A Modular Interface for Multimodal Data Annotation and Visualization with Applications to Conversational AI and Commonsense Grounding

Anonymous
Institution
City, State
anonymous@email.com

Anonymous
Institution
City, State
anonymous@email.com

Anonymous
Institution
City, State
anonymous@email.com

ABSTRACT

Artificial Intelligence (AI) research, including machine learning, computer vision, and natural language processing, requires large amounts of annotated data. The current research and development (R&D) pipeline involves each group collecting their own datasets using an annotation tool tailored specifically to their needs, followed by a series of engineering efforts in loading other external datasets and developing their own interfaces, often mimicking some components of existing annotation tools. We present a modular annotation, visualization, and inference software framework for computational language and vision research. Our framework enables researchers to set up a web interface for efficiently annotating language and vision datasets, visualizing the predictions made by a machine learning model, and interacting with an intelligent system. In addition, the tool accommodates many of the standard and popular visual annotations such as bounding boxes, segmentation, landmark points, temporal annotation and attributes, as well as textual annotations such as tagging and free form entry. These annotations are directly represented as nodes and edges as part of the graph module, which allow linking visual and textual information. Extensible and customizable as required by individual projects, the framework has been successfully applied to a number of research efforts in human-AI collaboration, including commonsense grounding of language and vision, conversational AI, and explainable AI.

CCS CONCEPTS

• **Human-centered computing** → **HCI**; • **Computing methodologies** → *Artificial intelligence*; • **Applied computing** → *Document management and text processing*.

KEYWORDS

HCI; explainable AI; conversational AI; commonsense grounding; multimodal annotation; language and vision

ACM Reference Format:
Anonymous, Anonymous, and Anonymous. 2018. A Modular Interface for Multimodal Data Annotation and Visualization with Applications to Conversational AI and Commonsense Grounding. In *Proceedings of Graphics Interface '20*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The integration of language and vision in machine learning applications has recently yielded new architectures and enabled applications areas which were not even considered just years before. This has lead to the emergence of a number of exciting research areas: image captioning [9], dense captioning using paragraphs [6], visual denotations of linguistic expressions [16], visual question answering in images [1, 7], grounding referring expressions in images [11], visual storytelling using images and text sequences [4], visual question answering in movies [21], describing situations in movies using vision and language [22], movie scene description [17], textually annotated cooking videos [26], and text and image correspondences [2].

With the rising interest in language and vision research and applications, there has been a plethora of new datasets released to the public. These datasets are annotated such that they highlight the visual and textual correspondence, referrals, context, questions and answers, or narrative. Annotating and visualizing this information requires a lot of effort: each research group ends up building a new tool to collect annotations which requires multiple iterations, is prone to failures and defects in software development, and is costly. There exist multiple text-only or vision-only open-source web annotation tools and services, however, to the best of our

Graphics Interface '20, May 21–22, 2020, Toronto, Canada
© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

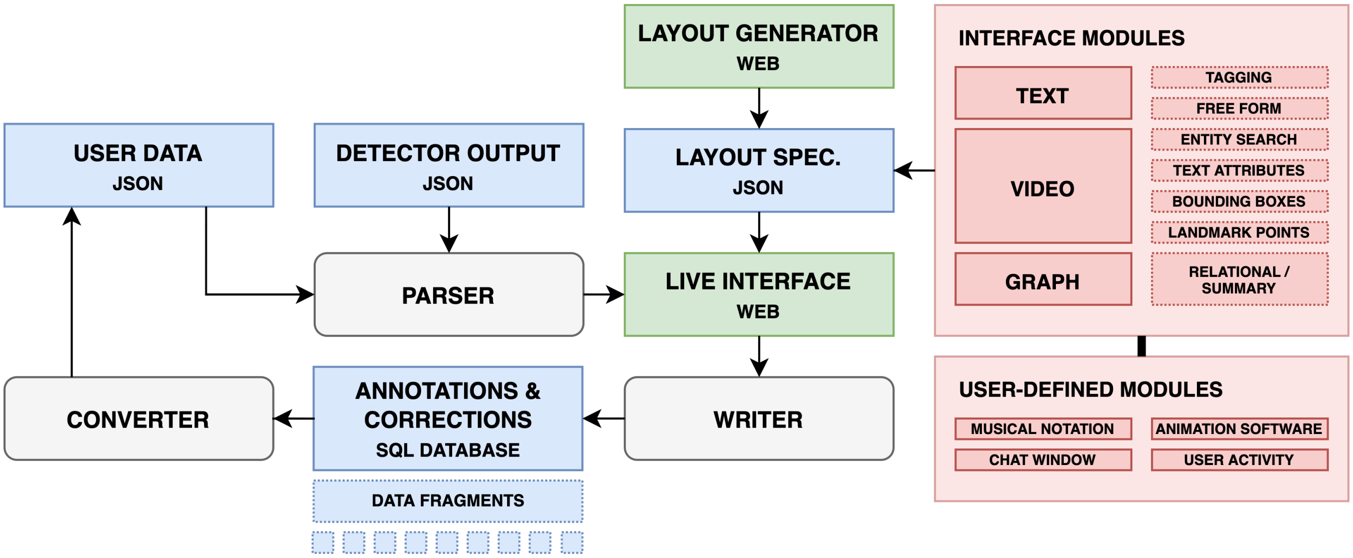


Figure 1: Our web-interface consists of text, video, and graph modules, which are arranged as a JSON-based specification file. The modules interact with a backend that loads the data schema and saves and loads annotations, which can be exported as a single output file. The user can also create and inject new project-specific modules as necessary.

knowledge, no joint text and vision annotation tool exists that would serve the language and vision community.

Furthermore, tools for dataset annotation are usually distinct from interfaces to machine learning systems — they are made for two different user groups: annotators and users of machine learning. However, many of the interface approaches are similar, and a common approach is possible. Recognizing the present gaps between dataset annotation interfaces and machine learning visualization solutions, we present a modular annotation and visualization platform that enables researchers to rapidly set up an interface for annotating new datasets and visualizing predictions made by a machine learning model. Our platform enables many of the standard and popular visual annotations such as: bounding boxes, segmentation, landmark points, temporal annotation and attributes, as well as textual annotations such as: tagging, weighted highlights, and free form entry. Relating visual and textual tokens is achieved using a 2D graph module. The spatio-temporal 2D graph tool is used to annotate and display an abstraction of the visual and textual annotations as well as annotation summary.

