Event Extraction in Video Transcripts

Amir Pouran Ben Veyseh[1], **Viet Dac Lai**[1],
Franck Dernoncourt[2], **and Thien Huu Nguyen**[1]
[1]Department of Computer Science, University of Oregon, OR, USA
[2]Adobe Research, Seattle, WA, USA
{apouranb, vietl, thien}@cs.uoregon.edu
{franck.dernoncourt}@adobe.com

Abstract

Event extraction (EE) is one of the fundamental tasks for information extraction whose goal is to identify mentions of events and their participants in text. Due to its importance, different methods and datasets have been introduced for EE. However, existing EE datasets are limited to formally written documents such as news articles or scientific papers. As such, the challenges of EE in informal and noisy texts are not adequately studied. In particular, video transcripts constitute an important domain that can benefit tremendously from EE systems (e.g., video retrieval), but has not been studied in EE literature due to the lack of necessary datasets. To address this limitation, we propose the first large-scale EE dataset obtained for transcripts of streamed videos on the video hosting platform Behance to promote future research in this area. In addition, we extensively evaluate existing state-of-the-art EE methods on our new dataset. We demonstrate that such systems cannot achieve adequate performance on the proposed dataset, revealing challenges and opportunities for further research effort.

1 Introduction

Event Extraction is an important task in the full pipeline of Information Extraction. In EE, the goal is to identify mentions/trigger words of events, and their participants and attributes of interest. For instance, in the sentence "*Joe Biden was born on November 20, 1942*", an event of *Birth* is mentioned. An event mention consists of two important components: (1) **Trigger**: the word(s) that most clearly refer to the occurrence of the event (e.g, "*born*" in the above example); and (2) **Argument**: the entity mentions involved in the event with some role (e.g., "*Joe Biden*" with the role of *Entity*) or attributes of the event (e.g., time and location).

Due to its importance, various methods and annotated datasets have been proposed for EE (Ahn, 2006; Nguyen and Grishman, 2015; Yang et al., 2019; Wang et al., 2020). Also, with the proliferation of EE methods, datasets for EE have been diversified to cover different domains and settings, e.g., multiple domains (Walker et al., 2006), multiple languages, (Mitamura et al., 2016), literary texts (Sims et al., 2019), cybersecurity texts (Man Duc Trong et al., 2020), and events in long documents (Ebner et al., 2020). However, despite all progress thus far, most of the available datasets for EE are restricted to the domains of formally written texts, e.g., news, reports, scientific papers, or books. As such, the challenges for EE in other domains with informal and noisy texts are less explored. One of such domains that has not been studied before for EE involves video transcripts obtained by automatic speech recognition (ASR) tools. Since the such transcripts might be noisy, e.g., incomplete sentences, incorrect words selected by the ASR tool, lack of correct punctuation and segmentation, repeated words or sentences, etc., existing EE models are not well evaluated and might not perform well in this domain. This is unfortunate as an effective EE model can be extremely helpful for downstream applications that utilize video transcripts. For instance, search engines can employ events detected in a transcript to locate relevant portion of a video to a query. It can also benefit video summarization, knowledge base construction, and script generation from videos. As such, it is necessary to study the challenges and potential directions for EE improvement in the video transcript domain.

Due to the lack of EE datasets for video transcript domain, we propose the first large-scale EE dataset annotated for transcripts of streamed videos on the popular video hosting platform Behance. Videos in this platform are streamed by artists who would like to share their creative projects using Adobe Creative Cloud products (e.g., Photoshop, Illustrator, etc). Videos have been first transcribed using the Microsoft Automatic Speech Recognition tool. In order to annotate the events mentioned in

Proceedings of the 29th International Conference on Computational Linguistics, pages 7156–7165
October 12–17, 2022.

the video transcripts, we first define a taxonomy of event types and their arguments for the domain of creative projects (e.g., Add a shape, Modify the color of an object, etc.). Using the pre-processed transcripts and the provided event ontology, we hire annotators with domain expertise to provide high-quality annotation for event triggers and their arguments. In addition, we employ the annotated dataset to evaluate the performance of the state-of-the-art (SOTA) EE models. Compared to the domains with formally written texts, our analysis shows that the current SOTA EE models fail to achieve comparable performance on video transcripts. This performance drop indicates the challenging nature of video transcripts and call for more research effort for EE in this domain. We will publicly release our dataset, called *TranscriptEE*, to foster future research in this area.

