Multimodal Conversation Modelling for Topic Derailment Detection

Zhenhao Li, Marek Rei, Lucia Specia
Language and Multimodal AI (LAMA) Lab, Imperial College London
{zhenhao.li18, marek.rei, l.specia}@imperial.ac.uk

Abstract

Conversations on social media tend to go off-topic and turn into different and sometimes toxic exchanges. Previous work focuses on analysing textual dialogues that have derailed into toxic content, but the range of derailment types is much broader, including spam or bot content, tangential comments, etc. In addition, existing work disregards conversations that involve visual information (i.e. images or videos), which are prevalent on most platforms. In this paper, we take a broader view of conversation derailment and propose a new challenge: detecting derailment based on the “change of conversation topic”, where the topic is defined by an initial post containing both a text and an image. For that, we (i) create the first Multimodal Conversation Derailment (MCD) dataset, and (ii) introduce a new multimodal conversational architecture (MMConv) that utilises visual and conversational contexts to classify comments for derailment. Experiments show that MMConv substantially outperforms previous text-based approaches to detect conversation derailment, as well as general multimodal classifiers. MMConv is also more robust to textual noise, since it relies on richer contextual information.

1 Introduction

Online conversations on social media can easily go off-topic (Churchill and Bly, 2000; Shepherd et al., 2015) and result in divergent and even harmful exchanges, ranging from off-topic discussions to spam information and personal attacks. Work in this field has focused on detecting or forecasting conversations that derail into toxic content (Zhang et al., 2018a,b; Chang and Danescu-Niculescu-Mizil, 2019; Kementchedjhieva and Søgaard, 2021; Lambert et al., 2022). However, such work is limited in two ways: 1) derailment is not limited to toxic comments or personal attacks; it can include topic shifts, spam posts, among others; 2) conversations in real applications can include modalities other than text, such as image, audio, and video, and the multimodal context can be crucial in deciding whether a conversation has derailed.

In this work, we focus on a broader view on detecting conversation derailment based on the change of conversation topic, making it a more challenging task, and propose the task of multimodal derailment detection. Figure 1 shows an example of multimodal conversation derailment detection. The conversation starts with a title and an image, with a sequence of utterances replying to each other and a target text which is classified to be a derailment or non-derailment.

Figure 1: An example of multimodal conversation derailment detection. The conversation starts with a title and an image, with a sequence of utterances replying to each other and a target text which is classified to be a derailment or non-derailment.

1https://github.com/Nickeilf/Multimodal-Conversation-Derailment
sational (MMConv) architecture which makes use of image and conversational contexts via a hierarchical structure to improve the performance of derailment detection. To simulate real situations on social platforms, we also perform a robustness test by evaluating the model under noisy texts. To summarise, our main contributions are:

- We propose a new task: multimodal conversation derailment detection, and create a new dataset for this challenging task.

- We propose a new hierarchical model architecture – MMConv – that uses visual and conversational contexts. Experiments show that our model outperforms existing multimodal architectures in this task.

- We evaluate the model robustness against noisy texts, showing that images and conversations can provide useful context and improve model robustness.

2 Related Work

Conversation derailment detection Previous work in conversation derailment detection focuses on derailment due to antisocial behaviours. Zhang et al. (2018a) propose the Conversations Gone Awry dataset and introduce the task of forecasting antisocial behaviours in conversations. Chang and Danescu-Niculescu-Mizil (2019) propose the Reddit CMV dataset and a CRAFT model that can potentially provide early warnings regarding abusive content. Kementchedjhieva and Søgaard (2021) further introduce dynamic training into BERT-based (Devlin et al., 2019) derailment forecasting models. Other studies investigate general toxicity in conversations (Almerekhi et al., 2020; Cécillon et al., 2020; Iye et al., 2021; Vidgen et al., 2021; Lambert et al., 2022) regardless of conversation derailment.

In this work, we study derailment detection from a broader angle, based on the change of conversation topics. The work by Park et al. (2021) is the closest to ours. They study conversations violating a set of community norms, such as uncivil behaviour, off-topic content, etc. However, their work only considers textual conversations, whereas we look at multimodal conversations.

