Multi-Modal Semantic Inconsistency Detection in Social Media News Posts

Scott McCrae

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2021-72
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-72.html

May 13, 2021

Acknowledgement

I would like to thank my advisor, Professor Avideh Zakhor, and my lab partners Kehan Wang, David Deng, and Jordan Grelling for their insightful discussions this past year. This work was supported by the DARPA Semantic Forensics program. Funding for computing resources was provided by Google Cloud.

Multi-Modal Semantic Inconsistency Detection in Social Media News Posts

Scott McCrae
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
mccrae@berkeley.edu

Abstract

As computer-generated content and deepfakes make steady improvements, semantic approaches to multimedia forensics will become more important. In this work, we introduce a novel classification architecture for identifying semantic inconsistencies between video appearance and text caption in social media news posts. We apply a multi-modal fusion framework to identify mismatches between videos and captions in social media posts by leveraging an ensemble method based on textual analysis of the caption, automatic transcription, semantic video analysis, object detection, named entity consistency, and facial verification. To train and test our approach, we curate a new video-based dataset of real-world Facebook news posts for analysis. Our multi-modal approach achieves 60.5% classification accuracy on random mismatches between caption and appearance, compared to accuracy below 50% for uni-modal models. Further ablation studies confirm the necessity of fusion across modalities for correctly identifying semantic inconsistencies.

1 Introduction

There has been a great deal of attention on misinformation and deepfakes recently, especially with regards to the ongoing COVID-19 pandemic and the 2020 US Presidential election. There are a variety of methods for detecting both manipulated media, such as Photoshopped images, and data which is machine-generated, such as images from generative adversarial networks (GANs). However, these tools tend to focus on a single modality, such as imagery, and look for clues that the image was manipulated using statistical methods or by leveraging metadata. While these tools are indisputably useful, we are interested in investigating multi-modal analysis, where we attempt to detect manipulations or misinformation using semantic clues from a variety of modalities.

The use of multiple modalities allows us to reason about the semantic content of each source. For instance, a caption describing an out-of-control protest would be inconsistent with a video of a candle-light vigil, and a video of a reporter in the midst of a hurricane in Florida would be inconsistent with a news article on the effects of deforestation in the Amazon. On their own, neither modality is manipulated, but together they represent an inconsistency. This might model the threat of "cheapfakes," where an attacker lazily sources pairs of material to output misinformation at scale; an attacker attempting to misrepresent some original source; or an attacker with one or more undetectably altered modalities generated by a system unaware of high-level semantic consistency. While current methods are able to detect GAN-based images or deepfakes and text generated from language models, such generation techniques may continue to improve and begin to fool uni-modal detection approaches.

To analyze the semantic alignment of videos and captions, we need three main ingredients. First, and most importantly, we need pristine data as ground truth. Second, there needs to be a way to extract semantic feature representations from each modality and its constituents, such as transcripts and named entities. Third, there needs to be a way to jointly reason about semantic content. In the following sections, each of these components will be addressed in turn. Section 2 describes related work in natural language processing, computer vision, and multi-modal analysis. Section 3 describes our data collection and pre-processing methods. Section 4 describes experimental results and ablation studies. Section 5 provides a conclusion and a discussion of future work for this project.

2 Related Works

The field of natural language processing has seen a rapid shift in recent years towards transformer-based methods, introduced in [31], with large language models achieving state of the art performance [5, 16, 24, 2]. Machine learning in computer vision has been dominated by convolutional methods, with 2D methods such as ResNet [10] becoming standard backbone networks. Several later works have extended 2D convolutional networks to process videos [36, 9, 34]. Approaches such as [36] extend convolution into three dimensions, while [34] introduces separable computations over the spatial and temporal domains to increase efficiency. [21] adapts [34] to include text embeddings which are jointly learned with video embeddings, and is trained on a very large corpus of instructional videos [22]. Recent research has shown promising results adapting transformer methods to process videos [1], opening the door to processing video clips which are longer than a few seconds.

Research in multi-modal learning with text and imagery has demonstrated the efficacy of learning modality-specific embeddings [7]. New methods have been developed with the goal of leveraging transformers to jointly process text and imagery [17, 28, 14, 29]. [19] extends joint text and image transformer-based methods to process text and video clips. [15] employs cross-modal transformers with video frame and text embeddings for multi-modal learning.

A variety of methods have been introduced recently for detecting computer-generated content and semantic inconsistencies. [35] detects neural fake news by modeling a joint distribution over a news article's domain, date, authors, headline, and body. [32] demonstrates the relative ease of detecting GAN-generated images from a variety of state-of-the-art generators at the time of publication. [30] checks for consistency between a news article and its images and captions. [27] attempts to identify and attribute inconsistencies between images and their captions. [18] introduces and evaluates detection methods on a new dataset for the task of identifying various semantic inconsistencies between images and captions.

3 Data and Representation

3.1 Dataset Design

We construct our dataset using raw data accessed via CrowdTangle [3], a public insights tool owned and operated by Facebook. The platform can surface public Facebook posts, including sources such as posts by celebrities and news outlets. It does not include paid advertisements unless they began as organic, non-paid posts that were subsequently “boosted” using Facebook’s advertising tools. It also does not include activity on private accounts, or posts made visible only to specific groups of followers.

We used the historical data function of the platform to construct our dataset. With the historical data feature, we downloaded all public Facebook posts which had videos in the last decade from the US General Media group, for a total of 647,009 posts. This list of organizations was curated by CrowdTangle, and ranges from large, relatively non-partisan sources such as The Associated Press to smaller, more partisan sources such as Breitbart News. A full list of organizations and precise query time ranges are provided in Appendix A.1.

