Modelling the sentimental correlation of multi-modal information

Name: Pradeep Kumar
Registration ID: 2105711

A thesis submitted for the degree of Master of Science in Artificial Intelligence

Supervisor: Dr. Yunfei Long

School of Computer Science and Electronic Engineering University of Essex

August 2022

Acknowledgement

I would like to express my special thanks and gratitude to my supervisor, Dr Yunfei Long for taking me under his tutelage and giving me this opportunity to work on 'Modelling the sentimental correlation of multi-modal information'. This project provided me an exciting and challenging problem statement to apply my knowledge in the field of artificial intelligence and expand my skills and knowledge-base further in this domain which was only possible because of Dr Yunfei Long and him providing me with his invaluable guidance and support throughout the research. He inspired me to work more diligently by teaching me the methodology to carry out the research as clearly as possible. I would also like to thank him for his empathy and his extensive and in-depth knowledge in this domain. It was an honour and privilege to work and study under his guidance.

This page is left blank

Abstract

Multi modal sentiment analysis is a hot topic for researchers, especially by multi billion dollar companies like Meta and Alphabet. Sentiment analysis tries to enable intelligent systems to learn, recognize and interpret human emotions. In multi modal sentiment analysis the intelligent system tries to perform the above actions by uncovering hidden and tightly coupled mappings between the multiple modalities of data being fed as input. BERT (Bidirectional Encoder Representation from Transformers) is a deep and large pre trained model used for language representation. BERT was limited to text only applications where Fine tuning yielded state of the art scores in multiple natural language processing tasks such as question answering and language inference tasks. But in the recent past BERT is also being used for multi modal inputs which has yielded state of the art results in multi modal sentiment analysis too. In this paper the input for BERT is being extended to include all three modalities i.e. text, audio and visual inputs to accurately predict the sentiment using the fully multi modal inputs. The core idea of Fully Cross Modal BERT (FCM – BERT) relies on multi modal attention mask which is a function designed to dynamically learn and assign weights for words based on the fused information representations of textual, acoustic, and visual inputs. The proposed model is evaluated against CMU – MOSI dataset which is available publicly. The experimental results show significant performance improvements over previous baseline models when all three modalities are taken into account. This shows that masked multimodal attention is able to capture and map the relationship between text, audio and visual features.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

With the advent of social media platforms like meta and its products, YouTube, Twitter, etc. the amount of user generated multi modal data has increased tremendously. Such rich information can be used to create, train and improve the performance of models which can then be used to create a profile of the mental health of the end user and protect the person from cyberbullying based on the type of videos or tweets the end user put out. Such models can not only save the lives of people but also help reduce such issues by automating the prediction of the sentiment of the end user. Another use case is to suggest the ads based on the end users's sentiments. This will also help companies increase their revenue through targeted advertising which will be based on the end users' emotions and not just on their previous preferences. [30] Recent advances in deep learning techniques like the use of transformers and self-attention layers have enabled researchers to map sophisticated co-relationships between different modalities with a heavy emphasis on text modality. This is because research on text sentiment analysis has yielded very promising results by using deep transformer-based models such as BERT. This not only shows that text holds quite a lot of information and is able to predict the sentiments on its own to a certain extent when the text data contains words that express the emotions explicitly but adding additional modalities like speech and/or video/images can augment the inputs for the model to create feature rich information which can then be exploited by the model during training to mapping the tightly coupled sentiment information between the different modalities which results in far more accurate predictions of the sentiments in real world applications.[38] Another issue with using only text modality is

that it cannot predict the sentiment effectively when the emotions are not expressed implicitly which is usually the case with human beings where the facial expressions also implicitly express the emotions which cannot be expressed in words or text alone. Using more than just one modality provides semantic information which is tightly coupled together with the different modalities and learning these underlying representations would give better results than using just one modality and losing this tightly coupled information. For example, a video might contain some words which taken in its own context would mean something differently when compared to the same thing but taken into context with a visual modality. For example the video where a happy face of a person is coupled with the words would mean that the words were rather friendly than hostile. Such multi modal data helps in detecting emotions more effectively as the multi modal data adds in more semantic information, which would have been lost if each of the modalities were taken individually for analysis and classification. In general cases, while dealing with human emotions the speech modality is usually accompanied by visual modality i.e., speech (tone and timing of words) along with visual modality (facial expressions). The information in audio modality is contained in the word being spoken, the tone of the words used, the tempo or timing of words being spoken, the intensity of the spoken words [17]. A combination of these features holds explicit information regarding the sentiments. The information in visual modality is contained in the way the face expresses the emotions i.e. anger, sadness, happiness, etc. and to what extent these expressions are expressed which also determines the intensity of the emotions. For example, in a text transcribed video consider a case where a person is angry. There would be pauses in the sentence and in those pauses the facial expressions would express the emotion more clearly. Thus the text modality alone would not be holding all the information required for an accurate representation of emotion and its intensity. Thus the combination of these visual, audio and textual modalities can paint a far more accurate picture of the human emotion than using just one or a combination of two of these modalities. Hence using multi modal information for sentiment analysis can yield more accurate results. In the pursuit to capture and map the human emotions and sentiments by machine models, in this paper I have proposed a model which can take into account all three modalities i.e. text, speech and images to learn the hidden and complex relationships between the modalities to effectively predict the sentiments. This model uses BERT output for text representation. The audio and video representations are learned by the model which is then used to create the fusion