2 RELATED WORK

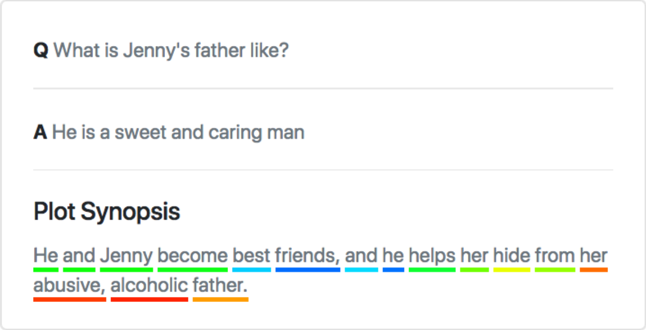
Existing annotation tools fall into three general categories: text annotaton, pixel annotation (for images and videos), and multi-modal tools. We will discuss each type in this section.

Text Annotation Tools

There exist open-source, web-based, text tagging, annotation, and visualization tools such as BRAT [20], Webanno [3], and Knowtator [13]. BRAT provides features for structured annotations, with fixed-form text. Webanno provides a focused set of features for linguistic annotations such as morphological, syntactical, and semantic annotations with a multi-user interface. Its multi-user feature enables measurement of inter-annotator agreement, with the ability to review, accept, reject, or modify the annotations. Knowtator offers an ontology integration with Protégé, which enables inputting hand-crafted ontologies for a domain-specific annotation task. These tools focus only on text and each of them provide a different set of features to facilitate annotation.

Pixel Annotation Tools

Similarly, there are multiple open-source, web-based, image annotation tools such as LabelMe [18], Annotorious [19] and video annotation tools such as LabelMe Video [25], and VATIC [23]. LabelMe offers a flexible tool to annotate objects in images using polygons and a single class label. Annotorious offers an image annotation tool with a bounding box or polygons options associated with free-form text annotation. Similarly, LabelMe Video offers a segmentation level annotation associated with class labels, while VATIC offers bounding boxes associated with class labels and attributes.



Q What is Jenny's father like?

A He is a sweet and caring man

Plot Synopsis

He and Jenny become best friends, and he helps her hide from her abusive, alcoholic father.

Figure 2: Our interface enables tagging specific words. Here, the user is asked to inspect the question-and-answer pair, along with a series of relevant word tokens and visualized importance weights.

Multi-modal Annotation Tools

ELAN [8] and NOVA [24] enable non-verbal communication coding. ELAN supports audio-visual-textual data, while NOVA supports audio, face, and gestures. Both interfaces do not support bounding boxes or polygons nor link the multimodal data together, except through a time-line annotation and tagging. The recently published Universal Data Tool [5] features a large set of annotation methods, ranging from bounding box and audio clips, but does not allow for any multimodal experience, forcing the users to dedicate a project to a specific annotation type.

None of the aforementioned tools enables joint text-, pixel-, and graph-based annotation. Each of these tools addresses a very specific line of research and thus cannot serve projects that cut across both computational language and vision.

3 INTERFACE MODULES

In this section, we specify the different modules of our interface. Our interface hosts a number of independent, context-agnostic components that can be freely arranged and combined as the research and dataset need arises. These are organized into three distinct categories: text, video/image, and graphs. These modules offer a variety of ways to annotate a dataset or visualize results, and the user may create custom modules specific to project needs. The starting page is a layout generator that produces a JSON-based specification file that defines the specific use-case schema. Figure 1 summarizes our system and in the following sections we will elaborate on each component.

Text Modules

Consisting of simple word token tagging and free-form text input components, our text module enables each user to mark different terms or create comments pertaining to a specific part of a text-based dataset. These user-provided annotations can also introduce an element of collaboration (or contention) as the interface offers a visual indication of previously tagged terms, as well as a visualization of agreement amongst uploaded entries.

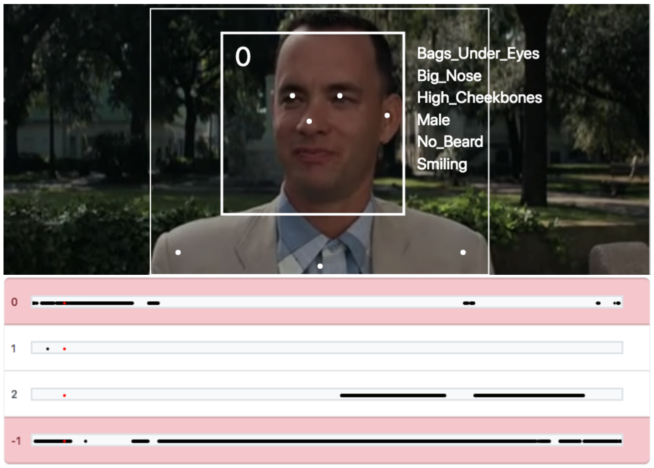


Figure 3: Our video module (top) with the entity seek module (bottom) displaying different entities present in the video and their temporal annotation. Each bar in the entity seek module indicates the presence of individual entities, marked by their index numbers, across the timeline.

Tagging. The tagging component, illustrated in Figure 2, tokenizes individual words, paragraphs, and sentences to enable token-based tagging across the interface. Upon discovering a notable word or phrase in the dataset, the user can click to select one or more individual tokens. The component also captures other types of metadata such as time stamps or presence of other interface modules for persistent storage. Our module also enables colored tokens to allow for importance annotation or visualization of output data from probabilistic models producing importance weights per token, for example, models that incorporate learned attention.

Free-Form. This component allows the user to submit free-form text entries that further annotate or describe the dataset. Each submission serves as an accompanying annotation to token tags or as a standalone comment, and contains the same set of comprehensive metadata as token tags.

Video Modules

The video module features a full-motion video player with a set of interactive overlay components. The user can activate each component to reveal more insights pertaining to the video, and further interact with individual entities to insert annotations or correct the original dataset.

