The Laughing Machine: Predicting Humor in Video

Yuta Kayatani[1] Zekun Yang[1] Mayu Otani[2] Noa Garcia[1]
Chenhui Chu[3] Yuta Nakashima[1] Haruo Takemura[1]
[1]Osaka University [2]CyberAgent Inc. [3]Kyoto University
{yuta.kayatani,yang.zekun}@lab.ime.cmc.osaka-u.ac.jp, otani_mayu@cyberagent.co.jp
chu@i.kyoto-u.ac.jp, {noagarcia@ids,n-yuta@ids,takemura@cmc}.osaka-u.ac.jp

Abstract

Humor is a very important communication tool; yet, it is an open problem for machines to understand humor. In this paper, we build a new multimodal dataset for humor prediction that includes subtitles and video frames, as well as humor labels associated with video's timestamps. On top of it, we present a model to predict whether a subtitle causes laughter. Our model uses the visual modality through facial expression and character name recognition, together with the verbal modality, to explore how the visual modality helps. In addition, we use an attention mechanism to adjust the weight for each modality to facilitate humor prediction. Interestingly, our experimental results show that the performance boost by combinations of different modalities, and the attention mechanism and the model mostly relies on the verbal modality.

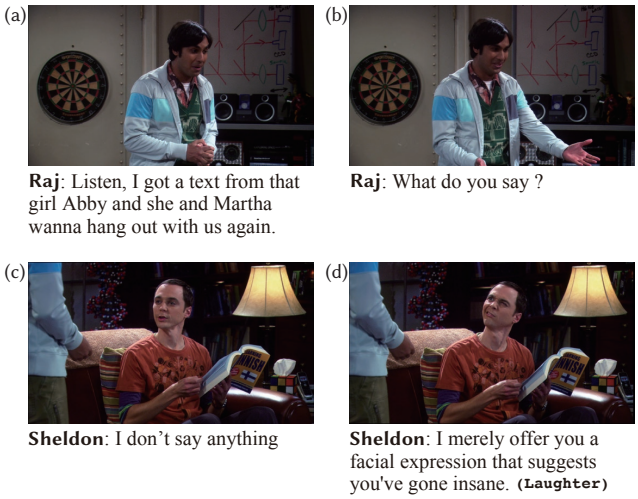

Figure 1. An example of a scene in which laughter happens. We can see that not only the utterance but also the facial expression are crucial cues for humor prediction.

1. Introduction

Humor plays an essential role in communication [24]. For example, some jokes in a presentation may draw the audience's attention. Even in a formal conversation, humor may make a person look more attractive and thus may lead to a better conclusion. Explicitly or implicitly knowing this, people try to provoke humor in their talks. This may also apply to human-machine interfaces. For instance, Apple's Siri has some repertories of jokes, which may imply that even commercial products try to acquire the capability to synthesize/understand humor.

Recently, a number of efforts have been done for understanding various aspects of a video (*e.g.* [39, 43]). Among them, understanding humor offers an interesting challenge in both the computer vision (CV) and natural language processing (NLP) fields: Firstly, humor is induced not only verbally but also visually (facial expressions, gestures, *etc.*) and vocally (tones, *etc.*). Any signals and their combinations that human bodies emit can cause laughter. Secondly, it obviously requires a deep understanding of semantics in *e.g.*, verbal, visual, and vocal signals. Thirdly, a joke has the dependency on the context that the signals have been formed, as pointed out by Raskin's work [33] through analysis of jokes' structure.

Existing literature, mostly in the NLP community, has tried to predict humor mainly focusing only on the textual modality [40, 32, 19, 26, 42, 3, 11, 21, 28], and other modalities have rarely been considered in spite of their potential importance for humor prediction. Figure 1 exemplifies this, in which an actor provokes humor by his facial expression. In this case, the actor's facial expression is a mandatory factor that causes laughter together with his utterance.

In this paper, we first construct a dataset on a popular TV sitcom series, "The Big Bang Theory" [10], consisting not only of sequences of subtitles (transcription of characters' utterances) but also of video frames. Humor labels are associated with timestamps so that humor involving only visual modality can be also predicted. We then present our model

for humor prediction, which additionally uses facial expression and who presents in the scene as extra modalities. We also adopt an attention mechanism so that the model can find informative modalities based on the context. Our main contribution is threefold:

- We create a new dataset for humor prediction. The dataset includes sequences of utterances and original video frames, as well as humor labels with timestamps that locate when laughter starts and ends in the video. This enables the prediction of silent humor solely invoked by the visual modality.
- We propose to use the visual modality to model humor in addition to the verbal modality. This enriches the context available for prediction. We use facial expressions and characters appeared in the scene as new modalities.
- We present the performance of our model with an attention mechanism for humor prediction over our new dataset yet with a similar set up to an existing work [2], which shows that the verbal modality provided as a sequence of subtitles is still the most informative cue.

2. Related Work

2.1. Humor Prediction

Humor prediction has drawn attention for a long time. Raskin [33] assumed that most humor is compatible with two or more sentences. These sentences are contrary to each other and then provoke humor. Raskin [33] further built the script-based semantic theory of humor, which impacts the research community of verbal humor and provides a basis for humor prediction.

Humor prediction has been formulated as a binary classification task. Purandare and Litman [32] predicted humor using a decision tree. Yang *et al*. combined word2vec [26] with human-centric features and used the nearest neighbor to predict humor [42]. Mihalcea and Strapparava [25] further classified humor into three types, *i.e*., alliteration, antonymy, and adult slang. Based on this taxonomy, they predict humor using naive Bayes classifiers and SVMs.