2 Dataset

Data Collection: In this work, we propose to employ the videos streamed on the popular video-hosting platform Behance[1] to obtain transcripts to be annotated for event mentions. Behance is a platform in which artists can share their creative projects using Adobe Creative Cloud products (e.g., Photoshop, Illustrator, etc). Most of the information is transmitted verbally in English in these videos. Each video lasts from few minutes to several hours. In the first step, we collect 370 videos with a total duration of 500 hours. On average, a video lasts 48 minutes. Next, for each video, we employ the Microsoft Automatic Speech Recognition tool to obtain transcripts of the videos. A video transcript on average contains 7,219 words. To facilitate the annotation process, following prior work for EE dataset creation (Ebner et al., 2020), we split the transcripts of the videos into chunks (called paragraphs) of 5 consecutive utterances. In total, 25,492 paragraphs are obtained for EE annotation.

Annotation: To annotate the data, we first define an ontology of event types and argument roles for the domain of creative projects. Specifically, an event is defined as an action that results in a visual change in the project (e.g., changing the color of an object, adding a new shape to an illustration, modifying the texture of the surfaces in an image, etc.). Concretely, we categorize the events into four types: (1) **Add**: A new visual element is added to the project; (2) **Modify**: One of the attributes of an existing element (e.g., color, size, texture, etc.) is changed; (3) **Select**: Some objects in the project are selected using tools of editing programs (e.g., the "Lasso" tool in Photoshop); and (4) **Remove**: An object is removed from the project. For each event, we also define their arguments, e.g., Tool, Object, Color, etc. We present a description of event types and arguments, along with their examples in Appendix A. To select paragraphs for annotation, we design a set of keywords that are relevant to our event types (e.g., "*add*", "*modify*", "*select*", "*pick*", "*remove*", "*delete*"). Paragraphs with the highest matching rates for the keyword set are retained for EE annotation. Overall, to accommodate our budget, 2,162 top paragraphs are annotated in our dataset.

We employ human annotation to find event mentions in the paragraphs of video transcripts. To annotate event triggers, we follow prior work on ED (Walker et al., 2006) to ask the annotators to select the word or phrases (e.g., a phrasal verb) that most clearly mention the occurrences of events. Also, for event arguments, we require the annotators to select the head words of the noun phrases that refer to the arguments of events. Event triggers and arguments can belong to different sentences in the paragraphs in our dataset.

In this work, we leverage Upwork, a freelancer platform, to hire expert annotators. The hired annotators have experience in both creative projects (e.g., using photo editing programs such as Photoshop) and data annotation. We train the annotators with the event ontology and designed examples. Based on the performance of the annotators in a pilot study, we select the final pool of five annotators. To compute the inter-annotation agreement (IAA) scores, 20% of the paragraphs are shared by the annotators for co-annotation while the remaining 80% is distributed evenly for the annotators. As such, annotators first independently identify event triggers in the shared paragraphs, achieving an agreement score of 0.812 for the Krippendorff's alpha (Krippendorff, 2011) with MASI distance metric (Passonneau, 2006). Afterward, the annotators discuss to resolve conflict cases for the co-annotated data, and then perform annotation individually on the remaining data to produce the final version of event triggers in our dataset. In the next step, given the annotated triggers, annotators also independently annotate event arguments for the shared

[1] www.behance.net

Statistics	Total	Train	Dev	Test
# Paragraphs	2162	1729	216	217
# Triggers	3180	2580	283	317
# Arguments	3427	2798	295	334
Avg. # Triggers / Sample	1.47	1.49	1.31	1.46
Avg. # Arguments / Trigger	1.59	1.62	1.37	1.54