Bounding Boxes. Visual entities, ranging from algorithmically detected objects to manually annotated counterparts, can be represented as 2D bounding boxes illustrated in Figure 3. Each box displays above the player component, moving in real-time along with the video. Based on the user-provided parameters, each box may be marked in a different color, display in a varying opacity, or contain text-based labels such as attributes. The user can also directly interact with bounding boxes to adjust their size or position, edit their labels, or create additional boxes as necessary.

Polygons. The user can create and modify polygonal annotations. Taking cue from popular graphics editor applications, the component allows the user to click on specific parts of the frame to construct a polygon and place it upon an object. The generated polygon serves the same function as bounding boxes, and will remain as a persistent annotation that supports spatial interpolation between keyframes.

Landmark Points. The video module also offers the ability to visualize a group of anchor points over video illustrated in Figure 3. Suitable for representing skeletal and facial feature tracking data, the resultant overlay dots can also vary in their opacity, size, and color as per user-specified parameters, and are also available for user modifications and annotations.

Text Attributes. In addition to visual entities that display (and move) in synchronization with on-screen objects, text-based overlay options are available for the user. Text-based subtitles and captions coincide with dialogues in the video, and offer the user the same degrees of interaction as the token tagging component found in the text module: the user can click on one or more individual word tokens to simply mark as notable or annotate with free form text comments.

Entity Seek. This module allows the user to look for occurrences of a certain entity across the video timeline illustrated in Figure 3. Characterized by a timeline visualization situated below the video progress bar, the feature displays an on-screen or in-script presence of an entity with a series of dots, indicating that the user can "scrub" the video to a specific point of the timeline to discover that particular entity. As the individual timeline components "light up" when the corresponding entities are displayed on-screen, the module also allows the user to easily identify co-occurrence of two or more distinct entities.

Graph Module

This module serves as a versatile method of generating 2D node-link graphs in direct relation to on-screen visual entities and/or text token nodes. Each generated graph can be presented as a single static image or a spatio-temporal animation displayed in synchronization with video playback.

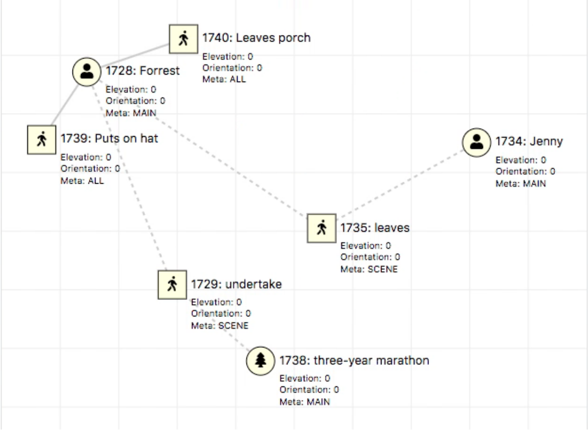


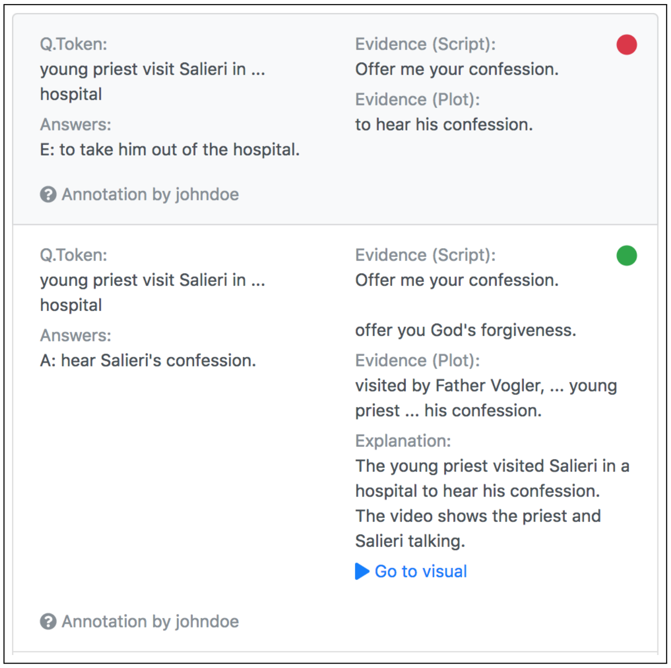
Figure 4: 2D graphs visualizing the relationship of different on-screen entities and text token nodes using nodes, icons, and edges.

Utilizing the popular SVG standard, the 2D graph component features an ability to generate vector-based network graphs that visualize relationships between different entities, including individual on-screen visual tokens and text tokens, in the dataset illustrated in Figure 4.

Each entity is represented as a node, with its various visual attributes — including size, opacity, color, and icon — mapped to user-defined characteristics of the corresponding entity. Two or more nodes may be linked using one or more path objects, each equipped with its own set of modifiers: path type (dotted, solid), direction (bidirectional, unidirectional, or non-directional), and a text-based annotation. Finally, the resultant 2D graph can visualize hierarchical information by using a tree-like approach: each of the larger, main nodes can have a series of child nodes, which in turn have the capacity to have children of their own.

While each node in the graph can be placed randomly in the canvas, the user may toggle a trigger that maps the position of each node to the corresponding entity in a video clip allowing the graph to capture the on-screen spatio-temporal information as well.

Instead of simply inspecting the resultant graph in a passive manner, the user may actively interact with nodes and links to induce changes to the dataset. The user may click and drag a child node and simply migrate it from one parent node to another in order to swap the two entities' characteristics at the dataset level. Alternatively, the user may remove a link between two nodes to sever the relationship between the two entities, or reverse the direction of the inter-node link to update the nature of the relationship.



Q.Token:
young priest visit Salieri in ... hospital

Answers:
E: to take him out of the hospital.

Evidence (Script):
Offer me your confession.

Evidence (Plot):
to hear his confession.

Annotation by johndoe

Q.Token:
young priest visit Salieri in ... hospital

Answers:
A: hear Salieri's confession.

Evidence (Script):
Offer me your confession.

offer you God's forgiveness.