Recently, neural network-based approaches have been introduced for humor prediction. For example, Chen and Lee [11] represents text by word2vec and applies a 1D convolutional neural network (CNN). Bertero and Fung [3] assumes that the current utterance depends on its previous utterance and used long short-term memory (LSTM) [17] to model the subtitle sequence. Liu and Zhang [21] adopts syntactic structure features, such as complexity metrics, phrase length ratios, modifiers changes, which perform better compared to baselines without these features. Kiddon and Brun [19] proposed to use characteristics of humor in an erotic domain, *i.e*., nouns that serve as euphemisms for sexually explicit nouns and the sentence structure commonly appears in that domain. Taylor and Mazlack's work [40] is specialized to find humor in the domain limited to "Knock Knock" jokes. Although most of these studies have worked on humor prediction in English, Ortega-Bueno [28] proposed an attention-based LSTM [17] model with linguistic knowledge from Spanish social media data.

Although few, previous studies also have tried to use multimodal data for humor prediction. Bertero and Fung [2] proposed a model using not only natural language but also audio features. Kamrul *et al*. used natural language, audio, and visual features [16]. To evaluate their model, they constructed a multimodal dataset on TED talks. Ours is also on this line but with more focus on the multimodality of humor in TV drama series. Specifically, these existing datasets assign humor labels to sentences in subtitles/transcripts, whereas ours provides timestamps of laughter in the video, so that humor that is not caused by utterances can be also predicted.

2.2. Humor Generation

Some works have been dedicated to generating humor. Humor generation may be beneficial to promote smooth and natural NLP-based interfaces between humans and machines [4]. The *JAPE* system developed by Binsted and Ritchie [5] can automatically generate punning riddles based on some schemata and templates. The *standup* system [35], inspired by the ideas in the *JAPE* system, is a pun generator developed for children with complex communication needs. Stock and Strapparava's *HAHAcronym* system [38] can generate humorous acronyms.

Sjöbergh and Araki [37] automatically generated Japanese stand-up comedy. The generated scripts are further converted to speech, and the comedy routine is performed by robots. Labutov and Lipson [20] proposed a generalized humor generation approach based on the semantic script theory of humor [33]. Yoshida *et al*. [44] proposed a model that generates humor using the funny image and captions from the *Bokete* website.[1] Their model is based on image captioning but can generate captions provoking humor. Being different from previous studies, Petrović and Matthews [30] proposed an unsupervised humor generation model that can generate "I like my *X* like I like my *Y*, *Z*"-style jokes, where *X*, *Y*, and *Z* are variables.

2.3. Sarcasm Detection

Sarcasm conveys implicit information in a message, which is usually the opposite of what the person is saying. Sarcasm detection is especially crucial for sentiment analysis in social media, and research efforts have been made to

[1] https://bokete.jp/

build models for sarcasm detection using the textual modality. Sarcasm and humor share similar styles, such as exaggeration, irony, and satire; thus its detection may have some affinity with humor prediction. Many previous studies manually design features for classifying sarcasm and non-sarcasm [31] [6]. Davidov *et al*. [12] used semi-supervised learning on both social media and product reviews. Riloff *et al*. [34] focused on the contrast of positive sentiment in a negative situation, and they proposed a bootstrapping model for sarcasm detection. Mehndiratta [23] proposed to find sarcasm in tweets using a CNN.

Similar to humor prediction, sarcasm also depends on multimodal information besides utterances. Mishra *et al*. [27] studied if readers can understand sarcasm through modeling their gaze behavior. Similarly, Filik *et al*. [15] explored the behavior of both gazes and electrical brain activities when exposed to irony. Schifanella *et al*. [36] used visual features together with textual features for sarcasm detection. Castro *et al*. [8] created a multimodal dataset on TV shows and worked on sarcasm detection using multimodal information from videos. Cai *et al*. [7] used both texts and images for sarcasm detection and proposed a hierarchical fusion model.

3. Our Dataset

We build a dataset from a sitcom TV drama series "The Big Bang Theory," but the labels are associated with timestamps to better fit to the CV community, which allows us to make predictions when laughter is visually induced. Figure 2 shows an excerpt from our dataset, where video frames are resampled for illustration.

3.1. Laughter Extraction

To acquire humor/non-humor labels, we exploit the audio track of the video clips following [3]. This is because sitcom comes with a *laugh track*, in which audience laughter or *canned laughter* is recorded. Our original videos have audio tracks that are a mixture of the laugh track and the music track. We use this audio track for laughter extraction.

Let $s_l(\tau)$ and $s_r(\tau)$ denote the τ-th sample in the left and right channels of the audio track, respectively. To pinpoint laughter, we subtract these channels, *i.e*., $s(\tau) = s_l(\tau) - s_r(\tau)$ to cancel characters' utterances, which are usually located at the center, and compute the envelop of $s(\tau)$ using the Hilbert transform. By this, the signal can be decomposed to its envelop and instantaneous phase. Formally, the envelop is given by:

$$e(\tau) = |\mathcal{H}[s(\tau)]|, \tag{1}$$

where $\mathcal{H}[\cdot]$ denote the Hilbert transform, and $|\cdot|$ computes the magnitude of a complex value. Envelop $e(\tau)$ has the same sampling frequency as $s(\tau)$. Therefore, we downsample it to the original video's frame rate (*i.e*., 24 fps) and apply a low-pass filter with the cut-off frequency of 6 Hz to remove noises. We denote the signal after applying the low-pass filter as $e'(n)$ (Figure 3).

