Understanding Effects of Events on News and Public Opinion

Akshay Sasikumar, Gayatri Belapurkar, Sagar Palao, Saloni Chalkapurkar*
{asasikumar, gbelapurkar, spalao, schalkapurka}@umass.edu

1 Introduction

The media and news organizations play a large role in how the public perceives events and entities through their reporting. Even if the expectation for the news is to be unbiased and present facts, in reality, the opinions of audiences could and are often influenced and formed through the reporting of the news. A few ways bias is presented in the news is through focus on different sorts of content, or through subtle word changes which change the polarity of the text. Through their coverage, news media agencies can have far-reaching effects on the public and society. Understanding the sentiment and emotion in media coverage and how these change with time, is a vital step for society to take more informed decisions. Much Natural Language work has been done on studying media bias in a political setting, such as using RNNs (Iyyer et al., 2014) and Headline Attention (Gangula et al., 2019), but in this project, we will extend this work to study opinions and biases (if any) for multiple entities, including political and non-political ones. We look to using data of multiple events of an entity and their public reception, so as to be able to visualize such sentiments and study how the perception of the media and the public to an entity changes as time progresses and with various related happenings or events.

Quantitatively describing media reporting and its effects on public opinion is not a well defined task. Hence, in this project we first want to understand the sentiments and emotions portrayed in articles and blogs released by a set of media houses. We then will further take into account how the article has affected the sentiments or perception of the audiences they reached (for that specific entity). Finally, we look to understanding how the views of the media house and the public evolves over time, and to include the effect of previous related major events.

We formulated a formal task to understand these interactions. The task is to predict the sentiment and emotions of the news from different media houses and their readers' comments on twitter for the next event of an entity. To predict this we take into account (a) reporting by the media house and comments on their reporting for last k events for that entity and (b) the next event's description (not the news or comments). We aim at generalizing this across entities. During inference the model produces prediction for events of an entity it has not seen during training. This task helps us understand the media and public interactions and their response for future events for different entities. Figure 1 shows a visualization of the output generated by our model. It shows the sequence of events for an entity: Novak Djokovic, the ground truth sentiment and emotions of different media house news and their readers' comments, and our model's prediction of these sentiments and emotions.

2 What we proposed, changes, and what we accomplished

- Collect and annotate data
- ~~Transfer Learning with RoBERTa + RNN~~ → Based on feedback from the instructors, we changed direction and decided to use T5 and treat it as a text generation problem to perform multilabel classification.
- ~~Summarization of tweet comments~~ → Based on feedback from the instructors, we selected the most liked comments instead of summarizing tweet comments.
- Build and train baselines on collected dataset and examine their performance

*In alphabetical order

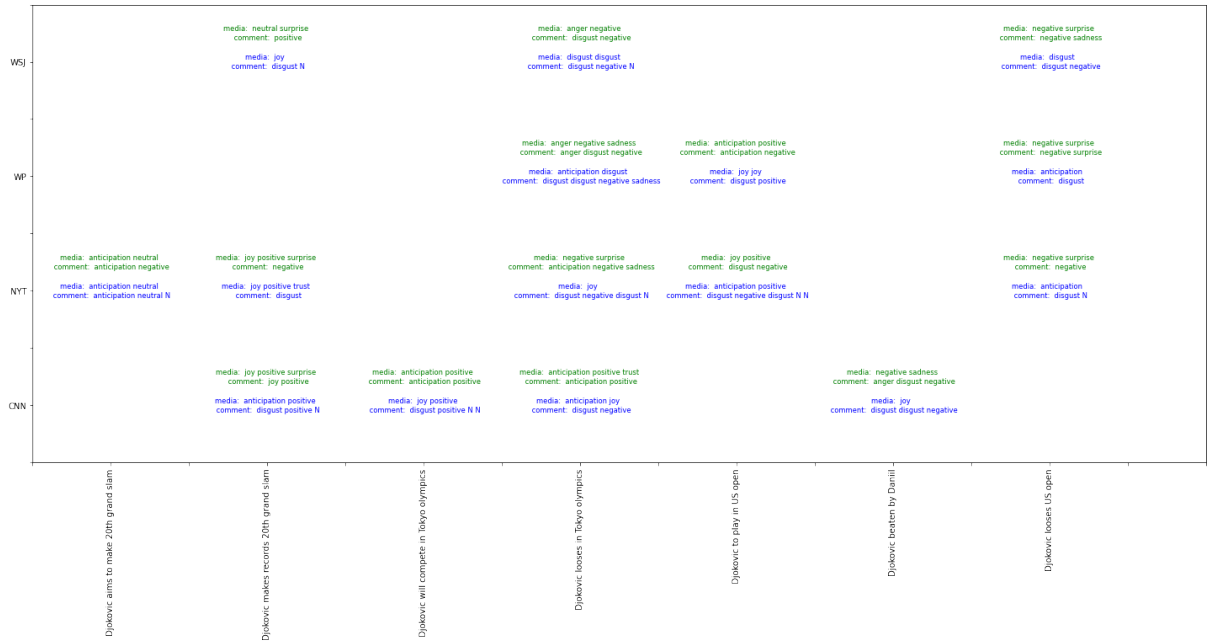

Figure 1: The figure above shows the sequence of events for an entity: Novak Djokovic over time. It shows the true sentiment & emotions (in green) and the predicted sentiment & emotions (in blue) for different media houses and their readers' comments.

- Training using transformer and examine its performance
- Training using transformer and intermediate training on emotion and sentiment dataset and examine its performance
- + Training using transformer, intermediate training, and generated article summary and examine its performance
- + Training using transformer, intermediate training, and data augmentation using back translation and examine its performance
- Visualization
- Error Analysis

3 Related work

Over the years, there has been a lot of work done in the fields of sentiment analysis and opinion mining. Chen et al. (2016) uses an LSTM model, followed by attentions over different semantic levels to get the overall sentiment in a document about a product. Neural Networks using multiple-attention mechanisms (Chen et al., 2017) have been used to have memory of different opinion targets of a sentence. Research has been done to show that different word embeddings can cause contrasting results and affect model performance, especially when using BERT (Han and Kando, 2019). Feng et al. (2021) proposes Target-Specified Sequence Labeling with multi-head self attention for Target oriented Opinion Words Extraction (TOWE), which experimentally outperformed benchmark methods.

