are extracted in two major steps: First, relation phrases are identified that meet syntactic and lexical constraints. Then, for each identified relation phrase, a pair of noun phrase arguments is identified.

Contrary to REVERB, ClausIE (clause-based Open IE) uses hand-crafted extraction rules based on dependency parsing. Introduced by Del Corro and Gemulla [8], ClausIE does not make use of any training data and does not require any postprocessing like filtering out low-precision extractions. Given an input sentence, ClausIE performs the following steps. First, the dependency parse of the sentence is computed. Then, using the dependency parse, a set of clauses is determined. The authors define seven clause types, where each clause consists of one subject, one verb and optionally of an indirect object, a direct object, a complement, and one or more adverbials. In the third step, for each clause, a set of coherent clauses is derived based on the dependency parse and domain-independent lexica. Finally, for each clause, one or more propositions are generated. Since dependency parsing is used, ClausIE is computationally more expensive compared to REVERB. However, it achieves improved precision and recall.

2.4 Text Classification

As mentioned in Section 2.1, Algur et al. [3] state that the proper category identification of a video is essential for efficient query-based video retrieval. This task is traditionally posed as a supervised classification of the features derived from a video. The features used for video classification can be of visual nature only, but if user-provided textual metadata (i.e., title, description, tags) is available, it can be used in a profitable way [4]. However, in our proposed framework the video category is predicted only with the textual information the DVC model provides. So although we address the video classification problem, we do this by utilizing existing work in the text classification area. Text classification

models can be roughly divided into two categories [23]. Traditional statistics-based models such as k-Nearest Neighbors and Support Vector Machines require manual feature engineering. On the other hand, deep learning models consist of artificial neural networks to automatically learn high-level features for better results in text understanding. Here, among the most popular is TextCNN [18], which solves the text classification problem using a Convolutional Neural Network (CNN). Also, pre-trained BERT (Bidirectional Encoder Representations from Transformers) [10] models can be fine-tuned for state-of-the-art text classification results [37]. However, Galke and Scherp [12] argue that simple Multi-layer Perceptrons (MLP) are overlooked as strong baselines for the topical categorization of text. They showed that MLPs are oftentimes on-par or even better than recently proposed models that synthesize a graph structure from the text.

3 SEMANTIC METADATA EXTRACTION FROM VIDEOS

Section 2.1 gives an overview over the various types of semantic metadata that video data models consider. In our framework, however, the extraction of semantic metadata is based only on the captioned events generated by its DVC model. As a result, certain metadata types such as emotions are difficult to extract. This depends mainly on how detailed the DVC model is able to describe video semantics. Considering the capabilities of current video description models, in this work we focus on four semantic metadata types that we aim to extract from a video: the depicted **entities** (i.e., persons, objects, locations) and their **properties**, observable visual **relations** between entities, and the video **category** (see Figure 2). We distinguish between *event-level* and *video-level* semantic metadata depending on whether semantic metadata is assigned to a specific time interval or not. For example, assume that at some point in a video there is a *cat* visible. On video-level, the corresponding entity item only stores the information that there is a cat occurring in the video. On event-level, the metadata item does not only store the name of the entity, but also a time interval in which the entity is visible. Ultimately, this means that event-level semantic metadata refers to exactly one captioned event generated by the framework’s DVC model. In contrary, video-level semantic metadata is extracted from information provided by all video events.

Revisiting Figure 1, it can be seen that our framework consists of several methods. For an input video, a DVC model generates captioned events (Section 3.1), which are then processed into text to make them accessible for different methods (Section 3.2). The natural language parser produces linguistic annotations, namely part-of-speech (POS) tags, a dependency parse and coreference clusters (Section 3.3). The extraction methods for the semantic metadata types entities (Section 3.4), properties (Section 3.5), and relations (Section 3.6) use these linguistic annotations, while the text classification method determines the category of a video using only the processed captioned events (Section 3.7). The lexical database WordNet [29] is used at various points to ensure that extracted semantic metadata consists of linguistically correct English nouns, verbs, adjectives, and adverbs.

3.1 Dense Video Captioning

Dense video captioning is a multi-task problem that involves detecting multiple individual events in a video and understanding of their context. From an input video, DVC models generate a set of captioned events. Each captioned event consists of the

event itself, a temporal segment which potentially overlaps with segments of other captioned events, and a natural language sentence that captions the event. While introducing the task of DVC, Krishna et al. [19] proposed a model which consists of a proposal module for event localization, and a separate captioning module, an attention-based Long Short-Term Memory network for context-aware caption generation.

Zhou et al. [46] argue that the model of Krishna et al. is not able to take advantage of language to benefit the event proposal module. To this end, they proposed an end-to-end DVC model with masked transformer (MT) that is able to produce event proposals and event descriptions simultaneously. Like many methods that tackle the DVC task, Zhou et al.’s model consists of mainly three components (see Figure 1b). The video encoder, composed of multiple self-attention layers, extracts visual features from video frames. The proposal decoder takes the features from the encoder and produces event proposals, i.e., temporal segments. The captioning decoder takes input from both, the visual encoder and the proposal decoder, and finally captions each event proposal.

Wang et al. [40] state that methods like the model of Zhou et al. follow a two-stage “localize-then-describe” scheme, which heavily relies on hand-crafted components. Instead, not following the usual structure of DVC models depicted in Figure 1b, they proposed a simpler framework for end-to-end DVC with parallel decoding (PDVC). Their model directly decodes extracted frame features into a captioned event set by applying two parallel prediction heads: localization head and captioning head. Furthermore, they proposed an event counter which is stacked on the top of the decoder to predict the number of final events. They claim that, in such way, their model is able to precisely segment the video into a number of events, avoiding to miss semantic information as well as avoiding replicated caption generation.

We implement and evaluate our framework for semantic metadata extraction from videos using the two described models: MT and PDVC. To summarize, within our framework, the DVC model is responsible for generating captioned events, each consisting of a temporal segment and a sentence describing the event, which are then forwarded to the event processing module.

3.2 Event Processing

The event processing module processes the captioned events generated by the preceding DVC model into text in order to make semantic information accessible to the natural language parser and the text classification method. In detail, the sentences of the captioned events are sorted in ascending order of the start times of the corresponding events. Afterwards, the sentences are concatenated, resulting in a single text per video, and forwarded to the language parser and text classification method, respectively. By not passing the sentences separately to the language parser, this enables it to use coreference resolution (see Section 3.3). When extracting entities and relations on event-level, we annotate each entity and relation with the temporal segment of the captioned event whose sentence contains the name of the entity or the words of the relation, respectively (see Figure 3).

3.3 Language Processing

The extraction of semantic metadata of the types entities, properties, and relations is done through syntactic analysis based on the linguistic annotations generated by the natural language parser of our framework. More precisely, the language parser is required to provide POS tags of tokens, a dependency parse, and

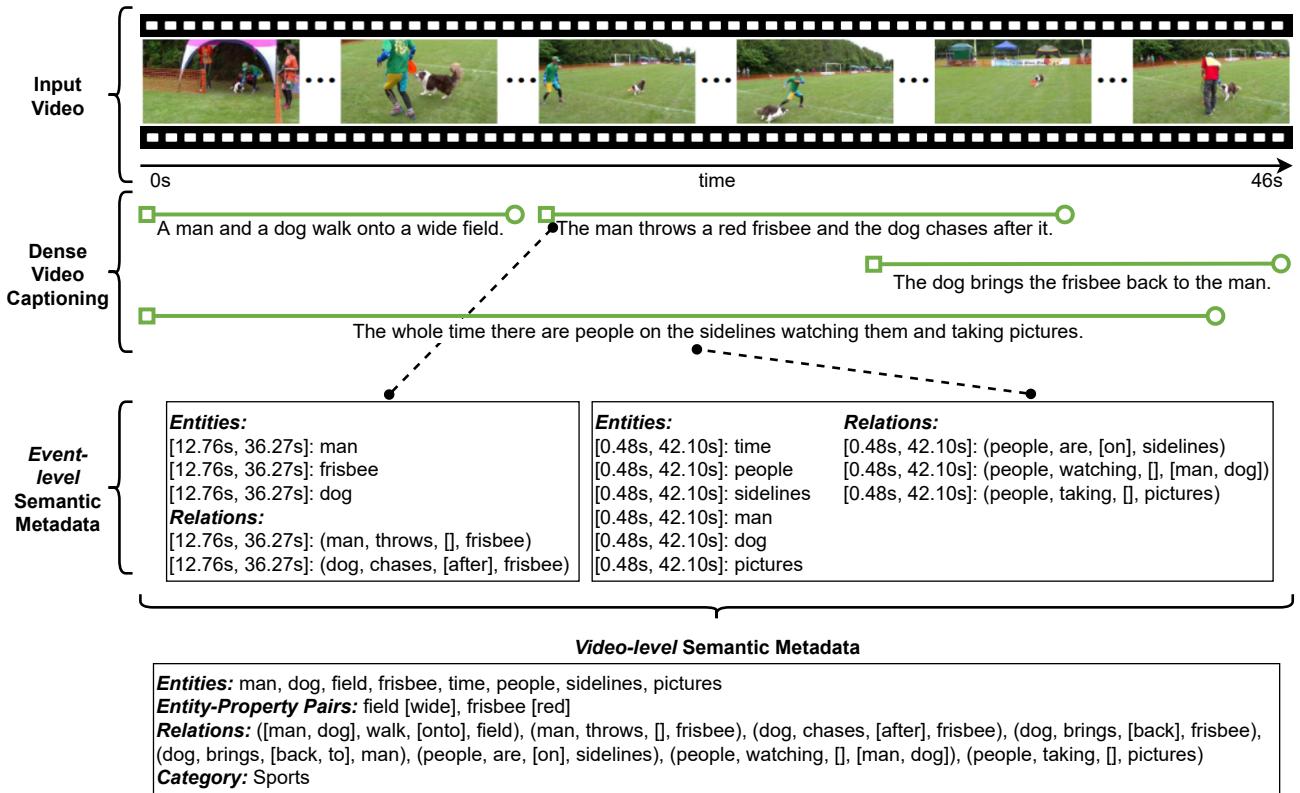


Figure 2: Our framework extracts semantic metadata of the types entities, properties, relations, and the video category from automatically generated dense video captioning events. We distinguish between event-level and video-level semantic metadata, depending on whether semantic metadata is assigned to a specific time interval or not. Image adapted from [24].

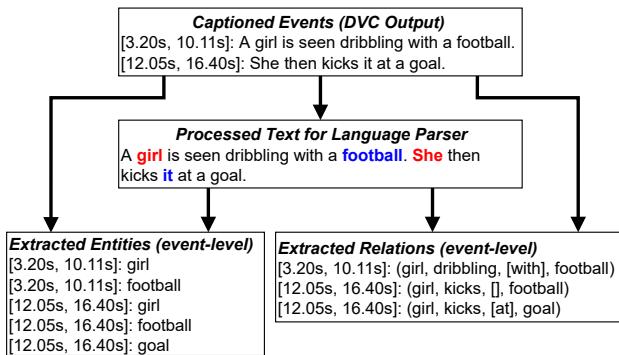


Figure 3: Captioned events are processed into text to make their semantic information accessible to the language parser and text classifier. This also allows the language parser to use coreference resolution.

coreference clusters. For a sentence token (i.e., word), the POS tag is a label assigned to the token to indicate its part of speech. The dependency parse consists of a set of directed syntactic relations between the tokens of the sentence. Coreference clusters are the result of coreference resolution, which aims to find all language expressions that refer to the same entity in a text.

When implementing our framework (see Section 4), we use a CNN based model from spaCy², an open-source natural language processing library. The model generates the desired linguistic

²spaCy is available at: <https://github.com/explosion/spaCy>.

annotations in a pipe-lined manner (see Fig. 1c). First, the tokenizer segments the input text into tokens. Afterwards, the tagger and dependency parser assign POS tags and dependency labels to tokens, respectively. Finally, coreference clusters are determined. For this, we use NeuralCoref³, a pipeline extension for spaCy, which performs coreference resolution. NeuralCoref uses a neural network, which adapts the scoring model of Clark and Manning [6], to rank and determine pairs of mentions. The resulting coreference clusters are sets, where each set contains all mentions that are co-referring to the same entity.

For an input sentence, the dependency parse produced by spaCy models is a tree where the head of a sentence, which is usually a verb, has no dependency. Every other token of the dependency tree has a dependency label that indicates its syntactic relation to its *parent*. The *children* of a token are all its immediate syntactic dependents, i.e., the tokens of the dependency tree for which it is the parent.

In the following sections, when explaining how the semantic metadata extraction methods use linguistic annotations, we use the Universal Dependencies POS tags⁴ [31]. For dependencies however, we use the ClearNLP dependency labels⁵ due to the fact that spaCy makes use of them. Note that the ClearNLP dependency labels are slightly different from the syntactic relations of Universal Dependencies. Table 12 and Table 13 list and explain the POS tags and dependency labels that are used in this work.

³NeuralCoref is available at: <https://github.com/huggingface/neuralcoref>.

⁴Universal Dependencies POS tags: <https://universaldependencies.org/u/pos/>.

⁵ClearNLP dependency labels: https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md.

3.4 Entity Extraction

The entity extraction method determines which (video-level) and when (event-level) entities like persons, objects and locations are visible in the video. A **video-level entity** only consists of a *name* describing the entity. On **event-level**, the entity additionally consists of a *temporal segment* containing the information about when the entity is seen. Entities are extracted by determining (compound) nouns with the POS tags and dependency labels of tokens, and coreference clusters, which are all provided by the language parser.

First, compound nouns are detected, i.e., nouns that consist of more than one word token. Candidates for compound nouns are lists of tokens. Such list is formed for a token that is not marked with the compound dependency, when it has a child (see Sec. 3.3) marked as compound. The candidate list then consists of the token itself and all recursively found children that are marked as compound. Now, each candidate list is checked whether it consists of a compound noun. For this to hold, the candidate list needs to have at least one token that is a NOUN or PROPN. While using the tokens in order of occurrence in the corresponding sentence, it is checked if the concatenation of the words using either "_", "-" or an empty string is an open, hyphenated, or regular compound noun, respectively. This is the case when the formed word or its lemma is a WordNet noun. If no compound noun is found, then the procedure repeats with the two sublists that do not contain the first or the last token of the candidate list. This recursive procedure ends either when a compound noun is found or no compound noun can be formed anymore. In such way, not only NOUN+NOUN (e.g., *living_room*) compounds are found, but also other combinations like ADJ+NOUN (*plastic_bags*), NOUN+VERB (*mountain_climbing*), or PROPN+PROPN+PROPN (*new_york_city*). The determination of the remaining nouns, i.e., nouns consisting of one token, is simple. For this, only tokens which are not part of a detected compound noun are used. Such token is a noun, if the language parser marked it as a NOUN or PROPN, and WordNet recognizes the token or its lemma as noun.