This can represent the degree of laughter for the n-th video frame; however, the audio track also contains sound effects that also elevates $e'(n)$. We manually exclude such sound effects. First, we empirically set the threshold over $e'(n)$ and roughly identify candidates of laughter segments (*i.e*., segments in which all $e'(n)$ is larger than the threshold). All candidates are reviewed, and non-laughter segments are removed. After removal, each extracted laughter segment is assigned with a humor label and is associated with the timestamps at which the laughter begins and ends.

Table 1. Statistics of our dataset.

# seasons	10
# episodes	228
Total time of videos	77 h 42 m
# humor labels	31,852
Total duration of laughter	19 h 55 m
# subtitles	74,212
# subtitles (humor)	32,791
# subtitles (non humor)	4,1426
Avg. # words in a subtitle	9.91
Avg. # words in a subtitle (humor)	9.46
Avg. # words in a subtitle (non-humor)	10.26
Avg. duration of a subtitle (*i.e*., an utterance)	2.51 s
Avg. duration of a subtitle (humor)	2.57 s
Avg. duration of a subtitle (non-humor)	2.47 s

3.2. Subtitles and Characters

Subtitles can be a critical hint for humor prediction because humor is often invoked verbally. In addition, who speaks in the story can also facilitate the prediction. Therefore, we include both subtitles and characters in our dataset. We found the subtitles available on the Internet with associated timestamps when utterances are made in the video; however, they do not have the identities of the speakers. On the other hand, we also found transcripts on the Internet, which come with the name of the character who makes each utterance. In order to associate each subtitle with the character name, we used a dynamic programming-based approach to align each line of subtitles and that of the transcript, and retrieved the character name for each subtitle. All subtitles are also associated with the timestamps at which the corresponding utterances begin and end.

3.3. Dataset Statistics

Table 1 shows the statistics of our dataset. The total duration of the videos in our dataset is over three days. Over 25% of the duration is associated with humor labels.

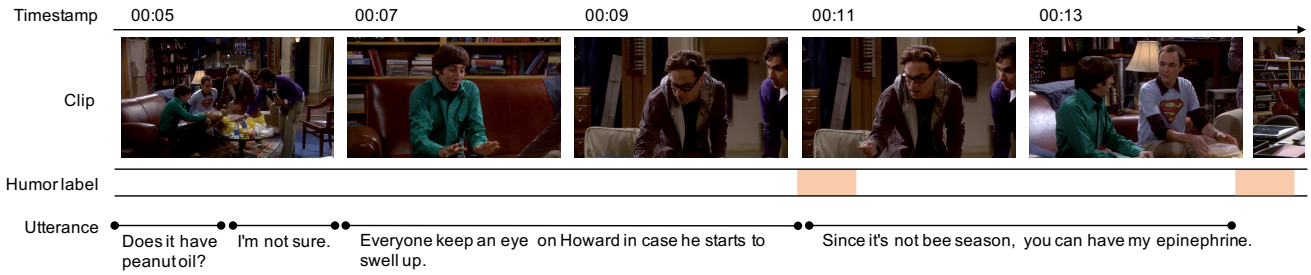

Figure 2. An example of video frames, timestamps, humor labels, and utterances in our dataset.

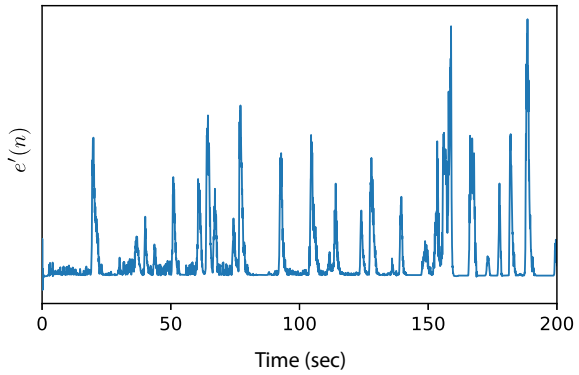

Figure 3. An illustrative example of envelop $e'(n)$ after down-sampling and low-pass filtering.

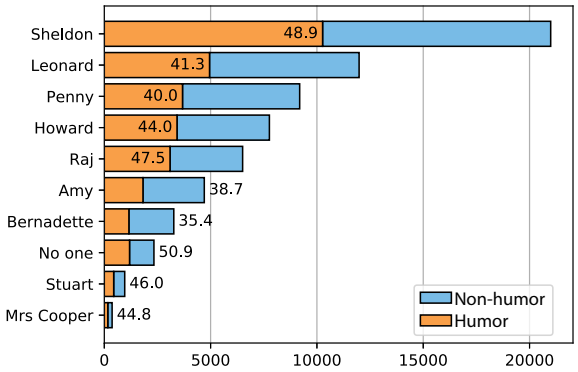

Figure 4. The number of subtitles of each character. The orange and blue bars indicate the number of subtitles by the character that involve humor and non-humor, respectively.

We also assign humor labels to subtitles to further investigate our dataset in the same way as [3]. For each subtitle with timestamp $(t^{\mathrm{b}}, t^{\mathrm{e}})$, we check if there is a humor segment in the duration from t_{e} to $t_{\mathrm{e}} + 1$; if there is, we assign a humor label (otherwise non-humor label) to the subtitle. The number of subtitles that are associated with the laughter label occupies 43.17% of all subtitles. We found that the average numbers of words in a single subtitle are not significantly different when it is associated with a humor label or not, which also applies to the average duration of a subtitle. This implies that there is no strong bias in this respect. The distribution of the numbers of subtitles per character is shown in Figure 4. We can observe that the proportions of subtitles that involve humor are not very different among the main characters. Therefore, who speaks itself cannot be a strong prior for humor prediction.

