A Modular Interface for Multimodal Data Annotations and Visualization

Chris Kim and Christopher Collins
University of Ontario Institute of Technology
{chris.kim,christopher.collins}@uoit.ca

Boris Knyazev and Graham W. Taylor
University of Guelph and Vector Institute for Artificial Intelligence
{bknyazev,gwtaylor}@uoguelph.ca

Tim Meo and Mohamed R. Amer
SRI International
{tim.meo,mohamed.amer}@sri.com

Abstract

We present a modular annotation and visualization tool for computational language and vision research. Our tool enables researchers to set up a web-interface for annotating new language and vision datasets, visualizing the predictions made by a machine learning model, or conducting user-studies. 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. It also includes a graph module to link visual and textual information. To further illustrate this, we showcase our interface applied to the MovieQA and MovieGraphs datasets.

1 Introduction

Recent work has shown that the integration of language and vision can substantially improve the predictions of a machine learning model. In response, there emerged a large number of research topics combining language and vision in recent years: image captioning (Lin et al., 2014), dense captioning using paragraphs (Krause et al., 2017), visual denotations of linguistic expressions (Rashtchian et al., 2010), visual question answering in images (Antol et al., 2015; Krishna et al., 2017), grounding referring expressions in images (Mao et al., 2015), visual storytelling using images and text sequences (Huang et al., 2016), visual question answering in movies (Tapaswi et al., 2016), describing situations in movies using vision and language (Vicol et al., 2018), movie scene description (Rohrbach et al., 2017), textually annotated cooking videos (Zhou et al., 2018), and text and image correspondences (Aytar et al., 2016).

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

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 visualize 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

Text Annotation Tools. There exist open-source, web-based, text tagging, annotation, and visualization tools such as BRAT (Stenetorp et al., 2012), Webanno (Eckart de Castilho et al., 2016), and Knowtator (Ogren, 2006). BRAT provides features for structured annotations, with fixed-

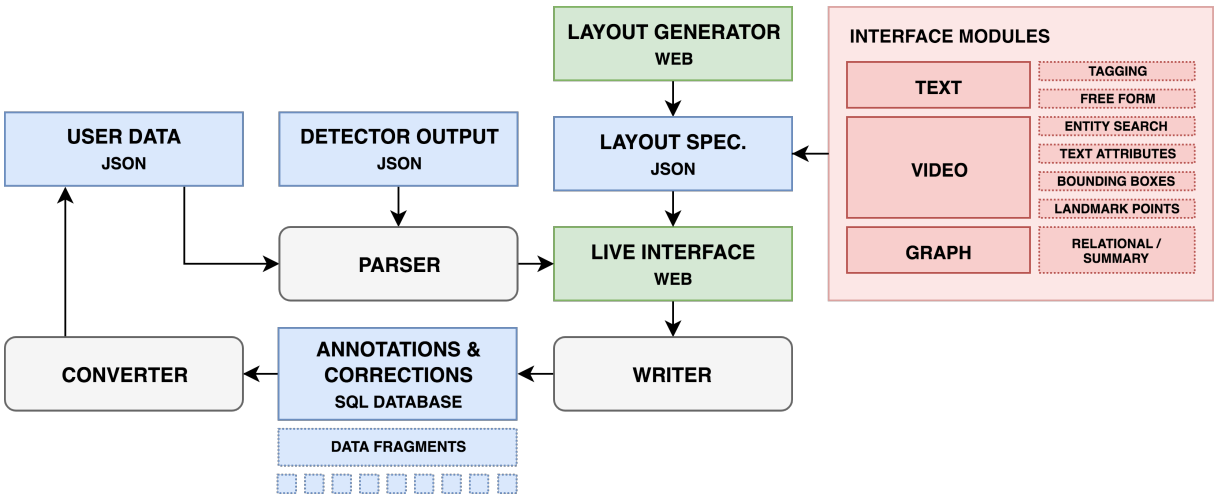

Figure 1: Our web-interface consists of text, video, and graph modules, which are arranged as a JSON-based specification file. The modules interact with a backend that loads the data schema and saves and loads annotations, which can be exported as a single output file.

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. All of 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 (Russell et al., 2008), Annotorious (Simon, 2018) and video annotation tools such as LabelMe Video (Yuen et al., 2009), and VATIC (Vondrick et al., 2013). 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.

Multi-modal Annotation Tools: ELAN (Lausberg and Sloetjes, 2009) and NOVA (Wagner et al., 2018) 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.

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. The starting page is a layout generator that produces a JSON-based specification file that defines the specific use-case schema. Fig. 1 summarizes our system and in the following sections we will elaborate on each component.

3.1 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 the uploaded entries.