While CrowdTangle provides access to large amounts of Facebook posts, it has two limitations that impact this project. First, it does not provide labels for whether or not a post contains misinformation, nor can one construct a search explicitly filtering for misinformation. Second, it does not provide video files; they must be scraped from Facebook using other tools. CrowdTangle was therefore used to source possible posts to scrape, while video files were scraped using the open-source `youtube-dl` tool [4]. Due to this limitation, we were only able to scrape a sample of 4651 videos.

Figure 1: Example videos and captions from our dataset.

To construct a labelled dataset for multi-modal semantic alignment, we treat the original caption-video post pairs as pristine examples and randomly swap in new captions from other posts to generate inconsistent examples. Examples are shown below in Figure 1. In this manner, a pristine example features a real-world video and a real-world caption which were intended to relate to each other by the media organization which posted them. We assume that pristine examples are semantically consistent across modalities. An inconsistent example still features a real-world video and caption, except the video and caption are not taken from the same post. In an inconsistent example, the caption is the only modality which is manipulated. For the additional modalities described in following subsections, such as a video’s transcript and Facebook reactions, each example video is always paired with its matching transcript and reactions. This assumes that a random swap of caption would result in some amount of semantic mismatch between the new caption and the original video. In practice, half of the examples in our dataset are pristine and half are inconsistent.

We opt to perform swaps on real-world captions instead of creating inconsistencies by generating captions using large language models in order to avoid reducing the problem of identifying semantic inconsistencies across modalities to the problem of detecting whether or not a caption is computer-generated. We do not filter out newsrooms which may be using computer-generated captions in social media posts, as the proposed random swapping approach means any computer-generated content, if present, would not be correlated with semantic inconsistency.

This task is challenging because of the abstract relationships between captions, videos, and other modalities. The captions in our dataset do not represent dense descriptions of video clips, nor are they necessarily a literal description of a video. Our transcripts are noisy due to automatic generation, and are not guaranteed to be faithful representations of what is said. We do not have audio descriptions of videos. Our videos cover a wide range of styles and subjects, and are not necessarily well-lit and well-produced, as one might expect in datasets with movie clips. However, the random swapping approach does make this task easier than some more adversarial swapping strategies. We hope to strike a balance between perceived human difficulty and the challenge of learning abstract associations between modalities from a small set of noisy data.

3.2 Video Pre-Processing

After collecting video data, there are several steps taken to standardize formats and to prepare the files for input to our system. Each video is transcoded to a constant resolution of 256x256 pixels and a constant frame rate of 10 frames per second. All files are converted to mp4 videos, regardless of the original format. Audio is left unchanged. Video transcoding is handled using FFmpeg [6].

Because videos are scraped at random from Facebook, there is a very wide range of video lengths, styles, and subjects. In our dataset, the minimum video length is 1 second, the maximum length is 14 hours, and the mean is 8.5 minutes. To handle the long and variable video lengths, we adopt a keyframe-based approach. Each video is broken up into a sequence of 32-frame-long clips, with each clip beginning at a keyframe.

A keyframe is intended to be a point in the video where there is a change in scene or camera angle. These keyframes should be well-aligned to the starts of semantically consistent clips. In practice, keyframes are identified as timestamps in a video where the FFmpeg [6] scene detection filter is triggered, with the scene detection threshold set at 0.4. If no keyframes are detected, which might be the case with very short videos or videos which are all one shot, we create placeholder keyframes every 3.2 seconds, corresponding to 32 frames. In this manner, a clip with no detected keyframes is split into 32-frame-long clips every 32 frames. We use 16 keyframes per video, and 73% of videos in our dataset have at most 16 keyframes. We did not observe a significant difference in performance between using 8 or 16 keyframes.

Every video is transcribed using the DeepSpeech [8] transcription system. Before passing a video's audio stream into DeepSpeech, we transcode it using FFmpeg to the PCM signed 16-bit little-endian format with a sample rate of 16kHz, apply a highpass filter with cutoff 200Hz, and apply a lowpass filter with cutoff 3kHz. Using these filters, the generated transcripts are generally faithful and fluent, although they are imperfect and tend to misspell named entities. Below is an excerpt from an example audio transcript with typos generated using DeepSpeech:

> the fourth democratic presidential debate wrapped up in ohio on tuesday minnesota senator amicable no time getting back on the campaign trail she picked off with a tour of new

hampshire traveling to all ten counties and just thirty hours overcasting a wave of support after snagging the spotlight on tuesday night going head to head against fortune elizabeth warehouses not even the billionaire to protect billionaire wreaking time locked in and more than thirteen minutes that prefer in sir behind warren and former vice president joined some people like what they heard on twitter cobhouse received one point one million dollars in campaign donations in the twenty four hours after the debate . . .

While our generated transcripts are mostly correct, they tend to include misspelled names, along with other misidentified words. In this case, misspelled names include "amicable," "warehouses," and "cobhouse." The correct names are "Amy Klobuchar," "Warren," and "Klobuchar." These errors make it difficult to compare named entities in captions and transcripts, as transcript typos might not correspond to common human mistakes which might be corrected by reverse search or edit distance methods.

While some videos provide closed captions, we use automatically generated transcripts uniformly across our dataset to avoid introducing any linguistic biases in the fluency or style of transcripts from different sources.

3.3 Named Entity Verification

3.3.1 Facial Verification

We implement facial verification for named entities in order to check semantic consistency between modalities. This subsection will describe the implementation of our facial verification system.