Evidence (Plot):
visited by Father Vogler, ... young priest ... his confession.

Explanation:
The young priest visited Salieri in a hospital to hear his confession. The video shows the priest and Salieri talking.

▶ Go to visual

Annotation by johndoe

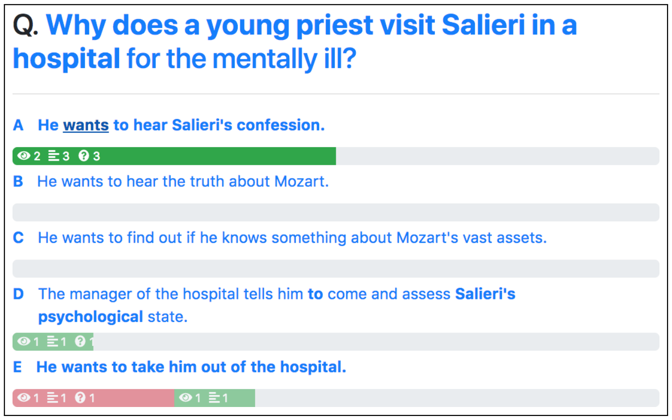
Figure 5: A user-defined module that provides a list of annotations created by different annotators.

User-Defined Module

Beyond the original offerings of our interface, the user may extend an existing module to perform new functions or define entirely new modules that are specific to project needs. Below are some of the user-defined modules that rely on previously created user annotation and provide data summary.

Summary View. With all user-provided annotations stored in a database table, the module embraces and promotes collaboration and collective efforts by visualizing the summary of prior user activities and the level of agreement amongst the participants. The module also features a component that dynamically generates a simple network visualization of relationships among the user annotations, and also offers a download link that allows the user to download all the accumulated annotations in a JSON format.

User Activity. Upon entering the interface, the user can immediately identify the in-text keywords that have been annotated or tagged by other participants. Each previously-tagged token is displayed with varying font weight (default), opacity or color saturation, alerting the user to how popular (or unpopular) that token is. The user can then click on the token to reveal all submissions associated with the term illustrated in Figure 5. The existing video player module may be extended to visualize the level of user activity across the timeline, indicating more popular (or contentious) points of each clip.



Q. **Why does a young priest visit Salieri in a hospital** for the mentally ill?

A **He wants to hear Salieri's confession.**

B He wants to hear the truth about Mozart.

C He wants to find out if he knows something about Mozart's vast assets.

D The manager of the hospital tells him **to** come and assess **Salieri's psychological** state.

E **He wants to take him out of the hospital.**

Figure 6: A user-defined summary module, where green indicates entries that support a specific question-answer pair, while red indicates those that serve as refuting evidence.

Opacity or saturation of each color block represents varying degrees of user activity at a more granular level.

Level of Agreement. This module also provides a more thorough visualization of user submissions, clustered by their response categories. Each bar, illustrated in Figure 6 , below the corresponding statement displays the makeup of associated responses, allowing users to quickly identify the popular opinion (or lack thereof) pertaining to each statement.

4 BACK END

Built with simplicity and extensibility in mind, our interface relies on a series of popular core web technologies with little dependence on niche plugins. The Bootstrap framework serves as a basis to the various interface modules, while PHP and MySQL serve as backend supports that handle record storage and retrieval. Finally, JavaScript and jQuery serve an instrumental role in integrating the various interface modules into a cohesive experience.

General Data Structure. The JSON files are structured to support weighted text tokens, free-form text, bounding boxes, text attributes, and landmark points.

Weighted text tokens consist of the token identifier along with the associated weight. In the case of un-weighted tokens, the value of the associated weight is simply set to "null". Associated free-form text is saved directly as a string.

Bounding boxes consist of the (x,y) coordinates of the box's upper-left corner, the box's width and height, a label for the object, along with a confidence score (automatically generated by a machine learning model) and tracked identity.

Landmark points for facial or body pose comprise a list of (x,y) coordinates along with the confidence value for each point in addition to a confidence score (provided by a pose estimation algorithm), tracked identity, and attributes.

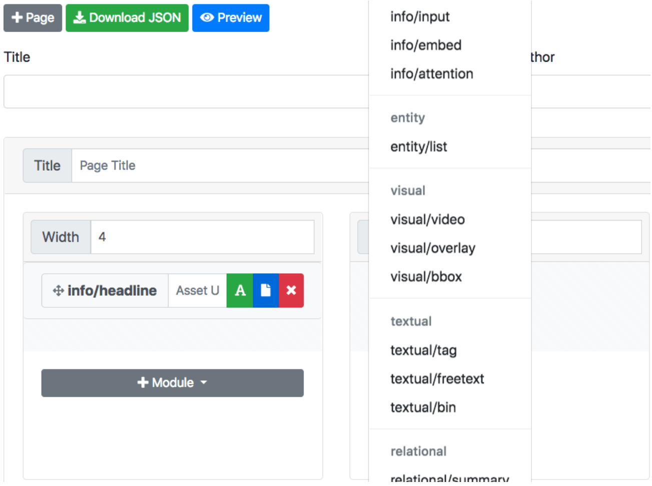


Figure 7: Each interface iteration can be easily constructed and customized using our web-based layout generator, eliminating the need for ad hoc web development and easing the initial learning curve.

Database. While the user is expected to directly manipulate the visible data using the various interface modules, the underlying JSON file remains intact without any permanent, irreversible changes. Instead, the interface pushes an incremental change, or a "delta," to the database table.

Each delta entry consists of three main components: the timestamp, the session identifier (which doubles as a username), and the JSON text fragment designed to replace the original counterpart. Upon detecting the user's interaction with the applicable entity, whether it be a text token or a bounding box, the interface creates a copy of the underlying data object's schema. This schema is then populated with the user-generated values, and then committed to the table.

When initializing the interface, the database module loads all the incremental changes and the original dataset into the memory. Upon completion, the interface proceeds with merging the two datasets by "injecting" each relevant delta into the dataset, producing a merged version for the interface to reference. This process takes place regularly under the hood, as the user continues to make corrections and annotations.

