Multimodal Semantic Mismatch Detection in Social Media Posts

Kehan Wang, Seth Z. Zhao, David Chan, Avideh Zakhor and John Canny
University of California, Berkeley
Berkeley, USA
{wang.kehan, sethzhao506, davidchan, avz, canny}@berkeley.edu

***Abstract*—Short videos have become the most popular form of social media in recent years. In this work, we focus on the threat scenario where video, audio, and their text description are semantically mismatched to mislead the audience. We develop self-supervised methods to detect semantic mismatch across multiple modalities, namely video, audio and text. We use state-of-the-art language, video and audio models to extract dense features from each modality, and explore transformer architecture together with contrastive learning methods on a dataset of one million Twitter posts from 2021 to 2022. Our best-performing method benefits from the robustness of Noise-Contrastive loss and the context provided by fusing modalities together using a cross-transformer. It outperforms state-of-the-art by over 9% in accuracy. We further characterize the performance of our system on topic-specific datasets containing COVID-19 and Russia-Ukraine related tweets, and shows that it outperforms state-of-the-art by over 17% in accuracy.**

***Index Terms*—Multimedia Forensics, Semantic Mismatch, Multimodal Representation Learning, Deep Learning for Videos, Social Media**

I. Introduction

Short videos are becoming increasingly popular on social media these days - viral videos on TikTok, Instagram Reels, Twitter, and YouTube Shorts are receiving millions of views from all over the world. Video social media posts often have one short video as their main component, accompanied by a few sentences as a description or a reaction to the video. Both video and text components are displayed and consumed by users at the same time. In recent years, a great deal of research has been devoted to the study of understanding video and language together in the context of different tasks, such as action localization, video retrieval, video captioning, video question answering, and video-text inference.

However, the detection of semantic mismatches across modalities has received little attention in the context of video-and-language models. In this paper, we develop methods to identify social media posts that contain semantic mismatches among their modalities, e.g. video, audio, and text. Semantic mismatches can lead to misinformation, especially those generated at large volumes by automated engines. Many of them are known as cheap fakes, whereby either modality has been crudely manipulated. An example of a semantic mismatch is shown in Figure 1.

The challenges of semantic mismatch detection in social media video posts are two-fold: (1) learning a joint representation of video, audio, and text effectively; (2) lack of a large, labeled dataset for semantic matching. To address the joint representation learning problem, we propose a deep-learning-based method for learning accurate video-language joint distributions. To address the data issues, we introduce a novel training and evaluation method through random-mismatch, which does not require human labeling effort. Specifically, we collect one million social media video posts from Twitter to use as a large self-supervised training corpus and introduce two datasets on specific topics, namely COVID-19 and Russia-Ukraine Crisis related tweets. For labels used in self-supervision, we consider all collected tweets as semantically matched video and text pairs, and construct mismatched video and text pairs through random mismatch - given a video, we randomly select another tweet's text to construct a mismatched pair. Our method of representation learning outperforms state-of-the-art methods[9, 15] by 9.03%.

Fig. 1. **Example of semantic mismatch in a social media video post –** the video's mismatched text description in contains activity mismatch and topical shift.

The outline of this paper is as follows: Section II covers related work; Section III introduces our method; Section IV details experiments on semantic mismatch detection; and Section V concludes and discusses potential future directions.

II. Related Work

There is a large body of literature on detecting multi-modal semantic mismatch. Luo et al. [7] leverage the expressiveness of a large pre-trained contrastive model CLIP [13] to classify mismatch based on retrieval. While methods based on billion parameter models can be powerful, many users do not have access to the computing resources or data required to train such models. Recently, several methods have been proposed for detecting differences in image and text semantics. Singhal et al. [15] leverage a learned joint embedding space. However,

978-1-6654-7189-3/22/31.00 © 2022*IEEE*

they require both labeled positives and negatives in the data, and the work is specifically restricted to the news domain. Pan et al. [12] and Mayank et al. [8] focus explicitly on the textual description, detecting fake news using knowledge-graph based approaches. Tan et al. [16] and Fung et al. [5] focus on detecting synthetically generated news using text, image and knowledge element extraction. While these methods are feasible in situations where large labeled datasets of paired and unpaired semantic images and text exist, they do not transfer well to the more complex and sparsely labeled video domain.