4. Humor Prediction Modeling

As mentioned above, we consider that humor inherently involves signals in various modalities that humans emit, as non-verbal signals play an important role in human-to-human communication. In this paper, we model humor using both visual and verbal modalities. Specifically, our model focuses on facial expression as the visual modality together with subtitles as the verbal modality. We incorporate character information into our model as well, as it can be also a hint for humor prediction. Our multimodal model thus predicts humor based on the subtitles, facial expressions, and characters. Figure 5 shows the network architecture of our model, which consists of four main components: punchline modeling, in-story laughter modeling, character-based modeling, and modality attention.

Given a sequence of subtitles $\{u_i | i = 1, \ldots, N\}$ and a sequence of video frames $\{v_n | n = 1, \ldots, K\}$, where N and K are the numbers of subtitles and video frames in an episode, our model predicts if subtitle u_i invokes laughter or not. In what follows, we detail each component.

4.1. Punchline Modeling

Humor, or more specifically a joke, usually consists of two parts: setups to construct the context and punchlines that actually provoke the humor [9]. Figure 1 is an excerpt of subtitles from our dataset. In this example, the first three subtitles serve as setup, and the last one is the punchline. In fact, the humor is formed not only by the last subtitle. The first three subtitles build up the context to make the final line a punchline, and the punchline *per se* is not always enough to provoke humor. That is, we need to model the temporal dependency of a sequence of subtitles. Therefore, we model subtitles with LSTM [17] that can handle dependency among each element in sequences.

As shown in Figure 6, our punchline model has a hierar-

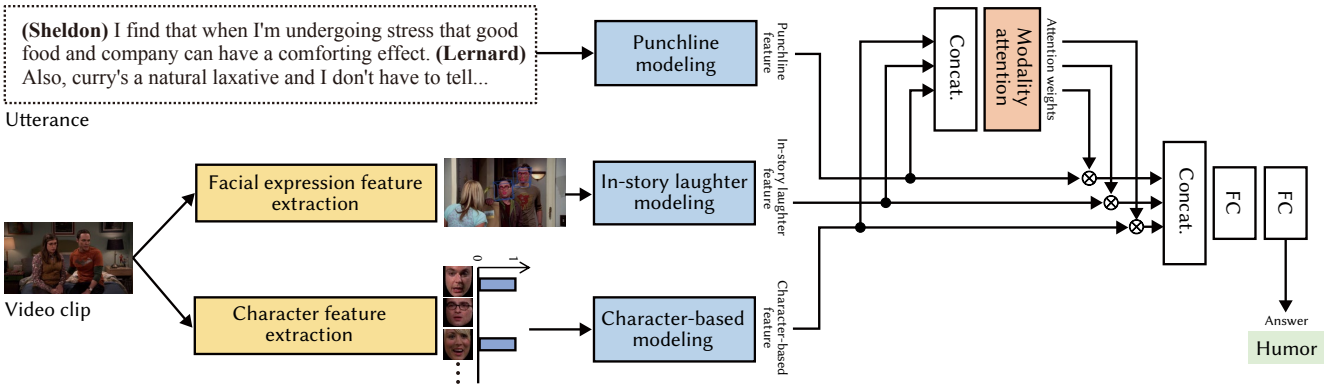

Figure 5. Our network architecture.

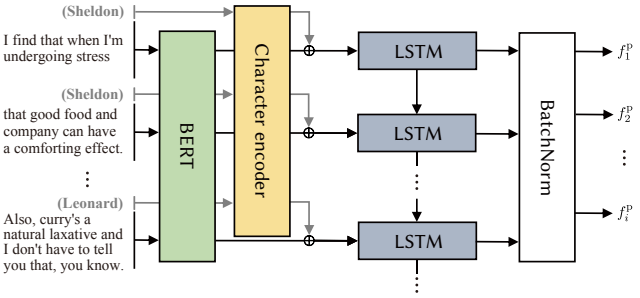

Figure 6. Punchline model.

chical structure of BERT [13] and LSTM [17]. Each subtitle u_i in an episode is encoded by BERT and it is fed to the LSTM cell. Through this structure, we model the long-term dependency that the humor may have. More specifically, we first preprocess the subtitles to remove textual descriptions of sound effects, environmental sounds, and character behaviors, such as "DOOR OPEN" and "PENNY NODS" as they can be noises for prediction. After removing them, we feed u_i to the BERT model. The output b_i of BERT corresponding to `[CLS]` token is then fed to the LSTM cell. The output of each LSTM cell goes through batch normalization [18], which is used as punchline feature f_i^{p}.

According to Figure 4, no apparent correlation between characters and humor labels is observed. Yet, the combination of the subtitle and the corresponding character may provide beneficial cues for prediction (*e.g.*, some characters prefer a certain type of humor to others). We extend BERT's output b_i with character encoding. Specifically, a character embedding vector c_i (based on a trainable embedding matrix) is fed to a batch normalization layer, followed by FC, batch normalization, and hyperbolic tangent layers, output c_i is added to b_i and then go through LSTM cells to get character-extended punchline feature $\hat{f}_i^{\mathrm{p}}$.

4.2. In-story Laughter Modeling

Non-verbal signals from humans can be informative cues for humor prediction. For example, funny facial expressions that occur in the story can be highly correlated with our humor labels. Inspired by this observation, we employ in-story laughter feature f_i^{s} derived from facial expression.