Conversation modelling Existing approaches to modelling conversations can be divided into two categories: hierarchical modelling and concatenated modelling. In the first category, each utterance in the conversation is encoded first into an utterance vector, and these utterance vectors are further encoded to generate a conversation vector. Sordoni et al. (2015) propose the HRED architecture, which uses two RNNs as the utterance encoder and context encoder. The utterance and context encoder have also been modelled with GRUs (Yang et al., 2016), LSTMs (Chang et al., 2019), and BERT (Zhang et al., 2019; Pappagari et al., 2019; Li et al., 2020; Gu et al., 2021; Santra et al., 2021). The second category treats the conversation as a concatenation of all utterances, separated by special tokens, and feeds it to Transformer-based (Vaswani et al., 2017) pre-trained language models to generate a vector representing the whole conversation (Lai et al., 2020; Zhong et al., 2021). This approach can be more effective, but requires more computational resources to handle the long inputs. We adopt the first type of approach to model the conversations.

3 MCD Task and Dataset

3.1 Task definition

On popular social platforms such as Reddit, a conversation often starts with an original post, which consists of a segment of text, sometimes with images or a paragraph of text. Users can then either comment under the post or reply to the comments. This means the post has a tree structure, where each branch can be seen as a conversation. Following this natural feature of a conversation, we define the multimodal dialogue derailment task, which has four types of inputs: a title $T$, an image $I$, a sequence of previous utterances (shortened to “utterances” in later paragraphs) $U = [U_1, U_2, ..., U_n]$ where $n$ is the length of the sequence of utterances. The title and the image are posted by the same user and are complementary to convey the topic. Each utterance $U_i$ is the reply to the previous utterance $U_{i-1}$, and the target text $X$ is the reply to the last utterance $U_n$. The utterances sequence can be empty, and in this case the target text is the direct reply to the title/image. This task aims to detect whether the target text has derailed the conversation. The target text is considered as a derailment if:

1) The target text does not directly relate to the initial topic. The initial topic can include the subject of the image, the image itself (its quality, colour, lighting, etc.), and the title itself (the way it is formulated).

2) The target text refers to the content of the
image or title, but starts a different conversation topic on them.

3) The target text is spam (e.g. advertising, unintelligible, etc.).

4) The target text is toxic (e.g. personal attack, insulting, etc.).

Figure 2 shows examples of the four types of derailments.

Figure 2: Example of different types of derailments.

3.2 Data collection

Our data is sourced from Reddit using the Pushshift API\(^2\) (Baumgartner et al., 2020). The data is taken from two subreddits with abundant images: /r/itookapicture and /r/pics. To select conversations with potential off-topic target texts, we focus on conversations based on three criteria:

1) Target text with keywords "off-topic", "off topic".\(^3\)

2) Target text removed by moderators. We first collect target texts with “[removed]” as the content using Reddit API\(^4\), and retrieve the texts before removal using Pushshift.

3) Target text with a Reddit score below -10. The score is calculated with the number of upvotes minus the number of downvotes.

For potential negative examples (on-topic target text), we select the target text with a Reddit score above 10. After collecting the target texts, we use the Reddit API to iteratively fetch the previous utterances in the reply chain, until the title and the image.

\(^2\)https://github.com/pushshift/api
\(^3\)We remove the sentences containing these keywords in the text to avoid potentially biasing the annotators.
\(^4\)https://github.com/praw-dev/praw

3.3 Annotation procedure

The data is annotated by qualified US-based annotators from Amazon Mechanical Turk (MTurk). The worker qualification process can be found in Appendix A. For each conversation, the annotator is asked to label whether the target text is a derailment with respect to the initial topic of the conversation, which is determined by the title and the image. The possible labels include “derailment”, “non-derailment”, and “skip (hard to decide)”. If a derailment is chosen, the annotator is asked to also select the reason why the target text is a derailment. The possible reason labels are: “unrelated to title”, “unrelated to image”, “unrelated to title and image”, “new conversation topic”, “spam comment”, and “toxic comment”.

For quality control, we also include gold data with gold labels in the annotation process. The annotators’ performance on this gold data is monitored during the annotation. If an annotator skips more than 10% conversations or has an accuracy below 60% on the gold data, this annotator will be blocked from further annotations. Each conversation is annotated by three annotators. The annotation interface is shown in Appendix B.