In the video/text domain, Shang et al. [14] use video, audio, text, and metadata in TikTok videos to detect misleading COVID-19 video posts by fusing features from pre-trained models. However, they do not leverage representation learning, and their method requires strong supervision, leading to generalization issues in low-resource domains. McCrae et al. [9] extract video, text, transcript, and named entity information from a news post, and utilize pretext-task learning on randomly permuted data to supervise an LSTM-based model. Since the method directly fuses video and text at each keyframe through concatenation and does not learn a joint model of video and text, the model is unable to build complex joint representations. In this work, we use state-of-the-art video-language understanding methods, including cross-encoder, Noise Contrastive Estimation loss, and achieve 9.03% higher accuracy on random mismatch detection as compared to McCrae et al. [9]. We further discover that without joint representation, state-of-the-art methods are merely detecting topic mismatch in video and text, as shown in Section IV-D. In detecting mismatch on tweets of the same topic, our methods greatly outperform state-of-the-art methods and exhibit capabilities to detect semantic mismatch, rather than topic mismatch alone.

III. Method

Given a video post consisting of a video and a corresponding text description, we first use pre-trained models to extract video and text features. Then, the features are projected to a common representation space, which are learned through contrastive learning. Lastly, we use the projected video and text representations to classify whether the pair of video and text is a match or a mismatch.

A. Video and Text Feature Extraction

We first preprocess videos into the input format of our video model. We convert all videos into 10 frames-per-second and break each video into segments of 32 frames. For video model, we use S3D [10], pre-trained on activity recognition, to extract one 512-dimensional video feature per video segment, resulting in $v = (v_1, \dots v_n)\ v_i \in \mathbb{R}^{512}$. For the text input, we use DeBERTa-v3-Large [6], pre-trained using Masked Language Modeling, to extract token-level features, where a text feature is generated corresponding to each text token. This results in $t = (t_1, \dots t_m)\ t_i \in \mathbb{R}^{1024}$.

Given video features of $v = (v^1, \dots v^n), v^i \in \mathbb{R}^{512}$ and $t = (t^1, \dots t^m), t^i \in \mathbb{R}^{1024}$, we first use a linear projection to project all features onto the same dimension space $\mathbb{R}^{1024}$.

$$\begin{aligned} v' &= W_v v \\ t' &= W_t t \end{aligned} \tag{1}$$

where $W_v \in \mathbb{R}^{1024\times512}$ and $W_t \in \mathbb{R}^{1024\times1024}$ are learned parameters. Since video and text are both sequences of data, it is important to process the temporal information within each modality, rather than naïvely averaging all features together. As seen in Figure 2, we use transformers[17] to embed features of each modality and to retrieve temporal information. Video features $v' = (v'_1, \dots v'_n)$ and text features $t' = (t'_1, \dots t'_m)$ each are passed through a transformer to generate features of video and text context, $h = (h_1, \dots h_n), h_i \in \mathbb{R}^{1024}$ and $k = (k_1, \dots k_m), k_i \in \mathbb{R}^{1024}$, respectively. Next, we apply global mean pooling on features of each modality, h and k, and retrieve aggregated modality features, v^{embed} and t^{embed}:

$$\begin{aligned} v^{\text{embed}} &= \frac{1}{n}\sum_{i=1}^{n} h_i \\ t^{\text{embed}} &= \frac{1}{m}\sum_{i=1}^{m} k_i \end{aligned} \tag{2}$$

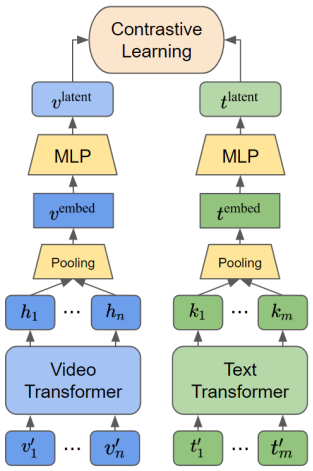


Fig. 2. **Contrastive Learning** – separate transformers for each modality, aggregation through mean pooling, contrastive learning on aggregated features.

B. Loss Functions for Contrastive Learning

We use contrastive learning to learn to project features onto a representation space, such that elements of matched semantics are close to each other, while those of mismatched semantics are away from each other.