Raffel et al. (2020) introduced a unified framework that converts all text-based language problems into a text-to-text format and achieved state-of-the-art results on many benchmarks including text classification. We combine the benefits of all the previous work done in this field and use T5 as our pretrained model for transfer learning. The model will take previous interactions into account and will predict the sentiment and emotions for the next event of an entity. The sentiment and emotions will be generated by the T5 model.

To improve performance on low-resource task, Phang et al. (2019) proposed the paradigm of intermediate task fine-tuning: first, fine-tune model on an data-rich supervised tasks, and then fine-tune the resulting model on the target task. However, the conditions for successful intermediate-task fine-tuning (i.e., which tasks make good intermediate tasks) remain unclear. Pruksachatkun et al. (2020) observe that intermediate tasks that require high-level inference and reasoning abilities tend to work best, while Vu et al. (2020) in-

dicate that the similarity between the intermediate task and the target task is crucial for successful intermediate task fine-tuning. We used two data-rich intermediate tasks: to predict emotions and to predict sentiments in tweets and studied the effects of these intermediate tasks on our predictions.

Data Augmentation is another popularly used strategy for various image tasks to make the model more robust and increase its ability to generalize. Edunov et al. (2018) investigates different data augmentation strategies, particularly back translation, which can be applied to textual data to help with low-resource tasks. Back translation has been used widely in Natural Machine Translation tasks to generate more parallel data (Sennrich et al., 2016). However, we plan to use it as a tool to generate more augmented data for our task.

Extractive and abstractive, both the methods of summarization have been used for summarising opinions and sentiments. Angelidis and Lapata (2018) uses weakly abstractive aspect extraction and sentiment prediction to form a summary of online product reviews. Opinion Digest (Suhara et al., 2020) uses an Aspect Based Sentiment Analysis model to extract opinion phrases from reviews. The most popular ones are then given as input to train a Transformer model to reconstruct reviews from these extractions. Bražinskas et al. (2020) uses a generative mechanism in an unsupervised setting to create fluent and coherent summaries reflecting common opinions expressed in reviews. Since news articles are very long and combining past news articles to predict sentiments and emotions for next events would lead to a very long input which is not feasible to be consumed by T5, we use an abstractive summarization model to summarize the articles before passing it to our model.

4 Dataset

We ran all the experiments on a dataset that we created for our task by collecting the headline, article and top comments for chronologically sequenced news events for an entity. A snapshot of the dataset can be seen in table 1.

Our task is to predict the news media agencies' emotions and its readers' emotions for the next event. But we are not looking at the headline, article body or comment for the next event. We rely on the past events' headlines, article bodies, comments and the next event's description to make the prediction for the next event. This makes the task challenging. The model has to learn the sentiment and emotion for historical events (which can be different for different events in the history) and to compose this information to make prediction for the next event.

4.1 Data Source and Collection

We collected the data from Twitter. For each entity we collected articles from about 4-5 news media agencies. For each media agency, we looked at their Twitter posts about the given entity within a time frame. This was done using the twitter search query: "[*entity*] (from:[*media house twitter handle*]) until:[*end date*] since:[*start date*]". The headline of the article was added to the 'Headline' column of our dataset as is without any change. The link to the article was added to the 'Article Link' column of our dataset. We wrote a simple python script for retrieving the entire content of the article from its link. We collected 1-2 comments for each entry in our dataset. These are from the comments on the Twitter post of the respective news article. We selected the most popular comments i.e. the comments that had the highest number of likes. For posts that did not have any clearly popular comments, we selected comments that best represented the majority of the comments in the comments section. The comments are in the 'Comment 1' and 'Comment 2' columns of our dataset.

For example: For the entity 'Djokovic', we looked at events covered between March and October 2021 for all media agencies. We thus have events (news articles) for 'Djokovic' from four media agencies - CNN, NYT, WP, WSJ. For each of these media agencies, we ordered the sequence of events in the 'Sequence Number' column of our dataset based on the date the article was published. Table 1 shows a sample of first event for three such media agencies for this entity.

The data in columns 'News Sentiment and Emotion', 'Readers' Sentiment and Emotion' and 'Event' was not collected but annotated by us.

4.2 Data annotation

The content of the columns in 'News Sentiment and Emotion', 'Readers' Sentiment and Emotion' and 'Event' was generated by the annotators while annotating. For the sentiment labels, we chose one sentiment from the three possible labels 'Positive', 'Negative' and 'Neutral'. For

Media	Entity	#	Event	Headline	Article Body	C1	C2	NSE	RSE
CNN	Djokovic	1	Djokovic beats Nadal	Novak Djokovic beats Rafael Nadal ...	Novak Djokovic dethroned Rafael ...	Awesome	The King of Tennis	Positive Joy	Positive Joy
NYT	Djokovic	1	Djokovic beats Nadal	French Open 2021: Djokovic Beats ...	Novak Djokovic, the world No. 1,...	I can't be more happy ...	The seeding was ...	Positive Anticipation Surprise	Positive Negative Surprise
WP	Djokovic	1	Djokovic beats Nadal	Novak Djokovic outlasts Rafael ...	Even though he was getting ...	Shit	GREAT #####	Positive Anticipation	Negative Disgust

Table 1: Examples from dataset. NSE: News Sentiment and Emotion, RSE: Readers' Sentiment and Emotion, #: Sequence Number, C1: Comment 1; C2: Comment 2

the emotion label, we chose zero, one or more labels from the 8 possible labels 'Joy', 'Trust', 'Fear','Surprise','Sadness','Anger','Disgust' and 'Anticipation'. These 8 emotions were taken based on Plutchik's Wheel of Emotions. For 'Events', the annotators were asked to describe the event without any bias or expression. We distributed the task of data annotation among ourselves. We had 2 annotators annotating one part of our data and 2 annotators annotating the remaining part of it.