After the annotation is done, we filter the data by removing the examples annotated by blocked workers and examples with “skip” as the majority label. The average annotation accuracy on the gold data is 82%, and the Fleiss Kappa is 0.368. This Kappa score reflects a fair agreement. This is because the task of derailment detection can be very subjective. Therefore, we use the majority labels from the three annotators as the final label.

3.4 Statistics

The final MCD dataset includes 12,653 conversations with 5,269 images. Figure 3 shows the conversation length distribution, where the length includes the number of context utterances plus the target text. When the conversation is short, non-derailment (NON) accounts for 60% of the examples. When the conversation length is longer than three, derailment becomes the main label. This behaviour is expected – the likelihood of a target text going off-topic increases as the conversation expands into more utterances. The longest conversation chain in the MCD dataset contains 56 messages.

We randomly split the dataset into train/dev/test sets with a ratio of 0.8/0.1/0.1 (Table 1).
4 Model

In this section, we introduce the architecture of our proposed MMConv model, which consists of a pre-trained text encoder, a visual encoder, a context encoder, and a late fusion component. Figure 4 gives an overview of the model architecture.

Pre-trained text encoder We use a pre-trained BERT (Devlin et al., 2019) model as our text encoder, but note that any other pre-trained language model can be used. Given an input text sequence \( \omega = [\omega^1, \omega^2, ..., \omega^k] \), BERT encodes the input into a sequence of vectors \( \mathbf{e} = [e^1, e^2, ..., e^k] \). For each utterance \( U_i \) and the target text \( X \), we use the first hidden vector (i.e. the [CLS] token representation) as the sentence representation \( s_{u_j} = e^{\text{CLS}}_{u_j} \) where \( j \in \{1, 2, ..., n\} \) and \( s_X = e^{\text{CLS}}_X \).

For the title, the input is provided as “[CLS] Title [SEP] Text [SEP]”. Instead of using the [CLS] representation, we keep the hidden representations of all tokens in the title input, thus yielding \( \mathbf{e}_t \).

Visual encoder We use an image object detector as our visual encoder. Given an input image \( I \), the visual encoder detects local regions in the image as object proposals and extract \( m \) pooled visual features \( \mathbf{O} = [o^1, o^2, ..., o^m] \) for each object region proposal. An MLP layer with gelu activation function (Hendrycks and Gimpel, 2016) is used to map the image features into the same space as the text, thus generating the visual hidden vectors \( \mathbf{V} = [v^1, v^2, ..., v^m] \):

\[
o^i = \text{Enc}_v(I),
\]

\[
v^i = \text{gelu}(W_v o^i + b_v), \quad \forall i \in \{1, 2, ..., m\}
\]

Given that both title and image form the initial topic of a conversation, we concatenate the title token vectors \( \mathbf{e}_t \) with the visual hidden vectors \( \mathbf{V} \) and fuse them using a self-attention layer (Vaswani et al., 2017) to generate a multimodal title-image hidden vector \( h_{mm} \) (the title [CLS] position is used for the hidden vector):

\[
h_{mm} = \text{SA}([e^{\text{CLS}}_t, e^1_t, ..., e^k_t, v^1, v^2, ..., v^m])
\] (2)

where \( k \) is the length of the title input.

Context encoder To better model the chronological order in the sequence of utterances, we use a 1-layer uni-directional GRU network (Cho et al., 2014) as our context encoder. Given the sentence representations for each utterance \( [s_{u_1}, s_{u_2}, ..., s_{u_n}] \), the context encoder iteratively computes the context hidden state \( h_{u_i} \) up to the \( i \)-th utterance:

\[
h_{u_i} = \text{GRU}(h_{u_{i-1}}, s_{u_i})
\] (3)

where \( i \in \{1, 2, ..., n\} \). Therefore, the last hidden state \( h_{u_n} \) can be viewed as the encoding of the whole sequence of utterances.