For the architecture shown in Figure 2, we apply two different methods of learning the joint video and text representation using Contrastive Learning: (a) Cosine Embedding Loss[4] and (b) Noise Contrastive Estimation(NCE) loss[11].

We first project each representation $v^{\text{embed}}, t^{\text{embed}}$ into a latent space $v^{\text{latent}}, t^{\text{latent}} \in \mathbb{R}^{1024}$. It has been shown in recent

self-supervision studies[2, 3] that this approach learns a more disentangled representation space than not projecting the representations. We use 2-layer MLPs for projecting $v^{\text{embed}}, t^{\text{embed}}$ to a latent space, but still use the before-projection features as part of our representation. After projection, we obtain:

$$\begin{aligned} v^{\text{latent}} &= W_{2,v}\max(0, W_{1,v}v^{\text{embed}}) \\ t^{\text{latent}} &= W_{2,t}\max(0, W_{1,t}t^{\text{embed}}) \end{aligned} \tag{3}$$

where $W_{1,v}, W_{1,t} \in \mathbb{R}^{1024\times1024}$ are learned weight matrices in first layers, $W_{2,v}, W_{2,t} \in \mathbb{R}^{1024\times1024}$ are learned weight matrices in second layers, and $v^{\text{latent}}, t^{\text{latent}} \in \mathbb{R}^{1024}$.

1) Cosine Embedding Loss: Our first method uses Cosine Embedding Loss [4] to build the representation space of video and text. Given the embedded video and text features $v^{\text{latent}} \in \mathbb{R}^{1024}$ and $t^{\text{latent}} \in \mathbb{R}^{1024}$, we apply cosine embedding loss, $\mathcal{L}_{\text{cos\_video-text}}$ to construct the representation space of video and text:

$$\mathcal{L}_{\text{cos\_video-text}}(v^{\text{latent}}, t^{\text{latent}}, y) = \begin{cases} 1-\cos(v^{\text{latent}}, t^{\text{latent}}) & \text{y=0} \\ \max(0, \cos(v^{\text{latent}}, t^{\text{latent}})) & \text{y=1} \end{cases} \tag{4}$$

where y denotes the label of the pair of video and text, 0 for a match, and 1 for a mismatch. Cosine embedding loss $L_{\text{cos\_video-text}}$ encourages the vector angle between a matching pair of video and text to be smaller, and the angle of a mismatching pair to be larger. This allows the models to learn feature embeddings v^{embed} and t^{embed} by contrasting matching pairs with mismatching pairs.

2) NCE Loss: One problem with Cosine Embedding Loss is that it only considers one instance of positive or negative sample at a time. The process of constructing negative examples through random mismatch makes the authenticity of the negative sample noisy. For example, it is likely to generate one negative sample that is in fact matching, thereby learning on this sample would push the originally close video and text away from each other. Therefore, to reduce noise presented in the negative samples, we sample and learn on multiple negative samples at a time using a variant of NCE loss[11], similar to NCE loss used in CLIP[13].

Given a batch of B matching video and text, we have embedded video and text features $v_i^{\text{latent}}, t_i^{\text{latent}} \in \mathbb{R}^{1024}; i \in [1, B]$. We contrast each matching pair with all other mismatching pairs for both video and text. Specifically, given v_i^{latent}, the matching text feature is t_i^{latent}, and all other mismatching text features are $t_j^{\text{latent}}, j \neq i, j \in [1, B]$. Out of all B text features, we learn to classify the text feature matching v_i^{latent}. Therefore, we minimize the cross entropy of each video feature and its matching text feature, versus other text features in the batch. Following conventions used in CLIP[13], we l_2-normalize $v^{\text{latent}}, t^{\text{latent}}$ first:

$$\begin{aligned} \hat{v}^{\text{latent}} &= \frac{v^{\text{latent}}}{||v^{\text{latent}}||_2} \\ \hat{t}^{\text{latent}} &= \frac{t^{\text{latent}}}{||t^{\text{latent}}||_2} \end{aligned} \tag{5}$$

and also scale their dot product using a learned temperature parameter T. This results in the following video-to-text loss function:

$$\mathcal{L}_{video\to text} = -\frac{1}{B}\sum_{i=1}^{B}\left(\log\frac{\exp(T\hat{v}_i^{\text{latent}}\cdot\hat{t}_i^{\text{latent}})}{\sum_{j=1;i\neq j}^{B}\exp(T\hat{v}_i^{\text{latent}}\cdot\hat{t}_j^{\text{latent}}))}\right) \tag{6}$$