We define facial verification in our context as checking whether or not people named in the caption of a video actually appear in the video. To accomplish this, we need to identify people in captions and build a database of representations for them. People are identified by using the named entity recognition (NER) feature available in the spaCy [12] natural language processing library. Using spaCy's `en_core_web_trf` language model, which implements RoBERTa [16], we run NER on our dataset of captions, and take all strings with the `PERSON` label as names of people. These strings are compiled into a set of people whose names appear in our dataset.

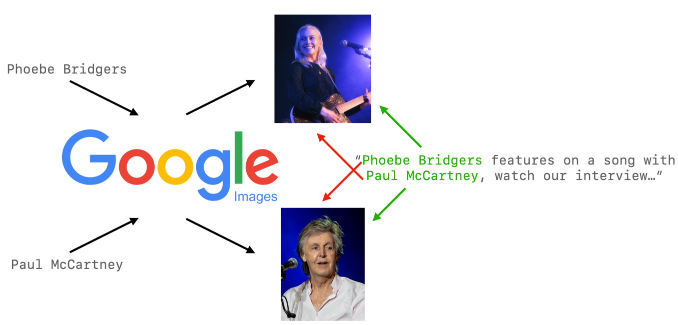

Figure 2: Representations of named entities in a caption, generated by querying Google Images, are compared against frames of a video. In this example, musicians Phoebe Bridgers and Paul McCartney could be verified by checking for their faces in the video, although there will also be apparent mismatches between each person's name and the other face present. Images courtesy [13, 25]. Best viewed in color.

Once all named people are identified, we need to compute a representation for each person. To this end, we query Google Images for the top 10 results for each name. These images are considered ground-truth references for how each named entity should appear, as shown in Figure 2. Having multiple images per name allows our dataset to contain examples of each person with potentially diverse lighting conditions, poses, ages, and camera angles.

Once reference images are collected, we use FaceNet [26] to compute facial recognition features for each image. The features for each set of 10 reference images are averaged to create a general representation for each name. At inference time, FaceNet features are also computed for a video's keyframes. We then take the cosine similarity between the representations for names appearing in the caption and the features for each keyframe in the video. (In practice, these keyframe features are pre-computed for efficiency.) The similarity scores are passed on to our model's classification head to be used alongside features from other modalities.

This approach to person identification has some drawbacks. This set of images is not manually curated, which introduces some issues such as the appearance of multiple people in a reference image. And in some cases, an individual might be referenced first by their full name, i.e. "Alice Appleseed," and then only by their first name, "Alice." Our NER approach does not account for this, and "Alice" would not be associated with "Alice Appleseed." In this case, we may try to verify the appearance of "Alice" in a video, without knowing which "Alice" we should look for. We would then try to compare the representation of the Google Images results for the common first name "Alice" against the keyframes of a video.

This is less of a problem for individuals who are commonly referred to by a single name, or a variety of distinctive names. For instance, celebrities can often be uniquely identified by their first or last name, and many politicians are referred to by their last names. While there will be separate reference images for the named entities "Kanye West" and "Kanye," or the entities "Nancy Pelosi" and "Pelosi," they will be faithful representations of the same person.

3.3.2 Name Verification

While we can attempt to verify the appearance of named entities from captions in videos, we can also try to compare captions to audio transcripts. This can help address the problem where an individual might be a topic of discussion, rather than a subject appearing in a video.

To accomplish this, we compute character-based embeddings for the names which appear in captions and/or transcripts. The intuition behind this operation is that we want to focus on misspellings, rather than any semantic concepts associated with names. Given a string representing a named entity, we convert each character in the string to its lower-case ASCII numerical value and pad to a maximum length of 64 characters. In our dataset, 100% of strings identified as names have at most 64 characters. We then feed this vector into a 2-layer fully connected network, with hidden size 64 and output size 32.

These name embeddings are then passed on to our classification head for use along with other modalities. By using learned embeddings, we are able to make comparisons between captions and audio transcripts, even when there are transcription errors in named entities.

3.4 Facebook Reactions

Since our data is collected from Facebook posts, we also have access to the Facebook reactions for each post. In Facebook, users are able to select the following reactions in response to a post: Like, Love, Wow, Haha, Sad, Angry, and Care. We hypothesize that these reactions can provide a coarse measure of the perceived semantics of an entire post, taking into consideration all of its modalities. In that case, the semantic inconsistency between an uplifting video paired with a sad or inflammatory caption might be reflected in an inconsistency between the post as a whole and its reactions.

We take the normalized reactions to a post as an input feature to our model. To normalize reactions, we divide the raw count of each reaction, such as Love, by the total number of reactions a post received. In this manner, viewers' reactions to content are separated from the popularity of a post, since without normalization it would be difficult to know what the order of magnitude should be for each reaction. One problem with this approach is that our data is collected from 2010 to 2020, but reactions were first introduced in 2016. So, for some posts in our dataset, users would not have been able to choose a reaction other than Like. Additionally, the Care reaction was added in 2020, so only the most recent posts can have non-zero Care reactions.

3.5 Ensemble Feature Extraction

We adopt a uni-modal ensemble approach to multi-modal fusion. To classify whether or not a post has a semantic inconsistency, we take as input the video split into clips starting at keyframes, the audio transcript, the normalized reactions to the video's pristine post, and a potentially inconsistent caption. In addition to the named entity verification features described in Section 3.3, we compute features for the caption, transcript, and video clip inputs.

Both the audio transcript and caption are processed using a pre-trained BERT [5] language model, implemented by HuggingFace [33]. When using the language model, inputs are truncated to their first 1024 characters, and split into two sets of characters with length 512. We need to create these splits because the maximum input length to the language model is 512 characters. In our dataset, 60% of audio transcripts and 99.97% of captions have at most 1024 characters.