On video-level, the method is finished already. Here, for each noun from the set of all detected nouns an entity is created with the noun as the entity's name. On event-level, however, before assigning a temporal segment to each name in order to obtain the entities, each pronoun is analysed whether it refers to a noun or not (see Figure 3). First, the tokens of the text are filtered for pronouns, i.e., for tokens having the PRON tag. Using the computed coreference clusters, for each pronoun it is checked whether it refers to a noun that has been determined in the previous step or not. If this is the case, then the pronoun is replaced with the corresponding noun, i.e., the name of a new entity. Finally, the event-level entities are built by assigning temporal segments to the names of the entities. Here, for an entity, the temporal segment is the segment of the corresponding sentence in which its name (or the pronoun that was previously replaced) occurs.

3.5 Property Extraction

The property extraction method determines properties of entities such as color, size, and shape. The method is only used to extract video-level information in order to collect information for an entity from different captioned events. An **entity-property pair** is a tuple consisting of an *entity* (i.e., its name) and a *property*, which further describes the entity linguistically.

Coreference clusters are unimportant for the property extraction method. Therefore, only the tokens' POS tags and dependencies, computed by the language parser, are utilized here. For each (video-level) entity, extracted using the method described in the previous section, properties are determined as follows. Using the dependency labels of tokens, the candidate properties for an entity are the children of the corresponding token (or tokens for compound nouns) that are marked with the *amod* dependency. If WordNet recognizes the token or its lemma as an adjective, then such candidate property is considered as a property of the entity. In this case, an entity-property pair is formed with the name of the entity and the detected property. In such way, the method results in a set of video-level entity-property pairs. There is no restriction on the tag a property token needs to have. Therefore, properties may be marked by the language parser as ADJ (e.g., *round ball*), VERB (e.g., *provoking film*), or other tags.

3.6 Relation Extraction

As seen for visual relation tagging (Section 2.2) or Open IE (Section 2.3), relations are usually formulated as triples. In this work, however, we define relations as follows. A **video-level relation** is a 4-tuple of the form (*subjects*, *verb*, *modifiers*, *objects*). With a *temporal segment*, an **event-level relation** has an additional element containing the information about when the relation is observed. The first element *subjects* is a list containing the names of the relation's acting entities. *Objects*, also a list of names of entities, contains the entities that are the receiver of the action. Note that usually subjects and objects contain only a single entity. In some cases, however, for example in the sentence "*A boy and a girl are seen playing football.*", multiple entities are the actors in a relation: (*[boy, girl], play, [], football*). With the above definition we are able to capture relations with different types of verbs: single-word verbs and multi-word verbs, i.e., prepositional verbs (verb+preposition), phrasal verbs (verb+particle), and phrasal-prepositional verbs. In both cases, *verb* contains the verb. *Modifiers* is an empty list for single-word verbs. For multi-word verbs, however, *modifiers* contains the verb's particles and prepositions. Both, particles and prepositions, provide information about how the verb and the object are related to each other. For example, the relation (*girl, catches, [up, with], kids*) for the sentence "*The girl catches up with the other kids.*" is better understood than the relation (*girl, catches, kids*).

As for entity extraction, the relation extraction method utilizes POS tags and dependency labels of tokens and coreference clusters. In fact, the entity extraction method is used here in order to determine valid subjects and objects for relations. The relation extraction method proceeds in three steps: 1) search for candidate verbs, 2) search for candidate tuples consisting of a subject and a verb, and 3) search for corresponding objects and modifiers for each verb in a candidate tuple.

First, candidate verbs are determined using Algorithm 1. For the above definition of relations to make sense, passive verbs are ignored. Verb tokens are tagged either as VERB or AUX (1), whereby passive verbs have at least one child token that has either the *nsubjpass*, *auxpass*, or *agent* dependency (2). The candidate verbs are obtained using those characteristics. Also, a candidate verb or its lemma needs to be included in WordNet to be considered as such (3).

Now, for each candidate verb a subject is determined using Algorithm 2, resulting in candidate tuples. At this stage, a subject can either be an entity or pronoun. Using coreference resolution,

Algorithm 1: Determine non-passive verbs of a token list.

Input: Text tokens $T = [t_1, \dots]$
Output: List $V = [v_1, \dots]$ of verbs v_i
Constants: $Tags = [\text{VERB}, \text{AUX}]$ (indicate verbs),
 $Deps = [\text{nsubjpass}, \text{auxpass}, \text{agent}]$
 (indicate passive verbs)

```
1 Function find_verbs( $T$ ):  
2    $V = []$   
3   for  $t$  in  $T$  do  
4     // (1) Search for verbs  
5     if  $\text{tag}(t) \in Tags$  then  
6       // (2) Ignore passive verbs  
7       for  $c \in \text{child\_tokens}(t)$  do  
8         if  $\text{dep}(c) \in Deps$   
9           then continue with next  $t$   
10      // (3) Verb included in WordNet?  
11      if  $\text{is\_wordnet\_verb}(t)$  then  
12         $V.append(t)$   
13  return  $V$ 
```

a pronoun can later be resolved into an entity, thus creating a valid relation. A distinction is made between three different cases of how subjects can be found. A nominal subject is a child of its corresponding verb, annotated with the nsubj dependency and therefore is easily determined (1). If the verb is a participle (e.g., gerund, past participle) and its parent is an entity or pronoun, then the parent token is returned as subject (2). When with both previous cases no subject is found, then a subroutine (referred to as `find_subject_of_parent()` in Algorithm 2) is used with the verb's parent token as input to recursively search for a subject (3). For a token, the subroutine checks if the token itself or any child of it is a subject. Here, a token is considered to be a subject when it has the nsubj or nsubjpass dependency and when it is an entity or pronoun. If no subject is found, the subroutine is called with the parent of the token as input. This procedure terminates and returns no subject if the root of the dependency tree is reached and no subject was found. Although passive verbs are ignored for relations, as mentioned, the subroutine searches for subjects also using the nsubjpass dependency. This is because a subject of a passive verb can act as a subject for a non-passive verb later in the sentence (e.g., "A child is seen riding a skateboard.").

Algorithm 2: Identify a subject for an input verb.

Input: Verb v , sentence entities E
Output: Subject of v

```
1 Function find_subject( $v, E$ ):  
2   // (1) Nominal subjects  
3   for  $c \in \text{child\_tokens}(v)$  do  
4     if  $\text{dep}(c) = \text{nsubj}$  and ( $c \in E$  or  $\text{tag}(c) = \text{PRON}$ )  
5       then return  $c$   
6   // (2) Verb is a participle  
7    $p = \text{parent\_token}(v)$   
8   if  $\text{dep}(v) = \text{acl}$  and ( $p \in E$  or  $\text{tag}(p) = \text{PRON}$ ) then  
9     return  $p$   
10  // (3) Recursively search for a subject  
11    using the parent token  
12  return find_subject_of_parent( $p, E$ )
```

In the third step, for each verb in a candidate tuple, Algorithm 3 is used to determine the verb's corresponding objects and modifiers. As for subjects, at this stage an object can be either an entity or pronoun. Objects are found in two different ways. A direct object is a child of its corresponding verb, annotated with the dobj dependency and therefore is easily determined (1). Objects of prepositions, which are annotated with the pobj dependency, are not as easily determined (2). This is because objects of prepositions are not direct children of the corresponding verbs in the dependency tree. In the dependency tree, on the path between an object of preposition and its verb, there is usually one, and sometimes multiple tokens. Therefore, a recursive subroutine (referred to as `find_objects_of_prepositions()` in Algorithm 3) is used to find objects of prepositions for a verb. Starting with the verb, the subroutine searches recursively for objects of prepositions by looking at child tokens. If an object of preposition is found, i.e., a token which is an entity or pronoun annotated with the pobj dependency, then the object is returned. In addition to the object, all tokens on the path between the verb and the object are returned. These tokens are used as modifiers and therefore will be included in the corresponding element of the relation. Modifiers between the verb and an object of preposition in the dependency tree cannot be of arbitrary nature. For a relation, we limit those to tokens that have a dependency and tag combination of [prep, ADP], [prt, ADP], [advmod, ADV], [conj, ADV], or [conj, ADP]. If a coordinating conjunction, i.e., a modifier with the conj dependency, is on the path between the verb and the object, then the relation is split up into two relations: one relation without the coordinating conjunction in the modifiers list, and one without its parent token. In addition, each modifier tagged as ADV needs to be included in WordNet as adverb. Since WordNet does not list prepositions, each modifier tagged as ADP is validated with a custom list of prepositions. Finally, the verb's leaf modifiers are determined, i.e., modifiers of the verb that do not have any children (3). These are limited to tokens that have a dependency and tag combination of [prt, ADP] or [prep, ADP].

Algorithm 3: Identify corresponding objects and modifiers for an input verb.

Input: Verb v , sentence entities E
Output: Tuple-list $T = [t_1, \dots]$, where a tuple
 $t_i = (o_i, m_i)$ contains an object o_i and modifiers
 $m_i = [m_{i,1}, \dots]$ for verb v

```
1 Function find_objects( $v, E$ ):  
2    $T = []$   
3   // (1) Verb has direct object(s)  
4   for  $c \in \text{child\_tokens}(v)$  do  
5     if  $\text{dep}(c) = \text{dobj}$  and ( $c \in E$  or  $\text{tag}(c) = \text{PRON}$ )  
6       then  $T.append((c, []))$   
7   // (2) Verb has object(s) of preposition(s)  
8   for  $t$  in find_objects_of_prepositions( $v, E$ ) do  
9      $T.append(t)$   
10  // (3) Search for leaf verb modifiers  
11  for  $c \in \text{child\_tokens}(v)$  do  
12    if ( $\text{dep}(c), \text{tag}(c) \in [(\text{prt}, \text{ADP}), (\text{prep}, \text{ADP})]$   
13      and  $\text{child\_tokens}(c) = \emptyset$ ) then  
14      for  $t = (o, m)$  in  $T$  do  
15         $m.append(c)$   
16  return  $T$ 
```

This almost completes the relation extraction method. So far, the subjects and objects lists of a relation each contain only a single entity or pronoun, although multiple subjects or objects may exist for it. For example, in addition to an already determined subject (i.e., the head of a noun phrase), another entity can be an actor of a relation. In this case, such subject is found by looking at the child tokens of the already determined subject. If a child token is either an entity or pronoun and is annotated with the `conj` dependency, then it is also considered a subject.

The subjects and objects lists of a definitive relation should only contain entities. However, pronouns have also been taken into account so far. As with the entity extraction method, each pronoun is checked whether it refers to an entity or not by using coreference resolution. When this is the case, then the pronoun is replaced with the corresponding entity. If for a pronoun no related entity is found, then it is simply removed from the relation’s corresponding subject or object list. If a relation does not have at least one subject or object anymore, then the relation is ignored and not returned as one of the videos relations.

The video-level relation extraction is finished here. Event-level relations are built by assigning temporal segments to the relations. Here, for a determined relation, the temporal segment is the segment of the corresponding sentence from which the relation was extracted.

3.7 Text Classification

Recall that, although our proposed framework performs video classification, it predicts the category of a video only with the textual information its DVC model provides. Therefore, the framework’s method for video category prediction is a text classification model. For this, we decided to combine a BERT (see Section 2.4) model with a classifier, aiming to reliably categorize an input video. As seen in Section 3.2, the captioned events generated by the DVC model are processed into text. Before forwarding the text to the BERT model, its corresponding preprocessing model transforms the text to numeric token ids. The BERT model then converts the preprocessed text into embedding vectors. Those are fed into the classifier, a simple neural network consisting of a dropout layer and a dense layer with as many nodes as there are categories.

4 EXPERIMENTAL APPARATUS

We run experiments for videos that are included in the ActivityNet Captions [19] dataset. We evaluate each of our semantic metadata extraction methods separately by collecting multiple metrics and compare results for our framework in two settings, i.e., while using two different DVC models. With MT⁶ (End-to-End Dense Video Captioning with Masked Transformer) [46] and PDVC⁷ (End-to-End Dense Video Captioning with Parallel Decoding) [40], we use two different DVC models that provide high-quality captioning results and precede all other methods of our framework. We implemented our framework in Python, using spaCy² as the language parser of our choice and TensorFlow⁸ for the implementation of the text classification method. Within spaCy, we are bound to use version 2.3.7 and the `en_core_web_lg-2.3.1` language model⁹. This is because NeuralCoref³, a pipeline extension for spaCy which we use for

⁶MT is available at: <https://github.com/salesforce/densecap>.

⁷PDVC is available at: <https://github.com/ttengwang/PDVC>.

⁸TensorFlow is available at: <https://github.com/tensorflow/tensorflow>.

⁹The spaCy model that we use is available at: https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-2.3.1.

coreference resolution, is not compatible with versions 3.0.0+ yet. The language model achieves accuracies for fine grained POS tags of 97.22 and labeled dependencies of 90.28. As stated before, WordNet [29] is used at various points for lemmatizing words and to ensure that extracted semantic metadata consists of linguistically correct English nouns, verbs, adjectives, and adverbs¹⁰. In addition to that, WordNet is used for the evaluation of semantic metadata extraction methods to compare words in terms of whether they are synonyms of each other or not.

4.1 Datasets

4.1.1 Dense Video Captioning. We use the large-scale benchmark dataset ActivityNet Captions¹¹ [19] to train and evaluate the underlying video captioning models. ActivityNet Captions consists of 20k YouTube videos of various human activities split up into train/val/test sets of 0.5/0.25/0.25, with the test set currently being withheld. Each video is annotated with captioned events, each consisting of a descriptive sentence and a specific temporal segment to which the description refers. On average, each video is annotated with 3.65 temporally-localized sentences. Each captioned event covers 36 seconds and composes 13.5 words on average. Temporal segments referring to the same video can overlap in time, which enables DVC models to learn complex matters. Note that ActivityNet Captions consists of two different annotation files for the validation set, `val_1.json` and `val_2.json`, while most times in this work we only use the first of the two annotation files. In this work, we make it clear at all times which annotation files of the ActivityNet Captions dataset validation set we are using.

4.1.2 Entity, Property, and Relation Extraction. The information that ActivityNet Captions provides for each video is limited to captioned events, i.e., pairs of temporal segments and sentences. To be able to evaluate the entity, property, and relation extraction methods, we need information about depicted entities, properties of entities, and relations of videos. For this purpose, we utilize the gold standard captioned events of ActivityNet Captions validation videos (using only the first of the two annotation files), use our proposed entity extraction, property extraction, and relation extraction methods on the processed captioned events and treat the results as gold standards for semantic metadata. This procedure is discussed further in Section 6.2. In such way, we generate five different gold standards for videos in the ActivityNet Captions validation set: one each for event-level entities, video-level entities, entity-property pairs, event-level relations, and video-level relations. Using these generated gold standards, we are then able to evaluate the framework’s capability to extract various types of semantic metadata and to compare the quality of the semantic metadata extracted from captioned events generated by different DVC models.