We also learn the reverse loss, namely given t_i^{latent}, we learn to classify which video feature is matching it among all B video features:

$$\mathcal{L}_{text\to video} = -\frac{1}{B}\sum_{i=1}^{B}\left(\log\frac{\exp(T\hat{t}_i^{\text{latent}}\cdot\hat{v}_i^{\text{latent}})}{\sum_{j=1;i\neq j}^{B}\exp(T\hat{t}_i^{\text{latent}}\cdot\hat{v}_j^{\text{latent}}))}\right) \tag{7}$$

We optimize using the mean of these two losses:

$$\mathcal{L}_{\text{NCE\_video-text}} = \frac{1}{2}\mathcal{L}_{text\to video} + \frac{1}{2}\mathcal{L}_{video\to text} \tag{8}$$

C. Cross-Transformer

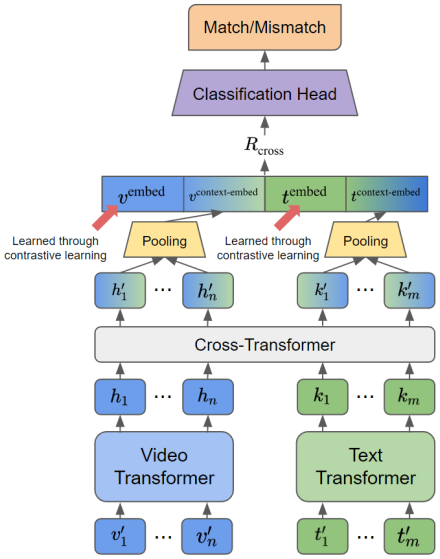


Fig. 3. Fusing video and text features using a Cross-Transformer.

To fully capture the temporal information in each modality, and allow video and text features to interact with each other, we further apply one cross-transformer as shown in Figure 3. It takes in $h_1, \ldots h_n$ and $k_1, \ldots k_m$, the last hidden state outputs of both video and text transformers before the global mean pooling, and uses attention mechanism to contextualize video and text features. Cross-transformer outputs new hidden states of video and text, $h'_1, \ldots h'_n$ and $k'_1, \ldots k'_m$, respectively. We then apply global mean pooling on features of each modality to obtain contextualized aggregated modality features, $v^{\text{context-embed}}$ and $t^{\text{context-embed}}$:

$$\begin{aligned} v^{\text{context-embed}} &= \frac{1}{n}\sum_{i=1}^{n}h'_i \\ t^{\text{context-embed}} &= \frac{1}{m}\sum_{i=1}^{m}k'_i \end{aligned} \tag{9}$$

We concatenate both contextualized and learned feature embeddings to obtain fused representation $R_{\text{cross}} \in \mathbb{R}^{4096}$:

$$R_{\text{cross}} = v^{\text{embed}} \oplus v^{\text{context-embed}} \oplus t^{\text{embed}} \oplus t^{\text{context-embed}} \quad (10)$$

With representation R_{cross}, we use a 4-layer MLP over the joint representation R_{cross} to regress the probability of matching $\hat{y}$:

$$\hat{y} = \sigma(\text{MLP}(R_{\text{concat}})) \quad (11)$$

which we supervise with binary cross-entropy loss L_{BCE}:

$$\mathcal{L}_{\text{BCE}} = y \cdot log(\hat{y}) + (1 - y) \cdot log(1 - \hat{y}) \quad (12)$$