Table 1: Statistics of *TranscriptEE*.

paragraphs, achieving the Krippendorff's alpha of 0.783. Finally, conflict argument examples in the co-annotated data are resolved and individual annotation on the rest of the data is done by the annotators to generate the final version of our dataset *TranscriptEE*. As such, we achieve strong agreement scores for both event trigger and argument annotation, showing the high quality of *TranscriptEE*. To facilitate future research, we randomly split the dataset into separate training, development, and test sets with the ratio of 80/10/10, respectively. The statistics for each data split along the entire dataset are presented in Table 1.

Annotation Challenges: This section describes some major challenges we encounter in the annotation process for *TranscriptEE*. (1) **False Triggers**: In some cases, the streamer discusses a general action without actually doing the action. For instance, in the sentence "*Croping the images is very easy in Photoshop*", the streamer mentions an edit action (i.e., "*cropping*") which can be considered as a Modify event without implying actual implementation of such actions in his/her work. These examples cause disagreements among the annotators on whether an event trigger should be annotated or not. To resolve this situation, we ask the annotators to not annotate event triggers that the streamer does not clearly imply their occurrence. (2) **Confusing Triggers**: Depending on the object of consideration, some event triggers can be interpreted as either an Modify or Add event, thus bewildering the annotators for correct annotation. For instance, in the sentence "*First, I **create** some shadows for letters.*", the word "*create*" can refer to an Add event with the object "*shadow*"; however, it can also evoke the event type Modify with the object "*letters*". To resolve this conflict, we require the annotators to select the more general event type, i.e., the type Modify in our example.

Dataset Challenges: In addition to the typical challenges of EE (e.g., ambiguous triggers that can trigger different event types depending on contexts), an unique challenge of *TranscriptEE* for EE models involves background knowledge. In particular, recognizing domain knowledge about technical terms for editing programs in creative projects is necessary for the models to make correct predictions in *TranscriptEE*. For instance, in the sentence "*We prefer **burn** to make shadows darker*", to select the word "*burn*" as the argument for the Modify event trigger "*make*", it is important for the models to realize that "*burn*" is a tool name in Photoshop.

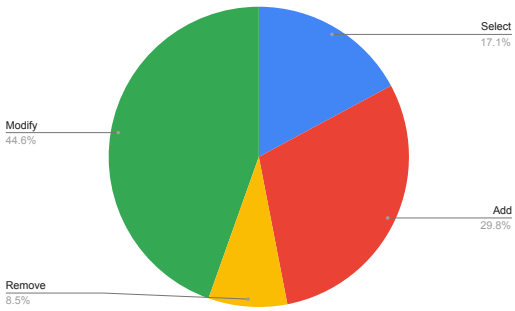

Figure 1: Distribution of event types in the proposed dataset *TranscriptEE*.

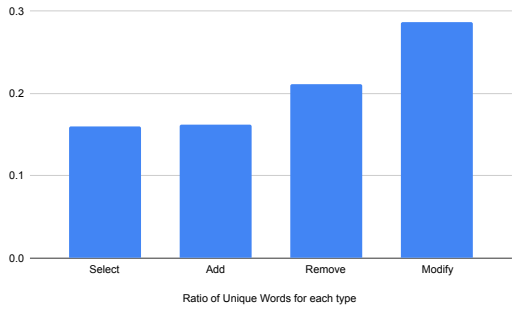

Figure 2: Ratio of number of unique trigger words to their frequency for each event type.