4.1.3 Text Classification. We build a new dataset to train and evaluate our text classification method. Here, we take advantage of the fact that ActivityNet Captions consists of YouTube videos. For each video of the ActivityNet Captions train and validation sets, we query the corresponding category from the YouTube Data API. We were able to query the category of 12,579 ActivityNet Captions videos (on March 20, 2022). The remaining videos

¹⁰When using the WordNet lemmatizer, we make exceptions for few nouns and verbs which WordNet does not lemmatize as expected. Example: WordNet lemmatizes `men` to `men`. We make a hard-coded exception and lemmatize `men` to `man`.

¹¹ActivityNet Captions is available at: <https://cs.stanford.edu/people/ranjaykrishna/densevid/>.

ID	Category	# videos	%
1	Film & Animation	213	1.69
2	Cars & Vehicles	191	1.52
10	Music	637	5.06
15	Pets & Animals	342	2.72
17	Sport	3,920	31.16
19	Travel & Events	526	4.18
20	Gaming	78	0.62
22	People & Blogs	2,098	16.68
23	Comedy	645	5.13
24	Entertainment	1,414	11.24
25	News & Politics	226	1.80
26	How-to & Style	1,661	13.20
27	Education	414	3.29
28	Science & Technology	146	1.16
29	Non-profits & Activism	68	0.54
		12,579	100%

Table 1: Categories of YouTube videos and their frequency in our classification dataset.

are no longer (publicly) available on YouTube. We split the videos up into train/val/test splits of 0.6/0.2/0.2, while ensuring that the category distribution is the same for all splits. An overview of the YouTube video categories and their frequencies in our dataset is shown in Table 1.

4.2 Procedure

4.2.1 Dense Video Captioning. We train MT and PDVC using the ActivityNet Captions dataset. For action recognition, both models adopt the same action recognition network, a pre-trained temporal segment network [39], to extract frame-level features. To ensure consistency, we evaluate both models on the ActivityNet Captions validation set (using both annotation files of the validation set) and compare the performances with those reported in the corresponding works.

For our proposed semantic metadata extraction methods, the number of captioned events forwarded to each method is an important parameter. With increasing number of captioned events forwarded to an extraction method, it can extract more semantic information. In this work, we denote the number of captioned events that are generated by a DVC model and forwarded to a specific method as $|E|$. MT and PDVC, the dense video captioning models of our choices, internally calculate confidence scores for their generated captioned events. If the number of considered captioned events is limited, then the generated captioned events with highest best confident scores are used. For high values of $|E|$, DVC models tend to produce identical sentences for different temporal segments. However, our property extraction, and video-level entity and relation extraction methods rely mainly on the textual information that the captioned events provide. Therefore, captioned events with duplicate sentences do not provide further semantic information. Because of that, we only forward events with distinct captions to these methods. We denote the number of distinct captioned events that are forwarded to a specific method as $|dist(E)|$. If two events share the exact same caption, then the event with the higher confidence score is forwarded. We finally evaluate MT and PDVC with $|E|$ set to 10, 25, 50, and 100, and $|dist(E)|$ set to 1, 3, 10, and 25.

ActivityNet Captions	Entity frequency threshold f			
	0	10	25	50
# Entities	5,428	1,358	794	468

Table 2: Number of different video-level entities in the ActivityNet Captions dataset per entity frequency threshold. The frequency of an entity is the number of different ActivityNet Captions videos in which it occurs.

4.2.2 Entity, Property, and Relation Extraction. For both trained dense video captioning models, MT and PDVC, we extract event-level entities and relations from captioned events that they generate for ActivityNet Captions validation videos with $|E|$ set to 10, 25, 50, and 100. Video-level entities and relations, and entity-property pairs are extracted with $|dist(E)|$ set to 1, 3, 10, and 25. We then evaluate the entity, property, and relation extraction methods by comparing the extracted semantic metadata with our generated gold standards for entities, entity-property pairs, and relations.

For the evaluation of video-level entity extraction, we introduce the entity frequency threshold f . Using the corresponding annotation files, we extract video-level entities from ActivityNet Captions train and validation videos. Then, the frequency of a video-level entity is the number of different videos in which it occurs. We then evaluate the generated video-level entities with f set to 0, 10, 25, and 50, meaning that the gold standard only contains those entities which have a frequency higher than the frequency threshold. Using a higher f means that the gold standard only contains entities that are occurring at least multiple times, which gives the DVC models a fairer chance to learn and reproduce the entities of videos (see Table 2).

4.2.3 Text Classification. We train and evaluate the text classification model of our framework in three different settings: using the captioned events generated by the MT and PDVC model, respectively, and using the gold standard captioned events provided by ActivityNet Captions (using only the first of the two annotation files for validation videos). This allows us to analyse how useful the automatically generated captioned events are for text classification compared to gold standard captioned events. For each data input, we set $|dist(E)|$ to 10. Note that the text classification model is also trained and evaluated on videos, for which the DVC models were already trained on. This is because our dataset for classification contains train and validation videos of ActivityNet Captions, which are randomly split up into train/val/test sets. The reason we also include ActivityNet Captions train videos into the classification dataset is to ensure that there are enough videos for all video categories to train, validate and test the text classification method on.

4.3 Hyperparamters and Hyperparameter Optimization

4.3.1 Dense Video Captioning. For the training of the DVC models, we use largely the same parameters that were used to train the models in their original works (refer to MT⁶ [46] and PDVC⁷ [40] or their latest codebases). For MT, we have to switch to a batch size of 84 and a learning rate of 0.06 to make sure that the model training converges. In the default configuration, PDVC generates a maximum of 10 events per video. We change the corresponding parameter such that 100 captioned events are generated to

ensure that there is enough information for our metadata extraction methods to work on. For both, MT and PDVC, we use the models' best states with respect to their achieved METEOR score performances (see Section 4.4). For MT, this was achieved after 34 epochs, and for PDVC after 13 epochs.

4.3.2 Entity, Property, and Relation Extraction. For the entity, property, and relation extraction methods, no hyperparameter optimization is necessary.

4.3.3 Text Classification. In our classification model, we use an uncased BERT model consisting of 12 transformer blocks, a hidden size of 768, and 12 attention heads¹². Each input text is truncated to 128 tokens. For training, we use cross-entropy loss and the Adam optimizer with 10% warmup steps. We use class weights to help the model learn on the imbalanced data of our classification dataset (see Table 1). While using captioned events from ActivityNet Captions as training data, we perform grid search over dropout rates in $\{0.0, 0.1, \dots, 0.5\}$, maximum number of training epochs in $\{1, 2, \dots, 5\}$, learning rates between $5e-6$ and $1e-4$, and a batch sizes in $\{1, 2, 4, 8\}$. The lowest validation loss was achieved after epoch 2 while using dropout of 0.1, 2 maximum training epochs, learning rate of $2e-5$, and batch size of 4. Using this parameter configuration, we train the model with each data input separately. While repeating the training three times for each data input, we select the model that achieved the lowest final validation loss.

4.4 Measures and Metrics

4.4.1 Dense Video Captioning. To evaluate the dense video captioning models MT and PDVC, we use the official evaluation toolkits provided by ActivityNet Captions Challenge 2017¹³ and ActivityNet Captions Challenge 2018¹⁴, which measure the capability to localize and describe video events. The reason we use these two toolkits is that MT and PDVC were originally evaluated using the 2017 and the 2018 toolkit, respectively, although the calculation of scores has changed from 2017 to 2018¹⁵. For evaluation of dense video captioning, we report METEOR [21] and BLEU [32] scores. BLEU is a precision-based metric, which accounts for precise matching of n-grams between a generated and ground truth sentence. METEOR, however, first creates an alignment by comparing exact tokens, stemmed tokens, and paraphrases of the prediction and ground truth sentence. It also takes semantically similar matches into account by using WordNet for synonym analysis [1]. Using temporal Intersection over Union (tIoU) thresholds at $\{0.3, 0.5, 0.7, 0.9\}$, for captioned events we report recall and precision of their temporal segments and METEOR and BLEU scores of their sentences. Given a tIoU threshold, if the event proposal has a segment overlapping larger than the threshold with any gold standard segment, the metric score is computed for the generated sentence and the corresponding gold standard sentence. Otherwise, the metric score is set to 0. The scores are then averaged across all event proposals and finally averaged across all tIoU thresholds.

4.4.2 Entity, Property, and Relation Extraction. For property, and video-level entity and relation extraction, we measure micro-averaged precision, recall, and their harmonic mean, F1 score.

¹²Available at: https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4.

¹³2017 eval. toolkit is available at: https://github.com/ranjaykrishna/densevid_eval/.

¹⁴2018 eval. toolkit is available at: https://github.com/ranjaykrishna/densevid_eval/.

¹⁵From 2017 to 2018 a critical issue was fixed where for each video only one out of multiple incorrect predictions was taken into account. This leads to a performance overestimation of models when they are evaluated with the 2017 toolkit.

Similarly to the evaluation of DVC models, when evaluating entity and relation extraction on event-level, we also take the quality of the predictions' temporal segments into account. Here, we compute micro-averaged precision and recall across all videos using tIoU thresholds at $\{0.3, 0.5, 0.7, 0.9\}$, and then average the results across all thresholds. Additionally, we report F1 score of averaged precision and recall.

For the evaluation of entities, entity-property pairs, and relations, we use WordNet to compare words in terms of whether they are synonyms of each other or not. We think this is fair as the variety of extracted semantic metadata is large (see Table 2). For example, "stand up" is a correct prediction for the verb "get up". However, this means that for a video the number of correct predictions TP_p , and the number of correctly predicted gold standard targets TP_g is not necessarily the same¹⁶. To ensure the validity of precision (proportion of correct predictions) and recall (proportion of correctly predicted targets), we use TP_p for calculating precision and TP_g for recall, respectively.

All entities, properties, and relations are compared using their word lemmas. On video-level, this means that potential duplicates of entities and relations are removed, i.e., they are lemma-unique. For example, $\{\text{man}, \text{men}\}$ results in $\{\text{man}\}$. We define $SYN(w)$ as the set of all synonyms listed by WordNet for the word w with respect to its part of speech, i.e., either noun, verb, adverb, or adjective. For w , a synonym is a word sharing a synset with w . We make sure that $w \in SYN(w)$.

The following rules define when a prediction for an entity, entity-property pair, or relation is correct:

Entities. A video-level entity e_s predicts e_g correctly when for its noun holds $n_s \in SYN(n_g)$. Similarly to the evaluation of DVC models, in the evaluation of event-level entity extraction, we also take the entities' temporal segments into account. Given a tIoU threshold, an event-level entity e_s predicts e_g correctly, if 1) $n_s \in SYN(n_g)$ and 2) e_s has a segment overlapping with the segment of e_g larger than the threshold.

Properties. An entity-property pair (e_p, a_p) predicts (e_g, a_g) correctly, if $e_p \in SYN(e_g)$ and $a_p \in SYN(a_g)$.

Relations. On video-level, a relation r_p predicts a target relation r_g correctly, if 1) $subjects_p$ contains at least one entity that predicts any entity of $subjects_g$, and 2) $objects_p$ contains at least one entity that predicts any entity of $objects_g$. In addition to that, 3) the verb $verb_p$, or any combination of it with one of its modifiers in $modifier_p$ has to predict the $verb_g$, or any combinations of it with one of its modifiers in $modifier_g$. Again, if a synonym of a target subject, verb or object is predicted, then the corresponding prediction is correct. If we use the above rules, then, e.g., $([\text{man}, \text{woman}], \text{go}, [\text{in}], \text{building})$ is a correct prediction of $(\text{man}, \text{enter}, [], \text{building})$. For the evaluation of event-level relation extraction, we take the relations' temporal segments into account in the same way as we do for entities.

4.4.3 Text Classification. Precision, recall, and F1 score are vital metrics for the evaluation of (single-label, multi-class) classification methods on unbalanced test sets [23]. We report weighted (with weights equal to the category probability) and macro-averages of precision, recall, and F1 score across all categories. This includes the calculation of accuracy, as this is equal to the weighted recall.

¹⁶Example: Using *person* and *individual* as predictions for *person* results in $TP_p = 2$ and $TP_g = 1$, as *individual* is a synonym of *person*.

5 RESULTS

Given the experimental setup from Section 4, we evaluate the results of dense video captioning, entity, property, and relation extraction, and text classification. For further experimental results refer to Section B of the Supplementary Materials. For example, Table 14 lists the video-level entities, entity-property pairs, and video-level relations that are most frequently extracted by our framework for ActivityNet Captions videos.

5.1 Dense Video Captioning

Recall that when implementing and training the DVC models MT [46] and PDVC [40], we changed few parameters compared to their original works (see Sec. 4.3). To ensure consistency, we evaluated both models in event localization and dense video captioning according to their original evaluation setup. Our trained models achieved similar, and for dense video captioning even better performances (see Supplementary Materials, Sec. B).

We introduced the parameters $|E|$ and $|dist(E)|$, which are primarily used in our framework to control the amount of information that is forwarded from the DVC model to the semantic metadata extraction methods. Using these parameters, we can also compare the DVC models in a fair way. Table 3 shows the event localization and dense video captioning performances of MT and PDVC with respect to $|E|$ and $|dist(E)|$, the number of (distinct) captioned events that are generated by each DVC model and used for the evaluation. This enables us to later analyse whether there are correlations between the performance of the DVC model that is used by our framework and the quality of the extracted semantic metadata. For event localization, PDVC constantly achieves better recall than MT, while MT constantly outperforms PDVC in terms of precision. For both models, higher $|E|$ and $|dist(E)|$ results in better recall and worse precision performance in event localization. Looking at the results for dense video captioning, we can state that the METEOR performances of both models degrade with increasing number of captioned events, both distinct and non-distinct, except for when increasing $|dist(E)|$ from 1 to 3. When increasing $|E|$ from 10 to 100, the METEOR performance of PDVC drops by 3.71 points in total, while the METEOR performance of MT drops only by 1.21 points. This is a consequence of the declining precision for event localization for higher $|E|$, which affects the PDVC model more heavily than the MT model.

5.2 Entity Extraction

Table 4 shows precision, recall, and F1 score performances of our framework for video-level entity extraction. As for all semantic metadata types, we evaluate our framework in two different settings: while using the MT model, and while using the PDVC model for generation of semantic information. We evaluate the extracted video-level entities with entity frequency threshold f set to 0, 10, 25, and 50, meaning that the gold standard only contains those video-level entities that have a frequency higher than the threshold. One observation is that, with increasing $|dist(E)|$, precision decreases and recall improves for both models and all thresholds. For both DVC models and all thresholds, the best F1 score performance is achieved with $|dist(E)| = 10$. This indicates a limited level of semantic information that any further generated captioned events provide. For higher entity frequency thresholds, for both DVC models and all $|dist(E)|$, our framework is able to predict video-level entities with improved recall, at the cost of only slightly lower precision. When our framework is using

the PDVC model, it achieves better precision and F1 scores for all $|dist(E)|$. On the other hand, it achieves better recall performances when using the MT model for $|dist(E)|$ set to 10 and 25. Overall, for video-level entity extraction, our framework achieves its highest F1 scores when using the PDVC model with $|dist(E)|$ set to 10. Here, the achieved F1 scores range from 31.27 for $f = 0$ to 34.21 for $f = 50$.