D. Incorporating Audio

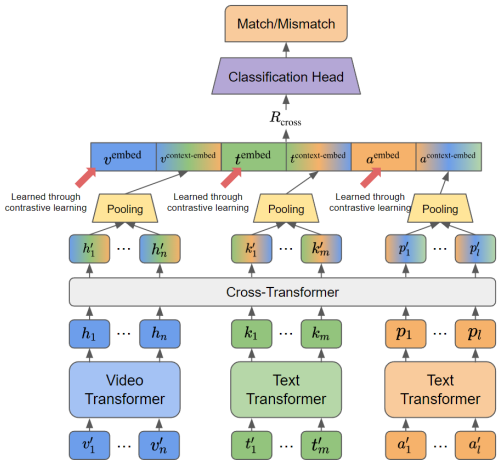


Fig. 4. Model architecture with audio feature input

We also explore adding audio as the third input feature to our pipeline, as shown in Figure 4. Specifically, we use Wav2vec 2.0[1] to transcribe the video's audio into text transcription, and use the same method, DeBERTa-v3 + text transformer, to extract transcription's features, $a' = (a_1, \ldots a_l) a_i \in \mathbb{R}^{1024}$. To build representation space for audio features, we use a modified NCE loss as follows:

$$\begin{aligned}\mathcal{L}_{\text{NCE\_video-audio-text}} = \frac{1}{2}(&\mathcal{L}_{text \to video} + \mathcal{L}_{video \to text} \\ &+ \mathcal{L}_{text \to audio} + \mathcal{L}_{audio \to text} \\ &+ \mathcal{L}_{audio \to video} + \mathcal{L}_{video \to audio})\end{aligned} \quad (13)$$

IV. Experiments

A. Datasets

In our dataset, we collect 1 million tweets using the Twitter API. We only consider tweets that contain both video and text and are not retweets/replies/quotes to other tweets. Our tweets' post time range from January 2021 to March 2022. To ensure an even data distribution, for each hour in the range, we collect 100 tweets that are posted within the hour.

For data cleaning, we remove any retweets/replies/quotes, and also tweets that are marked as "possibly_sensitive" and "possibly_sensitive_appeal", labeled by Twitter API. We then remove any tweets that contain a video shorter than 3 seconds or greater than 61 seconds, or a text shorter than 3 words. Twitter also imposes a 280-character length upper limit. Such removals remove 11% of the originally collected tweets. After data cleaning, there remain 943,667 tweets in total, with an average video length of 25 seconds. The data distributions for video and text after cleaning are shown in Figure 5. We use 80/10/10 split for train/validation/test. Given a video, we construct a random mismatch pair by randomly selecting another post's text from the same dataset split. For half of the tweets in each split, we construct random mismatch with replacement, where multiple videos could be mismatched with the same text.

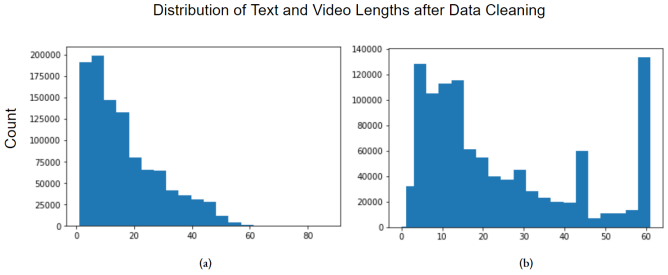


Fig. 5. **Video and Text Length Distribution in Collected Twitter Dataset after Filtering.** – (a) Number of words in Text; (b) Length of videos in seconds

Loss	Accuracy	Precision	Recall
$\mathcal{L}_{\text{BCE}}$	80.85%	78.92%	84.28%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{cos\_video-text}}$	81.45%	79.53%	84.76%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$	**85.43%**	**85.24%**	**85.44%**

TABLE I
EFFECT OF CONTRASTIVE LEARNING FOR VIDEO-TEXT MODALITIES

We use accuracy, precision, and recall to measure model performances. Precision and Recall are in terms of matching-post detection, where a match is considered positive and a mismatch is negative. We refer to our models by the losses used in each model's training: (1) $\mathcal{L}_{\text{BCE}}$; (2) $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{cos\_video-text}}$; (3) $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$; (4) $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-audio-text}}$, where the first three refer to the architecture in Figure 3, and the last is in Figure 4.

B. Different Losses for Contrastive Learning

We train three different models on Twitter 1M dataset to evaluate contrastive learning's effectiveness on semantic mismatch detection. Results are shown in Table I.

The model trained with only $\mathcal{L}_{\text{BCE}}$ does not construct any representation space and simply learns to classify semantic mismatch. If we add $\mathcal{L}_{\text{cos\_video-text}}$ to learn a noisy representation space, the model accuracy when measuring random-mismatch detection improves slightly by 0.60%. Adding $\mathcal{L}_{\text{NCE\_video-text}}$, on the other hand, learns a robust representation space and improves semantic mismatch detection accuracy by 4.6%.