From the pilot annotation experiment, we found that a major issue arose while annotating the sentiment and emotion of the comments. Since we chose the comments based on popularity, we did not control what the comments were addressing. While for certain articles the comments were addressing the event or the entity covered in the news article, for others the comments addressed the news agency itself. Hence, there was confusion with assigning labels to these because in most cases a comment showing 'negative' sentiment and 'disgust' emotion towards the news agency did not show 'negative' sentiment or 'disgust' emotion towards the event or entity. Therefore, we limited our annotation to the sentiment of just the content of the comments and did not consider who or what the comments were addressing.

The inter-annotator agreement score (Cohen's Kappa scores) for different labels were: Readers' Sentiment and Emotion: 0.79; News Sentiment and Emotion: 0.63; Event Description: 0.4. For sentiment and emotions of news, we observed that differences arose due to overlapping sentiments such as 'disgust' and 'anger'. Event description had low score because even though the descriptions were good, they were paraphrases of each other, hence lower score.

We resolved the inter annotator disagreements by having the annotators discuss and arrive at a final label.

4.3 Data preprocessing

We preprocessed our labels to lower case. In sentence case, due to subword tokenization, the tokenizer was splitting words into subwords. During prediction, we observed some words were mixture of different subwords, like 'Antitral' which is a combination of two subwords: 'Anti' from 'Anticipation' and 'tral' from 'Neutral'. To prevent this, we found that converting the labels to lowercase was turning every word into a separate token.

4.4 Statistics

4.4.1 Our data

We collected a total of 692 events (Train:Val:Test = 428:121:143) for 25 entities (Train:Val:Test = 17:4:4). Table 2 shows the distribution of the labels (sentiment and emotions for news and com-

ments) for train, validation, and test set. Table 3 shows how many entities were collected for each media agency. Figure 3 shows the distribution of length of characters for the headlines and its readers' comments. Figure 2 shows the numbers of events per entity from a single media house.

Label	News			Comments		
	T	V	Te	T	V	Te
Sadness	32	5	15	51	7	11
Anticipation	151	43	40	47	10	17
Joy	63	4	11	27	3	20
Fear	99	30	35	39	5	8
Surprise	65	5	14	15	6	7
Disgust	63	36	5	194	48	50
Trust	71	13	10	23	11	7
Anger	39	17	15	52	23	41
Positive	152	26	30	74	28	35
Negative	183	57	74	256	69	79
Neutral	93	38	40	102	27	42

Table 2: No. of labels. T: Train; V: Val; Te: Test

Media	T	V	Te
CNN	17	4	4
NYT	15	4	4
Reuters	14	2	11
WP	14	3	3
WSJ	4	2	3
AP	1	0	0

Table 3: No. of entities per media agency. T: Train; V: Val; Te: Test

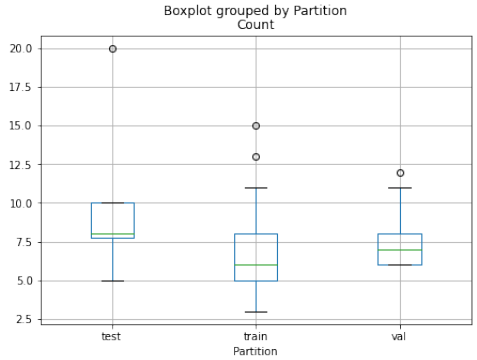

Figure 2: No. of events for an entity by a media agency

4.4.2 Emotion Dataset

In our experiments we used the emotion dataset (Saravia et al., 2018) for intermediate fine-tuning.

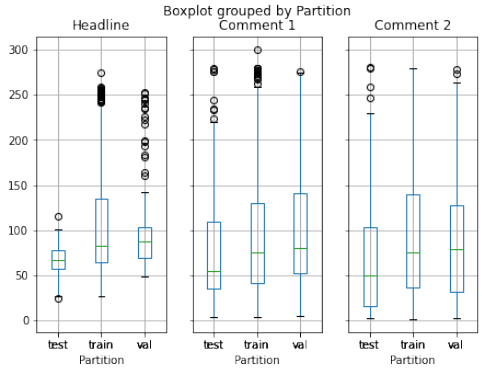

Figure 3: Length of characters of news headline and its readers' comments

Emotion	T	V	Te
Sadness	4666	550	581
Joy	5362	704	695
Love (Joy + Trust)	1304	178	159
Anger	2159	275	275
Fear	1937	212	224
Surprise	572	81	66

Table 4: Statistics of labels for emotion dataset. T: Train; V: Val; Te: Test

It represents six emotions as shown in table 4.

4.4.3 Sentiment Dataset

In our experiments we used the sentiment dataset (Barbieri et al., 2020) for intermediate fine-tuning. It represents three sentiments as shown in table 5.

Sentiment	T	V	Te
Negative	7093	312	3972
Neutral	20673	869	5937
Positive	17849	819	2375

Table 5: Statistics of labels for sentiment dataset. T: Train; V: Val; Te: Test

5 Task definition

We formally define our task as predicting the sentiment and emotions of the news media and its readers' comments for the next event of an entity without looking at the next news article or readers' comments. An entity can be considered to be similar to a theme for the events. For example, all the events and news related to 'Novak Djokovic' would come under an entity of the same name. To assist with this task, we provide the model past news headlines/articles, their readers' reactions,

and the next event's unbiased description; and ask it to predict the sentiment and emotion for the next event. It can also be thought of as forecasting the media's and their readers' opinions when the described event happens.

5.1 Input

The format for the input is,
Media: [*Media house name*] [*History*] E: [*Next event's description*]
where,

1. *Media house name* is the label of the media house, like NYT, WP, WSJ, Reuters, etc.

2. *Next event's description* is the description of the next event for which we want to predict media house's and its readers' sentiments and emotion.

3. *History* refers to the past events (in chronological order) headlines/articles by the media house and its readers' comments on Twitter for that entity. If the next event is at timestep t, it is formatted as,
   N1: [news headline/article for event $t-k$] C1: [tweet comment for event $t-k$] N2: [news headline/article for event $t-k+1$] C2: [tweet comment for event $t-k+1$] ... NK: [news headline/article for event $t-1$] CK: [tweet comment for event $t-1$]
   where, K is a hyperparameter.