Table 5 shows the results of event-level entity extraction of our framework. Depending on the framework’s used DVC model and the number of generated captioned events $|E|$, we report precision and recall for different temporal Intersection over Union (tIoU) thresholds. F1 score is calculated using the averages of precision and recall. Note that for a prediction to be correct for an event-level entity, a condition is that its temporal segment overlaps with the gold standard entity’s temporal segment larger than the tIoU threshold. Therefore, precision and recall decreases at higher tIoU thresholds. Regardless of the DVC model used, the framework’s precision decreases for higher $|E|$, while at the same time it benefits in recall performance. When using the PDVC model, the framework’s precision drops more (1.13 on average) when increasing $|E|$ from 10 to 100 as compared to when its using the MT model (0.57 on average). Note that we made the same observation when we evaluated the event localization and dense video captioning performances of the DVC models. For event localization, the PDVC model could not translate the large drops in precision for higher $|E|$ into better recall performance. Thus, on the one hand, this results in worse dense video captioning performance. On the other hand, when our framework is using the PDVC model for event-level entity extraction, this results in degrading F1 score performance for higher $|E|$. Still, our framework achieves its highest F1 score with 16.19 when it uses the PDVC model with $|E| = 10$. On the other hand, when using the MT model, the highest achieved F1 score is 15.48 for $|E| = 50$.

5.3 Property Extraction

The results of the extraction of entity-property pairs with our framework are shown in Table 6. In general, increasing $|dist(E)|$ leads to drops in precision, however, our framework benefits from much improved recall. Consequently, our framework achieves its highest F1 scores for $|dist(E)| = 25$. In contrast to video-level entity extraction where highest F1 scores were achieved for $|dist(E)| = 10$, we can observe that increasing $|dist(E)|$ from 10 to 25 leads to even better property extraction performance with respect to F1 score, indicating that the DVC models still provide meaningful semantic information about the properties of entities when many captioned events are generated. For all $|dist(E)|$, our framework achieves better recall and F1 score when using the MT model for captioned events generation and better precision when using the PDVC model. The highest achieved precision is 9.08 when using PDVC with $|dist(E)| = 1$. Using the MT model with $|dist(E)| = 25$ results in the framework’s highest achieved recall (8.86) and F1 score (4.94).

5.4 Relation Extraction

Table 7 shows the results of video-level relation extraction with our framework. Again, an increase in $|dist(E)|$ leads to drop in precision, while recall performance improves. In general, using the PDVC model leads to better precision performance except for $|dist(E)| = 1$, while using the MT model leads to better recall performance except for $|dist(E)| = 3$. For both DVC models, our framework achieves its highest F1 scores for $|dist(E)| = 10$. This

Event Localization & Dense Video Captioning						
DVC Model	E	avg. Recall	avg. Precision	2018 eval. toolkit		
		B@3	B@4	M		
MT	10	43.61	48.96	2.18	1.02	5.89
	25	56.55	46.15	2.33	1.14	5.74
	50	67.70	41.60	2.31	1.13	5.35
	100	76.33	34.78	2.12	1.04	4.68
PDVC	10	61.88	45.41	3.10	1.59	6.34
	25	73.81	36.89	2.54	1.23	5.17
	50	78.76	25.75	1.92	0.90	3.88
	100	82.24	15.63	1.36	0.63	2.63

(a) Results for MT and PDVC for varying numbers of generated captioned events $|E|$.

Event Localization & Dense Video Captioning						
DVC Model	dist(E)	avg. Recall	avg. Precision	2018 eval. toolkit		
		B@3	B@4	M		
MT	1	22.04	50.99	1.55	0.68	5.37
	3	32.72	49.65	1.97	0.90	5.76
	10	50.08	44.75	2.22	1.05	5.58
	25	63.67	36.23	2.05	0.97	4.80
PDVC	1	20.05	49.63	2.52	1.30	5.86
	3	40.28	48.26	2.98	1.54	6.46
	10	62.38	44.10	2.85	1.41	6.01
	25	71.62	31.69	2.14	1.00	4.54

(b) Results for MT and PDVC for varying numbers of generated distinct captioned events $|dist(E)|$.

Table 3: Event localization and dense video captioning results for different numbers of generated captioned events on the ActivityNet Captions validation set. Only the first of the two gold standard annotation files of the validation set is used, as this contains the data from which we generate our gold standards for semantic metadata from. We report recall and precision of temporal segments, and METEOR (M) and BLUE@N (B@N) performances of generated captioned events.

Video-level Entity Extraction													
DVC Model	dist(E)	Precision(@f)				Recall(@f)				F1(@f)			
		0	10	25	50	0	10	25	50	0	10	25	50
MT	1	39.94	39.88	39.66	38.91	15.38	16.49	17.57	18.99	22.21	23.33	24.35	25.52
	3	33.91	33.83	33.61	32.78	23.44	25.12	26.72	28.70	27.72	28.83	29.77	30.60
	10	26.44	26.37	26.16	25.43	32.59	34.90	37.06	39.67	29.20	30.04	30.67	30.99
	25	20.76	20.69	20.49	19.82	40.69	43.55	46.16	49.15	27.49	28.05	28.38	28.25
PDVC	1	45.13	45.05	44.91	44.43	15.44	16.56	17.68	19.28	23.01	24.22	25.37	26.89
	3	39.01	38.95	38.83	38.28	23.33	25.03	26.72	29.04	29.20	30.48	31.66	33.03
	10	31.77	31.71	31.60	31.03	30.78	33.01	35.21	38.11	31.27	32.35	33.30	34.21
	25	26.30	26.25	26.14	25.61	36.18	38.79	41.33	44.66	30.46	31.31	32.02	32.55

Table 4: Experimental results for video-level entity extraction with our framework. We evaluate our framework for two different dense video captioning models: MT and PDVC. Reported are precision, recall, and F1 score on our gold standard for video-level entities for different numbers of generated distinct captioned events $|dist(E)|$ and entity frequency thresholds f .

Event-level Entity Extraction													
DVC Model	E	Precision(@tIoU)				Recall(@tIoU)				F1			
		0.3	0.5	0.7	0.9	Avg	0.3	0.5	0.7	0.9	Avg	F1	
MT	10	28.97	20.08	8.55	1.26	14.72	21.11	15.31	9.59	2.94	12.24	13.37	
	25	28.12	18.44	7.40	1.03	13.75	26.48	20.66	14.29	5.58	16.75	15.10	
	50	26.30	15.93	6.10	0.83	12.29	31.15	25.32	18.69	8.47	20.91	15.48	
	100	23.31	12.70	4.52	0.59	10.28	36.35	30.05	22.89	11.04	25.08	14.58	
PDVC	10	30.47	19.79	9.11	2.53	15.47	26.36	21.13	14.52	5.87	16.97	16.19	
	25	27.16	15.83	6.41	1.53	12.73	31.14	26.34	20.01	8.21	21.42	15.97	
	50	20.75	10.49	3.84	0.86	8.98	33.66	28.89	22.40	8.75	23.42	12.98	
	100	13.76	6.36	2.22	0.48	5.71	35.35	30.65	24.00	9.02	24.75	9.28	

Table 5: Experimental results for event-level entity extraction with our framework. Reported are precision and recall performance on our gold standard for event-level entities for different numbers of generated distinct captioned events $|dist(E)|$ and tIoU thresholds. Furthermore, F1 score is calculated using the averages of precision and recall across all thresholds.

Property Extraction					
DVC Model	$ dist(E) $	Prec	Rec	F1	
MT	1	6.48	0.82	1.45	
	3	6.64	1.85	2.90	
	10	5.66	3.54	4.36	
	25	4.53	5.43	4.94	
PDVC	1	9.08	0.72	1.34	
	3	8.76	1.61	2.72	
	10	6.96	2.68	3.87	
	25	4.79	3.76	4.21	

Table 6: Experimental results for the extraction of entity-property pairs with our framework.

Video-level Relation Extraction					
DVC Model	$ dist(E) $	Prec	Rec	F1	
MT	1	5.76	2.07	3.04	
	3	4.88	4.07	4.44	
	10	3.64	6.61	4.70	
	25	2.89	8.86	4.35	
PDVC	1	5.64	2.06	3.02	
	3	5.02	4.18	4.56	
	10	4.09	6.50	5.02	
	25	3.47	8.36	4.91	

Table 7: Experimental results for video-level relation extraction with our framework.

suggests that any further generated captioned events can provide more semantic information for relations only at a higher cost of precision. This is the same observation as we made for video-level entity extraction. However, for property extraction, the highest F1 scores were achieved for $|dist(E)| = 25$. The highest achieved F1 score of our framework for video-level relation extraction is 5.02 while using the PDVC model with $|dist(E)| = 10$.

The results of event-level relation extraction are shown in Table 8. Very similar observations can be made as for event-level entity extraction. The framework’s precision decreases for higher $|E|$ while benefiting in recall performance. When using the PDVC model, the framework’s precision performance suffers more for higher $|E|$ compared to when its using the MT model. Again, as observed before for event-level entity extraction, this is not translated into much higher recall, resulting in degradation of the framework’s F1 score performance. Therefore, for $|E|$ set to 50 and 100, i.e., high numbers of generated captioned events, our framework achieves its highest F1 scores when using the MT model, while for $|E|$ set to 10 and 25 the highest F1 scores are achieved when using the PDVC model. As for event-level entity extraction, for event-level relation extraction our framework achieves its best performance with respect to F1 score when it uses the PDVC model with $|E|$ set to 10.

5.5 Text Classification

Finally, Table 9 shows the weighted and macro-averages of precision, recall, and F1 score performances of the framework’s text classification method, while trained and evaluated in three different settings: using the captioned events generated by the MT

or PDVC model, respectively, and using the captioned events provided by ActivityNet Captions. The most noteworthy observation is that the classification performance of our framework, when using captioned events generated by MT and PDVC, is not far from the classification performance that is achieved when using gold standard captioned events of ActivityNet Captions. Therefore, we can state that the DVC models generate specific semantic information for videos of different categories at a similar level as the captioned events of the gold standard provide. This is important for the text classifier to categorize videos successfully.

When using automatically generated captioned events for video classification, our framework achieves its best performances for all metrics, both weighted and macro-averaged, when using the PDVC model. Here, for weighted precision, recall, and F1 score, our framework performs around 2 points better as compared to when using the MT model. When using PDVC, our framework achieves an overall accuracy (i.e., weighted recall) of 50.22, which is only 0.59 points lower than the accuracy achieved when classifying videos using the gold standard captioned events of ActivityNet Captions. Taking the category imbalance in our dataset into account, we observe that the achieved macro-averages of precision, recall and F1 scores are much lower as compared to their weighted-averages. For extended results in text classification refer to the Supplementary Materials, Section B.

6 DISCUSSION

In this work, we proposed a framework for the automatic extraction of multiple semantic metadata types from videos, namely the depicted *entities* and their *properties*, the observable *relations* between entities, and the video *category*. For each metadata type, we presented a method which extracts the corresponding semantic information either directly from the captioned events that the framework’s DVC model generates for an input video, or the linguistic annotations produced by the framework’s language parser for the captioned events. With the results presented in Section 5, we now discuss 1) the capability of our framework to extract semantic metadata, 2) if and how the quality of extracted metadata is affected by the amount of information that is generated by its DVC model, and 3) whether there are correlations between the performance of the framework’s DVC model and the quality of semantic metadata that is extracted from the captioned events it generates.

6.1 Key Results

The experiments show that our proposed framework is able to automatically generate multiple types of semantic metadata in a meaningful way. We have to keep in mind that in our framework semantic metadata is extracted only from captioned events that are automatically generated by its DVC model, i.e., a model that is designed and trained for a different task. To extract higher quality semantic information, for each semantic metadata type a dedicated computer vision method could be used, such as video object detection and video visual relation tagging. Here, however, our framework trades quality for effectiveness as it only requires training for one computer vision model, the DVC model. We must also bear in mind that the variety of entities, entity-property pairs, and relations that occur in our gold standards, and that are extracted by our framework, is large, in particular when compared to related computer vision tasks. For example, the VidOR (Video Object Relation) dataset [35, 38], which is used to train models that target the task of video visual relation detection. It contains

Event-level Relation Extraction											
DVC Model	$ E $	Precision(@tIoU)					Recall(@tIoU)				
		0.3	0.5	0.7	0.9	Avg	0.3	0.5	0.7	0.9	Avg
MT	10	3.76	2.52	1.08	0.14	1.87	3.81	2.81	1.73	0.42	2.20
	25	3.61	2.28	0.93	0.12	1.74	4.89	3.90	2.56	0.87	3.05
	50	3.35	1.99	0.77	0.10	1.55	5.87	4.78	3.29	1.32	3.81
	100	2.96	1.59	0.56	0.07	1.30	7.06	5.60	3.91	1.63	4.55
PDVC	10	3.64	2.33	1.14	0.32	1.86	4.87	3.87	2.63	1.06	3.11
	25	3.33	1.92	0.80	0.20	1.56	6.01	4.95	3.49	1.37	3.95
	50	2.66	1.33	0.50	0.11	1.15	6.53	5.37	3.81	1.41	4.28
	100	1.75	0.82	0.30	0.07	0.73	6.75	5.53	3.91	1.41	4.40

Table 8: Experimental results for event-level relation extraction with our framework.

Text Classification						
Average	Captioned Events Input		Prec	Rec	F1	
Weighted	MT		43.72	48.24	44.80	
	PDVC		45.75	50.22	46.96	
	ActivityNet Captions		46.97	50.81	48.23	
Macro	MT		27.51	30.45	28.08	
	PDVC		34.31	32.30	30.88	
	ActivityNet Captions		33.42	34.74	33.19	

Table 9: Experimental results for the classification of videos with our framework. The text classifier is trained and evaluated in three different settings: using captioned events generated by the video captioning models MT and PDVC, and gold standard captioned events from ActivityNet Captions. For MT and PDVC, $|dist(E)|$ is set to 10. Reported are weighted and macro-averages of precision, recall, and F1 score across all categories on our classification test set.

annotations of 80 categories of objects and 50 categories of relation predicates. In our gold standard for relations, however, 2,493 different entities acted either as subject (905 entities) or object (2,299 entities), while 905 different verbs occur in the relations. Also, for video-level entity extraction, we observed that for higher entity frequency thresholds f the framework’s precision decreases only slightly, while recall performance improves greatly. This observation is not surprising, as the number of different entities in our gold standard for entities, which is based on ActivityNet Captions, is large, and many entities occur only a few times (see Table 2). This leads to the conclusion that the DVC models are only able to learn and integrate entities in their generated captions when they see them sufficiently enough during training. Out of the semantic metadata types entities, properties, and relations, our framework achieves its best results in entity extraction. Looking at the video classification results, it has to be said that our framework, with the DVC models that we used and how we trained them, could not reliably predict the category of a video with the text classification model we chose. This is due to the captioned events that are generated by the framework’s DVC model, which certainly contain specific semantic information for videos of different categories. However, they do not at a level where high-quality classification results are achievable.