Method	Accuracy	Precision	Recall
SpotFake[15]	72.05%	70.67%	74.96%
McCrae et al. [9]	76.40%	75.35%	78.56%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$	**85.43%**	**85.24%**	**85.44%**

TABLE II
PERFORMANCE COMPARISON FOR VIDEO-TEXT MODALITIES

Method	Accuracy	Precision	Recall
SpotFake[15]	50.09%	50.8%	5.57%
McCrae et al. [9]	50.43%	50.3%	72.76%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$	**69.5%**	**64.29%**	**92.9%**

TABLE III
EXPERIMENT ON COVID-19 RELATED TWEETS

C. Performance Comparison

We compare $\mathcal{L}_{\text{BCE}}+\mathcal{L}_{\text{NCE\_video-text}}$, with existing state-of-the-art methods, namely SpotFake[15] and McCrae et al. [9]. In our experiments, we trained both SpotFake and McCrae using the same random mismatch dataset. For McCrae's model, we removed the input branch of transcripts, because transcribing 1 million videos takes too long. As seen in Table II, our method outperforms [9] and [15] in semantic mismatch detection accuracy by 9.03% and 13.38% respectively.

D. Topic-Specific Random Mismatch

With the same models trained on 1 million Twitter dataset, we also test their performance on video-text random mismatches on a specific topic, namely COVID-19 or Russia-Ukraine crisis, as shown in Tables III and IV respectively. To avoid any training/testing data overlap, we recollect 41,000 twitter posts of COVID related terms from March to May 2022, and 60,000 twitter posts of Russia-Ukraine related terms from Feburary to April 2022 for testing purposes. We conduct the same data cleaning and random mismatching procedures in Section IV-A to the collected test data used.

In both COVID and Russia-Ukraine experiments, we see that previous state-of-the-art methods can only achieve random-guess accuracy at 50%. We speculate that these methods only learn to detect topic-mismatch, where video and text are on unrelated topics. Thus, they do not perform well on one-topic random mismatch testing. Our methods outperform state-of-the-art methods by 17.40% for COVID-19 and 18.47% for the Russia-Ukraine datasets, perhaps implying that they understand the fine-grain details in video and text, rather than only inferring using the general topic in video and text.

Note that the best performance on topic-specific mismatches is 18.06% and 16.98% lower for COVID-19 and Russia-Ukraine respectively, compared to ransom mismatches on all tweets from Table II. We believe this is caused by (1) the increasing difficulty of topic-specific mismatches, and (2) the out-of-distribution test data.

Method	Accuracy	Precision	Recall
SpotFake[15]	49.98 %	49.58%	2.96%
McCrae et al. [9]	49.97%	49.97%	69.72%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$	**67.37%**	**64.99%**	**75.27%**

TABLE IV
EXPERIMENT ON RUSSIA-UKRAINE RELATED TWEETS

Method	Accuracy
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-audio-text}}$	**76.24 %**
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$	74.63%
$\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_audio-text}}$	65.82%

TABLE V
TOPIC-SPECIFIC MISMATCH ON COVID-19 - DIRECTLY TRAINED ON TOPIC-SPECIFIC MISMATCH

E. Audio

We separately collect 60,000 COVID-19 related tweets with video and audio from Jan 2020 to Dec 2021, conduct data-cleaning, and split them into 80/10/10 train/valid/test. In Table V, we compare $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$, $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-audio-text}}$, and $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_audio-text}}$. $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_audio-text}}$ is a model that only considers audio and text. It uses the same architecture as Figure 4, but without the video branch. $\mathcal{L}_{\text{NCE\_audio-text}}$ is only computed between t^{embed} and a^{embed}. All models are trained on the 60,000 COVID-19 dataset's training split and tested on its test split. As seen in Table V, adding audio improves accuracy by 1.61%, compared to using only video and text as input. Furthermore, dropping the video branch results in a 10% drop, as seen in Table V.

F. Qualitative Examples of Topic-specific Mismatch Detection

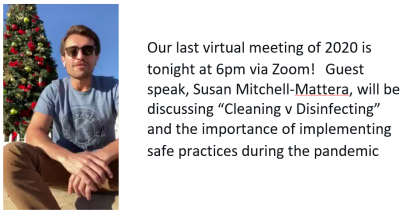


Fig. 6. A matched tweet https://twitter.com/HermosaKiwanis/status/1339260084388433923 correctly detected by our model.