Data Exporter / Loader. Similar to version control systems such as Subversion or Git, this feature allows the user to identify the differences and revert to previous annotations or original data and download all the accumulated annotations, along with the schema supporting them, in a JSON format. Once the user acquires the file, they would be able to load it back to the interface to visualize, modify or augment.

Usability. While each iteration of our interface can be manually constructed using a JSON-based layout specification file, our layout generator interface eases the burden of web development for typical lay-users illustrated in Figure 7. Inspired by website building tools such as Squarespace and Wix, the layout generator allows the user to customize the placement and the size of individual modules, and also dictate how the whole experience unfolds using pagination. Using the interface, each user can conveniently create a simple annotation tool, visualize datasets, or deploy a complex user study.

	Annotation	Visualization	User Studies
Text	Word Token Tags	Language Parser	Freetext Entry
Video	Landmark Points	Entity Detector Overlay	AI-generated Clips
Graph	Text-Visual Links	Spatio-Temporal Links	Audience Clustering

Table 1: Sample manifestation of interface modules applied to various research efforts, including conversational AI, explainable AI, and commonsense grounding.

Extensibility. With the source code to be made available on Github, we also invite other developers to contribute new interface modules to our repository. Each module consists of HTML (element), CSS (style), and JS (behavior) files, and upon passing a series of checks, can be readily added to the user-facing interface. HTML and CSS files work to define the look-and-feel of an individual module, while the more complex JS file is responsible for establishing the available actions in each module, as well as its behavior in relation to other co-located modules. For example, the user may build a custom module that predicts visual interestingness of a specific image, and ensure that this module refreshes and refers to each frame of the video clip should the video player be present in the layout.

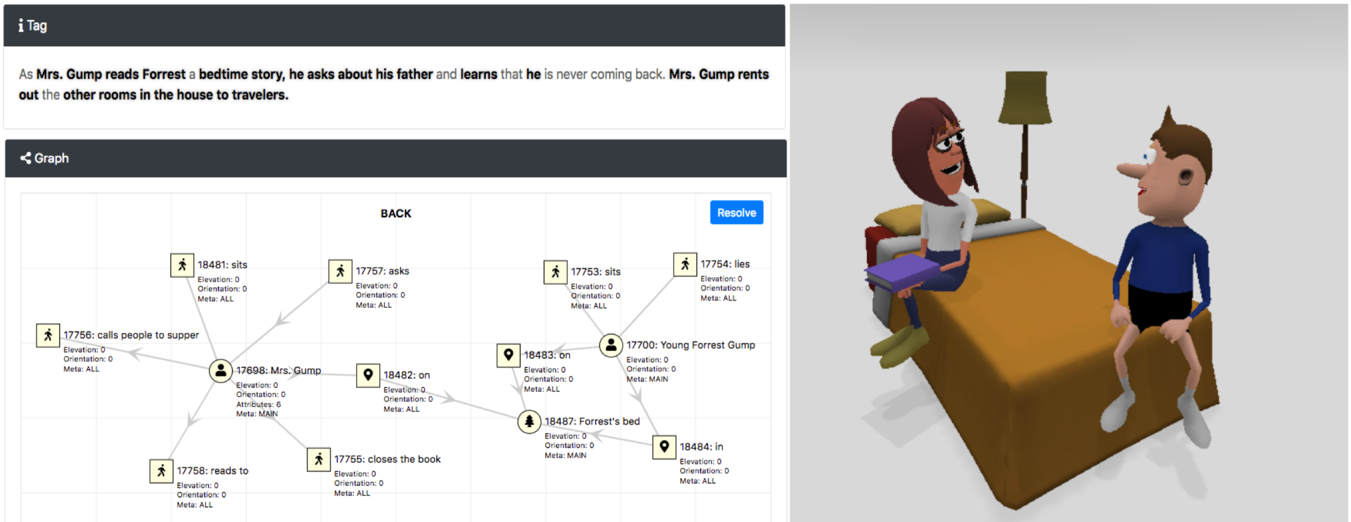
5 USE CASES

Described in Table 1, this framework can applied to a number of different aspects in interdisciplinary research efforts across visualization, annotation, and user studies.

Visualization. The user can build an interactive visualization of available datasets or results with ease. After constructing the layout and inserting the necessary modules, the user can attach a JSON-based dataset or results, a plain text file, or a video file to each relevant module. The resultant interface will automatically load the assets as specified in the modules.

Annotation. Beyond passive visualizations, the user can actively interact with the modules and create new annotations to build a new dataset or contribute to an existing one. The user can watch a video, identify a series of on-screen entities, and create a series of bounding boxes. All the activities are recorded and become available for download as a JSON file.

User Study. Our toolbox also supports a lengthier, more complex experience where the user is guided through a series



As Mrs. Gump reads Forrest a bedtime story, he asks about his father and learns that he is never coming back. Mrs. Gump rents out the other rooms in the house to travelers.

Figure 8: An example of grounding a scene (left). The sentence is about "Mrs. Gump reads Forrest a bedtime story." This would entail that Mrs. Gump and Forrest is in a bed, most likely sitting or laying on bed, with Mrs. Gump holding a book and reading from it. Grounding such knowledge graphs, created from textual and visual knowledge, augments the knowledge of the AI clarifying the implicit information. A user-define module, powered by an external animation application, is used to visualize the AI system's understanding of the original scene (right).

of different annotation, inference, and analysis tasks. Spanning multiple pages and equipped with a variety of editorial content, the interface presents an opportunity for deploying large-scale user studies without deep technical knowledge.

6 APPLICATION

In this section, we describe various projects in which our framework served as a core component in gathering annotations, visualizing AI inference, and promoting human-AI collaboration. These projects have been recently featured in AI Communication [12] and ECCV [10], with more publication plans underway.