For this, we collect video frames (resampled at 1 frame per second) in $\{v_n\}$ that are in a temporal range of $(t_i^{\mathrm{b}}, t_i^{\mathrm{e}} + 1)$, where t_i^{b} and t_i^{e} are the timestamps of subtitle u_i at which it starts and ends. The range is extended by one second because facial expression may appear after the subtitle. We denote the set of video frames in this range by V_i. We use OpenFace[2] [1] to detect all faces in $V_i = \{v_{ij} | j = 1, \ldots, K_i\}$, where K_i is the number of video frames, and compute the facial action coding system (FACS) [14]. The FACS encodes displacement in a face into 35-dimensional vector, called action units, which are correlated to facial muscle movement (both continuous and binarized ones). Let d_{ij}^m denote the m-th facial region in v_{ij}, whose face detection confidence is $c_{ij}^m \in [0, 1]$. We compute action units a_{ij}^m for each detected face and then their weighted sum with confidences over all faces in V_i, *i.e.*,

$$a_i = \sum_{j,m} c_{ij}^m a_{ij}^m, \tag{2}$$

which goes through a fully-connected (FC) layer and batch normalization to obtain in-story laughter feature f_i^{s}.

4.3. Character-based Modeling

As mentioned in Section 4.1, who speaks, or who are in the scene, can be a cue for humor prediction. In addition to the character-extended punchline feature $\hat{f}_i^{\mathrm{p}}$, we incorporate the characters into our model to leverage richer context. We use a face recognizer to n order to complement $\hat{f}_i^{\mathrm{p}}$.

[2] `https://github.com/TadasBaltrusaitis/OpenFace`

We utilize an implementation of a face recognizer[3] to identify 17 main characters appear in the show. We apply the recognizer to each frame in V_i and built an 18-dimensional multi-hot vector q_i (17 characters and *unknown*), each element of which is set to 1 if the corresponding character is recognized in V_i. The vector is fed to an FC layer and batch normalization to obtain character-based feature f_i^{c}.

4.4. Modality Attention

Our overall model is built on top of three different models for feature extraction. These features may not be always equally informative depending on the context. We thus introduce an attention mechanism to weight each modality as shown in Figure 5. Specifically, after concatenating all three features into a single vector $f_i = f_i^{\mathrm{p}} + f_i^{\mathrm{s}} + f_i^{\mathrm{c}}$, we compute an attention weight by:

$$\alpha_i = \mathrm{softmax}(\mathrm{MLP}(f_i)), \tag{3}$$

where $\alpha_i = (\alpha_i^{\mathrm{p}}, \alpha_i^{\mathrm{s}}, \alpha_i^{\mathrm{c}})$ and $\mathrm{MLP}(f_i)$ is a multi-layer perceptron that consists of two FC layers with ReLU nonlinearity and batch normalization inserted between the FC layers. The feature vectors are fused with a_i as follows:

$$\bar{f}_i = (1+\alpha_i^{\mathrm{p}})f_i^{\mathrm{p}} + (1+\alpha_i^{\mathrm{s}})f_i^{\mathrm{s}} + (1+\alpha_i^{\mathrm{c}})f_i^{\mathrm{c}}.$$

$\bar{f}_i$ is fed to two FC layers with ReLU nonlinearlity and then the sigmoid function to produce a score. Note that f_i^{p} can be replaced with $\hat{f}_i^{\mathrm{p}}$

5. Experiments

5.1. Settings

We split the dataset into 80%, 10%, and 10% for training, validation, and test splits, respectively. We implemented our model with PyTorch [29]. We used the BERT implementation of [41].[4] A certain subtitle with its four preceding subtitles were fed to the model as context when training, whereas all subtitles in an episode were fed to the model when testing. The dimensionalities of feature vectors f_i^{p}, f_i^{s}, and f_i^{c} were all set to 512. The loss was binary cross-entropy applied to the score, and AdamW [22] was adopted as an optimizer. The mini-batch size and learning rate were set to $1,000$ and 10^{-5}, respectively. Dropout was inserted before the last two FC layers with a ratio of 0.5.

As for face recognition for character-based modeling, we retrieved a single face image from the Internet for each of the ten main characters of the show as exemplars. The recognized faces may be correlated with the character name associated with each subtitle. Taking these character names as ground-truth for our face recognizer, we computed the F1-score as well as precision and recall, shown in Table 2. The relatively high recall rates imply that characters appear in the scene are mostly covered by the recognized faces, although they are not necessarily the actual speakers.

[3] https://github.com/ageitgey/face_recognition

[4] https://github.com/huggingface/transformers

5.2. Results and Discussion

We again used accuracy, F1-score, precision, and recall as performance metrics for our humor prediction. To show the contribution of each module, we tried various combinations of them. We use Punch, Story, Char, Att, and Ext to represent the punchline model, in-story laughter model, character-based model, modality attention, and character extension of punchline features, respectively. Table 3 shows the performances of naive baselines. Performance for random prediction is the average over 100 trials.

5.2.1 One Modality Cases

Table 4 shows performance with a single modality (either Punch, Story, or Char). For showing the advantage of finetuned BERT in the punchline modeling (Punch w/ FT), we tested BERT without finetuning (Punch w/o FT), where BERT's parameters were fixed in training. The subscript 1 of the punchline modeling means that the model does not use any subtitle as context and use an FC layer instead of LSTM, which demonstrates the importance of the context. Punch w/ FT clearly outperformed the models without finetuning and without context. We use Punch w/ FT in the following experiments.

In-story laughter modeling Story offers almost no hint for humor prediction. Character-based modeling has a very limited capability. The latter is not surprising according to Figure 4, implying that the probability of humor-labeled subtitles given the character is almost the same for all characters. Comparing the modalities, punchline modeling is significantly better than others, which is also reasonable as the main structure of humor seems to be formed in the verbal modality in most cases.