Late fusion Finally, a self-attention layer is used to fuse the multimodal title-image hidden vector \( h_{mm} \), the utterances sequence hidden vector \( h_{ua} \), and the target text representation \( s_x \), generating a vector \( h_{conv} \) encoding the whole conversation. The three hidden vectors are averaged after the self-attention layer, then fed to an MLP classifier to make the final prediction:

\[
h_{conv} = \text{avg}(\text{SA}([h_{mm}, h_{ua}, s_x]))
\]

\[
p = \text{sigmoid}(W_c h_{conv} + b_c)
\] (4)

5 Experiments

5.1 Baselines

We compare to a number of text-only, conversational and multimodal baselines:
WordOverlap \((T, X)\)\(^5\) This is a rule-based approach which simply defines a derailment when there are no overlapping contents between the title \(T\) and the text \(X\).

BERT-simple \((T, X)\) (Devlin et al., 2019) This uses a pre-trained BERT for classification. The title \(T\) and the text \(X\) are encoded using BERT separately, and their \([CLS]\) token representations are concatenated and fed to a classifier head.

BERT-concat \((T, U, X)\) This approach makes use of the sequence of utterances by concatenating the title \(T\), utterances \(U\), and the text \(X\) as a sequence, which is then given as input to BERT. The sequences are separated by the \([SEP]\) token. The previous utterances are truncated when the concatenated sequence exceeds the 512 tokens.

CRAFT \((T, U, X)\) (Chang and Danescu-Niculescu-Mizil, 2019) The CRAFT model employs the HRED architecture for modelling conversations. The CRAFT model is first pre-trained on the contextual dialog generation task using Reddit data. When fine-tuned on the derailment task, the decoder is removed and a classifier is attached to make the prediction.

ContextRNN \((T, U, X)\) (Ive et al., 2021) This approach first encodes the title \(T\), each utterance \(U_i\) in the conversation history, and the target text \(X\) separately using BERT. The \([CLS]\) representations of the utterances are summarized into a context vector using a GRU network. Finally, the context vector, title vector, and the text vector are then concatenated to make the final prediction.

ToBERT \((T, U, X)\) (Pappagari et al., 2019) This approach first encodes title \(T\), utterances \(U\), and the text \(X\) separately, but the context vector is modelled with an additional Transformer encoder block (Vaswani et al., 2017) with a positional embedding, taking the hidden representations of the utterances as input.

BERT-ViT \((T, I, X)\) (Dosovitskiy et al., 2021) This approach encodes the concatenation of title \(T\) and text \(X\) (separated by \([SEP]\) token) with BERT to get the \([CLS]\) representations. The image \(I\) is encoded with the ViT model and is represented as a pooled visual feature vector. The joint title-text vector and the visual vector are concatenated to make the final prediction.

MMBT \((T, I, X)\) (Kiela et al., 2019) Given that the title and image constitute the initial topic of the conversation, this approach early fuses title \(T\) and image \(I\) using BERT. The visual feature is extracted using ResNet-152 (He et al., 2016) and is concatenated with the joint title-text embedding (separated by \([SEP]\) token) before the BERT layers.

---

\(^5\) The symbols denote which types of inputs the model receives.
ViLT ($T, I, X$) (Kim et al., 2021) The ViLT model is pre-trained on multiple vision-and-language tasks for better multimodal representation. The title $T$ and the image $I$ are encoded using the pre-trained ViLT model to get a fused vector, which is then concatenated with the text vector extracted from BERT for the prediction.6

5.2 Pre-processing and training
We lowercase all texts, replacing emojis and emoticons with corresponding word phrases (e.g. “=D” is replaced with “<happy>”) using the ekphrasis toolkit (Baziotis et al., 2017). Due to computational constraints, we only use the first and last 5 previous utterances if the length of utterances sequence is above 10. We extract the “bottom-up-top-down” (BUTD) (Anderson et al., 2018) object features for the input images with a pre-trained Faster R-CNN ResNet-101 (Ren et al., 2015) object detector. The visual features are represented with 36 object vectors $O \in \mathbb{R}^{36 \times 2048}$.