The entity name is never provided as input as we want our model to generalize across entities.

5.2 Output

The output is the sentiment and emotion for the next event. We are building it as a generation task. So the format for the output is,
N: [*Sentiment or emotion labels*] C: [*Sentiment or emotion labels*]
where, *Sentiment or emotion labels* is one or more space separated labels: sadness, anticipation, joy, fear, surprise, disgust, trust, anger, positive, negative, neutral.

6 Approach

Our approaches are based on the fact that previous experiences shape perceptions. Some impressions about a particular 'entity' as defined above would have been influenced by the impressions and emotions associated with it previously.

6.1 Classification as a generation task

We propose to predict the media house's and its readers' emotions and sentiments for an event of an entity, given the media house label, past reporting, comments on the reporting, and the event's description. To achieve this, we are treating our task as a multi-label classification task with 11 labels (including 8 emotions and 3 sentiments). Following the recent work on building a unified framework that converts all text-based language problems into a text-to-text format we planned to convert the problem into a text generation problem (Raffel et al., 2020). Thus, the model generates the output labels which it finds suitable for the next event. We used a T5 model for the generation task, which is a transformer based encoder-decoder model. It is trained using cross-entropy loss at each time-step and we used greedy decoding scheme to output the most probable outcome at each output time-step as we are trying to solve a classification problem.

For this base version, we selected only the news headlines and ignored the news article's body.

6.2 Intermediate fine-tuning

We have limited data for training, and the model has to do a lot of work to get better predictions. We hypothesize that the model is doing two main tasks:

1. Understanding the sentiment and emotions of the series of historical news and comments for the entity.

2. Composing this information to learn what should be the sentiment and emotion for the next event.

Intermediate fine-tuning is a medium by which we are making the first task for the model a bit easier. We hypothesize that by training the model on a data-rich similar task it will find it easier to understand the sentiment and emotions for the historical news and the model can rather focus on just learning the composition and the prediction for the next event.

We trained our model on two data-rich task which are very similar to our task. Details of the datasets are in section 4.

1. A task to predict emotion from tweets (Saravia et al., 2018). We trained our model to generate the emotion label given the tweet.

2. A task to predict sentiment from tweets (Barbieri et al., 2020). We trained our model to generate the sentiment given the tweet.

We tried out different variants, training on only sentiment, on only emotions, and on both in different orders. Finally, we settled with training on emotion dataset and on sentiment dataset separately which gave the highest performance in the validation set on our task.

6.3 Article Summarization

Until now, [News headline/article for event] in the [*History*] only consisted of the news headline, as specified in section 5. When media houses present news, some of the sentiment details and emotions are present in the headline. However, many subtle details of these emotions become obvious only when the readers read the news article. Moreover, we observed that when readers commented on the news on Twitter, they made references to important news details present in the article. We hypothesized that including the news articles will help our model further understand the sentiment and emotions in the historical events and make it better in predicting sentiment and emotions for the next event.

However, incorporating the entire news article is infeasible. Some news articles are very long. Even if we include only the primary paragraphs of the news, it is still long to fit multiple historical events in our model. To address the issue, we used a model which has been trained on generating abstractive summary from huggingface (Rothe et al., 2020). The model was trained to generate abstractive summary on the CNN Dailymail dataset (See et al., 2017). We generated the summaries for news articles in our dataset and incorporated the summary for the articles as input along with the news headline in the [News headline/article for event] section in the input.

6.4 Data augmentation using back translation

We have limited data. One approach popularly used in image-based dataset it to augment the data. Following the recent work in Neural Machine Translation for data augmentation using back translation (Edunov et al., 2018) (Sennrich et al., 2016), we planned to use back translation to augment our dataset. Back translation translates the data from English to another language, lets say x, and then translates it back from this language x to English. In the process it retains the key information we care for our task, like the meaning of the sentence, its sentiment, emotions, etc. But, it generates a paraphrase of the sentence. This allows the model to see multiple versions of the inputs as it might see in the real world. Thus making it more robust and possibly high performing.

7 Experiments and Results

7.1 Train-Val-Test Split

We wanted our model to generalize across entities. So we stratified our splitting based on "entity". In other words, an entity and the sequence of events for an entity seen in test set will not be seen in training or validation set. We split the data into train:val:test = 62:18:20. The ratio looks a bit odd as we had to satisfy the constraint that an entity should be completely present in one of these splits.

7.2 Evaluation

To evaluate the performance of our model we computed the F1 Score for sentiments and emotions for news and comments. Since there are 8 emotions and 3 sentiment polarities, we have a total of 11 classes. We aggregate the result using micro and macro aggregation. We also report per class F1 Score as the results help in justifying our different approaches. The model was trained to generate output in the form, N: [*Sentiment or emotion tags*] C: [*Sentiment or emotion tags*]. We split the generated output to two segments representing news and comments. And then we say the model has predicted a label for news or comments if the label is present in the respective segment. We were initially concerned if the model will generate gibberish text. But, we observed that the models (both, baseline and T5) only generate outputs from the target set and it always generates two segments. However, we occasionally saw the model generate 'N' and 'C'. The model evaluation was designed to ignore them.

7.3 Baselines

For our baselines, we stick with the theme of building classification as a generation task. We define two strong baselines for our task.

7.3.1 B1: Consider only the next event

Here we consider only the next event and do not take into consideration any history. So the inputs

here take the form,
Media: [*Media House Name*] E: [*Next Event's description*]
The idea was that the model has seen multiple events for different media houses during training and will learn how media houses portray these events and how its readers react to it.

To train the model we selected a GRU (Gated Recurrent Unit) (Chung et al., 2014) based Sequence-to-Sequence model with attention to maintain long-term memory. It uses cross-entropy loss at each timestep to generate labels. The architecture has an embedding size of 254 with 1024 units in the GRU cell. We selected the maximum vocabulary size of 5000. The code was adapted from a tensorflow tutorial on neural machine translation[1]. We adopted most hyperparameters directly and tuned two hyperparameters: the learning rate and the number of epochs and selected the one which had the highest performance in the validation set. We settled with a learning rate of 0.001 and 20 epochs. In addition, we changed the decoding scheme to greedy to solve the classification problem.