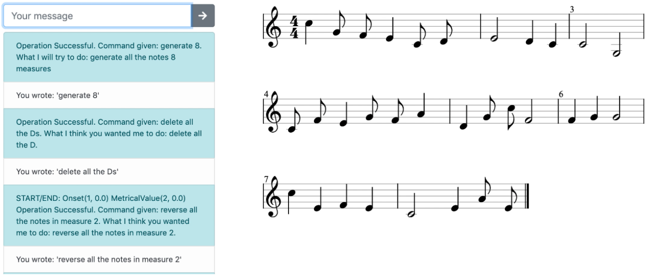
Commonsense Grounding via Data Annotation

Commonsense is a platform for grounding knowledge graphs on vision and language. Using this platform, one can reverse engineer movies, clips and scripts, and ground them on animations to learn common sense from human users to support automatic movie creation. User-provided annotations can also introduce an element of collaboration as the interface offers a visual indication of previously tagged terms, as well as a visualization of agreement among the uploaded entries.

Visual and textual annotation modules offer a variety of methods to annotate text tokens for individual frames in a single video clip. Bounding boxes and polygons can also be created to mark different on-screen entities. As shown in Figure 8, the user can also connect otherwise discrete annotations, represented as individual nodes, using the graph module in order to further augment the AI model.

Conversational AI in Creative Practices

Visual storytelling: Aesop is a system with the goal of content creation by conversing with a set of AI agents using verbal and non-verbal communication to co-create animations. Aesop provides a rich platform that enables research in language, gestures, vision, and planning in the context of storytelling. Aesop uses shared knowledge graph representations created from language and vision, using our interface, to generate a 3D animation sequence. The user also can engage



Operation Successful. Command given: generate 8. What I will try to do: generate all the notes 8 measures

You wrote: 'generate 8'

Operation Successful. Command given: delete all the Ds. What I think you wanted me to do: delete all the D.

You wrote: 'delete all the Ds'

START/END: Onset(1, 0.0) MetricalValue(2, 0.0) Operation Successful. Command given: reverse all the notes in measure 2. What I think you wanted me to do: reverse all the notes in measure 2.

You wrote: 'reverse all the notes in measure 2'

Figure 9: Equipped with two distinct modules that can seamlessly communicate with each other, one iteration of the MUSICA interface supports both chat and point-and-click operations, allowing the user to interact with the same piece of music in two distinct methods.

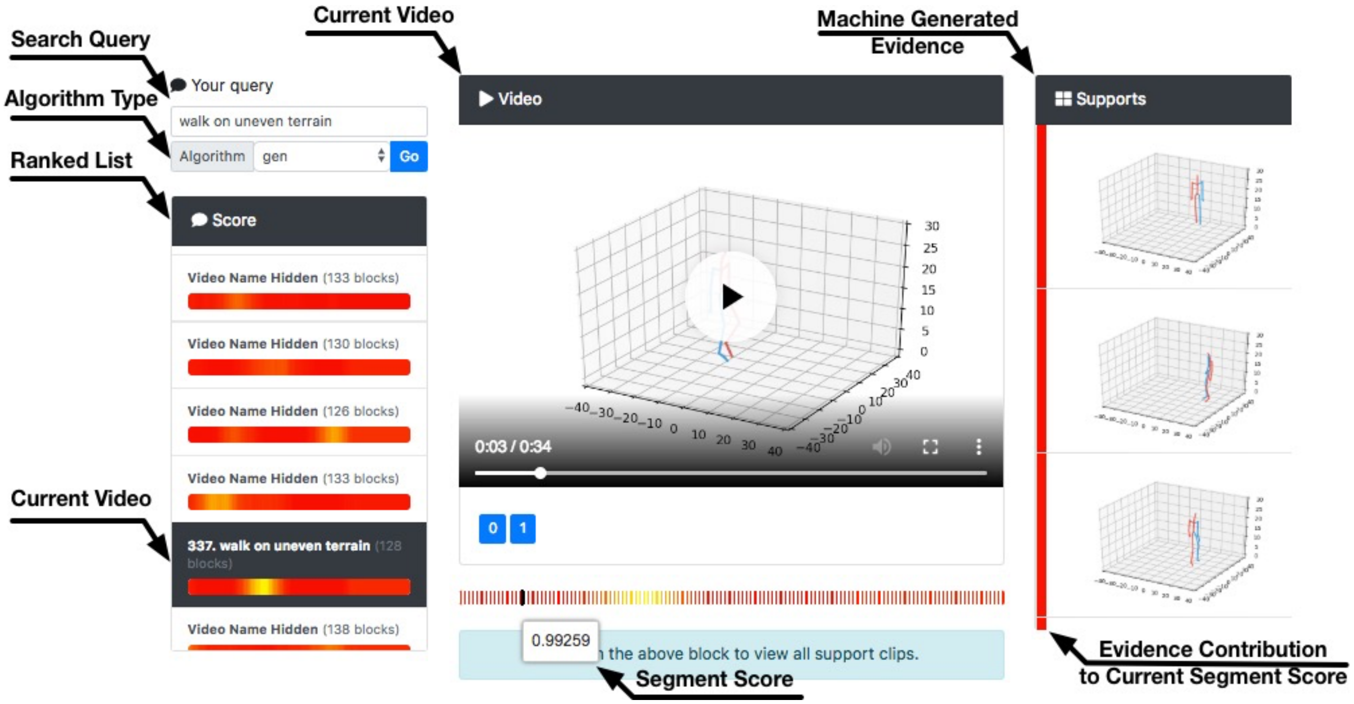

Search Query
Algorithm Type
Ranked List
Current Video
Machine Generated Evidence
Your query
walk on uneven terrain
Algorithm
gen
Go
Score
Video Name Hidden (133 blocks)
Video Name Hidden (130 blocks)
Video Name Hidden (126 blocks)
Video Name Hidden (133 blocks)
337. walk on uneven terrain (128 blocks)
Video Name Hidden (138 blocks)
Current Video
Video
0:03 / 0:34
0
1
Supports
0.99259
the above block to view all support clips.
Segment Score
Evidence Contribution to Current Segment Score

Figure 10: Annotated view of the explanation interface. The user starts by inputting a query and selecting the ranking algorithm of interest. Once one of the videos is selected from the ranked list, the video becomes available for the user to view, along with a segmented confidence bar (red indicating higher confidence) located below the navigation bar. Upon clicking on the confidence bar in a certain segment, the score for that segment is presented to the user, and the list of evidence in the right column gets sorted showing the most important evidence first.

with Aesop and receive corresponding animations using a chat window similar to today’s messenger applications.