We use the pre-trained BERT model (Devlin et al., 2019) (bert-base-uncased) as our text encoder. The model hidden size is 768, and the dropout rate (Srivastava et al., 2014) is set to 0.2. We use binary cross-entropy loss as our training loss and Adam (Kingma and Ba, 2015) as optimiser with an initial learning rate of 3e-5. The training batch size is 64. We train the model for 15 epochs, with early stopping when the model does not improve on the development set for 5 epochs. The best checkpoint on the development set with highest macro-F1 score is selected for evaluation on the test set. All models are trained with 5 random seeds and the average results are reported. The experiments are done on an RTX A6000 GPU.

6 Results
6.1 Benchmark
We first compare the performance of the baselines and our MMConv model on the test split. The results are shown in Table 2, where we report the accuracy, precision, recall, F1, macro-weighted F1 (F1-mw), and the area-under-curve (AUC) score.

The results for text-only models (taking only title/utterances/text as inputs) are presented in the second and third rows. The rule-based baseline WordOverlap only achieves a performance of 0.34 F1 score, which is even lower than random guessing (0.44 F1), suggesting that the task of detecting derailment is challenging and more advanced methods are needed. The BERT-simple baseline outperforms the rule-based methods by a large margin.

Comparing the conversational models (taking title/utterances/text as inputs) with the text-only models, we find that the context utterances needs to be handled separately. By simply concatenating all textual inputs into a whole sequence, the BERT-concat model shows a slight performance drop when utilising the previous utterances, compared to the BERT-simple baseline. The drop might be caused by the long sequence length resulting from concatenation, thus the model might have fewer attention weights for title and text tokens as there are too many utterance tokens for attention. Models that encode each utterance separately – CRAFT, ContextRNN, and TbBERT – all improve over the BERT-simple baseline, which shows that utilizing previous context is beneficial for this task.

It is also noticeable that CRAFT, although much smaller in model size, performs on par with BERT-based models. Gururangan et al. (2020) show that pre-training on in-domain data and a related task could result in significant benefits, therefore we hypothesize that the improvement is from the pre-training on Reddit data (same domain) and the dialogue generation task (contextual task). To test that, we train CRAFT from scratch without loading the pre-trained checkpoint from Chang and Danescu-Niculescu-Mizil (2019). The performance drops by 0.03 F1 score compared to the pre-trained CRAFT, which confirms the benefits of in-domain pre-training.

Regarding the multimodal models (taking title/image/text as inputs), both BERT-ViT and MMBT show better performance than the BERT-simple baseline from using the visual information. However, the performance of ViLT is similar to that of BERT-simple, with a lower F1 score, but higher F1-mw score. Limited by the max length of 40 tokens for text inputs, ViLT seems to lose information when encoding the title, thus causing the performance to be worse than other multimodal counterparts.

Finally, MMConv, which takes all inputs, outperforms all other approaches, improving over BERT-simple by around 0.03 F1. Inspired by Bouritsas et al. (2018), we also train a version of MMConv by replacing the visual object features with the se-
Table 2: Comparison of baselines and MMConv on the test split. *-VC: model using visual concepts instead of visual features. visual: whether the model uses images as inputs. conv: whether the model uses previous utterances as inputs.