attention output which uses all 3 learned representations of text, audio and image which in turn is used by top layer of the model to predict the sentiment.

CHAPTER 2

Related Work

Multi modal sentiment analysis is a hot topic for research due to the availability of human generated data through the use of social media platforms. Considering the fact that these multi modal data are tightly coupled which hold useful information which can effectively capture the sentiments of human beings and that would've been lost otherwise if only one modality is used [5]. A number of techniques are used to utilize multi modal data, some of the most widely used ones are feature fusion and decision fusion [12]. In the first case the features or feature representations are combined to get a unified representation which can then be used by model to make predictions. In the second case the decisions taken by multiple individual models or each model for each of the modalities are combined to reach a common final decision. Fusing the features of the different modalities usually represents more information as no information is lost by the model as is the case with decision fusion. [43] Zhou et al took the approach of feature fusion where the features of audio and text were combined and then fed into a semi-supervised multi-path generative neural network. In this approach, the features were first extracted by local classifiers and then these high-level features were then fed into another global classifier. [39] Zadeh et al uses a tensor fusion approach to where the product of the features of multi modal data is used for the representation of multi-modal information. In this case, initially a network of LSTMs is used whose output is then fed into a network of attention layers. The model then learns the cross interactions between the different modalities. [19] similar to feature fusion Liu et al uses a low rank feature fusion. The issue with tensor or feature fusion is that these are usually in

high dimensions, and such computations require a lot of time and compute resources. This issue becomes more pronounced when deep and large transformers models are used. In order to overcome these operational issues low rank tensor fusion is used which requires less compute resources and is more efficient when compared to high dimension tensor or feature fusion techniques. The paper concludes that low rank tensor fusion operations not only improve efficiency but also improve the accuracy of the model. [23] Poria et al uses a LSTM which is contextual i.e. it takes into account the utterance level information to more accurately represent the multi-modal information. As stated above decision fusion is the technique where each modality is analyzed and classified individually and the output from each model is then fused or the collective output vector is then used by another global level classifier to make the final decision. Dobrišek et al. [10] used the decision fusion technique and experimented with weighted sum and weighted product for audio and video decision fusion. The paper concluded that weighted product performs better than weighted sum.[39] Zadeh et al uses a multi attention recurrent network to learn the tightly coupled hidden information between different modalities. This model uses attention blocks to map the relationship between different modalities and then is fed into a modified LSTM block called Long-short Term Hybrid Memory (LSTHM). The paper concludes that the experiment showed SOTA performance at that time. [33] Tsai et al uses a cross modal attention in their transformer model which attends to pairwise sequences in a uni directional manner. This enables the model to learn the interaction between the multi-modal sequences irrespective of the time-step. Some other techniques involve a hybrid approach which uses both feature fusion and decision fusion like the one [1] implemented by Arjmand et al. For audio, a novel and customized layer known as Lightweight Attentive Aggregation (LAA) is used which is basically a bidirectional GRU (Gated Recurrent Units) followed by attention network. This enables the audio features to be extracted without any constraints on the sequence length. The decision fusion occurs when the text modality is fed into a BERT model and then the outputs from LAA and BERT are used to predict the sentiment. The paper concludes that the implemented model is able to perform better than uni-modal models used for sentiment prediction. Another approach similar to TEASEL is MCTN (Multi modal Cyclic Translation Network). The model learns the multi modal representation from source modality to target modality by translating between modalities[22]. It can be described as for a given Source Modality M_s and Target Modality M_t the model learns an intermediate representation that

encapsulates the fused information of both the modalities. When a cyclic translation loss function is used which involves both the forward translation loss which is the loss from source to target modality and the backward translation loss which is the loss from target back to source modality. This technique is extended to include multiple target modalities in MCTN. The MCTN model is trained first to learn the multimodal relationship between the different modalities and then is trained on specific tasks such as sentiment analysis to learn the multimodal representation for the required task.