Dataset Analysis: In order to shed more light for the proposed dataset *TranscriptEE*, we report the distribution of the event types in Figure 1. This figure shows that the Modify event type has the

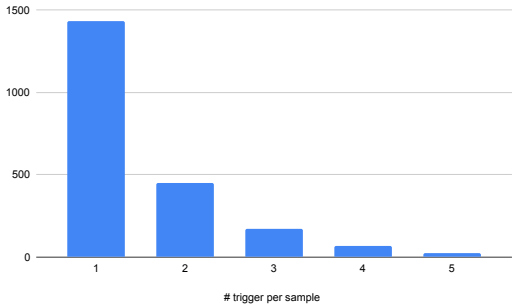

Figure 3: Distributions of number of event triggers per paragraph.

highest frequency in the dataset, followed by Add, Select and Remove. Moreover, to study the challenging nature of each event type, we study the ratio of the number of unique event triggers to the frequency of each event type. The higher this ratio, the more challenging the event type is as it expresses more diverse ways to present an event in the dataset. The results are presented in Figure 2. This figure demonstrates that the Modify event type employs the most diverse set of triggers followed by the event type Remove. Considering the low frequency of Remove with its high ratio of trigger diversity, it also implies the challenging nature of this event type. Finally, Figure 3 shows the distribution for the numbers of trigger words per paragraph in *TranscriptEE*. As can be seen, while the majority of the paragraphs have one trigger word, nearly 30% of the dataset involves more than one trigger words. It thus suggests an opportunity to exploit the corelation of event triggers and types to improve EE performance on *TranscriptEE*.

3 Experiments

We study the performance of existing state-of-the-art EE systems on the proposed dataset *TranscriptEE*. Since EE consists of two sub-tasks, i.e., Event Detection (ED) and Event Argument Extraction (EAE), we consider two types of baselines:

(1) **Pipeline Modeling**: In this category, an ED system is first employed to identify event triggers with their types in the input text. Next, given a predicted event trigger, an EAE system is utilized to recognize arguments and their roles for the event trigger. As such, we use the **BERT** model (Devlin et al., 2019) as the baseline model for ED and EAE in the pipeline approach as in prior work (Yang et al., 2019; Wang et al., 2020). Specifically, the input paragraph, in the form of $D = [[CLS], w_1, \ldots, w_n, [SEP]]$ with n words, is first fed into the BERT model (the base cased version) of the ED system to predict the triggers and their types. The ED task is modeled as a sequence labeling problem using the BIO tagging schema to encode the labels for each word in D. Next, the predicted event type and trigger will be concatenated with the input document D to be consumed by the BERT model of the EAE system for argument prediction, i.e., $[Type, Trigger, [SEP], w_1, \ldots, w_n]$. Here, $Type$ and $Trigger$ are predicted by the ED system. The EAE task is also modeled as an sequence labeling for argument roles. In addition, we study the performance of **BERT+CRF** model, where a conditional random field (CRF) layer is added on top of the BERT models for the ED and EAE architectures for sequence labeling.

Model	ED			EAE		
	P	R	F1	P	R	F1
BERT	57.47	59.93	59.69	39.21	44.74	41.79
BERT+CRF	56.42	59.94	58.13	41.14	39.45	40.28

Table 2: ED and EAE performance of pipeline models on the test set of *TranscriptEE*.

(2) **Joint Modeling**: In this category, the systems jointly predict event triggers and their arguments for an input text in an end-to-end fashion. As such, we consider four typical joint models for EE. The first two models employ similar architectures as **BERT** and **BERT+CRF** where BERT is utilized to encode the input paragraph D to produce representation vectors for each word. The word representations are then fed into a feed forward layer (and a CRF layer in case of **BERT+CRF**) to identify event trigger and argument spans with sequence labeling. Next, trigger and argument representations are obtained by averaging the word representations inside the detected spans. Finally, the trigger representations are sent to a feed-forward network for event type prediction while pairs of trigger and argument representations are consumed by another feed-forward network to predict argument roles. **BERT** and **BERT+CRF** are trained end-to-end with the combined loss from different components. Our third joint baseline involves **OneIE** (Lin et al., 2020) that is similar to the **BERT+CRF** joint baseline. However, instead of using greedy decoding as in **BERT+CRF**, **OneIE** manually designs global features to capture label dependencies among different IE tasks to improve beam search decoding. Our fourth joint baseline explores **FourIE** (Nguyen et al., 2021) that differs **OneIE** in that **FourIE** exploits instance and label dependencies to improve representation learning in the training step (via Graph Convolutional Networks and consistency regularization). Note that we leverage the original implementations and remove the relation extraction task from **OneIE** and **FourIE** for our joint EE problem. **OneIE** and **FourIE** are among the current state-of-the-art (SOTA) methods for EE.