<table>
<thead>
<tr>
<th>Models</th>
<th>visual</th>
<th>conv</th>
<th>Acc</th>
<th>Pre</th>
<th>Rec</th>
<th>F1</th>
<th>F1-mw</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>.484</td>
<td>.402</td>
<td>.478</td>
<td>.437</td>
<td>.487</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WordOverlap</td>
<td>.455</td>
<td>.346</td>
<td>.338</td>
<td>.342</td>
<td>.452</td>
<td>N/A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT-simple</td>
<td>.731</td>
<td>.668</td>
<td>.709</td>
<td>.688</td>
<td>.731</td>
<td>.810</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT-concat</td>
<td>✓</td>
<td></td>
<td>.727</td>
<td>.673</td>
<td>.679</td>
<td>.676</td>
<td>.727</td>
<td>.803</td>
</tr>
<tr>
<td>CRAFT</td>
<td>✓</td>
<td></td>
<td>.744</td>
<td>.705</td>
<td>.674</td>
<td>.689</td>
<td>.743</td>
<td>.812</td>
</tr>
<tr>
<td>CRAFT w/o pre-train</td>
<td>✓</td>
<td></td>
<td>.714</td>
<td>.661</td>
<td>.657</td>
<td>.659</td>
<td>.714</td>
<td>.788</td>
</tr>
<tr>
<td>ContextRNN</td>
<td>✓</td>
<td></td>
<td>.747</td>
<td>.685</td>
<td>.730</td>
<td>.707</td>
<td>.747</td>
<td>.827</td>
</tr>
<tr>
<td>ToBERT</td>
<td>✓</td>
<td></td>
<td>.740</td>
<td>.683</td>
<td>.709</td>
<td>.696</td>
<td>.741</td>
<td>.820</td>
</tr>
<tr>
<td>BERT-ViT</td>
<td>✓</td>
<td></td>
<td>.750</td>
<td>.705</td>
<td>.697</td>
<td>.701</td>
<td>.750</td>
<td>.837</td>
</tr>
<tr>
<td>MMBT</td>
<td>✓</td>
<td></td>
<td>.749</td>
<td>.705</td>
<td>.689</td>
<td>.697</td>
<td>.748</td>
<td>.824</td>
</tr>
<tr>
<td>ViLT</td>
<td>✓</td>
<td></td>
<td>.735</td>
<td>.684</td>
<td>.676</td>
<td>.680</td>
<td>.733</td>
<td>.817</td>
</tr>
<tr>
<td>MMConv</td>
<td>✓ ✓</td>
<td></td>
<td>.756</td>
<td>.704</td>
<td>.724</td>
<td>.714</td>
<td>.756</td>
<td>.840</td>
</tr>
<tr>
<td>MMConv-VC</td>
<td>✓ ✓</td>
<td></td>
<td>.753</td>
<td>.699</td>
<td>.721</td>
<td>.710</td>
<td>.753</td>
<td>.841</td>
</tr>
</tbody>
</table>

Table 3: Ablation study when only one model component is changed each time.

<table>
<thead>
<tr>
<th>Models</th>
<th>Pre</th>
<th>Rec</th>
<th>F1</th>
<th>F1-mw</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMConv</td>
<td>.704</td>
<td>.724</td>
<td>.714</td>
<td>.756</td>
</tr>
<tr>
<td>w/o SA fusion</td>
<td>.704</td>
<td>.716</td>
<td>.710</td>
<td>.755</td>
</tr>
<tr>
<td>w/o context enc</td>
<td>.700</td>
<td>.690</td>
<td>.695</td>
<td>.746</td>
</tr>
<tr>
<td>w/o visual enc</td>
<td>.725</td>
<td>.662</td>
<td>.692</td>
<td>.751</td>
</tr>
<tr>
<td>w/ incongruent visual</td>
<td>.689</td>
<td>.720</td>
<td>.704</td>
<td>.747</td>
</tr>
</tbody>
</table>

6.2 Ablation study

To understand which components contribute the most to MMConv, we perform an ablation study by changing one model component at a time (Table 3). We first change the self-attention fusion, where the hidden vectors of title/image, context, and main text are fused, with a concatenation of the three vectors (w/o SA fusion). This change results in a small drop in F1 while also decreasing the recall. Removing the context/visual encoder (w/o context enc and w/o visual enc) causes a larger performance drop in F1, confirming the benefit of using image and previous utterances as additional context. In addition, using both context and visual encoders results in much higher recall than only using one. This indicates that the two sources of information are complementary and both provide useful information. Finally, to further analyse the effect of visual information, we perform an incongruent inference (Elliott, 2018), where we give as input visual features from a randomly selected image. Using incongruent visual features decreases both precision and recall, validating that the model indeed uses information from the provided image.

6.3 Forecasting future derailment

Zhang et al. (2018a) and Chang and Danescu-Niculescu-Mizil (2019) propose forecasting conversation derailment as a more challenging task for detecting online dialogue derailment. The task is to test if the model can predict a derailment given only historical conversation utterances. In other words, the model has no access to the actual target text, and has to predict derailment based on the sequence of conversation utterances. Following their settings, we evaluate the conversational models on the derailment examples in the test set and calculate the mean-H (average forecast horizon), which represents how many utterances ahead the model can signal a future derailment (Table 4). All the four conversational models, although not trained specifically for forecasting derailment, can achieve a mean-H above 3 utterances, i.e. the models are capable of early warning potential derailments. Taking images as additional inputs, MMConv exhibits the largest mean-H (the mean-H distribution for MMConv model is shown in Appendix D). CRAFT outperforms ContextRNN and ToBERT in forecast-
7 Robustness Analysis