Result The baseline model performed poorly. Although the model sees a lot of examples of emotions and sentiments for different events from media house, it cannot find any lingusitic property to tie it to. For media houses, it gave an F1 Micro score of 0.236 and F1 Macro score of 0.121. On the comments, the F1 Micro score was 0.409 and F1 Macro score was 0.189.

7.3.2 B2: Consider history and next event

For our second baseline, we consider not only the next event, but also take into consideration any history of events. So the input here takes the form as described in section 5. The idea echos our proposal that given past events' news and comments, the model should do a better task to predict the sentiment and emotion for a next event of an entity.

The architecture and training details are identical to baseline B1 as described in section 7.3.1. Here we tuned an additional hyperparameter k, which is the number of past events to consider in history. We settled with a learning rate of 0.001, 20 epochs and k as 2.

[1]https://www.tensorflow.org/text/tutorials/nmt_with_attention

Result This baseline model sees historical events for the entity reported by the media house and its readers' comments. This information is useful in making predictions for the next event. There are linguistic properties it can use to come up with a prediction. But, the model's performance was poor. For media house, it gave an F1 Micro score of 0.247 and F1 Macro score of 0.133. For the comments, the F1 Micro score was 0.376 and F1 Macro score was 0.196. This was an improvement from the first baseline, but the performance was still poor. As seen in tables 7 and 8, the baseline models are unable to detect emotions such as 'sadness', 'surprise', and 'trust'. We assume this is because the model is not complex enough to use historical event information and compose them to predict the sentiment and emotion for next event.

7.4 T5: Training using Transformer

We used a pre-trained T5 model as the base model for our task. The pre-trained model is picked from the huggingface checkpoint "t5-small" (Wolf et al., 2020). The original T5 paper suggested using a constant learning rate of 0.001 with Adam optimizer (Raffel et al., 2020). The huggingface documentation suggested using a learning rate of 1e-4 or 3e-4 which works well with classification task (Wolf et al., 2020). We tuned the model for these three learning rates. In addition, we tuned the model for k, which is the number of past events to consider in history. Following the paper's suggestion, we used Adam optimizer for optimizing the model and didn't used any learning rate decaying schedules. As we are trying to solve a classification problem, we used greedy decoding scheme to output the most probable outcome at each output token. We slightly modified the input to include a prefix, as suggested in the T5 paper (Raffel et al., 2020).
opinion: Media: [*Media House Name*] [*History*] E: [*Next Event's description*]
We used early stopping with patience level of 10, keeping track of the validation loss in order to avoid overfitting and to get the best checkpoint. We settled with a learning rate of 1e-4 and k as 2.

Result As shown in table 6, the T5 model performs significantly better than the baseline models. Firstly, it confirms our hypothesis that past events' reporting and readers' receptions are useful in predicting sentiment and emotions for next

Approach	F1 Score News		F1 Score Comments	
	Micro	Macro	Micro	Macro
Baseline 1 (Only the event)	0.236	0.121	0.409	0.189
Baseline 2 (Event and History)	0.247	0.133	0.376	0.196
T5	0.375	0.261	0.436	0.228
T5 + IT (Emot)	**0.401**	**0.321**	**0.457**	**0.282**
T5 + IT (Sent)	0.379	0.293	0.442	0.229
T5 + IT (Emot) + Summ	0.384	0.281	0.378	0.179
T5 + IT (Emot) + BT	0.391	0.301	0.452	0.222

Table 6: Evaluation of different approaches on the test set. Symbols used - IT: Intermediate Training; Emot: Emotion Dataset; Sent: Sentiment Dataset; Summ: Article Summary; BT: Back Translation

For News	B1	B2	T5	T5 + IT		T5 + IT (Emot) + Summ	T5 + IT (Emot) + BT
				Emot	Sent		
sadness	0	0	0	**0.25**	0	0.143	0
anticipation	0.349	0.286	0.407	0.354	0.318	**0.443**	0.4
joy	0.083	0.069	0.125	0.143	0.154	0.235	**0.261**
fear	0.087	0.1	0.211	0.278	**0.391**	0.143	0.228
surprise	0	0	0	**0.471**	0.2	0.333	0.2
disgust	0	0.051	0.258	0.214	**0.353**	0.308	**0.353**
trust	0	0	0.556	0.526	**0.625**	0.308	**0.625**
anger	0	0	0	0	0	**0.167**	0
positive	0.125	0.265	**0.467**	0.333	0.242	0.375	0.333
negative	0.359	0.559	0.617	**0.693**	0.638	0.637	0.639
neutral	**0.331**	0.136	0.233	0.272	0.3	0	0.269

Table 7: Evaluation of different approaches on the test set for different emotions and sentiments on news headlines/articles. Notations used - IT: Intermediate Training; Emot: Emotion Dataset; Sent: Sentiment Dataset; Summ: Article Summary; BT: Back Translation

For Comment	B1	B2	T5	T5 + IT		T5 + IT (Emot) + Summ	T5 + IT (Emot) + BT
				Emot	Sents		
sadness	0	0	0.133	**0.182**	0.143	0.118	0.154
anticipation	0.261	**0.324**	0.24	0.111	0	0.0714	0
joy	0	**0.261**	0	0.118	0	0	0
fear	0	0	0	**0.182**	0.154	0	0
surprise	0	0	0	0	0	0	0
disgust	0.497	0.438	0.492	0.5	0.387	0.478	**0.524**
trust	0	0	0	**0.286**	0	0	0
anger	**0.406**	0.158	0	0.059	0	0.056	0.054
positive	0.154	0.146	0.727	**0.776**	0.667	0.619	0.651
negative	0.65	0.676	0.684	0.677	**0.691**	0.632	0.641
neutral	0.113	0.154	0.231	0.218	**0.473**	0	0.413

Table 8: Evaluation of different approaches on the test set for different emotions and sentiments on readers' comments. Notations used - IT: Intermediate Training; Emot: Emotion Dataset; Sent: Sentiment Dataset; Summ: Article Summary; BT: Back Translation

event. For media house, it gave an F1 Micro score of 0.375 and F1 Macro score of 0.261. For the comments, the F1 Micro score was 0.436 and F1 Macro score was 0.228. Tables 7 and 8 shows that the emotion 'trust' was also first detected in the T5 approach. The 'positive' sentiment had a high representation here. However, there are many emotions that the model is not able to capture. We assume this is because of limited data. It fails to accommodate emotions which have low representation in training set.