The video clips are processed using both a video-understanding network and an object detection network. For video understanding, we use S3D-MIL-NCE (S3D) [21], and for object detection, we use ResNet [10]. S3D is run on the full 32-frame sequence in each of the video clips, while ResNet is run on the first frame of each clip. Implementation details are described in Appendix A.4.

3.6 Multi-Modal Fusion

For each modality, we learn an embedding to a shared semantic latent space. Figure 3 shows our full model architecture. Each embedding function is implemented as a 2-layer fully connected network, mapping from the output feature space of a modality's feature extraction network to a common 256-dimensional latent space. The learned semantic embeddings for video clips, object detection, audio transcript, and caption are concatenated and passed through a Long Short-Term Memory (LSTM) [11] module to condense information from the clips into one summary feature vector. This begins to fuse multi-modal content at the clip level, before the output of the LSTM is concatenated with named entity verification features. The final combined feature vector is passed on to our classification network. Our classifier is implemented as a 3-layer fully connected network. Additional details, including layer sizes, are described in Appendix A.4.

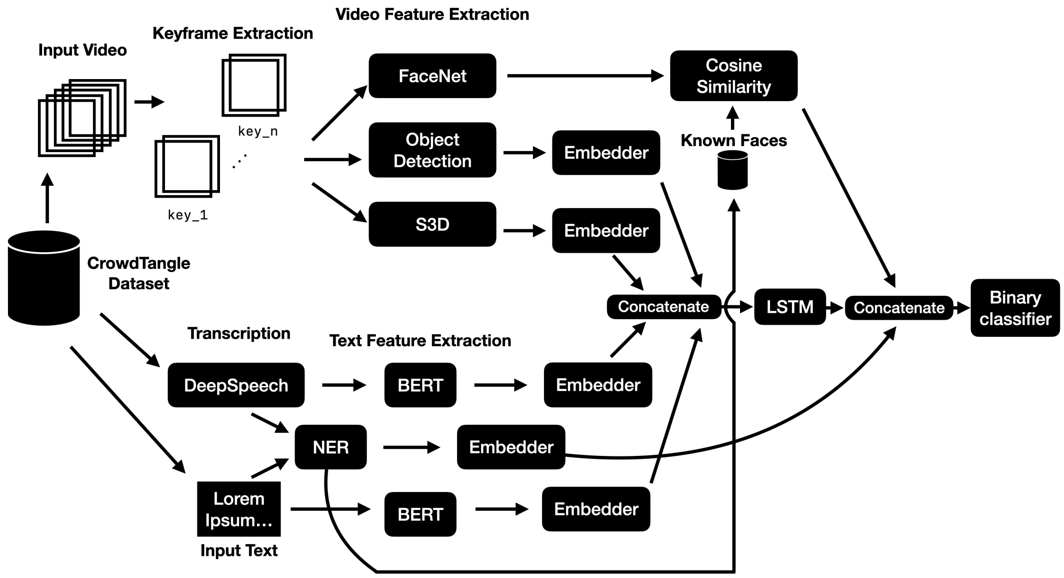

Figure 3: Our semantic inconsistency detection architecture. Modality-specific feature extraction is run in parallel, and features representing the content of each modality are concatenated with facial verification features in order to perform classification. In practice, expensive operations such as transcription are run in advance.

4 Experiments

4.1 Experimental Design

We train our model with the self-supervised dataset described in Section 3. We optimize the binary cross-entropy loss function, where our model attempts to classify caption, audio transcript, and video appearance tuples as either pristine or inconsistent.

We report classification accuracy for our experiments, calculated as the percentage of examples correctly identified as either pristine or inconsistent in our balanced test set. Our data is split such that 15% of the examples are reserved for the test set, and the other 85% for training and validation.

4.2 Results and Ablation Studies

Table 1: Binary classification accuracy (%) of heavily multi-modal models

	Modality or Feature Removed							
Model	None	Transcript	Caption	Video	Faces	Reacts	Names	Names & Faces
Full	58.3	57.0	54.2	54.7	56.9	57.4	52.4	49.8
No OD	**60.5**	59.5	51.5	56.5	59.6	**60.5**	54.8	49.9

We perform a variety of ablation experiments to characterize the impact of each modality on the accuracy of our model. Results are shown in Table 1, with each modality removed one-by-one. Note that "removing" a modality refers to removing its features or embeddings from our classifier. For instance, removing the video appearance makes the semantic video embeddings inaccessible to

Table 2: Binary classification accuracy (%) of uni- and bi-modal models

Modalities Used				
Caption	Video	Caption & Video	Name Verification	Facial Verification
49.9	49.8	49.6	**53.5**	51.7

Table 3: Full model confusion matrix (%)

	Predict Pristine	Predict Inconsistent
Pristine Examples	51.0	49.0
Inconsistent Examples	28.6	71.4

our classifier, although the video is still available for checking named entity consistency with facial verification.

Our best performance is achieved by using all modalities, except object detection features, and reaches classification accuracy of 60.5%. Table 3 shows the confusion matrix for this model. We observe that the model is more accurate when classifying inconsistent examples. Due to the fact that removing object detection features improved model performance, we perform one-by-one removal ablation studies again, with object detection features always removed. These experiments are referred to as "No OD" models in Table 1. Table 2 shows results for models using one or two modalities.

We observe that named entity verification is key to model accuracy. Without facial verification, classification accuracy decreases slightly to 59.6%. Without comparing names between captions and transcripts, classification accuracy falls to 54.8%. Without performing either consistency check, classification accuracy falls to 49.9%, essentially random.