Since the main application of dialogue derailment detection is automatic moderation on social platforms, where the user-generated texts can be noisy, it is important to evaluate the model’s robustness to textual noise, e.g. typos, code-switching, etc. Li et al. (2021) evaluate on test data injected with noise and show that multimodal models are more robust to textual noise than unimodal models. Following their settings, we select 7 types of common textual noise (details in Appendix E) from NL-Augmenter (Dhole et al., 2021) and apply noise injection to the test set. We evaluate under two settings: noise injected to both title and target text, and a more realistic setting where noise is injected to all textual inputs, including title, previous utterances, and the target text. The models’ F1 scores on different proportion of noise are shown in Figure 5.

When noise is only injected to title and text, all models show robustness against a small proportion (less than 5%). However, when the noise percentage reaches 15%, all models exhibit larger performance drops while the performance of MMConv is more stable and clearly shows less performance drop over the others, which indicates that the benefits from using image and context are consistent and can help robustness.

When noise is injected to all textual inputs, similar performance drops is observed for all conversational models. Comparing the performance with and without noise injected to previous utterances, we notice that the difference within each model type is small, showing that the models are robust to the noise in previous utterances. Note that the four conversational models use a vector to encode the whole sequence of utterances rather than each utterance separately, so they are less affected by noise in individual utterances.

8 Conclusions

In this paper, we explore the task of detecting online conversation derailment. We take a broader view of derailment and focus on conversation derailment based on topic changes. We create a new multimodal Conversation Derailment (MCD) dataset, which introduces multimodality into conversation derailment detection. Furthermore, we propose a novel hierarchical architecture – MMConv – that uses visual and conversational contexts. Our experiments show that the proposed model outperforms strong baselines in conversation derailment detection. Finally, we perform a robustness analysis by evaluating models with noisy texts. We find that models using visual and conversational information have increased robustness, which can help build more accurate practical moderation applications. Future work in this direction could involve extending the size of the MCD dataset to include more topics in texts/images, building pre-
trained multimodal conversational models for the multimodal derailment detection task, and applying distillation to MMConv model to reduce model size.

9 Limitations

The main limitation of this work is the requirement of large GPU resources. Note that the proposed dataset is sourced from Reddit where user-generated texts can be very long, the model requires larger GPU memory to process the long inputs. In the MCD dataset, the longest target text contains 1,678 words and the longest conversation contains 56 previous utterances. To fit the long inputs into the GPU, we have to truncate the previous utterances and the target text, which might cause information to be lost. In addition, in the proposed MMConv model, pre-trained vision models such as ViT, are not used because of the large cost of GPU resources compared to using pre-extracted object features. We also use gradient accumulation to perform large-batch training. These trade-offs to fit the model in smaller GPU usage limit the model performance. In addition, another potential limitation is that the annotation agreement between crowd-workers is not perfect. Finally, a potential limitation of the data could be the size/diversity of the dataset, where the texts might represent a specific domain. A possible extension would be to increase the number of subreddits to improve the diversity of the covered topics.

10 Ethical Considerations

In this paper, we study the task of detecting online conversation derailment. We create a new multimodal Conversation Derailment dataset. Our data is collected from Reddit using the PushShift API and Reddit API. The data is annotated by crowd-workers from Amazon Mechanical Turk. Before the annotation, we collect informed consent that the annotation will be used for research in conversations derailment. Given that the data contains a small proportion of toxic language, we require the annotators to pass the adult qualification and state clearly in the annotation interface that the workers can exit at any point, i.e. there is no minimal number of samples to be completed for payment. No personal information of the workers is collected, and the dataset does not include annotators’ IDs.

We propose a model architecture and perform experiments on conversation derailment detection. The technology can be potentially used in online social platforms as an auto-moderator tool. Therefore, human moderators could benefit from the technology by spending less time on moderation. In failure cases, false negative might cause failed moderation, thus allowing unwanted language to remain in the platform. We are aware that the current size of data and model might make them potentially biased to certain types of images and topics. Expanding the size of the dataset might help mitigate such biases.