Interactive music: MUSICA (MUSical Interative Collaborative Agent) is a project that focuses on interaction and communication between human musicians and machine assistants. Built on computational models of music and natural language processing for musical operations [14, 15], the project offers different methods in which the human user can interact with machine to generate new music, as well as various opportunities to inform future research regarding musical language and human-computer interaction. As illustrated in Figure 9, the user can ask the algorithm to automatically generate bars of music, remove a specific set of musical notes, and listen to the work-in-progress with simple text commands using a familiar chat window. In addition, the user can interact with the same piece of music using other methods, including using a mouse pointer to manipulate individual notes or playing a specific piece of music with a MIDI controller.

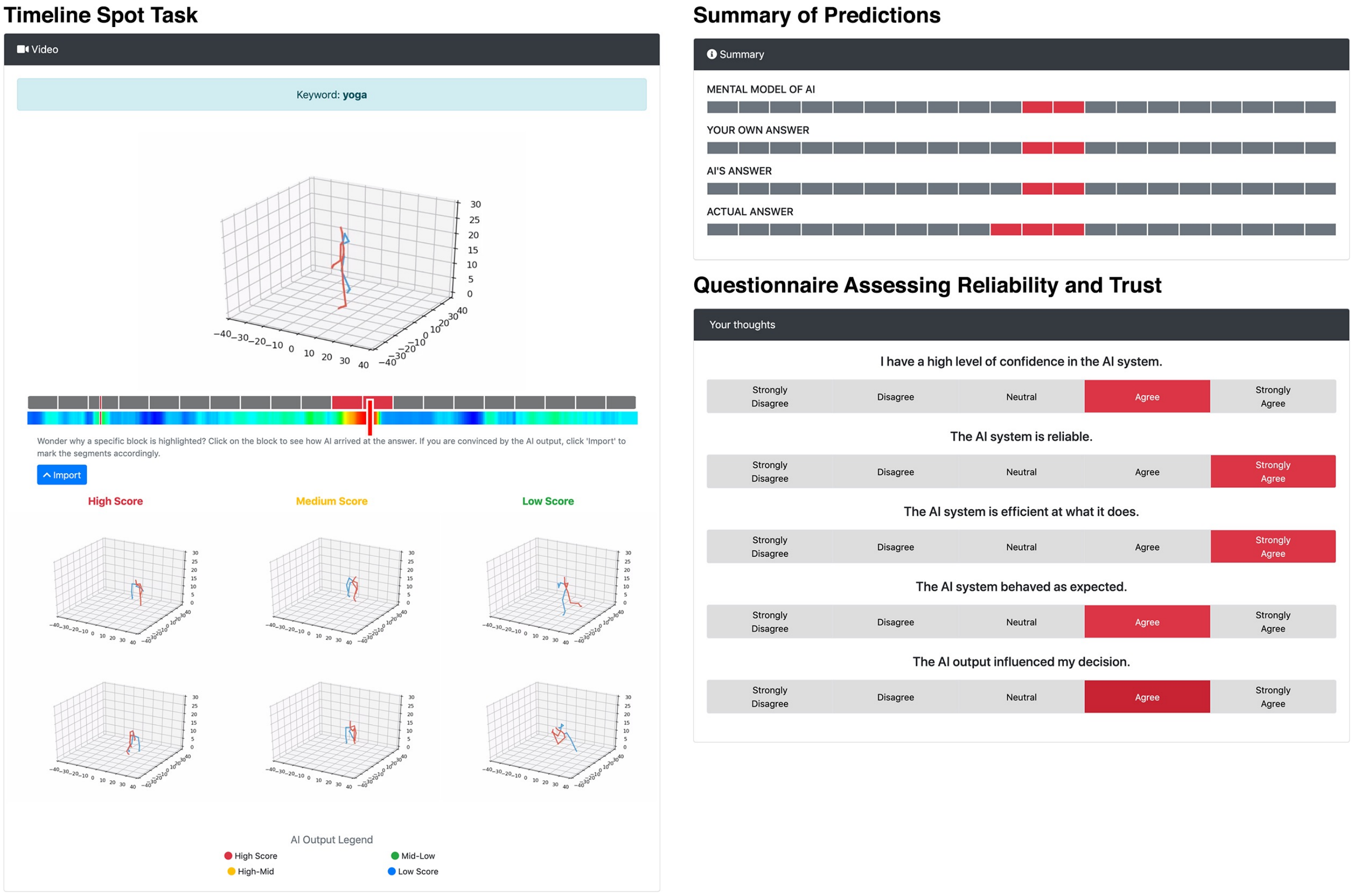
Explainable AI and Trust and Reliance Assessment

In this application we used the tool for created both an explainable AI (XAI) system for understanding how GANs can be used in video search, and modules for studying the efficacy of the XAI in a user study.

XAI Interface. The challenge of searching and ranking videos is well established and is usually addressed using discriminative models, but the decisions made by these “black-box” models tend to be inexplainable. On the other hand, generative searching and ranking models are more explainable, as the model learns human motion, generates realistic visual instances given textual inputs, and then uses the generated instances to search and rank videos in a database.

The interface, shown in Figure 10 acts as a mediator between the human and the AI, to help understand the AI’s rationale for decisions to be explained in a variety of ways. The explanation interface combines visualization of generator instances from the AI generator, as well as uncertainty in the outcomes from the ranker. The explanations consist primarily of relatable constructs such as animated motion sequences, rather than abstract visualizations of hidden model states or other low level features.

User Study. Through comparing two conditions — the XAI system and a black-box AI system, powered by generative and discriminative models respectively — the user study



Wonder why a specific block is highlighted? Click on the block to see how AI arrived at the answer. If you are convinced by the AI output, click 'Import' to mark the segments accordingly.

I have a high level of confidence in the AI system.

Strongly Disagree | Disagree | Neutral | [✓ Agree] | Strongly Agree

The AI system is reliable.

Strongly Disagree | Disagree | Neutral | Agree | [✓ Strongly Agree]

The AI system is efficient at what it does.

Strongly Disagree | Disagree | Neutral | Agree | [✓ Strongly Agree]

The AI system behaved as expected.