We find that named entities are not the only useful information provided by captions. When semantic embeddings for captions are removed, accuracy falls to 54.2% and 51.5%, depending on whether or not object detection features are present, respectively. When caption embeddings are removed, the names present in the caption are still made available to our named entity verification process. Combination of semantic embeddings and named entity verification is the best use of information in the caption modality.

We note that video embeddings from S3D are more important than object detection embeddings from ResNet. In fact, removing ResNet embeddings improves the performance of our model, while removing S3D embeddings lowers performance. When ResNet embeddings are present, removing S3D embeddings leads to 3.8% lower accuracy, and without ResNet embeddings, removing S3D embeddings leads to 4% lower accuracy.

This could be due to the fact that features from S3D contain representations of objects, so the contribution of object detection features is diluted. ResNet features are not temporally aware, and so they cannot contain all the information represented in S3D features. Furthermore, the ResNet50 model we take features from is trained for image classification, which may be too general of a task to be useful for modelling abstract video semantics.

We note that Facebook reactions do not seem to provide a useful signal, as removing them from our model did not decrease performance.

Finally, we observe that multi-modal fusion is necessary for achieving the best possible accuracy. Removing any one of our modalities decreases performance, with the exception of reactions, and no uni-modal model can perform significantly better than random. Caption- and video-only models achieve 49.9% and 49.8% classification accuracy, respectively, confirming that our dataset does not have linguistic or visual bias. A model combining caption and video clip embeddings achieves 49.6% accuracy, highlighting the importance of incorporating additional modalities and features. A model which solely compares named entities in captions and audio transcripts achieves 53.5% accuracy, and a model which compares named entities in captions with video frame facial verification features achieves 51.7% accuracy. While attending to named entities is important, named entities alone are not sufficient for our model to achieve the highest possible accuracy.

5 Conclusion

We have introduced a novel multi-modal semantic inconsistency detection system, along with a 4k-large dataset for self-supervising semantic alignment detection in real-world social media posts. We demonstrate the importance of making use of modalities beyond video appearance and captions, including transcription, facial verification, and possibly misspelled named entity comparison.

We observe that fusion across modalities is key to detecting semantic inconsistencies. We find that named entities can provide strong signals for verifying consistency across modalities, and that verifying named entities using both language-based and visual methods is better than only using one. Semantic consistency checks cannot be fully explained by named entity verification, however, highlighting the need to consider semantic embeddings for language and video.

Future work could explore aspects of attributing and characterizing inconsistencies, beyond detection. Including explainable modules for facial verification and author attribution of various modalities could take steps towards addressing this. Our approach would likely benefit from more data, and we are interested in expanding data collection to other social networks such as Instagram, Twitter, and TikTok. Increasing the size of our dataset by one or more orders of magnitude might also allow for more challenging inconsistencies during training time.

Twitter's Birdwatch program could provide especially relevant data, as the program is releasing publicly-available labels on misinformation which were annotated by volunteers. These annotations may be inaccurate, and Birdwatch also has a system where volunteers can rate annotations. Birdwatch notes, as they are called, contain rich annotations which could be used to train modules for attribution and characterization. Annotations include fields for how believable a post is, how harmful a post may be, how difficult it is to validate information in a post, and whether a post might be misleading because of media manipulations, factual errors, outdated information, missing context, unverified claims, or satire. Annotators also provide a short written summary of their note. Applying multi-modal analysis on these posts and their annotations is left for future work.

Acknowledgments and Disclosure of Funding

I would like to thank my advisor, Professor Avideh Zakhor, and my lab partners Kehan Wang, David Deng, and Jordan Grelling for their insightful discussions this past year. This work was supported by the DARPA Semantic Forensics program. Funding for computing resources was provided by Google Cloud.

References

[1] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. *Is Space-Time Attention All You Need for Video Understanding?* 2021. arXiv: 2102.05095 [cs.CV].

[2] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL].

[3] CrowdTangle Team. *CrowdTangle*. Facebook, Menlo Park, California, United States, 2021.

[4] youtube-dl Developers. *youtube-dl*. Version 2021.01.24.1. 2021. URL: https://youtube-dl.org.

[5] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].

[6] FFmpeg Developers. Version 4.3.1. 2020. URL: http://ffmpeg.org/.

[7] Amirhossein Habibian, Thomas Mensink, and Cees G. M. Snoek. "Video2vec Embeddings Recognize Events When Examples Are Scarce". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.10 (2017), pp. 2089–2103. DOI: 10.1109/TPAMI.2016.2627563.

[8] Awni Hannun et al. *Deep Speech: Scaling up end-to-end speech recognition*. 2014. arXiv: 1412.5567 [cs.CL].

[9] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6546–6555.

[10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[11] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[12] Matthew Honnibal et al. *spaCy: Industrial-strength Natural Language Processing in Python*. 2020. DOI: 10.5281/zenodo.1212303. URL: https://doi.org/10.5281/zenodo.1212303.

[13] David Lee. *Phoebe Bridgers (41599189180) (cropped).jpg*. [Online; accessed May 7, 2021]. 2018. URL: https://commons.wikimedia.org/wiki/File:Phoebe_Bridgers_(41599189180)_(cropped).jpg.

[14] Gen Li et al. *Unicoder-VL: A Universal Encoder for Vision and Language by Cross-modal Pre-training*. 2019. arXiv: 1908.06066 [cs.CV].

[15] Linjie Li et al. "HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2046–2065. DOI: 10.18653/v1/2020.emnlp-main.161. URL: https://www.aclweb.org/anthology/2020.emnlp-main.161.