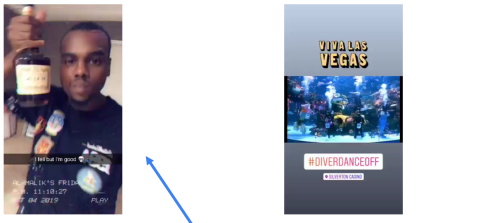
Fig. 7. A mismatched tweet correctly detected by our model. Video of the tweet on the left https://twitter.com/themohawkmike/status/1349877798509408257 is paired with text of the tweet on the right https://twitter.com/SteakImperator/status/1243657403654160384.

In this section, we show qualitative examples of our $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-audio-text}}$ model. In Figures 6 and 7, we show matched and mismatched posts that are classified correctly. In Figures 8

and 11, we show wrong predictions made by the same model. Figure 8 shows a matched post that is classified as mismatched. In this case, the video and text provided are unrelated, and the text seems to be a high-level comment of the video, which remains a challenge for vision and language understanding. Figure 11 shows a mismatched post that is classified as matched. The text focuses on "fun" and "entertainment", while the video shows a man dancing to the music. Likely, our model considers this relevancy between the text and video, and classifies them as matched.

Someone hand these guys some tin foil and ask em to make themselves a hat. #COVIDIDIOTS #COVID19 #WearAMask

Fig. 8. A matched tweet https://twitter.com/JillChipley/status/1276315836018499584 incorrectly detected by our model as mismatched.



I miss the me before covid

Here is some fun #OnlyVegas entertainment during quarantine! #DiverDanceOff #vegas

Fig. 9. A mismatched tweet incorrectly detected by our model as matched. Video of the tweet on the left https://twitter.com/kilamdead/status/1287614917420318720 is paired with text of the tweet on the right https://twitter.com/SilvertonCasino/status/1263982540802494465.

G. Detection of Semantic Mismatch in Russia-Ukraine Crisis

2.5 weeks later, I still have to applaud #Zelenskyy & the people from Ukraine. For their extraordinary amount of fortitude & their resilience.

I stand w/#Ukraine, in solidarity.

Fig. 10. A tweet https://twitter.com/LittleLeighXoxo/status/1502391215869726720 correctly detected as semantic mismatch by our model.

We use our $\mathcal{L}_{\text{BCE}} + \mathcal{L}_{\text{NCE\_video-text}}$ model to classify if the collected Russia-Ukraine related tweets contain semantic mismatch. In Figure 10, our model successfully detected semantic mismatch, since the text is in support of Ukraine and its people, but the video is only showing a sunflower. Figure 10 shows a mismatched post that is classified as matched. The text includes "send more supplies", while the video shows

Mr. Putin, try explaining YOUR ruthless war to this innocent child! @cnnbrk @CNN #UkraineRussianWar #UkraineRussiaWar #ukraine

#DEVELOPING: Global Empowerment Mission in #Doral, #Florida, is getting ready to send more supplies to those affected in #Ukraine

Fig. 11. A mismatched tweet incorrectly detected by our model as matched. Video of the tweet on the left https://twitter.com/KevinGLowery/status/1500676098312581123 is paired with text of the tweet on the right https://twitter.com/LeoFeldmanNEWS/status/1499388539305447426.

a child walking alone, in the background of people walking with bags. We hypothesize that the model considers the bags as supplies in the video, thus classifying them as a match.

V. Discussion and Future Work

To detect semantic mismatch across multi-modal social media posts, we developed effective representation learning methods. Our best-performing Contrastive Learning method achieves accuracy 9% and 13% higher than McCrae et al. [9] and SpotFake[15] respectively in random mismatch detection, and improves accuracy by 17% for COVID-19 and 18% for the Russia-Ukraine related topic-specific mismatch. We further show that learning a good representation is vital to improving semantic mismatch detection accuracy, and adding audio as an additional feature can lead to a performance increase.