5.2.2 Ablation Studies over Input Modalities

The first section of Table 5 shows the ablation results that demonstrate the advantage of modality combinations. We observed that our model solely using punchline modeling can achieve 70.50% accuracy, which is the best among all. The performance boost by additional modality was rather limited. These results demonstrate that, at least in our dataset, the punchline modeling provides a strong cue for humor predictions, while modeling the other modalities is challenging or these modalities may have limited signals for humor prediction.

The second section of Table 5 shows results on some combinations of modalities with modality attention. Modality attention does not improve the performance, or even

Table 2. The performance of face recognition for character-based modeling when the characters associated with subtitles can be viewed as ground-truth.

	Train			Validation			Test		
	F1	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**
Sheldon	66.05	51.39	92.42	64.67	49.91	91.83	64.56	49.59	92.49
Leonard	61.00	48.06	83.47	62.33	49.57	83.92	58.90	44.55	86.88
Penny	62.94	49.76	85.62	63.34	50.29	85.54	62.29	49.62	83.65
Howard	59.76	46.65	83.13	62.44	49.45	84.68	58.05	44.44	83.69
Raj	55.61	40.34	89.47	56.61	41.48	89.12	48.59	33.69	87.14
Stuart	48.28	33.90	83.86	54.65	39.33	89.52	44.00	30.2	81.05
Leslie	23.00	13.19	89.84	47.37	33.69	79.75	15.84	8.79	80.00
Bernadette	54.31	39.11	88.84	55.99	41.13	87.68	52.83	38.41	84.59
Amy	59.18	44.27	89.21	54.27	39.17	88.34	56.00	41.38	86.61
Emily	31.09	18.8	89.68	46.49	30.28	100.00	23.4	14.29	64.71
Overall	60.96	46.66	87.90	61.13	46.87	87.87	58.40	43.80	87.59

Table 3. Performance for naive baselines.

	Acc.	**F1**	**Pre.**	**Rec.**
Random	49.98	46.29	43.14	49.93
All positive	43.17	60.30	43.17	100.00
All negative	56.83	0.00	0.00	0.00

Table 4. Experimental results for one modality.

	Acc.	**F1**	**Pre.**	**Rec.**
Punch$_1$ (w/o FT)	65.61	53.07	64.58	45.04
Punch (w/o FT)	67.01	56.03	65.97	48.69
Punch$_1$ (w/ FT)	69.31	62.51	66.13	59.26
Punch (w/ FT)	**70.50**	**64.21**	**67.40**	**61.30**
Story	56.73	0.83	39.39	0.42
Char	56.93	12.28	50.82	6.98

Table 5. Ablation results evaluating the benefit of each modality. The best and second best scores are shown in **bold** and *italic*.

Punch	Story	Char	Att	Ext	**Acc.**	**F1**	**Pre.**	**Rec.**
✓					**70.50**	64.21	67.40	61.30
✓	✓				69.06	58.35	**69.64**	50.21
✓		✓			*70.41*	**65.07**	66.34	63.85
✓	✓	✓			70.01	63.59	66.80	60.69
✓	✓		✓		69.34	60.69	*67.95*	54.83
✓		✓	✓		70.08	63.64	66.93	60.65
✓	✓	✓	✓		70.20	*65.03*	65.90	**64.18**
✓				✓	70.01	64.82	65.65	*64.02*
✓	✓	✓	✓	✓	70.33	63.58	67.61	60.01

slightly drops it. Figure 7 shows the distributions of attention weights over the test split for the Punch + Story + Char + Att model. As expected, the attention weight for punchline modeling is significantly larger than the others. In order to further investigate the behavior of attention weights, we

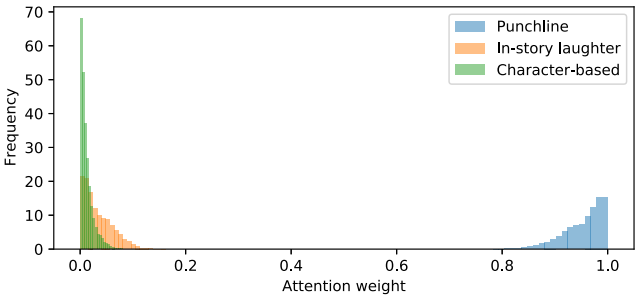

Figure 7. The distributions of attention weights for punchline (blue), in-story laughter (yellow), and character-based (green) modeling over the test split.

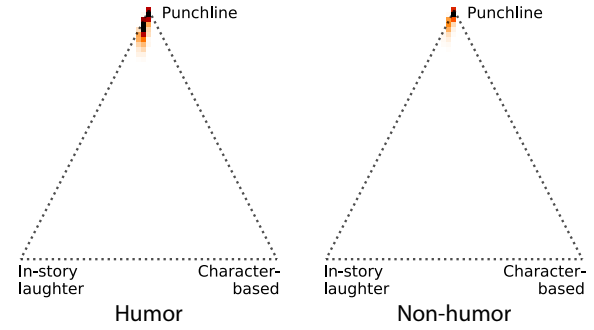

Figure 8. Heat maps of attention distributions for samples in the test split that gave humor (left) and non-humor (right) predictions. As attention weights sum up to 1, they lie in a triangle formed by $(1, 0, 0)$, $(0, 1, 0)$, and $(0, 0, 1)$ in the attention weight space. We converted attention weights into the barycentric coordinates, where the top, left, and right vertices are weight values for punchline, in-story laughter, and character-based modelling.