[16] Yinhan Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. 2019. arXiv: 1907.11692 [cs.CL].

[17] Jiasen Lu et al. *ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks*. 2019. arXiv: 1908.02265 [cs.CV].

[18] Grace Luo, Trevor Darrell, and Anna Rohrbach. *NewsCLIPpings: Automatic Generation of Out-of-Context Multimodal Media*. 2021. arXiv: 2104.05893 [cs.CV].

[19] Huaishao Luo et al. "UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation". In: *arXiv preprint arXiv:2002.06353* (2020).

[20] Ad Fontes Media. *Interactive Media Bias Chart*. [Online; accessed May 7, 2021]. 2021. URL: https://www.adfontesmedia.com/interactive-media-bias-chart/.

[21] Antoine Miech et al. "End-to-End Learning of Visual Representations from Uncurated Instructional Videos". In: *CVPR*. 2020.

[22] Antoine Miech et al. "HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips". In: *ICCV*. 2019.

[23] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[24] Alec Radford et al. "Language Models are Unsupervised Multitask Learners". In: (2019).

[25] Raph_PH. *Paul McCartney in October 2018.jpg*. [Online; accessed May 7, 2021]. 2018. URL: https://commons.wikimedia.org/wiki/File:Paul_McCartney_in_October_2018.jpg.

[26] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015). DOI: 10.1109/cvpr.2015.7298682. URL: http://dx.doi.org/10.1109/CVPR.2015.7298682.

[27] Ravi Shekhar et al. "FOIL it! Find One mismatch between Image and Language caption". In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, July 2017, pp. 255–265. DOI: 10.18653/v1/P17-1024. URL: https://www.aclweb.org/anthology/P17-1024.

[28] Weijie Su et al. *VL-BERT: Pre-training of Generic Visual-Linguistic Representations*. 2020. arXiv: 1908.08530 [cs.CV].

[29] Hao Tan and Mohit Bansal. "LXMERT: Learning Cross-Modality Encoder Representations from Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. 2019.

[30] Reuben Tan, Bryan A. Plummer, and Kate Saenko. "Detecting Cross-Modal Inconsistency to Defend Against Neural Fake News". In: *Empirical Methods in Natural Language Processing (EMNLP)*. 2020.

[31] Ashish Vaswani et al. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[32] Sheng-Yu Wang et al. "CNN-generated images are surprisingly easy to spot...for now". In: *CVPR*. 2020.

[33] Thomas Wolf et al. "Transformers: State-of-the-Art Natural Language Processing". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. URL: https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[34] Saining Xie et al. *Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification*. 2018. arXiv: 1712.04851 [cs.CV].

[35] Rowan Zellers et al. "Defending Against Neural Fake News". In: *Advances in Neural Information Processing Systems 32*. 2019.

[36] Chen Zhao, Jungang Han, and Yang Jia. "3D Inception Convolutional Neural Networks for Automatic Lung Nodule Detection". In: *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*. 2017, pp. 1649–1653. DOI: 10.1109/CSCI.2017.287.

A Appendix

A.1 Supplemental Details on Data Collection

Figure 4: The CrowdTangle data analytics platform [3].

The full list of news organization Facebook pages included in the US General Media group is: ABC News, AJ+, Al Jazeera English, AP, ATNN:, Axios, BBC News, Blavity, Bloomberg, Breitbart, Business Insider, BuzzFeed News, CBS News, CNBC, CNN, CNN Politics, Complex News, CRWN MAG, Daily Wire, Financial Times, Fox News, FRONTLINE | PBS, Guardian US, HuffPost Black Voices, HuffPost Latino Voices, Los Angeles Times, MSNBC, NBC News, New York Daily News, New York Post, NewsOne, Newsweek, Noticias Telemundo, NowThis, NPR, OZY, PBS NewsHour, POLITICO, Quartz, Quartz News, RedState, Reuters, Splinter, The Atlantic, The Christian Science Monitor, The Daily Beast, The Daily Caller, The New York Times, The Root, The Wall Street Journal, The Washington Times, TheBlaze, TheGrio, TIME, U.S. News and World Report, Univision, USA TODAY, USA TODAY Video, VICE, VICE News, Vox, Washington Examiner, Washington Post, and Yahoo News.

Historical data was requested for these pages in the time range between 2010-09-24 - 2020-09-24. The requests were filtered to only include video posts, and no other filters, such as topic or keyword, were applied. Historical data does not include posts which have been deleted.

A.2 Reputable and Disreputable Media

One avenue of research that did not pan out was an attempt to group news organizations into "reputable" and "disreputable" categories, such that we could have a more controlled approach to generating semantic inconsistencies. In this setting, mismatches would be generated by swapping in captions from the opposite reputability category. For example, a video post from The Associated Press (AP) would have its caption replaced by a sample from Breitbart. This might create an inconsistency between the sentiments, styles, and levels of polarization present in the given video and caption. Similarly, a video from Breitbart might be paired with a caption taken from AP.