Performance of the pipeline and joint models is presented in Tables 2 and 3, respectively. The joint models outperform their counterpart pipeline systems. Specifically, **BERT** and **BERT+CRF** enjoy 2.73 and 2.99 F1 point improvement for ED,

Model	ED			EAE		
	P	R	F1	P	R	F1
OneIE	60.72	54.38	57.38	50.70	45.82	48.14
FourIE	58.48	59.95	59.21	52.55	46.44	49.31
BERT	61.19	63.70	62.42	52.43	51.02	51.72
BERT+CRF	60.98	61.26	61.12	51.65	50.15	50.89

Table 3: ED and EAE performance of joint models on the test set of *TranscriptEE*.

and 9.93 and 10.61 F1 point improvement for EAE respectively. This can be attributed to the shared parameters of BERT in joint **BERT** and **BERT+CRF** that enrich the induced representation vectors to improve the prediction. Also, among the joint models, the simpler methods **BERT** and **BERT+CRF** actually perform better than the more complicated models **OneIE** and **FourIE** that exploit label dependency of the tasks for training and decoding. This indicates that the methods to capture label dependence in **OneIE** and **FourIE** are not helpful for EE in *TranscriptEE*, thus calling for more research effort to design more suitable EE models in this domain. Finally, the performance of existing SOTA methods for EE over *TranscriptEE* is still far from being perfect that presents much room for future research.

4 Related Works

Previous EE systems can be classified according to the representation construction methods, i.e., feature engineering (Ahn, 2006; Liao and Grishman, 2010; Li et al., 2013) vs. deep learning for representation learning (Nguyen and Grishman, 2015b; Chen et al., 2015; Nguyen and Grishman, 2018; Chen et al., 2018; Wang et al., 2019a), or the formulation approaches, i.e., pipeline (Ahn, 2006; Yang et al., 2019; Wang et al., 2019b) vs. joint models (Yang and Mitchell, 2016; Nguyen and Nguyen, 2019; Lin et al., 2020; Nguyen et al., 2021, 2022). A majority of prior EE work utilize the ACE 2005 (Walker et al., 2006) and TAC KBP datasets (Mitamura et al., 2016) that focus on newswire article domains. Recently, there have more efforts to create EE datasets for more diverse domains, including biomedical (Kim et al., 2011), literary (Sims et al., 2019), cybersecurity (Man Duc Trong et al., 2020), Wikipedia (Wang et al., 2020), multilingual (Pouran Ben Veyseh et al., 2022; Lai et al., 2022), history (Lai et al., 2021), and suicide understanding (Guzman-Nateras et al., 2022). However, none of such prior work and datasets explores EE for video transcripts.

5 Conclusion

We present *TranscriptEE*, the first manually annotated dataset for EE for video transcripts. The videos are obtained from the Behance platform which is dedicated to sharing creative projects. *TranscriptEE* contains more than 2,000 labeled paragraphs for various edit events in creative projects with high quality. Our analysis with state-of-the-art EE models reveals challenging nature of the dataset for future research.

Ethical Considerations

In this work we present a dataset on the transcripts of a publicly accessible video-streaming platform, i.e., "*Behance*"[2]. Complying with the discussion presented by Benton et al. (2017), research with human subjects information is exempted from the required full Institutional Review Board (IRB) review if the data is already available from public sources or if the identity of the subjects cannot be recovered. However, to protect the identity of the streamer and any other person whose information are shared in the video transcript, we impose extra processing on the presented dataset before presenting it to annotators and publicly releasing it later. First, in this dataset, we exclude username or any other identity-related information of the streamers in the transcripts to prevent disclosing their identity. Moreover, the proposed dataset only provides textual data (i.e., paragraphs), hence the other content of the videos (e.g., images, audios) are not revealed (to annotators or users) to protect human identity. Finally, to reduce the risk of disclosing the information of the people mentioned in the transcripts, in the final version of the dataset, we exclude the transcripts that explicitly or implicitly refer to the identify of the target people.