Throughout the experiments in this paper, we have assumed that there is no mismatch across the modalities of the downloaded tweets. Upon examination of small subsets of the data, we have empirically verified that about 5% of the tweets have modal inconsistencies. We anticipate this small percentage not to significantly affect the results presented in this paper. In addition, we acknowledge that random mismatch of text with video/audio clips is overly simplistic and might not be a good proxy for misinformation detection. As such, future work should focus on learning harder mismatches of higher semantic similarity by mismatching a video/audio with a text of high similarity to the original text. Another area of future work has to do with making our models explainable. In [18], we made some progress in this direction by detecting semantic mismatch through per text-token inference using probabilistic arguments. Future work should also investigate Masked Language Modeling, which was shown to perform worse than the methods presented in this paper [18]. Additionally, it is worthwhile to investigate whether poor performance on topic-specific mismatch in Section IV-D is caused by domain shifts.

Acknowledgment

We would like to graciously acknowledge Google for partially providing cloud computing resouces for this project.

References

[1] Alexei Baevski et al. *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*. 2020. DOI: 10.48550/ARXIV.2006.11477. URL: https://arxiv.org/abs/2006.11477.

[2] Ting Chen et al. "A Simple Framework for Contrastive Learning of Visual Representations". In: *arXiv preprint arXiv:2002.05709* (2020).

[3] Xinlei Chen et al. *Improved Baselines with Momentum Contrastive Learning*. 2020. DOI: 10.48550/ARXIV.2003.04297. URL: https://arxiv.org/abs/2003.04297.

[4] S. Chopra, R. Hadsell, and Y. LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 539–546 vol. 1. DOI: 10.1109/CVPR.2005.202.

[5] Yi Fung et al. "InfoSurgeon: Cross-Media Fine-grained Information Consistency Checking for Fake News Detection". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 1683–1698. DOI: 10.18653/v1/2021.acl-long.133. URL: https://aclanthology.org/2021.acl-long.133.

[6] Pengcheng He, Jianfeng Gao, and Weizhu Chen. *DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing*. 2021. arXiv: 2111.09543 [cs.CL].

[7] Grace Luo, Trevor Darrell, and Anna Rohrbach. "NewsCLIPpings: Automatic Generation of Out-of-Context Multimodal Media". In: *CoRR* abs/2104.05893 (2021). arXiv: 2104.05893. URL: https://arxiv.org/abs/2104.05893.

[8] Mohit Mayank, Shakshi Sharma, and Rajesh Sharma. "DEAP-FAKED: Knowledge Graph based Approach for Fake News Detection". In: *CoRR* abs/2107.10648 (2021). arXiv: 2107.10648. URL: https://arxiv.org/abs/2107.10648.

[9] Scott McCrae, Kehan Wang, and Avideh Zakhor. *Multi-Modal Semantic Inconsistency Detection in Social Media News Posts*. 2021. arXiv: 2105.12855 [cs.CV].

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

[11] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. 2018. DOI: 10.48550/ARXIV.1807.03748. URL: https://arxiv.org/abs/1807.03748.

[12] Jeff Z. Pan et al. "Content Based Fake News Detection Using Knowledge Graphs". In: *SEMWEB*. 2018.

[13] Alec Radford et al. "Learning Transferable Visual Models From Natural Language Supervision". In: *CoRR* abs/2103.00020 (2021). arXiv: 2103.00020. URL: https://arxiv.org/abs/2103.00020.

[14] Lanyu Shang et al. "A Multimodal Misinformation Detector for COVID-19 Short Videos on TikTok". In: *2021 IEEE International Conference on Big Data (Big Data)*. 2021, pp. 899–908. DOI: 10.1109/BigData52589.2021.9671928.

[15] Shivangi Singhal et al. "SpotFake: A Multi-modal Framework for Fake News Detection". In: *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*. 2019, pp. 39–47. DOI: 10.1109/BigMM.2019.00-44.

[16] Reuben Tan, Bryan Plummer, and Kate Saenko. "Detecting Cross-Modal Inconsistency to Defend Against Neural Fake News". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2081–2106. DOI: 10.18653/v1/2020.emnlp-main.163. URL: https://aclanthology.org/2020.emnlp-main.163.

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

[18] Kehan Wang. "Representation Learning in Video and Text - A Social Media Misinformation Perspective". MA thesis. EECS Department, University of California, Berkeley, May 2022. URL: http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-140.html.