7.5 T5 + IT: Intermediate Fine-tuning

As described in section 6.2, we are using two high-resource intermediate tasks to increase our model's performance.

7.5.1 Emotion dataset

We first fine-tuned our model on the intermediate task of emotion detection (Saravia et al., 2018). As described in section 4, the emotion dataset has six different emotions. We modified the input and output as:
Input: opinion: [*Tweet*]
Output: [*Emotion*]
For the intermediate fine-tuning tasks, we fixed the model's hyperparameters on our task, ported from the best hyperparameters learned above, and only tuned the hyperparameters for the intermediate task such that it gives high performance in the validation set of our task. We couldn't do a comprehensive tuning due to compute constraints. What we observed is that selecting a lower learning rate and fewer number of epochs in the intermediate task reaps a higher benefit in our task. We used the learning rate of 1e-5 for the intermediate task and optimized it using Adam optimizer for 3 epochs with no decay in learning rate.

Result This approach performed the best of all our proposed approaches. For the News data, the F1 Micro score was 0.401 and F1 Macro score was 0.321. For the comments, the F1 Micro score came out to be 0.457 and the F1 Macro score came out to be 0.282. Tables 7 and 8 show that it made the model more sensitive to emotions. 'Sadness' and 'surprise' were first detected here and the representation of 'fear' and 'negative' sentiment increased. Since there is a high correlation between emotions and sentiments, the increase in the representation of negative emotions such as 'fear' and 'sadness' may have led to an increase in the prediction of the 'negative' sentiment. However, for news, 'anger' is 0 and for comments 'surprise' is 0. This may be due to the fact that these emotions have one of the lowest representations in news and comments, respectively in our training set.

7.5.2 Sentiment dataset

We also tuned our model on the sentiment dataset (Barbieri et al., 2020). As described in section 4, the sentiment dataset has three different sentiments. We modified the input and output as:
Input: opinion: [*Tweet*]
Output: [*Sentiment*]
The hyperparameter tuning was identical to the one used for emotion dataset, and our observation of lower learning rate with fewer epochs in intermediate task helping with our task held in this experiment as well. We used the learning rate of 1e-5 for the intermediate task and optimized it using Adam optimizer for 3 epochs.

Result For the News data, the F1 Micro score was 0.379 and F1 Macro score was 0.293. For the comments, the F1 Micro score came out to be 0.442 and the F1 Macro score came out to be 0.229. Its performance is comparable to the one trained with emotion dataset. From table 8, it is evident that the model does perform better in detecting two of three sentiment polarities. In table 7 it can been seen to make emotion detection better for 'trust' and 'disgust'. We hypothesize this is because some emotions are well correlated with different sentiments.

7.6 T5 + IT (Emot) + Summ: Article Summarization

Amongst all the above methods, intermediate fine-tuning on the emotion dataset gave the best results. We now tried using the summary of the articles along with the headlines as the input to this architecture. In order to generate this summary, we selected a model checkpoint from huggingface "google/roberta2roberta_L-24_cnn_daily_mail" (Rothe et al., 2020) which was trained on CNN Dailymail dataset (See et al., 2017). This took the first 512 tokens of the article as input for summary generation and truncated rest of the article. We observed that most of the emotion and sentiment of the article can be captured from the initial paragraphs of the article. Due to lack of compute resources, we only tuned the maximum length of the output to see what worked the best for our

validation set, and went with the default values for all the other hyperparameters. The maximum length of the output was set to 30. The tuning of our model was done using the same approach as described in section 7.4. We settled with a learning rate of 0.001 and k as 2.

Result We added the article summary to improve the performance in predicting the sentiment and emotion of news for next event. We got an F1 Micro score of 0.384 and an F1 Macro score of 0.281 on the News data and an F1 Micro score of 0.378 and an F1 Macro score of 0.179 on the comments. It had a comparable performance. It didn't stand out, but it did give the model more context for news. In table 7, we can see that the model is the only one which captured 'anger' emotion. This is because it is a subtle emotion to find in a headline. Unless someone reads the news article, it is difficult to detect 'anger'. However, it makes predicting emotion and sentiment of next event's comments more difficult as seen in table 8. We assume this is because the model now sees more information about the news than comments.

7.7 T5 + IT (Emot) + BT: Back translation

Working with the checkpoint model of T5 and intermediate fine-tuning done using the emotion dataset, we now augmented our data by back translating. The pivot languages we used were German, Spanish, French, Italian, Russian, Czech and Hindi. The only principle we used while choosing pivot languages was to ensure diversity. The model didn't incorporate article summary and we back translated the news headline and the readers' comments. This essentially added seven times more datapoints to our training data. The tuning of the model was done using the same approach as described in section 7.4. We settled with a learning rate of 1e-4 and k as 2.

Result The model gave an F1 Micro score of 0.391 and an F1 Macro score of 0.301 on the news and an F1 Micro score of 0.452 and an F1 Macro score of 0.222 on the comments. We observe in tables 7 and 8 that in most cases back translation improves performance of emotions and sentiments which were easy to capture even for T5 model without data augmentation. However, it starts struggling with more subtle emotions which were difficult even for the T5 model. We assume this is because with many paraphrases some emotions and sentiment which were evident from the original text becomes more obvious. But, the subtle emotions and sentiment might get lost in the paraphrases, thus making it hard to predict the sentiment and emotions for next event.