We introduced the parameters $|E|$ and $|dist(E)|$, the number of (distinct) captioned events that are generated, to be able to

Semantic Metadata Type	DVC Model	$ E / dist(E) $
Entities (video-level)	PDVC	10
Entities (event-level)	PDVC	10
Properties	MT	25
Relations (video-level)	PDVC	10
Relations (event-level)	PDVC	10
Category	PDVC	10

Table 10: Combinations of DVC model and number of its generated captioned events that result in the highest achieved F1 scores of our framework for different semantic metadata types.

control the amount of information that the DVC model provides for the semantic metadata extraction methods. MT and PDVC, the dense video captioning models that we used for our framework, internally calculate confidence scores for their generated captioned events. If the number of considered captioned events is limited, then only the generated captioned events with the best confidence scores are used. The experiments show that the confidence scores serve their purpose. As expected, precision of predictions decreases for increasing $|E|$ and $|dist(E)|$ with only few exceptions. On the other hand, higher recall values are achieved when increasing $|E|$ and $|dist(E)|$. This applies to all, event localization and dense video captioning, as well as entity, property, and relation extraction. We assume the F1 score to be the metric that best measures the overall quality of extracted semantic metadata. For all extracted semantic metadata types, Table 10 lists the combination of the DVC model and the number of its generated captioned events our framework uses to achieve best performance with respect to F1 score. We observe that usually for 10 generated captioned events the highest F1 scores are achieved, except for property extraction, where this is the case for $|dist(E)| = 25$. This indicates that the quality of semantic metadata extracted by our framework degrades when more than 10 captioned events are generated by its DVC model.

We finally analyse whether there are correlations between the performance of the DVC model that is used by our framework and the quality of the extracted semantic metadata. Revisiting Table 10, we observe that our framework achieves highest F1 scores for all semantic metadata types when employing the PDVC model, except for property extraction. Looking at the dense video captioning results of MT and PDVC with respect to $|E|$ and $|dist(E)|$ (see Table 3), we also observe that PDVC usually outperforms

MT in METEOR score when generating distinct captioned events. For non-distinct captioned events, however, this is only the case for $|E| = 10$. For higher $|E|$ the achieved METEOR score of PDVC decreases more sharply as compared to the MT model due to declining precision for event localization. We made the exact same observations for event-level extraction of entities and relations in the previous section. Thus, all things considered, we conclude that the performance of the framework’s DVC model is certainly an indicator of the quality of semantic metadata that is extracted from the captioned events it generates.

6.2 Threads to Validity

There are several concerns related to the gold standards that we use to evaluate our framework’s capability in entity, property, and relation extraction. Since there is no existing dataset for this purpose, we generated a gold standard using the captioned events of ActivityNet Captions validation videos. As described in Section 4.1, we use our proposed entity, property, and relation extraction methods on the processed captioned events, and treat the results as gold standards for semantic metadata. This, however, requires these extraction methods to work sufficiently well, i.e., they in fact are able to extract semantic metadata using the linguistic annotations provided by the framework’s language parser. To this end, we annotated a small subset of 25 (total of 110 captioned events) ActivityNet Captions videos with the video-level entities, entity-property pairs, and video-level relations that we expect the methods to extract from captioned events. Table 11 shows the precision and recall performances of our entity, property, and relation extraction methods on these manually annotated videos. The results show that our entity and property extraction methods are certainly reliable. In some cases, however, our entity extraction method wrongly determines entities. For example, for the sentence “*the camera pans around the field*”, spaCy wrongly classifies *camera*, *pans* and *field* as nouns, while *pans* actually acts as verb in the sentence. Since *pan* is listed in WordNet as a noun, *pans* is still determined as an entity. The relation extraction method is able to determine only around 70% of the relations that we annotated. This is due to the complexity of relation extraction, which relies heavily on the dependency parse that the language parser of our framework produces. For more complex sentences, we observed that the language parser produces syntactic relations between tokens with wrong labels, thus leading the relation extraction method to determine either no relations, or incorrect relations.

It becomes clear that the quality of extracted semantic metadata depends heavily on the quality of the linguistic annotations that the language parser provides, i.e., the POS tags, dependency labels, and coreference clusters. The quality of linguistic annotations, again, depends on whether the sentences at hand are grammatically correct or not. This means that the DVC model used by our framework needs to generate proper sentences with respect to grammar. Only then the language parser is able to generate linguistic annotations of high quality, ensuring the entity, property, and relation extraction methods to extract semantic information from linguistic annotations.

In Section 3.4, we declared that, e.g., an object is only an entity when it can be seen in the video. In our framework, we extract entities from the captioned events generated by a DVC model. However, the entities that appear in the sentence descriptions of the captioned events are not necessarily visible in the video (e.g., *camera* in “*A man is speaking to the camera*”). To analyse

Semantic Metadata Type	Prec	Rec
Entities (video-level)	94.21	98.39
Properties	92.98	91.38
Relations (video-level)	78.36	70.62

Table 11: Precision and recall performances of the entity, property, and relation extraction methods on captioned events from 25 manually annotated videos.

how many entities that are extracted by our framework for ActivityNet Captions videos are actually not visible in the video, we used ActivityNet-Entities¹⁷ [44], a dataset based on ActivityNet Captions. ActivityNet-Entities augments 12,469 videos from the ActivityNet Captions train and validation sets with bounding box annotations, each grounding a noun phrase (NP). In particular, NPs are not annotated when they are abstract or not presented in the scene. By comparing the entities that appear in ActivityNet-Entities with those included in our generated gold standard for entities, we can estimate how many extracted entities are actually not visible in the input video. Predicting the video-level entities extracted from NPs from ActivityNet-Entities with the video-level entities extracted from captioned events from ActivityNet Captions results in a precision of 64.97 and recall of 98.17. Therefore, up two thirds of the entities in our gold standards for entities are actually abstract or not visible (e.g., *camera*, *front*, *end*, and *game*). Furthermore, when extracted from a captioned event, an event-level entity has the same temporal segment as the captioned event. Therefore, it is assumed that the entity is visible for the entire duration of the temporal segment, even though the entity may actually be visible for only a part of it (e.g., *glass* in “*the man puts a glass into a cupboard, then closes it and walks out of the room*”). Finally, it is also likely that many entities, properties, and relations that appear in a video are not captured by our framework. This is because the semantic information may not be relevant for a video, and therefore is not included in a caption event generated by the DVC model.

6.3 Generalizability

We stated that our framework is not able to differentiate whether an extracted entity is actually visible in a video or not. Since the gold standard captioned events of ActivityNet Captions also contain non-visible entities, the DVC model reproduces this characteristic, leading our framework to extract semantic metadata that is abstract or not presented in the video. In general, the quality of extracted semantic metadata depends on the quality of semantic information that the framework’s DVC model generates, which again depends on the quality or characteristics of the dataset on which the DVC model is trained on. YouCook2 [45], e.g., is a task-oriented video dataset in the domain of cooking. In this case, the annotated captioned events for videos are cooking instructions. The proportion of actually observable semantic information that YouCook2 annotations contain is much higher as compared to ActivityNet Captions annotations. Consequently, our framework would have the same characteristic when trained and used on YouCook2. Ultimately, this means that our framework produces semantic metadata of highest quality, when its video description model is trained with captioned events that explicitly only describe the observable entities, their properties, and how the entities relate to each other.

¹⁷ActivityNet-Entities: <https://github.com/facebookresearch/ActivityNet-Entities>.

Our proposed framework extracts semantic metadata from videos by first generating semantic information in textual form using DVC, and then applying NLP methods. We decided to use a DVC model for generating semantic information as this allows us to associate entities and relations with temporal segments, i.e., to generate event-level semantic metadata. When event-level information is not required, in our framework the DVC model can be easily replaced with other video description methods. For example, Video Paragraph Generation models [22, 42] automatically generate multiple natural language sentences that provide a narrative of a video clip, without having to localize individual events [1]. Furthermore, our framework can be used to extract semantic metadata from other multimedia types, e.g., from images. Here, the framework’s model for generating semantic information in textual form could be an Image Paragraph Captioning model [28, 41], which generate a detailed, multi-sentence description of the content of an image.

We conclude that our framework is easily adaptable because of its modularity: multiple configurations of our framework can be implemented, each serving its specific purpose. Different DVC models can be trained for different video domains (e.g., cooking, animals) to improve the quality of generated semantic information. Further video description methods can be used in our framework, allowing the extraction of only video-level semantic metadata. The framework’s text classification method, and entity, property, and relation extraction methods can be modified or replaced, and other semantic metadata extraction methods can easily be added. Lastly, by using an appropriate description model that generates semantic information in textual form, our framework can be adapted to extract semantic metadata from other multimedia types such as images.

6.4 Future Work

As the capabilities of our proposed framework are not yet exhausted, our work enables several paths for future work. The modular structure of our framework allows to replace its DVC model, the language parser, and also the text classification method at will. Iashin and Rahtu [15] argue that handling of multiple modalities is under-explored in the computer vision community. To this end, they presented a bi-modal transformer which uses both, visual features and the audio track, and achieved state-of-the-art performance for dense video captioning. Perhaps our framework produces higher quality semantic metadata when employing a DVC model that also uses the audio track. More generally, it would be interesting to see how well our framework performs for further DVC models, and different language parser. The training of the framework’s DVC model could be improved for the purpose of semantic metadata extraction by putting more attention to, e.g., entities and which properties they have. Also, instead of the spaCy library, in our framework the widely used Stanford CoreNLP toolkit [27] could be used to generate linguistic annotations from text. However, this would involve minor changes to the entity, property, and relation extraction methods as, unlike spaCy, Stanford CoreNLP uses the dependency labels of Universal Dependencies.

As mentioned before, in our framework the DVC model can be replaced easily with other multimedia description methods. In the domain of videos, it would be interesting to see how well our framework is able to extract video-level semantic metadata when using a video paragraph generation model instead of a DVC model [22, 42]. As opposed to video paragraph generation,

DVC models generate sentences that are not necessarily coherent. Therefore, when in our framework text is generated with a video paragraph generation model, the language parser could be able to produce linguistic annotations of higher quality, resulting in improved quality of extracted semantic metadata.

The entities that our framework extracts are generic. For example, when in a video *Lionel Messi* is seen standing in the *Camp Nou*, our framework would generate the entities *man* and *stadium*. To enable extraction of named entities, the framework’s video description model could be supported by models that are able to, e.g., identify persons, such as face recognition models [9]. This would also allow the language parser to use Named Entity Recognition [20], i.e., the tagging of named entities in text with their corresponding type like *person* or *location*, potentially improving the quality of the linguistic annotations it produces.

Finally, in this work we considered the semantic metadata types entities, properties, relations, and the video category. However, depending on the application, one might prefer a more detailed distinction of entities, e.g., between persons, objects, and locations, rather than just entities. Furthermore, other types of semantic metadata may be of interest, e.g., emotions. Our framework can easily be extended by further metadata extraction methods. It would be interesting to see where the limits of the framework’s video description model are in terms of what amount of useful semantic information it can generate for semantic metadata extraction.

7 CONCLUSION

In this work, we presented an end-to-end framework for automatic extraction of semantic metadata of various types from videos. Our framework combines methods from computer vision, namely dense video captioning, and methods from natural language processing, namely natural language parsing, open information extraction, and text classification. The modular structure of our framework allows its dense video captioning model and its language parser to be replaced at will. Our framework extracts semantic metadata of four different types, namely the depicted entities, their properties, the observable relations between entities and the video category. Experiments have shown that our framework is able to automatically extract semantic metadata from videos, while the quality of its extracted information depends on two important factors. This is primarily the event localization and video captioning performance of the framework’s dense video captioning model. Furthermore, we have shown that the number of captioned events that are forwarded from the dense video captioning model to each semantic metadata extraction method is an important parameter.

REFERENCES

- [1] Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019. Video Description: A Survey of Methods, Datasets, and Evaluation Metrics. *ACM Comput. Surv.* 52, 6, Article 115 (oct 2019), 37 pages. <https://doi.org/10.1145/3355390>
- [2] Harry Agius. 2008. *MPEG-7: Multimedia Content Description Interface*. Springer US, Boston, MA, 475–483. https://doi.org/10.1007/978-0-387-78414-4_41
- [3] Siddu Algur and Prashant Bhat. 2014. Metadata Construction Model for Web Videos: A Domain Specific Approach. *International Journal Of Engineering And Computer Science* 3 (01 2014), 2319–7242.
- [4] Siddu P Algur and Prashant Bhat. 2016. Web video mining: metadata predictive analysis using classification techniques. *IJ Information Technology and Computer Science* 2 (2016), 69–77.
- [5] Werner Bailer, Lionel Brunie, Mario Döller, Michael Granitzer, Ralf Klamma, Harald Kosch, Matthias Lux, and Marc Spaniol. 2008. *Multimedia Metadata Standards*. Springer US, Boston, MA, 568–575. https://doi.org/10.1007/978-0-387-78414-4_49