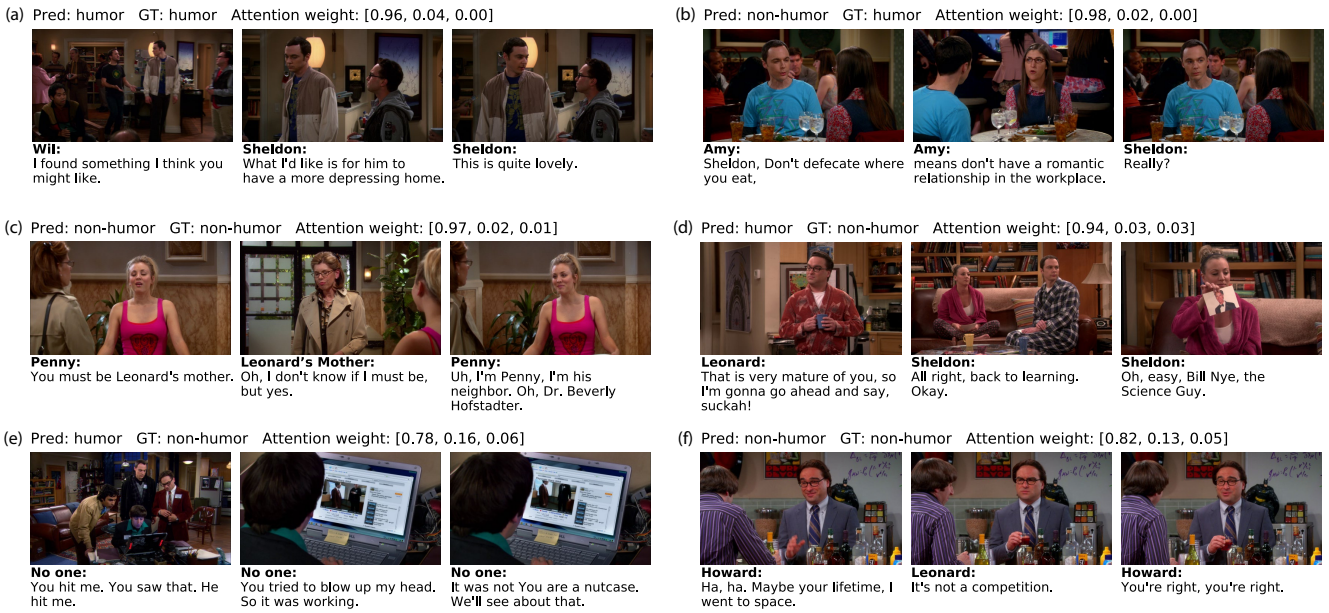

Figure 9. Some examples from our our prediction results. (a)–(c) gave successful prediction, while (d)–(f) did not. The dimensions of the attention weight vector are in the order of punchline, in-story laughter, and character-based modeling.

show the heat maps of attention distributions for humor and non-humor predictions in Figure 8. We can see that both distributions are concentrated around the punchline modeling and there is no significant difference in the distributions, although humor predictions' distribution is a little bit broader. This behavior of attention weights may be because the in-story laughter and character-based modeling do not always give informative cues and thus are overlooked even when they offer some cues. Therefore, the reasons why the modality attention does not work can include (i) the additional complexity introduced by the modality attention leads to overfitting and (ii) the visual modalities are overlooked due to their inconsistent signals.

The third section of Table 5 shows the performance when the character-extended punchline feature was used. The Punch + Char model gave slightly better performance than the Punch + Ext model. This may imply that who are in the scene can be more informative than who speak even though the character-based features are error-prone.

5.2.3 Qualitative Results

Figure 9 shows some examples of prediction results with our full (Punch + Story + Char + Att + Ext) model, where two context subtitles are also shown together with corresponding frames. (a) and (b) are successful and failure samples with the humor label, while (c) and (d) are with the non-humor label. (e) and (f) are humor and non-humor predictions when the visual modalities take the highest weights. Taking (a) as an example, we can see that, given the contextual subtitles, the subtitle "This is quite lovely" is likely to be a punchline. For (e) and (f), we can see that our modality attention does not work since, for example, (e) has no face can be seen in the third subtitle but still the model tries to focus on visual modalities.

6. Conclusion

We created a new dataset based on timestamps for humor prediction for a TV drama series. Different from most previous work, which only considers the verbal modality, our dataset associates humor labels with video's timestamps. This allows us to predict humor induced by non-verbal modalities like facial expressions. We also presented a baseline model for humor prediction, which uses three modalities that can correlate with humor, *i.e.*, subtitles, facial expression, and character names, to predict whether the subtitle involves humor. We adopted the attention mechanism to adjust weights for the three modalities. We found that the characters in the scene slightly increase the humor prediction performance, but the attention mechanism does not help improve the prediction. We plan to incorporate body gestures. From the results, we can conclude that there are still some difficulties in modeling visual modalities for humor prediction. Our current model predicts humor labels over subtitles, but it will be an interesting direction to develop a model to predict silent humor, such as funny costumes or gestures, aside from exploring more efficient ways to take the visual modalities into account.

Acknowledgment. This work was partly supported by JSPS KAKENHI No. 18H03264 and JST ACT-I.

References

[1] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. Openface: an open source facial behavior analysis toolkit. In *Proc. WACV)*, pages 1–10, 2016.

[2] Dario Bertero and Pascale Fung. Deep learning of audio and language features for humor prediction. In *Proc. LREC*, pages 496–501, 2016.

[3] Dario Bertero and Pascale Fung. A long short-term memory framework for predicting humor in dialogues. In *Proc. NAACL-HLT*, pages 130–135, 2016.

[4] Kim Binsted. Using humour to make natural language interfaces more friendly. In *Proc. AI, A-Life and Entertainment Workshop*, 1995.