7.8 Other Experiment Details

We used the Tensorflow library for our project. We used the model checkpoints ('t5-small', 'google/roberta2roberta_L-24_cnn_daily_mail') from huggingface, and many huggingface libraries were used throughout the project. For back translation, we used the googletrans 4.0.0rc1 Python API hosted in PyPi. For scraping article body from article links, we used the BeautifulSoup and requests libraries. All the models have been trained on Google Colaboratory. The code for all the experiments can be found in the file 'NLP_Project.ipynb' in our GitHub repository[2]. In order to use the available Colab resources efficiently, during development, the code was written without using the GPU and only used 10 datapoints with 1 epoch to make sure the code was working. Once that was confirmed, we switched to GPU to trained our models. We shifted to Colab Pro as we kept running out of resources.

8 Error analysis

We have a sequence of approaches that we followed with an aim to improve the model's performance. The approaches were guided by analysing the class of examples on which our model most commonly failed. Below we describe the class and give representative examples for different approaches. Table 9 shows these examples.

The first baseline (B1) had an ambitious goal to predict sentiment and emotion for an event of an entity without any historical information. This model did not perform very well. We saw that the model mostly generated text which were most commonly observed in the training data as seen in figure 4. These were pairs or triplets of sentiment and emotions like 'disgust' and 'negative', 'fear' and 'negative', 'neutral', etc. There were no specific linguistic properties it tied to.

The second baseline (B2) performed better than baseline 1, but we saw two issues with the model. Firstly, it was not able to represent how different media represented different events. In table 9, the first example for B2 shows that the prediction for the event is 'joy', 'positive', and 'an-

[2]https://github.com/sagarpalao/natural_language_processing

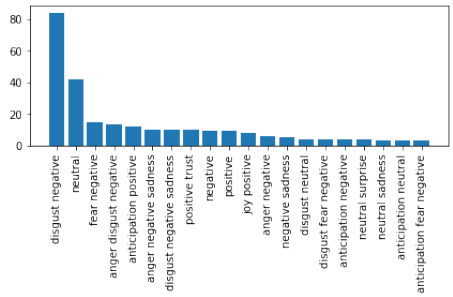

Figure 4: Train labels and its combinations top N distributions.

ticipation'. It was true with other media houses, but WSJ represented this event with 'neutral' sentiment and B2 failed to identify that. Secondly, it looked at the overall sentiment of the text, rather then studying the sentiment of history and using it to predict the sentiment of the next event. For example, in the second example for B2 in table 9, we see it predicts 'joy' and 'positive' for news. It is true for the historical events, but not true for the next event for which we need predictions. For comments also, B2 showed a similar tendency as B1 to generate commonly observed examples from the training data.

The T5 model did better than B2 in the the class of errors commonly made by B2. However, post error analysis, we found that it was plagued with two main issues. Firstly, it was generating less emotions labels. It mostly generated and correctly identified emotion labels which were represented well in the training set. In the first example for T5 in table 9, we can see that it didn't generate 'sadness' and 'anger' label. Second problem we observed was it gets confused between different emotions. In the second example for T5 in table 9, we see that it gets confused between 'joy', 'surprise' and 'anticipation'. It seems to get the direction right, but couldn't well classify the emotion. We observed very similar behavior for the model T5 with intermediate training on sentiment.

The model with intermediate training on emotion (T5 + IT (Emot)) was good at labeling emotions, but the major issue we observed with the model was it was very sensitive to emotions. The model was predicting more emotion labels. In example for T5 + IT (Emot) in table 9, we can see that the model predicts 'joy' and 'disgust'. But the true prediction for next event was quite neutral and had fewer emotion labels. It relied mostly on the historical events' texts in the input to come up with this extra emotion labels. In the example, the historical events' news represent 'joy' and the comments represent 'disgust'.

The model which had article summary (T5 + IT (Emot) + Summ) in input performed quite well in predicting sentiment and emotions of the next event for the media agency. But the major error we observed form the output of this model was that it was less verbose when it comes to predicting sentiment and emotions for the comments. In the example for T5 + IT (Emot) + Summ in table 9, we see that the model was predicting 'disgust' and 'negative' for this example. In fact it was predicting these labels for many comments. This may be because of the fact that these labels were the most common labels in the training data.

The model trained on augmented back translated data (T5 + IT (Emot) + BT) became quite sensitive to sentiments and emotions which were commonly represented in the training data. We observed that the model started to predict most commonly observed sentiments and emotions. In the example for T5 + IT (Emot) + BT in table 9, we see that it predicts 'disgust' and 'anticipation'. These were the most commonly observed labels which we observed sporadically appeared in many predictions generated by this model. These were labels which the base T5 model got right, and it was amplified with back translation leading to few errors in the test set. A possible fix could be to augment the emotions examples which are in lower counts relative to other samples. This could help fix the skewed data, and allow the model to learn more about other emotions.

We notice that none of our models do well in certain examples. One such example is demonstrated in table 10 where the input and comments contain a wide variety of emotions over the history provided. The combination of true labels for the media in this example is an uncommon set in the training data. Due to this, none of the models are able to decipher what the emotions or the sentiment is of the media for the event provided. Since the input history is widely varying, all models are confused about what to predict, and we see that from the wide variety of outputs across the models.

9 Contributions of group members

All team members collected and annotated data as well as contributed to writing the final report.