- [6] Kevin Clark and Christopher D. Manning. 2016. Deep Reinforcement Learning for Mention-Ranking Coreference Models. <https://doi.org/10.48550/ARXIV.1609.08667>
- [7] Alberto Del Bimbo. 1999. *Visual Information Retrieval*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- [8] Luciano Del Corro and Rainer Gemulla. 2013. ClausIE: Clause-Based Open Information Extraction. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) (WWW '13). Association for Computing Machinery, New York, NY, USA, 355–366. <https://doi.org/10.1145/2488388.2488420>
- [9] Jiankang Deng, Jia Guo, Nianan Xue, and Stefanos Zafeiriou. 2019. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4685–4694. <https://doi.org/10.1109/CVPR.2019.00482>
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 <http://arxiv.org/abs/1810.04805>
- [11] Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK, 1535–1545. <https://aclanthology.org/D11-1142>
- [12] Lukas Galke and Ansgar Scherp. 2022. Bag-of-Words vs. Graph vs. Sequence in Text Classification: Questioning the Necessity of Text-Graphs and the Surprising Strength of a Wide MLP. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Online.
- [13] Kenneth Haase. 2004. Context for Semantic Metadata. In *Proceedings of the 12th Annual ACM International Conference on Multimedia* (New York, NY, USA) (MULTIMEDIA '04). Association for Computing Machinery, New York, NY, USA, 204–211. <https://doi.org/10.1145/1027527.1027574>
- [14] M.-S. Hadid, C. Decleir, and J. Kouloundjian. 2000. A database approach for modeling and querying video data. *IEEE Transactions on Knowledge and Data Engineering* 12, 5 (2000), 729–750. <https://doi.org/10.1109/69.877505>
- [15] Vladimir Lashin and Esa Rahtu. 2020. A Better Use of Audio-Visual Cues: Dense Video Captioning with Bi-modal Transformer. In *British Machine Vision Conference (BMVC)*.
- [16] Ramesh Jain and Arun Hampapur. 1994. Metadata in Video Databases. *SIGMOD Rec.* 23, 4 (Dec. 1994), 27–33. <https://doi.org/10.1145/190627.190638>
- [17] Licheng Jiao, Ruohan Zhang, Fang Liu, Shuyuan Yang, Biao Hou, Lingling Li, and Xu Tang. 2021. New Generation Deep Learning for Video Object Detection: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* (2021), 1–21. <https://doi.org/10.1109/TNNLS.2021.3053249>
- [18] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1746–1751. <https://doi.org/10.3115/v1/D14-1181>
- [19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-Captioning Events in Videos. In *2017 IEEE International Conference on Computer Vision (ICCV)*. 706–715. <https://doi.org/10.1109/ICCV.2017.83>
- [20] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, 260–270. <https://doi.org/10.18653/v1/N16-1030>
- [21] Alon Lavie and Abhay Agarwal. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Prague, Czech Republic, 228–231. <https://aclanthology.org/W07-0734>
- [22] Jie Lei, Liwei Wang, Yelong Shen, Dong Yu, Tamara Berg, and Mohit Bansal. 2020. MART: Memory-Augmented Recurrent Transformer for Coherent Video Paragraph Captioning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Online, 2603–2614. <https://doi.org/10.18653/v1/2020.acl-main.233>
- [23] Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. 2020. A Survey on Text Classification: From Shallow to Deep Learning. *CoRR* abs/2008.00364 (2020). arXiv:2008.00364 <https://arxiv.org/abs/2008.00364>
- [24] Yehao Li, Ting Yao, Yingwei Pan, Hongyang Chao, and Tao Mei. 2018. Jointly Localizing and Describing Events for Dense Video Captioning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7492–7500. <https://doi.org/10.1109/CVPR.2018.00782>
- [25] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. 2018. Attention Clusters: Purely Attention Based Local Feature Integration for Video Classification. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7834–7843. <https://doi.org/10.1109/CVPR.2018.00817>
- [26] Matthias Lux, Werner Kleiber, Jutta Becker, Klaus Tochtermann, Harald Mayer, Helmut Neuschmied, and Werner Haas. 2002. XML and MPEG-7 for Interactive Annotation and Retrieval using Semantic Meta-data. *J. UCS* 8 (01 2002), 965–984.
- [27] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit.. In *ACL (System Demonstrations)*. The Association for Computer Linguistics, 55–60. <http://dblp.uni-trier.de/db/conf/acl/ac2014-d.html#ManningSBFBM14>
- [28] Luke Melas-Kyriazi, Alexander Rush, and George Han. 2018. Training for Diversity in Image Paragraph Captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, Belgium, 757–761. <https://doi.org/10.18653/v1/D18-1084>
- [29] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.
- [30] Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2018. A Survey on Open Information Extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3866–3878. <https://aclanthology.org/C18-1326>
- [31] Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic̄, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. *arXiv:2004.10643 [cs.CL]*
- [32] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, 311–318. <https://doi.org/10.3115/1073083.1073135>
- [33] Injy Sarhan and Marco Spruit. 2018. Uncovering algorithmic approaches in open information extraction: A literature review. In *30th Benelux Conference on Artificial Intelligence*. Springer CSAI/JADS, 223–234.
- [34] Risto Savvas, Erick Herrarte, Anita Wilhelm, and Marc Davis. 2004. Metadata Creation System for Mobile Images. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services* (Boston, MA, USA) (MobiSys '04). Association for Computing Machinery, New York, NY, USA, 36–48. <https://doi.org/10.1145/990064.990072>
- [35] Xindi Shang, Donglin Di, Junbin Xiao, Yu Cao, Xun Yang, and Tat-Seng Chua. 2019. Annotating Objects and Relations in User-Generated Videos. In *Proceedings of the 2019 on International Conference on Multimedia Retrieval* (Ottawa ON, Canada) (ICMR '19). Association for Computing Machinery, New York, NY, USA, 279–287. <https://doi.org/10.1145/3323873.3325056>
- [36] Xindi Shang, Tongwei Ren, Jingfan Guo, Hanwang Zhang, and Tat-Seng Chua. 2017. Video Visual Relation Detection. In *Proceedings of the 25th ACM International Conference on Multimedia* (Mountain View, California, USA) (MM '17). Association for Computing Machinery, New York, NY, USA, 1300–1308. <https://doi.org/10.1145/3123266.3123380>
- [37] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification?. In *Chinese Computational Linguistics*, Maosong Sun, Xuanjing Huang, Heng Ji, Zhiyuan Liu, and Yang Liu (Eds.). Springer International Publishing, Cham, 194–206.
- [38] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The New Data in Multimedia Research. *Commun. ACM* 59, 2 (2016), 64–73.
- [39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaou Tang, and Luc Van Gool. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 20–36.
- [40] Teng Wang, Ruimao Zhang, Zhichao Lu, Feng Zheng, Ran Cheng, and Ping Luo. 2021. End-to-End Dense Video Captioning with Parallel Decoding. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (2021), 6827–6837.
- [41] Ziwei Wang, Yadan Luo, Yang Li, Zi Huang, and Hongzhi Yin. 2018. Look Deeper See Richer: Depth-Aware Image Paragraph Captioning. In *Proceedings of the 26th ACM International Conference on Multimedia* (Seoul, Republic of Korea) (MM '18). Association for Computing Machinery, New York, NY, USA, 672–680. <https://doi.org/10.1145/3240508.3240583>
- [42] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. 2016. Video Paragraph Captioning Using Hierarchical Recurrent Neural Networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4584–4593. <https://doi.org/10.1109/CVPR.2016.496>
- [43] Hengshun Zhou, Debin Meng, Yuanyuan Zhang, Xiaojiang Peng, Jun Du, Kai Wang, and Yu Qiao. 2019. Exploring Emotion Features and Fusion Strategies for Audio-Video Emotion Recognition. *2019 International Conference on Multimodal Interaction* (Oct 2019). <https://doi.org/10.1145/3340555.3355713>
- [44] Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J. Corso, and Marcus Rohrbach. 2019. Grounded Video Description. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 6571–6580. <https://doi.org/10.1109/CVPR.2019.00674>
- [45] Luowei Zhou, Chenliang Xu, and Jason J. Corso. 2017. ProcNets: Learning Segment Procedures in Untrimmed and Unconstrained Videos. *CoRR* abs/1703.09788 (2017). arXiv:1703.09788 <http://arxiv.org/abs/1703.09788>
- [46] Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. 2018. End-to-End Dense Video Captioning with Masked Transformer. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8739–8748. <https://doi.org/10.1109/CVPR.2018.00911>

Supplementary Materials

A TAGS AND DEPENDENCY LABELS

In this work, we use the Universal Dependencies POS tags and ClearNLP dependency labels when explaining how the framework’s semantic metadata extraction methods use linguistic annotations produced by the language parser (see Sec. 3). Table 12 and Table 13 list and name the POS tags and dependency labels mentioned in this work.

POS Tag	Description
NOUN	Noun
PRON	Pronoun
PROPN	Proper Noun
VERB	Verb
AUX	Auxiliary verb
ADV	Adverb
ADP	Adposition
ADJ	Adjective

Table 12: List of the Universal Dependencies [31] part-of-speech Tags used in this work. For a complete list refer to the Universal Dependencies web page¹⁸.

B EXTENDED RESULTS

The following sections contain extended experiment results. In particular, we provide the entities, entity-property pairs, and relations that are most frequently extracted by our framework for ActivityNet Captions videos, present further experimental results for dense video captioning and video category extraction with our framework, and evaluate whether the performance of the framework’s DVC model for a particular video category correlates with the framework’s classification performance on the same category.

B.1 Frequently Extracted Semantic Metadata

Table 14 shows the entities, entity-property pairs, and relations that are most frequently extracted by our framework for ActivityNet Captions videos. Recall that the frequency of a video-level semantic metadata instance is the number of different videos in which it occurs. For each, video-level entities (Table 14a), entity-property pairs (Table 14b), and video-level relations (Table 14c), we list the instances that occur most frequently in our corresponding generated gold standard, and the instances that are most frequently extracted by our framework for ActivityNet Captions validation videos when using MT [46] or PDVC [40] as dense video captioning model. For entities, we also list the gold standard entities whose names are compound nouns that were detected with our entity extraction method (see Section 3.4).

B.2 Dense Video Captioning

When implementing and training the dense video captioning models MT and PDVC, we changed few parameters compared to their original works. To ensure consistency, we evaluate both models in event localization and dense video captioning according to their original evaluation setup. Therefore, we set $|E| = 3$

¹⁸Universal Dependencies POS tags: <https://universaldependencies.org/u/pos/>.

¹⁹ClearNLP dependency labels: https://github.com/clir/clearnlp-guidelines/blob/master/md/specifications/dependency_labels.md.

for PDVC, while for MT the number of generated captioned events is not limited, and evaluate both models on the entire ActivityNet Captions validation set, using the official evaluation toolkits provided by ActivityNet Captions Challenge 2017 and 2018. Compared to the results in their original works, for MT and PDVC we achieved similar, and for dense video captioning even better, performances (Table 15). Since for the MT model the number of generated captioned events is not limited, it generates 210 captioned events per video on average. This explains its high recall and low precision event localization performance compared to PDVC, which results in worse dense video captioning performance (using the 2018 evaluation toolkit). For our purpose, we think it is fairer to evaluate the DVC models as we did in Section 5.1, i.e., to specify the number of captioned events that they generate.

B.3 Text Classification

Revisiting Table 9, it shows the weighted and macro-averages of precision, recall, and F1 score performances of our framework’s text classification method, when trained and evaluated in three different settings: using the captioned events generated by the MT and PDVC model, respectively, and using the captioned events provided by ActivityNet Captions. Taking the category imbalance in our classification dataset into account, we see that the achieved macro-averages of precision, recall and F1 scores are much lower as compared to their weighted-averages. Looking at the classification performance per video category when using the PDVC model (see Table 18), we can observe that the highest F1 scores are achieved for the categories Sport, Music, and Pets & Animals. However, when looking at the corresponding confusion matrix (see Figure 5), we see that, even though we use class weights during training to overcome class imbalance, the categories Film & Animation, Science & Technology, and Non-profits & Activism are not even predicted once. This explains the poor performance of our framework for macro precision, recall and F1 score. Similar observations can be made when captioned events are used for classification that are generated by the MT model (see Table 17 and Figure 4), or captioned events from ActivityNet Captions (see Table 19 and Figure 6).

B.4 Analysing DVC and Text Classification Performances per Video Category

In this work, we showed that the quality of semantic metadata extracted by our framework depends heavily on the event localization and dense video captioning performance of the framework’s DVC model. Finally, we now want to evaluate whether a similar statement applies to the classification of videos with our framework, i.e., whether the performance of the framework’s DVC model for a particular video category correlates with the framework’s classification performance on the same category. Note that if a DVC model performs well on a video category, this does not logically translate into better classification performance of our framework. For DVC, a model performs well when it is able to describe the events of a video with appropriate natural language sentences. Theoretically, the gold standard annotations for videos of two different categories could contain similar words that, nevertheless, describe the videos well. For successful video category prediction, however, videos of different categories need to be distinguishable through unique features. These can be, e.g., unique entities that occur only for videos of a specific category.

Dependency Label	Description	Explanation by ClearNLP
nsubj	Nominal Subject	Non-clausal constituent in the subject position of an active verb.
nsubjpass	Nominal Subject (pass.)	Non-clausal constituent in the subject position of a passive verb.
auxpass	Auxiliary (passive)	Auxiliary verb that modifies a passive verb.
agent	Agent	Complement of a passive verb that is the surface subject of its active form.
acl	Clausal Modifier	Either an infinitival modifier, i.e., a phrase that modifies the head of a noun phrase, or a participial modifier, i.e., phrase whose head is a verb in a participial form that modifies the head of a noun phrase, or a complement.
dobj	Direct Object	Noun phrase that is the accusative object of a (di)transitive verb.
pobj	Object of Preposition	Noun phrase that modifies the head of a prepositional phrase, which is usually a preposition, but can be a verb in a participial form.
prt	Particle	Preposition in a phrasal verb that forms a verb-particle construction.
prep	Prepositional Modifier	Any prepositional phrase that modifies the meaning of its head.
amod	Adjectival Modifier	Adjective or an adjective phrase that modifies the meaning of another word.
advmod	Adverbial Modifier	Adverb or an adverb phrase that modifies the meaning of another word.
compound	Compound	Either a noun modifying the head of noun phrase, a number modifying the head of quantifier phrase, or a hyphenated word.
conj	Conjunct	Dependent of the leftmost conjunct in coordination.

Table 13: List of the ClearNLP dependency labels used in this work. For a complete list refer to the ClearNLP Guidelines¹⁹.

Table 16 shows the dense video captioning performances of MT and PDVC with respect to the video category. For both models, we set $|dist(E)| = 10$, the same number of distinct captioned events that is used for the classification of videos. Both models perform best on videos from the categories Pets & Animals, Music, and Gaming. They also share Film & Animation and Non-profits & Activism as categories for which they have some of the lowest METEOR scores. The PDVC model achieves its second lowest METEOR score for How-to & Style videos, although 1,661 videos are available for this category, while How-to & Style is among the top 5 categories for the MT model. From this we can conclude that, except for a few categories, different DVC models have a similar ease or difficulty to generate appropriate captioned events for videos of a certain category.

To answer the initial question, we now compare the DVC performance of the MT model per video category (see Table 16a) with the F1 score performance of our text classification method per video category when trained and evaluated with captioned events generated by the MT model (see Table 17). MT achieves

best METEOR performances for the categories Pets & Animals, Music, and Gaming. Pets & Animals and Music are also among the categories where highest F1 scores for text classification are achieved. For Gaming videos however, only a F1 score of 15.38 is achieved for text classification. For the categories Film & Animation, Non-profits & Activism, and News & Politics the MT model achieves its worst METEOR performances. For these categories, also poor performances are achieved for text classification, with a F1 score of 16.22 for News & Politics, and 0.00 else. Thus, if the MT model performs well (or poorly) for a certain video category, our framework seems to achieve better (or worse) classification performance in terms of F1 score for this particular video category. Nevertheless, there are exceptions, e.g., MT achieves relatively poor DVC performance for Sport videos, while the best F1 score performance for text classification with captioned events generated by MT is achieved for the Sports category. Similar observations can be made when classifying videos with the captioned events generated by the PDVC model.

Video-level Entity Extraction							
Gold Standard Entity	Videos	(only Compound Nouns)		Framework with MT		Framework with PDVC	
		Gold Standard Entity	Videos	Entity	Videos	Entity	Videos
man	2,450	diving_board	31	camera	3,022	man	3,477
people	1,262	living_room	28	man	2,956	camera	3,124
camera	1,144	t-shirt	26	people	1,506	people	1,300
woman	1,082	bumper_car	25	woman	1,016	woman	1,083
person	679	lawn_mower	25	group	772	group	799
hand	574	christmas_tree	25	person	618	person	757
girl	517	balance_beam	22	pan	545	room	683
front	498	tennis_racket	22	movement	544	game	615
group	458	parking_lot	20	game	476	field	399
water	419	jump_rope	19	angle	463	chair	377

(a) Entities that occur most frequently in our gold standard, and the entities that are most frequently extracted by our framework for different DVC models. Furthermore, we list the most frequent gold standard entities that consist of a compound noun.