Acknowledgement

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112 and the NSF grant CNS-1747798 to the IU-CRC Center for Big Learning. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program.

[2] www.behance.net

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*.

Adrian Benton, Glen Coppersmith, and Mark Dredze. 2017. Ethical research protocols for social media health research. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pages 94–102, Valencia, Spain. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Luis Guzman-Nateras, Viet Lai, Amir Pouran Ben Veyseh, Franck Dernoncourt, and Thien Nguyen. 2022. Event detection for suicide understanding. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1952–1961, Seattle, United States. Association for Computational Linguistics.

Jin-Dong Kim, Tomoko Ohta, Sampo Pyysalo, Yoshinobu Kano, and Jun'ichi Tsujii. 2011. Extracting bio-molecular events from literature—the bionlp'09 shared task. *Computational Intelligence*, 27(4):513–540.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Viet Lai, Minh Van Nguyen, Heidi Kaufman, and Thien Huu Nguyen. 2021. Event extraction from historical texts: A new dataset for black rebellions. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*.

Viet Dac Lai, Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2022. Meci: A multilingual dataset for event causality identification. In *Proceedings of the COLING 2022*. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Hieu Man Duc Trong, Duc Trong Le, Amir Pouran Ben Veyseh, Thuat Nguyen, and Thien Huu Nguyen. 2020. Introducing a new dataset for event detection in cybersecurity texts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H. Hovy. 2016. Overview of TAC-KBP 2016 event nugget track. In *Proceedings of the Text Analysis Conference (TAC)*.

Minh Van Nguyen, Viet Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4363–4374, Seattle, United States. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Thien Huu Nguyen and Ralph Grishman. 2015b. Event detection and domain adaptation with convolutional neural networks. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*.

Rebecca Passonneau. 2006. Measuring agreement on set-valued items (masi) for semantic and pragmatic annotation. In *LREC*.

Amir Pouran Ben Veyseh, Minh Van Nguyen, Franck Dernoncourt, and Thien Nguyen. 2022. MINION: a large-scale and diverse dataset for multilingual event detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2286–2299, Seattle, United States. Association for Computational Linguistics.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. In *Technical report, Linguistic Data Consortium*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019a. Adversarial training for weakly supervised event detection. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019b. HMEAE: Hierarchical modular event argument extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

A Event Types & Argument Roles

In this work, we define the event types based on edit actions performed during a creative project. Specifically, an event is change of state that can potentially refer to a visual change in an image. As such, we define four event types "*Add*", "*Remove*", "*Modify*", and "*Select*". Their description and examples are presented in Table 4. Moreover, each event type can involve multiple arguments. Argument are the objects, tools and properties employed for the edit action. The list of available argument roles for each event type is presented in Table 5.

B Annotation Instruction and Tool

We present the instructions provided to the annotators in Figure 4. In this work, we employ the BRAT[3] annotation tool (MIT License). A screenshot of the annotation tool is presented in Figure 5.

[3] https://brat.nlplab.org/

1. General Instructions

In this project, we annotate the transcripts of Behance videos. These are videos streamed on behance.net related to various Adobe Creative Cloud products (e.g., Photoshop, Fresco, XD, Illustrator, etc). In this project, we will focus on the Event Extraction task. An event is a specific occurrence involving participants. Specifically, an event in a Behance video transcript refers to some visual changes in the image that the streamer is working on and it is mentioned in the video transcript. We will not be tagging all events, only examples of a particular set of types are annotated. Specifically, we are interested in annotating events that can represent an edit action, i.e. Select, Add, Remove, Modify.