Model	Input	True	Prediction
B1	Media: CNN E: Djokovic beats Nadal	N: joy positive C: joy positive	n anticipation neutral c disgust negative
B2	Media: WSJ N1: Volatility Spurs Onslaught of Five-Minute Trading Halts in GameStop C1: Every time I read an article about Silver, ... E: GameStop stocks keep rising and this trend has now spread to Netflix and the big screen.	N: neutral C: anger negative sadness	n anticipation joy positive c disgust negative
B2	Media: Reuters N1: GameStop rallies back as U.S. regulators eye wild trading C1: Robinhood has been restricting transactions on shares 2 protect Wall St. hedge funds, actually stealing ... E: Brokers' Restrictions on GameStop to be reviewed.	N: anticipation neutral C: disgust negative	n joy positive c anger disgust negative
T5, T5 + IT (Sent)	opinion: Media: CNN N1: Novak Djokovic ends Tokyo 2020 without a medal after losing in singles and withdrawing from mixed... C1: same ... en Slam still in play C2: He can definitely do it. E: Djokovic beaten by Daniil	N: negative sadness C: anger disgust negative	N: neutral C C: neutral disgust negative
T5, T5 + IT (Sent)	opinion: Media: NYT N1: Novak Djokovic Aims to Win at Wimbledon, and His Side Hustle C1: It's actually a sad era for tennis. ... mentality to bounce back when many others would give up E: Djokovic makes records 20th grand slam	N: joy positive surprise C: negative	N: anticipation positive C C: disgust
T5 + IT (Emot)	opinion: Media: WP N1: Love him or not, Novak Djokovic embraces pressure at U.S. Open as he inches closer to history C1: BIG NOT N2: Novak Djokovic tosses, smashes racket in ... with stress better? E: Djokovic to play in US open	N: anticipation positive C: negative	N: joy joy C: disgust disgust
T5 + IT (Emot) + Summ	opinion: Media: CNN N1: 227 people were killed defending the environment last year, a new report shows. That's a record. C1: Put politics aside: Climate change is real. ... millions of years including warmer periods and ice ages. E: UN reports suggest global warming	N: fear negative C: fear negative	N: fear negative C: disgust negative
T5 + IT (Emot) + BT	opinion: Media: WSJ N1: GameStop CFO Was Forced Out as Activist Investor Pushes New Strategy C1: ... N2: Reddit Legend Keith Gill Boosts Stake in GameStop C2: He likes the stock 3 E: GameStop focuses on transformation and forms a new committee to do so.	N: joy neutral C: joy positive	N: anticipation positive C: disgust positive

Table 9: Error analysis: Representative examples from test set

Input	opinion: Media: NYT N1: Novak Djokovic, King of the Olympic Village, Loses Run at Golden Slam C1: Where's the Olympic spirit? Disgraceful. Champion don't behave this way N2: Novak Djokovic loses in a men's singles tennis semifinal. C2: Zverev played a superb game! Congrats! E: Djokovic looses in Tokyo olympics
True	N: negative sadness surprise C: disgust negative
B2	n joy positive c disgust negative
T5	N: anticipation C: disgust negative
T5 + IT (Emot)	N: disgust C: disgust negative
T5 + IT (Sent)	N: joy C: disgust negative
T5 + IT (Emot) + Summ	N: anticipation C: disgust negative
T5 + IT (Emot) + BT	N: joy C: disgust negative

Table 10: Error analysis: Difficult example for all models

We collectively performed error analysis of each model in order to decide the approach for the next models.

- Akshay Sasikumar: Built and trained baselines B1 and B2.
- Gayatri Belapurkar: Built and trained T5 and T5 + IT (Sent)
- Saloni Chalkapurkar: Built and trained T5 + IT (Emot)
- Sagar Palao: Built and trained T5 + IT (Emot) + Summ and T5 + IT (Emot) + BT. Code for scraping article content from article link and generating article summaries and visualizing model results.

10 Conclusion

We fine-tuned a T5 model to predict the sentiment and emotion of the media house and its readers for the next event of an entity. We further tried to improve the performance of our model using intermediate fine-tuning, article summarization, and back translation. We found that intermediate tuning on emotion task really helped the model to be more sensitive to different emotions. Initially, we tried to tune the hyperparameters of our model using arbitary values. This turned out to be very difficult and we were not even able to beat our baseline models. But, soon we realized that it would be best to look these up in papers which have used these models and try out their hyperparameters. This helped us a lot with fine-tuning. For our task we take into consideration the historical events to make prediction for the next event. We were restricted by the length of the context and were not able to try out with more historical information. We plan to extend this work by trying out different efficient and long range transformers.

References

Angelidis, S. and Lapata, M. (2018). Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Barbieri, F., Camacho-Collados, J., Espinosa-Anke, L., and Neves, L. (2020). TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Bražinskas, A., Lapata, M., and Titov, I. (2020). Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169, Online. Association for Computational Linguistics.

Chen, H., Sun, M., Tu, C., Lin, Y., and Liu, Z. (2016). Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659, Austin, Texas. Association for Computational Linguistics.

Chen, P., Sun, Z., Bing, L., and Yang, W. (2017). Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, Copenhagen, Denmark. Association for Computational Linguistics.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding back-translation at scale.

Feng, Y., Rao, Y., Tang, Y., Wang, N., and Liu, H. (2021). Target-specified sequence labeling with multi-head self-attention for target-oriented opinion words extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1805–1815, Online. Association for Computational Linguistics.

Gangula, R. R. R., Duggenpudi, S. R., and Mamidi, R. (2019). Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 77–84.

Han, W.-B. and Kando, N. (2019). Opinion mining with deep contextualized embeddings. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 35–42, Minneapolis, Minnesota. Association for Computational Linguistics.

Iyyer, M., Enns, P., Boyd-Graber, J., and Resnik, P. (2014). Political ideology detection using recursive neural networks. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1113–1122.

Phang, J., Févry, T., and Bowman, S. R. (2019). Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks.

Pruksachatkun, Y., Phang, J., Liu, H., Htut, P. M., Zhang, X., Pang, R. Y., Vania, C., Kann, K., and Bowman, S. R. (2020). Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work?

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer.

Rothe, S., Narayan, S., and Severyn, A. (2020). Leveraging pre-trained checkpoints for sequence generation tasks.

Saravia, E., Liu, H.-C. T., Huang, Y.-H., Wu, J., and Chen, Y.-S. (2018). CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.

See, A., Liu, P. J., and Manning, C. D. (2017). Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units.

Suhara, Y., Wang, X., Angelidis, S., and Tan, W.-C. (2020). OpinionDigest: A simple framework for opinion summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5789–5798, Online. Association for Computational Linguistics.

Vu, T., Wang, T., Munkhdalai, T., Sordoni, A., Trischler, A., Mattarella-Micke, A., Maji, S., and Iyyer, M. (2020). Exploring and predicting transferability across nlp tasks.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.