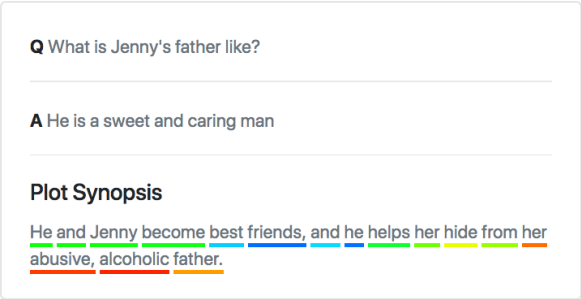

Figure 2: Our interface enables tagging specific words. Here, the user is selecting the seemingly correct answer, along with a series of relevant word tokens.

Tagging. The tagging component, illustrated in Fig. 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 weighted 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.

3.2 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 Fig. 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

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

to adjust their size or position, edit their labels, or create additional boxes as necessary.

Landmark Points. The video module also offers the ability to visualize a group of anchor points over video illustrated in Fig. 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 available for user modifications and annotations as other overlay elements.

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 Search. This module allows the user to look for occurrences of a certain entity across the video timeline illustrated in Fig. 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.

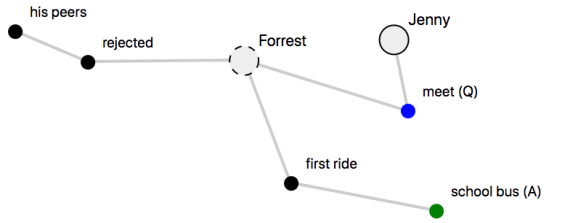

Figure 4: 2D Graphs visualizing the relationship of different on-screen entities and text token nodes using colored nodes and edges. A dotted stroke indicates absence of that entity in the current point in time.

3.3 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.

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 Fig. 4.

Each entity is represented as a node, with its various visual attributes – including size, opacity, and color – 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.

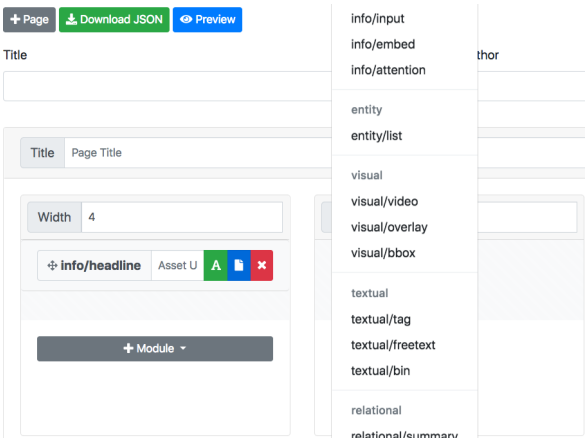

Figure 5: Each interface iteration can be easily constructed and customized using our web-based layout generator, eliminating the need for ad hoc web development and easing the initial learning curve.

4 Back End

Technologies. 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 confi-

dence score (provided by a pose estimation algorithm), tracked identity, and attributes.

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 and 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 or modify or add annotations to it.

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 Fig. 5. Inspired by modern 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 video annotation tool, visualize existing datasets, or deploy a complex user study.

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, CSS, and JS files, and upon passing a series of checks, can be readily added to the user-facing interface.

5 Application

We apply our interface to the MovieQA (Tapaswi et al., 2016) and MovieGraph datasets (Vicol et al., 2018). The goal of MovieQA is multimodal question answering using text provided in the script, subtitles, plot synopsis, as well as the video, while MovieGraphs datasets seek to map the relationships between different on-screen characters for situation recognition. The illustration, available at `http://modular.ckprototype.com`, features the interface modules in three distinct use cases: visualization, annotation, and user study.

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 the 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 persistent 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 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 Conclusion

We present a highly customizable, extensible tool for visualizing available model outputs, building and annotating new datasets, and setting up user studies. 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

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

Yusuf Aytar, Lluis Castrejon, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. 2016. Cross-modal scene networks. In *arXiv:1610.09003*.

R. Eckart de Castilho, . Mjdricza-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*.

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*.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *CVPR*.

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*.

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

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.

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.

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*, pages 273–275. ACL.

Cyrus Rashtchian, Peter Young, Micah Hodosh, and Julia Hockenmaier. 2010. Collecting image annotations using Amazon's Mechanical Turk. *NAACL HLT Workshop*.

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*.

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):157–173.

Rainer Simon. 2018. Annotorious - image annotation for the web. https://annotorious.github.io/. [Online; accessed 21-March-2018].

Pontus Stenetorp, Sampo Pyysalo, and Goran Topi. 2012. Brat rapid annotation tool. http://brat.nlplab.org/. [Online; accessed 21-March-2018].

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)*.

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)*.

Carl Vondrick, Donald Patterson, and Deva Ramanan. 2013. Efficiently scaling up crowdsourced video annotation. *Int. Journal of Computer Vision*, 101(1):184–204.

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 multi-modal corpora. *arXiv:1802.02565*.

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*, pages 1451–1458.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018. Towards automatic learning of procedures from web instructional videos. *AAAI*.