An event mention has two parts:

- **Trigger:** A word that most clearly mentions the occurrence of the event. For instance, in the sentence “What I’m doing now is to just **move** these boxes to the background, so we can see the texture here”, the word “move” is the event trigger and clearly mentions that an object is being modified in the image.
- **Argument:** In an event, multiple entities are involved. An entity is a text span that refers to something that is involved in the event. The participants of events play a role in the event. Technically, they are called “event arguments”. For instance, in the example above, the word “boxes” is an event argument with the role “Object” for the event “Modify”.

We will annotate both event triggers and event arguments. For each trigger, we are interested in knowing what type of event is evoked by that. For the arguments, we want to know the role of the argument in the event. Please note that it is possible that only a few argument roles are mentioned in the document and the rest might be missing.

Figure 4: Annotation Instruction

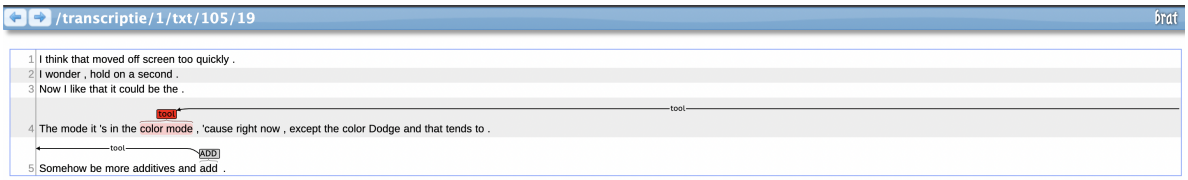

Figure 5: Annotation Tool

ID	Type	Description	Example (triggers are highlighted)
1	Select	A "Select" event happens when an object is selected using one of the selection tools.	• And this time I'm just going to be really, really kind of lazy about it and use my Lasso tool to do some **selections** • Let's **pick** these leaves and do some fun edits on them!
2	Remove	A "Remove" event happens when a part of the image is removed.	• We need to first **get rid of** these lower sections and add the new sketch there. • Okay, I just **deleted** all background shapes to make the image cleaner.
3	Add	An "Add" event happens when a new object is added to the image.	• We're going to be very. Very carefully. **Brushing** alongside. • The birds on the tree are easily **added** by my special brush.
4	Modify	A "Modify" event happens when an object of the image is updated (e.g., resize, recolor, blur, etc.).	• What I'm going to do is to **turn** that ball to blue so it will be matched with whatever we have over there. • I first **brightened** its front side to give more depth to the image.

Table 4: Event types along with their descriptions and examples in the proposed dataset.

ID	Type:Argument	Description	Example (arguments are highlighted and triggers are in *italic* font)
1	Select:Tool	The tool that is utilized to perform the select action.	And this time I'm just going to be really, really kind of lazy about it and use my **Lasso** tool to do some *selections*
2	Select:Object	The object that is selected.	I'm gonna *select* the **leaves** using the magic tool.
3	Remove:Tool	The tool that is utilized to perform the removal action.	Using the **perspective** crop, it's super easy to *get rid of* these buildings.
4	Remove:Object	The object that is being removed.	Using the perspective crop, it's super easy to *get rid of* these **buildings**.
5	Add:Tool	The tool that is utilized to add the new object.	First, let's *add* a single circle here using **ellipse**.
6	Add:Object	The object that is added to the image.	First, let's *add* a single **circle** here using ellipse.
7	Modify:Old_Color	Previous color of the object or image.	We first start with this **blue** sky and *turn* it to dark blue.
8	Modify:New_Color	New color of the object or image.	We first start with this blue sky and *turn* it to **dark blue**.
9	Modify:Old_Size	Old size of the object.	Let's *make* this giant **100-pixel** bar shorter.
10	Modify:New_Size	New size of the object.	The hat overhear should be *enlarged* to **10** cm.
11	Modify:Tool	The tool that is utilized to modify the object.	Use the **color replacement** tool to easily *change* the background color.
12	Modify:Object	The object that is modified.	We first start with this blue **sky** and *turn* it to dark blue.

Table 5: Argument roles for each event type along with their descriptions and examples in the proposed dataset.