[5] Kim Binsted and Graeme Ritchie. An implemented model of punning riddles. Technical report, University of Edinburgh, Department of Artificial Intelligence, 1994.

[6] Mondher Bouazizi and Tomoaki Ohtsuki. Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really? In *Proc. GLOBECOM*, pages 1–6, 2015.

[7] Yitao Cai, Huiyu Cai, and Xiaojun Wan. Multi-modal sarcasm detection in twitter with hierarchical fusion model. In *Proc. ACL*, pages 2506–2515, 2019.

[8] Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proc. ACL*, pages 4619–4629, 2019.

[9] Andrew Cattle and Xiaojuan Ma. Recognizing humour using word associations and humour anchor extraction. In *Proc. COLING*, pages 1849–1858, 2018.

[10] CBS, Inc. The Big Bang Theory Site. `https://the-big-bang-theory.com/`.

[11] Lei Chen and Chong Min Lee. Predicting audience's laughter using convolutional neural network. *arXiv preprint arXiv:1702.02584*, 2017.

[12] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. In *Proc. CoNLL*, pages 107–116, 2010.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Rosenberg Ekman. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997.

[15] Ruth Filik, Hartmut Leuthold, Katie Wallington, and Jemma Page. Testing theories of irony processing using eye-tracking and ERPs. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40(3):811, 2014.

[16] Md Kamrul Hasan, Wasifur Rahman, AmirAli Bagher Zadeh, Jianyuan Zhong, Md Iftekhar Tanveer, Louis-Philippe Morency, and Mohammed (Ehsan) Hoque. UR-FUNNY: A multimodal language dataset for understanding humor. In *Proc. EMNLP-IJCNLP*, 2019.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, pages 1735–1780, 1997.

[18] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[19] Chloe Kiddon and Yuriy Brun. That's what she said: Double entendre identification. In *Proc. HLT*, pages 89–94, 2011.

[20] Igor Labutov and Hod Lipson. Humor as circuits in semantic networks. In *Proc. ACL*, pages 150–155, 2012.

[21] Lizhen Liu, Donghai Zhang, and Wei Song. Exploiting syntactic structures for humor recognition. In *Proc. COLING*, pages 1875–1883, 2018.

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[23] Pulkit Mehndiratta, Shelly Sachdeva, and Devpriya Soni. Detection of sarcasm in text data using deep convolutional neural networks. *Scalable Computing: Practice and Experience*, 18(3):219–228, 2017.

[24] John C Meyer. Humor as a double-edged sword: Four functions of humor in communication. *Communication Theory*, 10(3):310–331, 2000.

[25] Rada Mihalcea and Carlo Strapparava. Making computers laugh: Investigations in automatic humor recognition. In *Proc. EMNLP*, pages 531–538, 2005.

[26] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. NeurIPS*, pages 3111–3119, 2013.

[27] Abhijit Mishra, Diptesh Kanojia, and Pushpak Bhattacharyya. Predicting readers' sarcasm understandability by modeling gaze behavior. In *Proc. AAAI*, 2016.

[28] Reynier Ortega-Bueno, Carlos E Muniz-Cuza, José E Medina Pagola, and Paolo Rosso. UO UPV: Deep linguistic humor detection in spanish social media. In *Proc. Workshop on Evaluation of Human Language Technologies for Iberian Languages*, 2018.

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *Proc. NIPS Autodiff Workshop*, 2017.

[30] Saša Petrović and David Matthews. Unsupervised joke generation from big data. In *Proc. ACL*, pages 228–232, 2013.

[31] Tomáš Ptáček, Ivan Habernal, and Jun Hong. Sarcasm detection on czech and english twitter. In *Proc. COLING*, pages 213–223, 2014.

[32] Amruta Purandare and Diane Litman. Humor: Prosody analysis and automatic recognition for f* r* i* e* n* d* s. In *Proc. EMNLP*, pages 208–215, 2006.

[33] Victor Raskin. *Semantic Mechanisms of Humor*. 1985.

[34] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *Proc. EMNLP*, pages 704–714, 2013.

[35] Graeme Ritchie, Ruli Manurung, Helen Pain, Annalu Waller, Rolf Black, and Dave O'Mara. A practical application of computational humour. In *Proc. ICCC*, pages 91–98, 2007.

[36] Rossano Schifanella, Paloma de Juan, Joel Tetreault, and Liangliang Cao. Detecting sarcasm in multimodal social platforms. In *Proc. MM*, pages 1136–1145, 2016.

[37] Jonas Sjöbergh and Kenji Araki. A complete and modestly funny system for generating and performing japanese stand-up comedy. In *Proc. COLING*, pages 111–114, 2008.

[38] Oliviero Stock and Carlo Strapparava. Hahacronym: A computational humor system. In *Proc. ACL*, pages 113–116, 2005.

[39] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proc. CVPR*, pages 4631–4640, 2016.

[40] Julia M Taylor and Lawrence J Mazlack. Computationally recognizing wordplay in jokes. In *Proc. Annual Meeting of the Cognitive Science Society*, 2004.

[41] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[42] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. Humor recognition and humor anchor extraction. In *Proc. EMNLP*, pages 2367–2376, 2015.

[43] Keren Ye, Kyle Buettner, and Adriana Kovashk. Story understanding in video advertisements. In *Proc. BMVC*, 2018.

[44] Kota Yoshida, Munetaka Minoguchi, Kenichiro Wani, Akio Nakamura, and Hirokatsu Kataoka. Neural joking machine: Humorous image captioning. *arXiv preprint arXiv:1805.11850*, 2018.