In order to gauge the reputability of a news organization, we turned to the Media Bias Chart produced by Ad Fontes Media [20], shown in Figure 5. Using the Media Bias Chart's "News

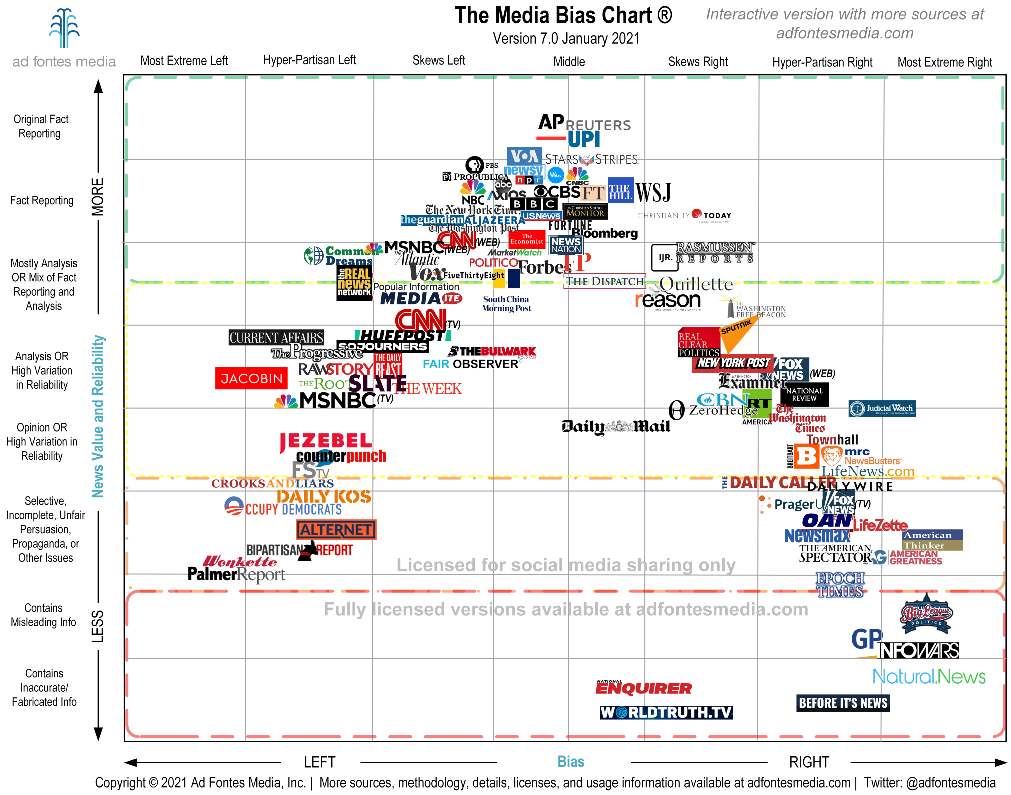

Figure 5: Ad Fontes Media Media Bias Chart, Version 7.0. Courtesy [20].

Value and Reliability" axis, groups deemed to be in the "Selective, Incomplete, Unfair Persuasion, Propaganda, or Other Issues," "Contains Misleading Info," and "Contains Inaccurate/Fabricated Info" categories were taken as "disreputable." Groups deemed to be in the "Mostly Analysis OR Mix of Fact Reporting and Analysis," "Fact Reporting," and "Original Fact Reporting" categories were taken as "reputable." Groups were not separated by partisan or political bias.

The full list of disreputable news pages is: AlterNet, American Greatness, American Thinker, beforeitsnews.com, Big League Politics, Bipartisan Report, Crooks and Liars, Daily Kos, Daily Wire, David Wolfe, Fox News, Gateway Pundit, Jacobin magazine, LifeZette, Newsmax, Occupy Democrats, One America News Network, Palmer Report, PragerU, REALfarmacy.com, The American Spectator, The Daily Caller, The Epoch Times New York, The National Enquirer, and Wonkette.

The full list of reputable news pages is: ABC News, Al Jazeera English, AP, Axios, BBC News, Bloomberg, CBS News, Christianity Today, CNBC, CNN, Financial Times, FiveThirtyEight, Forbes, Foreign Policy, Fortune, MarketWatch, MSNBC, Newsy, NPR, PBS, POLITICO, ProPublica, Rasmussen Reports, Reuters, Stars and Stripes, The Atlantic, The Guardian, The Hill, The New York Times, The Wall Street Journal, U.S. News and World Report, UPI News Agency, Voice of America - VOA, Vox, and Washington Post.

Historical data was requested for each of these groups. The time range was from 2010-02-15 to 2021-02-23. The requests were filtered to return video posts containing any of the following keywords: covid, coronavirus, covid-19, corona, rona, sars-cov-2, sars, pandemic, mask, masks, masking, vaccine, vaccinations, vaccinated, mrna, vax, vaxx, anti-vaxx, anti-vax, anti-vaccine, fauci, birx, moderna, and pfizer. Using these keywords, we hoped to model a threat scenario in which an actor takes a reputable and truthful video relating to the COVID-19 pandemic and misrepresents it by creating a more polarized or inflammatory caption presented to the user. Due to the topic constraints in the data request, random swaps are guaranteed to be at least somewhat related.

Using this approach to data collection, we were not able to generate results which were significantly better than random. This could be due to the fact that there was less data provided (2631 videos, to be precise) to solve a harder problem, as compared to random swapping on general news posts. We also had not added named entity verification methods to our model at the time of testing this data.

A.3 Keyframe Detection Details

The recommended range for scene detection thresholding is $[0.3, 0.5]$, and we opt for 0.4 in our dataset. While this is in some sense a model hyperparameter, it is prohibitively expensive to tune. The detected keyframes with this threshold appear qualitatively sufficient.