Strongly Disagree | Disagree | Neutral | [✓ Agree] | Strongly Agree

The AI output influenced my decision.

Strongly Disagree | Disagree | Neutral | [✓ Agree] | Strongly Agree

Figure 11: (left) One of the tasks dubbed *Timeline Spot*. Given a long video, the user's task is to highlight segments where the activity described by the given keyword query exists. The XAI system, unlike the AI system, provides assistance with explanations using model-generated clips. (top right) A summary of timelines with the location of the correct answer, the user's answer, the user's prediction of the system's answer, and the system's answer. (bottom right) A questionnaire per trial assessing the performance of the system.

aims to assess the benefit, if any, of XAI. Designed as a more linear, guided variation of the explanation interface, the study presents numerous instances of three main tasks:

(1) Identifying one or more video clips that best illustrate the displayed query.
(2) Spotting one or more segments in a single video clip that best illustrate the keyword.
(3) Collaborating with AI to solve a more complex challenge of identifying a longer video clip that best illustrates a complex query with multiple actions.

Featuring a variety of interactive modules, this web-based interface was refined through an internal pilot and was deployed as part of a randomized controlled study. One of the tasks is illustrated in Figure 11.

7 CONCLUSION

Driven by the increasing need for tools to work with language and image data together, we present a highly customizable, extensible tool for visualizing available model outputs, building and annotating new datasets, and setting up user studies. This toolbox brings together the usually disparate actions of data annotation and curation, as well as machine learning visualization and testing. Our toolbox enables vision and language researchers to seamlessly conduct their work without complex configuration of a web-interface. In addition, our toolbox invites other developers to contribute new modules to our public repository.

REFERENCES

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual Question Answering.

[2] Yusuf Aytar, Lluis Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Cross-Modal Scene Networks. In *arXiv:1610.09003*.

[3] R. Eckart de Castilho, É. Mújdricza-Maydt, S.M. Yimam, S. Hartmann, I. Gurevych, A. Frank, and C. Biemann. 2016. A Web-based Tool for the Integrated Annotation of Semantic and Syntactic Structures. *LT4DH Workshop at COLING* (2016).

[4] Ting-Hao K. Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Jacob Devlin, Aishwarya Agrawal, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual Storytelling. In *Proc. Conf. of the North American Chapter of the Assoc. for Computational Linguistics*.

[5] Severin Ibarluzea. 2020. Annotorious - Image Annotation for the Web. https://universaldatatool.com/. [Online; accessed 25-March-2020].

[6] Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A Hierarchical Approach for Generating Descriptive Image Paragraphs. In *CVPR*.

[7] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. 2017. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *IJCV* (2017).

[8] H. Lausberg and H Sloetjes. 2009. Coding gestural behavior with the NEUROGES-ELAN system. *Behavior Research Methods* 41, 841 (2009).

[9] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. *CoRR* abs/1405.0312 (2014).

[10] Xiao Lin, Chris Kim, Timothy Meo, and Mohamed R. Amer. 2018. Learn, Generate, Rank: Generative Ranking of Motion Capture.. In *European Conference on Computer Vision*.

[11] Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L. Yuille, and Kevin Murphy. 2015. Generation and Comprehension of Unambiguous Object Descriptions. *CoRR* abs/1511.02283 (2015). http://arxiv.org/abs/1511.02283

[12] Timothy Meo, Chris Kim, Aswin Raghavan, Alex Tozzo, David A. Salter, Amir Tamrakar, and Mohamed R. Amer. 2019. Aesop: A Visual Storytelling Platform for Conversational AI and Commonsense Grounding. *AI Communications* (2019).

[13] Philip V. Ogren. 2006. Knowtator: a protégé plug-in for annotated corpus construction. In *Proc. Conf. of the North American Chapter of the Assoc. for Computational Linguistics on Human Language Technology*. ACL, 273–275. https://doi.org/10.3115/1225785.1225791

[14] Donya Quick and Christopher N Burrows. 2018. Evaluating Natural Language for Musical Operations. (2018).

[15] Donya Quick and Kelland Thomas. 2019. A functional model of jazz improvisation. In *Proceedings of the 7th ACM SIGPLAN International Workshop on Functional Art, Music, Modeling, and Design*. 11–21.

[16] Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting Image Annotations Using Amazon's Mechanical Turk. *NAACL HLT Workshop* (2010).

[17] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. 2017. Movie Description. *Int. Journal of Computer Vision* (2017).

[18] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. 2008. LabelMe: A Database and Web-Based Tool for Image Annotation. *IJCV* 77, 1 (01 May 2008), 157–173. https://doi.org/10.1007/s11263-007-0090-8

[19] Rainer Simon. 2018. Annotorious - Image Annotation for the Web. https://annotorious.github.io/. [Online; accessed 21-March-2018].

[20] Pontus Stenetorp, Sampo Pyysalo, and Goran Topić. 2012. Brat Rapid Annotation Tool. http://brat.nlplab.org/. [Online; accessed 21-March-2018].

[21] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. MovieQA: Understanding Stories in Movies through Question-Answering. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[22] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. 2018. MovieGraphs: Towards Understanding Human-Centric Situations from Videos. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[23] Carl Vondrick, Donald Patterson, and Deva Ramanan. 2013. Efficiently Scaling up Crowdsourced Video Annotation. *Int. Journal of Computer Vision* 101, 1 (01 Jan 2013), 184–204. https://doi.org/10.1007/s11263-012-0564-1

[24] Johannes Wagner, Tobias Baur, Yue Zhang, Michel F. Valstar, Björn Schuller, and Elisabeth André. 2018. Applying Cooperative Machine Learning to Speed Up the Annotation of Social Signals in Large Multimodal Corpora. *arXiv:1802.02565* (2018). https://arxiv.org/abs/1802.02565

[25] Jenny Yuen, B. Russell, Ce Liu, and A. Torralba. 2009. LabelMe Video: Building a video database with human annotations. In *IEEE Int. Conf. on Computer Vision*. 1451–1458. https://doi.org/10.1109/ICCV.2009.5459289

[26] Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards Automatic Learning of Procedures from Web Instructional Videos. *AAAI* (2018).