References


Tianda Li, Jia-Chen Gu, Xiaodan Zhu, Quan Liu, Zhen-Hua Ling, Zhiming Su, and Si Wei. 2020. Dialbert: A hierarchical pre-trained model for conversation disentanglement.


A Annotator Qualification Process

Workers must pass the all of the following qualifications before they are allowed to annotate the data:

- The worker must be an adult since the data contains potential offensiveness.
- The worker’s number of approved annotations must be above 2,000 on AMT platform.
- The worker’s approval rate must be above 95% on AMT platform.
- The worker must pass a qualification test of 10 random conversations from the gold data and achieve an accuracy above 80%.

The worker is granted access to be the qualified worker. Note that we also perform quality control during the annotation and block annotators with poor annotations. Once a worker is blocked, the work is no longer able to annotate more data.

B Annotation Interface

An example of the annotation interface is shown in Figure 6.

C MMConv with different pre-trained encoders

In Table 5, the performance of MMConv model with different pre-trained encoders is presented. We notice that BERT performs better than the other two pre-trained language models, i.e. RoBERTa and XLNet. Such improvement might be introduced by the next-sentence-prediction objective used in BERT pre-training, which might help conversational coherence modelling. In terms of visual encoders, BUTD features outperform ViT by a small margin.

<table>
<thead>
<tr>
<th>Encoder</th>
<th>BUTD</th>
<th>ViT</th>
</tr>
</thead>
<tbody>
<tr>
<td>BERT</td>
<td>.756</td>
<td>.704</td>
</tr>
<tr>
<td>BERT</td>
<td>.753</td>
<td>.703</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>.729</td>
<td>.671</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>.715</td>
<td>.658</td>
</tr>
<tr>
<td>XLNet</td>
<td>.729</td>
<td>.653</td>
</tr>
<tr>
<td>XLNet</td>
<td>.725</td>
<td>.672</td>
</tr>
</tbody>
</table>

Table 5: Performance of MMConv model with different pre-trained text and visual encoders.

D Forecast Horizon Distribution for MMConv

Figure 7 shows the distribution of forecast horizon for MMConv model in predicting future comments. The model does not predict future derailment on 25% of true positive examples, but could give early signal on future derailments within 4 utterances for over 40% of the cases. Finally, the model is also capable of handling long conversations with over 13 utterances and predict future derailment at an early stage.

E Noise Injection Strategies

We used seven common noise injection strategies from NL-Augmenter:

- abbreviation transformation: words are randomly replaced with their abbreviations if the abbreviations exist.
- butter fingers perturbation: characters are randomly replaced with adjacent characters on a standard QWERTY keyboard.
- close homophones swap: words are randomly swapped to another word that look or sound similarly with the original word.
- multilingual dictionary based code switch: words are randomly replaced with its translation in another language.
- visual attack letters: characters are randomly substituted with another character (might from a different encoding) that looks similar.
- whitespace perturbation: whitespaces are randomly inserted or deleted in the text.

F Robustness to attack

The other aspect of robustness is on adversarial attacks. Although the model shows robustness on
noisy texts, users might escape the auto-moderation by iteratively paraphrasing the target text until the model misclassifies, and such a process can be seen as an adversarial attack. To simulate this, we use TextAttack (Morris et al., 2020) to perform adversarial attacks to the models. We randomly sample 200 examples from the test set and attack the model with textfooler attack (Jin et al., 2020). We calculate the successful attack rate and the average number of required queries to measure the model’s robustness. If a model is robust, the attack method would have difficulties finding an adversarial example, thus resulting in lower successful attack rate and higher query number. The model performance is presented in Table 6.

The BERT-simple baseline exhibits the least robustness to attack, with a successful attack rate of 92.8% and average query number of 137. Other conversational models, though not optimized with adversarial training, all show obvious improvements over the text-only baseline, indicating that the additional conversational context could increase model robustness. Among all models, ContextRNN shows the best robustness, decreasing the successful attack rate significantly to 75.2%, and increasing the average query number to 196.

Table 6: Results of successful attack rate and average query number with textfooler attack (Jin et al., 2020).