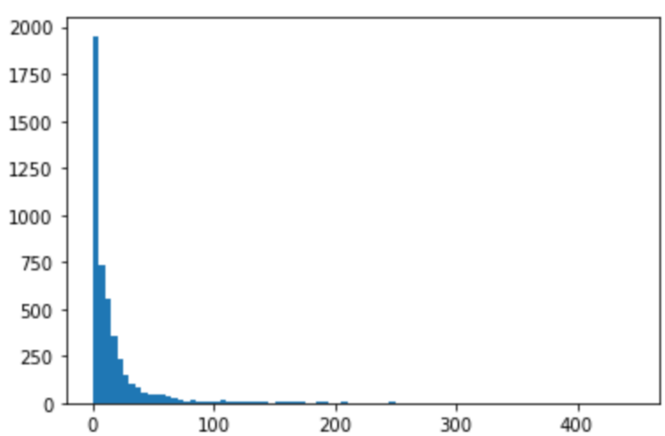

Figure 6: Histogram of the number of keyframes detected per video in our dataset. x-axis is keyframes per video, split into bins of width 5. y-axis is number of videos fitting into each bin.

As mentioned in Section 3.2, the lengths of videos in our dataset are highly variable. We observe that the distribution of detected keyframes per video is less variable, although has a very long tail. Figure 6 shows this distribution. The minimum, maximum, mean, and standard deviation of the distribution are: 0, 4629, 18.8, and 79.3, respectively. 54.6% of videos have at most 8 keyframes, and 73.1% of videos have at most 16 keyframes. To cover at least 95% of videos fully, we would need to process 66 keyframes per video.

We use 16 keyframes per video for analysis. If a video has more than 16 keyframes, we randomly sample a sequential subset of those keyframes. In other words, we do not necessarily process the first 16 keyframes, but hope to cover more of a video's semantic content, as adjacent keyframes might contain redundant information. The use of a fixed maximum number of keyframes was a practical consideration for our data processing and model training pipelines. Our model could be extended to process videos of arbitrary length.

A.4 Model Implementation Details

When processing text through BERT, we use the `BERT-base-uncased` language model implemented by HuggingFace [33]. Text is truncated to the first 1024 characters, and split into two sets of characters which are 512 long. We need to create these splits because the maximum input length to the language model is 512 characters. The features for these two passages are then averaged to represent the text as a whole. The feature vector is 768-dimensional.

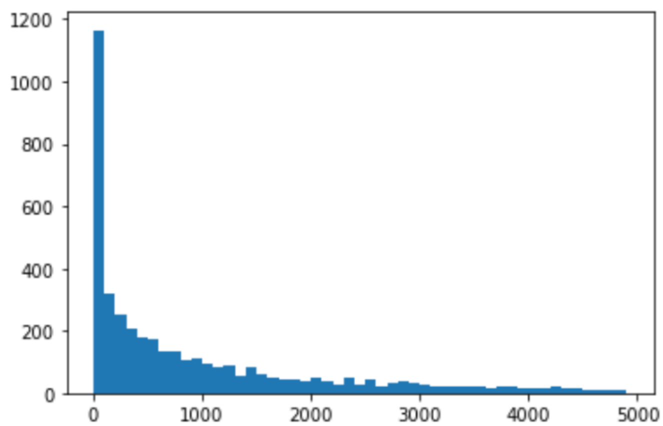

Figure 7: Histogram of the length of audio transcripts in characters. x-axis is characters per transcript, split into bins of width 50. y-axis is number of transcripts fitting into each bin.

The distribution of the lengths of audio transcripts is shown in Figure 7. There is a long tail, similar to the distribution of keyframes in Figure 6. The minimum, maximum, mean, median, and standard deviation of the distribution are 0; 100,785; 2,618; 618; and 6,940; respectively. 60.4% of audio transcripts have at most 1024 characters, and are therefore fully processed using our language model.

When processing video clips using S3D, we take the `mixed_5c` output of the video network as the clip feature, as recommended by the authors. We do not use the `video_embedding` output, as we want to learn our semantic embeddings which are specific to our task, rather than activity recognition. The `mixed_5c` output is 1024-dimensional, while the `video_embedding` output is 512-dimensional.

For object detection features with ResNet, we use the PyTorch [23] implementation of ResNet-50. Each feature vector is 1024-dimensional.

The learned embedding functions for each modality are implemented as 2-layer fully connected networks. They map from the input feature latent space to the shared semantic latent space. Formally, for each modality m, we have:

$$F_{\text{embed}}(x) = W_2^m(\sigma(W_1^m(x) + b_1^m)) + b_2^m$$

Where x is the input modality feature, σ is the leaky rectified linear unit nonlinearity (leaky ReLU), and $W_1^m, b_1^m, W_2^m, b_2^m$ are learned parameters. For each modality, if the input $x \in \mathbb{R}^n$, then $W_1^m \in \mathbb{R}^{n \times n}$, $b_1^m \in \mathbb{R}^n$ and $W_1^m \in \mathbb{R}^{s \times n}$, $b_2^m \in \mathbb{R}^s$, where n is the dimensionality of the input feature latent space and s is the dimensionality of the shared semantic latent space. In practice, $s := 256$, and n are as described above, per modality.

Our LSTM is uni-directional, with hidden size 1032 and dropout of 0.2.

Our classifier is implemented as a 3-layer fully connected network. The input size is 1096-dimensional, with hidden layer sizes 512 and 128, and output size 2. We use leaky ReLU as the activation function.

In practice, all modality features are pre-computed. Our learned parameters include all embedding functions, the LSTM, and the classifier. Without pre-computation, training is quite slow and cannot accommodate large batches on a single GPU. Computing facial verification features and generating transcripts are especially expensive operations.

A.5 Training Details

All models were trained using the ADAM optimizer with the StepLR learning rate scheduler. All models use an initial learning rate of 0.0001, decreasing by a factor of 0.1 every 10 epochs. All models are trained for 100 epochs.

All models were trained on a machine with an NVIDIA Tesla V100 GPU, a 4-core Intel Xeon CPU, and 54GB of RAM.