Property Extraction					
Gold Standard Entity [Property]	Videos	Framework with MT		Framework with PDVC	
		Entity [Property]	Videos	Entity [Property]	Videos
group [large]	126	angle [several]	463	shot [several]	344
people [several]	105	shot [several]	406	group [large]	306
man [young]	102	group [large]	387	shirt [black]	124
shot [several]	92	object [various]	132	clip [several]	98
people [other]	73	clip [several]	121	girl [little]	84
girl [little]	71	field [large]	94	girl [young]	71
girl [young]	69	gymnasium [large]	85	boy [young]	65
clip [several]	69	people [several]	64	shirt [blue]	58
time [several]	69	girl [young]	63	field [large]	53
motion [slow]	68	crowd [large]	38	people [several]	52

(b) Entity-property pairs that occur most frequently in our gold standard, and the entity-property pairs that are most frequently extracted by our framework for different DVC models.

Video-level Relation Extraction					
Gold Standard Relation	Videos	Framework with MT		Framework with PDVC	
		Relation	Videos	Relation	Videos
(man, speak, [to], camera)	137	(man, speak, [to], camera)	1,214	(man, speak, [to], camera)	1,684
(woman, speak, [to], camera)	92	(man, lead, [into], man)	587	(man, lead, [into], man)	1,273
(man, talk, [to], camera)	76	(woman, speak, [to], camera)	577	(woman, speak, [to], camera)	575
(man, hit, [], ball)	59	(camera, capture, [], movement)	488	(man, stand, [in], room)	312
(man, stand, [in], front)	51	(camera, capture, [from], angle)	463	(man, play, [], game)	312
(man, throw, [], ball)	39	(camera, capture, [], man)	226	(man, end, [with], man)	305
(people, watch, [on], side)	36	(man, play, [], game)	156	(man, lead, [into], shot)	250
(man, play, [], drum)	33	(group, lead, [into], group)	156	(man, begin, [to], camera)	219
(woman, talk, [to], camera)	31	(camera, capture, [], people)	153	(woman, lead, [into], woman)	219
(man, hold, [in], hand)	28	(man, lead, [into], clip)	152	(people, play, [], game)	214

(c) Relations that occur most frequently in our gold standard, and the relations that are most frequently extracted by our framework for different DVC models.

Table 14: Statistics on video-level semantic metadata that is most frequently extracted by our framework when using MT and PDVC as dense video captioning model. For both, MT and PDVC, we limit the number of generated distinct captioned events $|dist(M)|$ to 3. For comparison, we list the most frequent entities, entity-property pairs and relations of our corresponding gold standards. All entities, entity-property pairs, and relations are counted lemmatized.

Event Localization & Dense Video Captioning									
DVC Model	E	avg.		2017 eval. toolkit			2018 eval. toolkit		
		Recall	Precision	B@3	B@4	M	B@3	B@4	M
MT	-	91.48	32.11	5.47	2.63	10.21	2.65	1.30	5.19
PDVC	3	50.19	57.87	4.65	2.32	9.12	4.14	2.07	8.20

Table 15: Event localization and dense video captioning results on the ActivityNet Captions validation set. Both annotation files of the validation set are used for the evaluation. According to the setup in their original works, we set $|E|$ to 3 for PDVC, while for MT the number of generated captioned events is not limited.

Dense Video Captioning			Dense Video Captioning								
Category ID	Name	Videos	2018 eval. toolkit			Category ID	Name	Videos	2018 eval. toolkit		
			B@3	B@4	M				B@3	B@4	M
15	Pets & Animals	342	2.82	1.27	6.88	10	Music	637	4.25	2.31	7.40
10	Music	637	3.12	1.54	6.83	15	Pets & Animals	342	3.94	1.89	7.26
20	Gaming	78	3.26	1.46	6.22	20	Gaming	78	3.80	1.97	6.89
19	Travel & Events	526	2.57	1.18	6.02	2	Cars	191	3.13	1.45	6.86
26	How-to & Style	1,661	2.74	1.37	6.00	22	People & Blogs	2,098	3.46	1.71	6.72
22	People & Blogs	2,098	2.38	1.13	5.99	19	Travel & Events	526	3.38	1.68	6.69
23	Comedy	645	2.20	1.04	5.98	23	Comedy	645	3.26	1.57	6.67
24	Entertainment	1,414	2.56	1.28	5.97	24	Entertainment	1,414	3.30	1.69	6.61
27	Education	414	2.37	1.03	5.84	28	Science & Technology	146	3.20	1.57	6.45
2	Cars	191	2.19	0.94	5.84	17	Sport	3,920	3.45	1.78	6.43
17	Sport	3,920	2.47	1.24	5.83	25	News & Politics	226	3.50	1.83	6.36
28	Science & Technology	146	2.29	0.95	5.70	27	Education	41	3.19	1.54	6.26
1	Film & Animation	213	1.74	0.64	5.46	1	Film & Animation	213	2.63	1.12	6.07
29	Non-profits & Activism	68	2.25	1.02	5.42	26	How-to & Style	1,661	2.69	1.42	5.92
25	News & Politics	226	1.98	0.97	5.38	29	Non-profits & Activism	68	2.96	1.45	5.54
weighted avg. across categories		12,579	2.51	1.22	5.98	weighted avg. across categories		12,579	3.44	1.75	6.57

(a) Dense video captioning performance of MT per category.

(b) Dense video captioning performance of PDVC per category.

Table 16: Dense video captioning results of MT (a) and PDVC (b) per video category on ActivityNet Captions. Here, we evaluate on the train and validation set, while using only the first of the two annotation files of the validation set. This is the same data that we use for the classification of videos. For both, MT and PDVC, we set $|dist(E)| = 10$. The categories are sorted by the METEOR performances achieved on them.

Text Classification															
ID	Name		Videos	Prec.	Rec.	F1									
17	Sport		784	67.36	77.93	72.26									
10	Music		128	53.18	71.88	61.13									
15	Pets & Animals		69	52.69	71.01	60.49									
26	How-to & Style		333	47.33	71.77	57.04									
19	Travel & Events		106	28.85	42.45	34.35									
2	Cars & Vehicles		39	31.82	35.90	33.73									
22	People & Blogs		421	35.78	19.71	25.42									
24	Entertainment		284	28.65	17.96	22.08									
23	Comedy		129	16.33	18.60	17.39									
25	News & Politics		46	21.43	13.04	16.22									
20	Gaming		17	22.22	11.76	15.38									
27	Education		84	7.02	4.76	5.67									
1	Film & Animation		44	0.00	0.00	0.00									
28	Science & Technology		30	0.00	0.00	0.00									
29	Non-profits & Activism		15	0.00	0.00	0.00									
		weighted average		2,529	43.72	48.24	44.80								
		macro-average		2,529	27.51	30.45	28.08								

Table 17: Classification performance of our framework per video category when the text classifier is trained and evaluated with captioned events generated by MT. Categories are sorted according to the F1 score achieved for them.

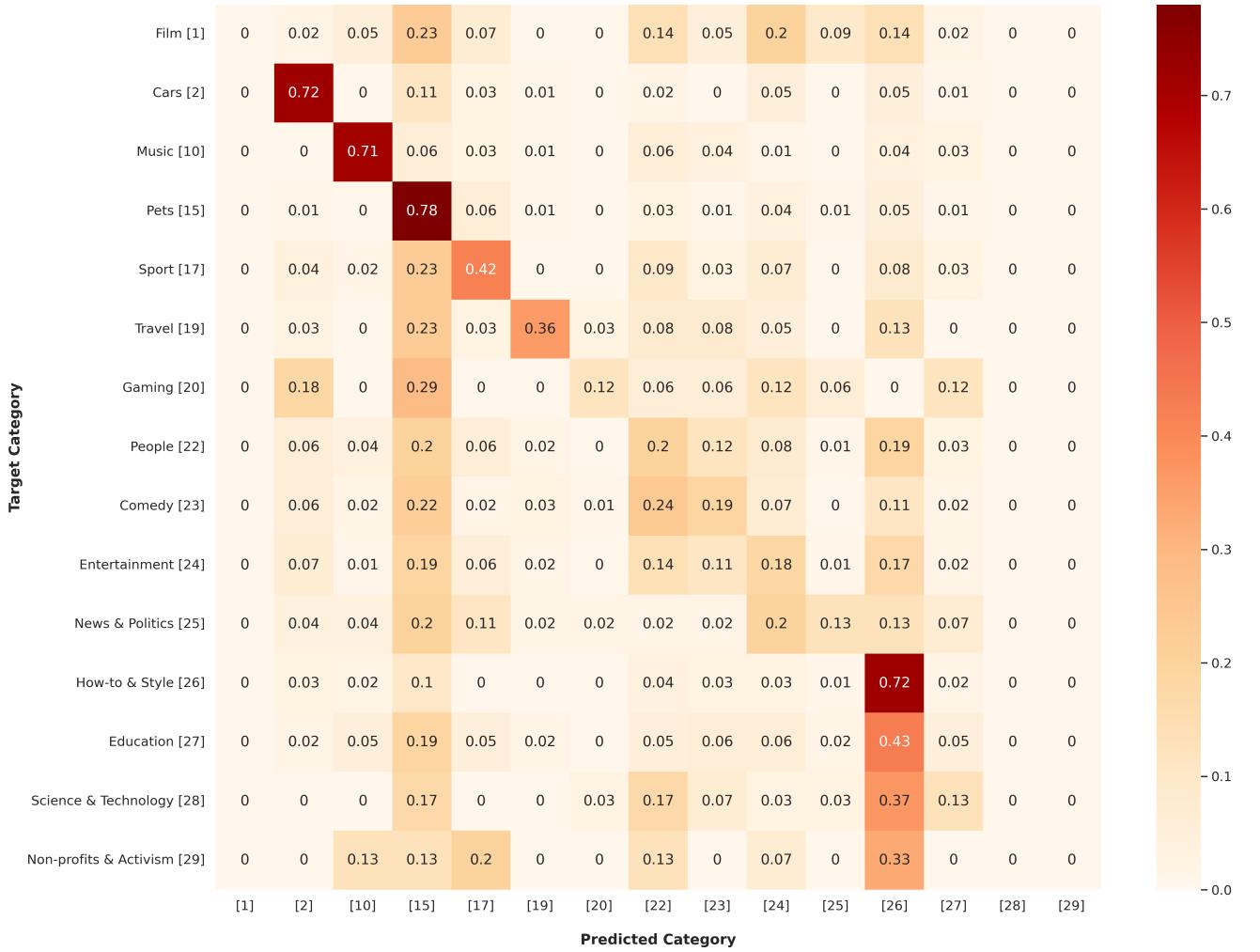


Figure 4: Confusion matrix for the classification method of our framework when trained with captioned events generated by MT.

Text Classification						
ID	Name		Videos	Prec.	Rec.	F1
17	Sport		784	68.00	82.65	74.61
10	Music		128	60.38	75.00	66.90
15	Pets & Animals		69	57.45	78.26	66.26
26	How-to & Style		333	50.00	66.37	57.03
2	Cars & Vehicles		39	37.50	38.46	37.97
19	Travel & Events		106	31.62	40.57	35.54
20	Gaming		17	100.0	17.65	30.00
22	People & Blogs		421	33.33	22.33	26.74
24	Entertainment		284	34.08	21.48	26.35
25	News & Politics		46	16.36	19.57	17.82
23	Comedy		129	19.63	16.28	17.80
27	Education		84	6.33	5.95	6.13
1	Film & Animation		44	0.00	0.00	0.00
28	Science & Technology		30	0.00	0.00	0.00
29	Non-profits & Activism		15	0.00	0.00	0.00
			weighted average	2,529	45.75	50.22
			macro-average	2,529	34.31	32.30
						46.96

Table 18: Classification performance of our framework per video category when the text classifier is trained and evaluated with captioned events generated by PDVC. Categories are sorted according to the F1 score achieved for them.

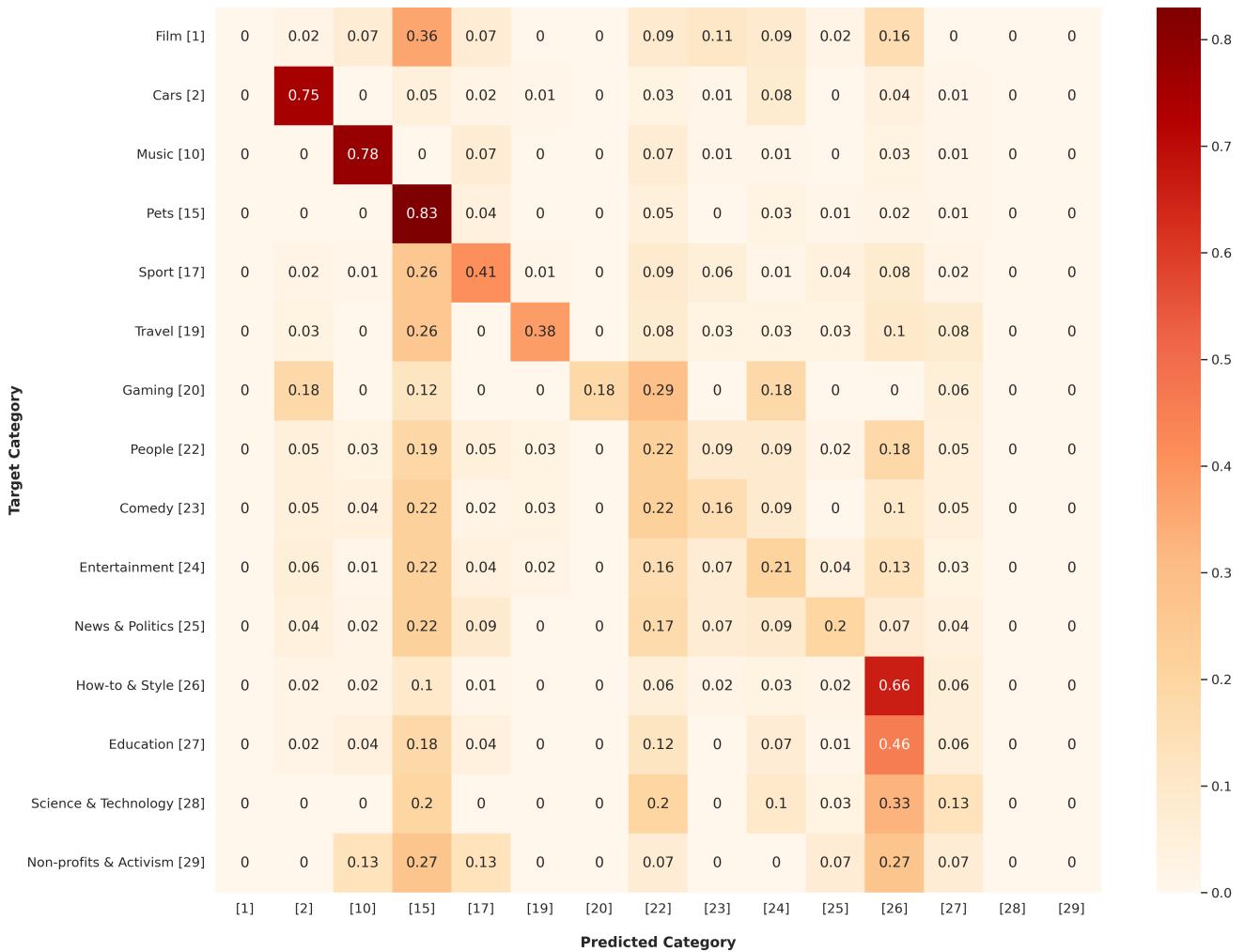


Figure 5: Confusion matrix for the classification method of our framework when trained with captioned events generated by PDVC.

Text Classification															
ID	Name		Videos	Prec.	Rec.	F1									
17	Sport		784	68.76	82.27	74.91									
15	Pets & Animals		69	59.78	79.71	68.32									
10	Music		128	58.82	70.31	64.06									
26	How-to & Style		333	55.91	62.46	59.01									
2	Cars & Vehicles		39	43.18	48.72	45.78									
19	Travel & Events		106	34.40	40.57	37.23									
25	News & Politics		46	32.61	32.61	32.61									
22	People & Blogs		421	32.70	24.47	27.99									
24	Entertainment		284	32.58	20.42	25.11									
23	Comedy		129	20.22	27.91	23.45									
20	Gaming		17	30.00	17.65	22.22									
27	Education		84	12.33	10.71	11.46									
28	Science & Technology		30	20.00	3.33	5.71									
1	Film & Animation		44	0.00	0.00	0.00									
29	Non-profits & Activism		15	0.00	0.00	0.00									
		weighted average		2,529	46.97	50.81	48.23								
		macro-average		2,529	33.42	34.74	33.19								

Table 19: Classification performance of our framework per video category when the text classifier is trained and evaluated with captioned events from ActivityNet Captions. Categories are sorted according to the F1 score achieved for them.

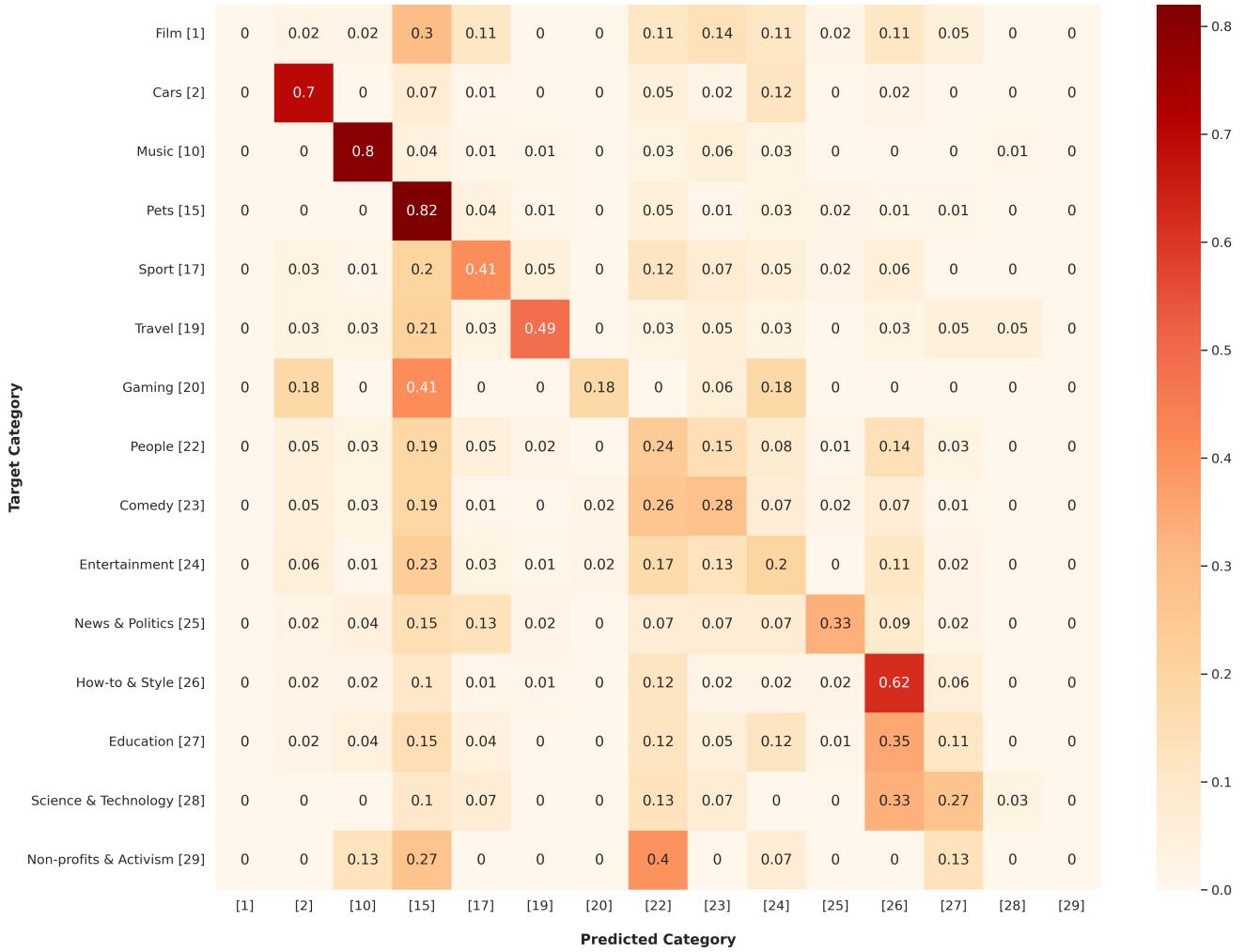


Figure 6: Confusion matrix for the classification method of our framework when trained with gold standard captioned events from ActivityNet Captions.

C USER'S GUIDE

This section describes how to use the implementation of our framework. In particular, we provide information on the installation of dependencies, and provide guidance on its usage.

Dense Video Captioning. Thanks to its modularity, any DVC models can be used for our framework. In this work, we make use of the MT [46] and PDVC [40] models. However, the source code that we provide for our framework does not include any implementation of the DVC models, mainly because the feature extraction of new videos, which are not contained in the ActivityNet Captions dataset, was not achieved with our hardware. Refer to the GitHub repositories MT²⁰ and PDVC²¹ for the source code of the used DVC models and extensive information about their installation and usage. Furthermore, refer to Section 4.3 for the hyperparameters that we used for training MT and PDVC. For both models, we generated captioned events for ActivityNet Captions train and validation videos for $|E|$ set to 10, 25, 50, 100, and $|dist(E)|$ set to 1, 3, 10, 25 (i.e., the numbers of (distinct) captioned events). For each such configuration of DVC model and number of generated captioned events, a JSON file is created containing the results for all videos. These JSON files are then forwarded to the remaining components of the framework for the extraction of semantic metadata, and the evaluation of our framework.

Semantic Metadata Extraction

The GitLab repository 2021ss-thesis-johannes contains the source code for the extraction and evaluation of semantic metadata from captioned events and all necessary data files. We provide information on the installation of dependencies, as well as guidance on how to use the provided files to extract semantic metadata and evaluate our framework.

Installation.

We recommend to use Anaconda²² for package management. In order for the semantic metadata extraction methods to work, python3 must be available. As mentioned in Section 4, we are bound to use version 2.3.7 within spaCy since NeuralCoref, a pipeline extension for spaCy which we use for coreference resolution, is not compatible with versions 3.0.0+ yet. After installation of Anaconda, run the following commands in the repository's root directory to install all necessary dependencies:

```
# Creating conda environment
> conda create -n Video2Metadata python=3.7
> conda activate Video2Metadata

# Install spaCy with NeuralCoref
> cd Libraries
> git clone \
  https://github.com/huggingface/neuralcoref.git
> cd neuralcoref
> pip install -r requirements.txt
> pip install -e .
> cd ../../

# Download the spaCy language model
> python -m spacy download en_core_web_lg
```

```
# Install WordNet
> conda install -c anaconda nltk

# Only necessary when you want to query categories
# from the YouTube Data API
> conda install -c conda-forge \
  google-api-python-client
```

If errors occur when installing spaCy with NeuralCoref, then run the following commands:

```
# Remove and create conda environment
> conda deactivate
> conda env remove -n Video2Metadata
> conda create -n Video2Metadata python=3.7
> conda activate Video2Metadata

# Install spaCy with NeuralCoref, and download
# the spaCy language model
> pip install spacy==2.1.0
> pip install neuralcoref
> python -m spacy download en_core_web_lg

# Install WordNet
> conda install -c anaconda nltk

# Only necessary when you want to query categories
# from the YouTube Data API
> conda install -c conda-forge \
  google-api-python-client
```

With the first sequence of commands, NeuralCoref is installed from source, while the second command sequence installs NeuralCoref via pip. NeuralCoref can be installed via pip only when spaCy version is 2.1.0 or lower. With spaCy 2.1.0, however, the language parser operates much slower. Therefore, the first installation process is to be preferred.

Files and Usage.

We now explain the repository's files and how they can be used to extract semantic metadata and to evaluate the extraction methods of our framework.

- example_extraction_from_text.py

Run this program to extract and print video-level entities, properties, and video-level relations from a given input text. Text can be passed with the text option. Example:

```
> python ./example_extraction_from_text.py \
--text "A man is standing in front of a \
fridge. He opens it and takes out a red glass."
```

- examples.py

To test the entity, property, and relation extraction methods on given captioned events, you can edit the script's examples at own will. Running this script with the following command will print the extracted video-level and event-level entities, entity-property pairs, and video-level and event-level relations:

```
> python example.py
```

- data_dvc_output/

As mentioned above, we do not provide any source code for the DVC models. Instead, we generated captioned events for all ActivityNet Captions train and validation videos with both models and saved those to JSON files. This directory contains the generated (distinct) captioned events for $|E|$ set to 10, 25, 50, 100, and

²⁰MT is available at: <https://github.com/salesforce/densecap>.

²¹PDVC is available at: <https://github.com/ttengwang/PDVC>.

²²Anaconda is available at: <https://www.anaconda.com/>.

$|dist(E)|$ set to 1, 3, 10, 25. `MT_train_and_val_10_no_dup.json`, e.g., contains 10 distinct captioned events per ActivityNet Captions train and validation video generated with the MT model.

- `dataset_activity_net_captions/`

Contains the ActivityNet Captions dataset²³, which is used for training of the DVC models and generating the gold standards for entity, property, and relation extraction.

- `Libraries/`

This directory contains Python files regarding the implementation of the semantic metadata extraction methods of Section 3, i.e., the algorithms for entity, property, and relation extraction (`entities_lib.py`, `relations_lib.py`), class definitions of entities, properties, and relations (`entity.py`, `entity_property.py`, `relation.py`), a file for building the spaCy language parser (`nlp_lib.py`), and a file that contains logic regarding WordNet (`wordnet_lib.py`).

- `Semantic_Metadata_Extraction/`

`Semantic_Metadata_Extraction/scripts/` contains all scripts for generating and evaluating the results for entity, property, and relation extraction presented in this work. Among others, the most important scripts are:

1) `extract_sm.py`: Extracts entities, entity-property pairs, and relations from the input captioned events. All results are saved to `Semantic_Metadata_Extraction/output_data/`. Using this script, the gold standards for entities, properties, and relations (see Section 4.1) can be generated using the following command:

```
> python extract_sm.py --type gt --id GT
```

To extract semantic metadata from generated captioned events, e.g., from 10 distinct captioned events generated with the PDVC model, the following command is used:

```
> python extract_sm.py --type submission \
--submission ../../data_dvc_output/PDVC_ \
train_and_val_10_no_dup.json --id PDVC_10_no_dup
```

2) `evaluate_event_entities.py`: After semantic metadata for entities, properties, and relations is now generated, we can evaluate all extraction methods. More generally, we can evaluate the capability of our framework for entity, property, and relation extraction. The performance for event-level entity extraction is evaluated using this script. By specifying the gold standard, the submission file, and the ActivityNet Captions set(s) on which to evaluate, we can measure precision, recall, and F1 score performances of our framework. Example:

```
> python evaluate_event_entities.py --gt \
./../output_data/sm_GT.json --submission \
./../output_data/sm_MT_10.json --split val
```

3) `evaluate_event_relations.py`: Evaluate event-level relation extraction. Example:

```
> python evaluate_event_relations.py --gt \
./../output_data/sm_GT.json --submission \
./../output_data/sm_PDVC_25.json --split val
```

4) `evaluate_properties.py`: Evaluate property extraction. Example:

```
> python evaluate_properties.py --gt \
./../output_data/sm_GT.json --submission \
./../output_data/sm_PDVC_25_no_dup.json \
--split val
```

5) `evaluate_video_entities.py`: Evaluate video-level entity extraction. Example:

```
> python evaluate_video_entities.py --gt \
./../output_data/sm_GT.json --submission \
./../output_data/sm_PDVC_10_no_dup.json \
--split val
```

6) `evaluate_video_relations.py`: Evaluate video-level relation extraction. Example:

```
> python evaluate_video_relations.py --gt \
./../output_data/sm_GT.json --submission \
./../output_data/sm_MT_3_no_dup.json \
--split val
```

7) `build_frequent_entities.py`: Needed to evaluate video-level entities. Calculates the frequency of entities in the ActivityNet Captions dataset (see Table 2).

```
> python build_frequent_entities.py
```

- `dataset_activity_net_entities/`

Contains the ActivityNet-Entities dataset²⁴, which is used in this work to analyse how many entities that are extracted by our framework for ActivityNet Captions videos are actually not visible in the video (see Section 6.2).

- `dataset_classification/`

Contains our dataset for video classification, i.e., the video categories of ActivityNet Captions videos, queried from the YouTube Data API, and the train, validation, and test splits (see Sec. 4.1).

- `Text_Classification/`

As opposed to the methods for entity, property, and relation extraction, the text classification method of our framework is implemented and evaluated in `classify_text_with_bert.ipynb`, i.e., a single Jupyter Notebook. Note that we ran the experiments for text classification in Google Colab²⁵. Therefore, to run the experiments on a local machine, minor changes to the code are necessary. This concerns the locations of used files (classification dataset, generated captioned events), and also the installation of dependencies. The text classification method requires the installation of dependencies different from those named above and is handled in the Jupyter Notebook.

- `mini_gold_standards/`

Contains the manually annotated gold standard of 25 ActivityNet Captions videos with the video-level entities, entity-property pairs, and video-level relations that we expect the semantic metadata extraction methods to extract from captioned events (see Sec. 6.2).

²³ActivityNet Captions: <https://cs.stanford.edu/people/ranjaykrishna/densevid/>.

²⁴ActivityNet-Entities: <https://github.com/facebookresearch/ActivityNet-Entities>.

²⁵Google Colab: <https://colab.research.google.com/>.