Earlier attempts for modelling multi modal sentiment analysis usually focused on using a pair of modalities like audio and video modality pairs or audio and text modality pairs. Some of these attempts used modified Convolution Neural Networks (CNN) to map the relationship between the modalities. SAL or Selective Additive Learning [36] is an architecture built on top of neural networks for improved performance. It is made of two phases namely the selection phase and the addition phase. In the Selection phase, the SAL identifies the dense representations learned by the neural network. In the addition phase the SAL removes the error features or confounding features by adding Gaussian noises. Some models used a hybrid architecture of using Convolution layers along with Recurrent layers. One such approach was implemented by Donohue et al [11]. This model used this approach so as to capture the long term dependencies between the sequences. When compared to CNN only architecture which has fixed length and fixed frequency (spatio – temporal) receptive fields, recurrent layers can be stacked in both the spatial and temporal regions. Such layers are useful for learning the representations when the training data is limited or if there are complicated non linear relationships which cannot be easily modelled by conventional CNN layers. In such cases, Long term RNNs are able to learn long term mappings between the sequences of variable length. Such networks can also be optimized using back propagation. Another novel application of multimodal models is in combating human trafficking. Tong et al [32] developed a model called Human Trafficking Deep Network (HTDN) and also developed a novel dataset for the same called Trafficking – 10k. This dataset contains two modalities which are text and images of advertisements which are suspected to contain information pertaining to human trafficking. The model uses the representation fusion technique where the dense representations of the language network and visual network are fused using the outer product and then it is passed to a convolution decision network to get the final outcome whether the ad is suspected to contain information regarding human

trafficking or not. The language network is primarily a Long Short Term Memory network with a dense layer on top to create dense neural representations. The input is a sequence of word embeddings made using skip gram technique. The visual network is a fine tuned VGG network called Trafficking VGG or T-VGG. Another approach is to use feature fusion along with hybrid architecture. As done by Sun et al [31] uses a 1D spatial convolution layer followed by LSTM to learn and extract dense audio representations and the same technique is used for video representations. Then the feature fusion stage takes places; where both of the learned representations are then used to get audio – text and video – text features using the outer product. The audio – text and video – text representations which are then passed into a convolution neural network where the CCA (Canonical Co-relation Analysis) loss is calculated. The advantage of using CCA function is that the model is able to learn the paired modality relationship more effectively than using other metrics like cosine similarity.

CHAPTER 3

Deep Language models and Pre Training

It has been shown that deep and large language models perform better than statistical models [42] however that is not always the case as the improvement from BERT large over BERT mini is only 4 percent. But that improvement decay is due to the fact that large models have higher variance which results in less than expected improvement in accuracy. However, after fine-tuning the large language models perform better in the downstream tasks and hence perform better than smaller language models. So, using large language models like BERT should perform better than smaller language models like GloVe. Also using large scalable language models like deep stacking of transformer blocks (BERT like and GPT (Generative Pre-Trained) like models) are shown to have better results which has been proven empirically and this trend is expected to continue as the shown in this paper [14]. Using deep language models has shown to improve performance on instance level and token level tasks and using transformer-based models have improved performance on contextual level tasks too. [21] Peters et al published the paper Embeddings from Language Models (ELMo). It is a large and deep, bidirectional model that was trained using a large general text corpus. The study showed that it can substantially raise performance on all six language inference tasks. These findings are in line [25] with the observations made by Radford et al which states that deep language models are generalized learners and hence with enough unstructured text data and deep enough model, the trained model should be able to handle multiple tasks and with fine tuning the model should be able to perform better than most models and achieve the desired results. In another paper published by Radford et al [24] introduced the now famous and

groundbreaking GPT (Generative Pre –Trained Transformer). This used pre training of the model on a general corpus using unsupervised learning and was then fine tuned on target language tasks using supervised learning. This is done so that the model can learn a general representation of the language and to transfer that learned knowledge into downstream tasks without much changes. The overall training process for GPT can be classified under semi supervised learning task where both unsupervised pre training on a large text corpus and supervised training for targeted tasks is used. In the unsupervised pre training process the model is trained to maximize the log likelihood of the tokens $U = \{u_i, ..., u_n\}$ using the objective function defined as

$$L_1(U) = \sum_i logP(u_i|u_{i-k}, ..., u_{i-1}; \Theta) \tag{3.0.1}$$

where Θ is the parameters of the network, k is the size of the window, P is the conditional probability modelled using the network. The architecture used in GPT is modified version of staked decoder part of transformers. It uses multi headed attention with position embeddings to produce an output represented as:

$$h_0 = W_e + W_p \tag{3.0.2}$$

$$h_l = transformer_{block}(h_{l-1}) \forall_i \in [1, n] \tag{3.0.3}$$

$$P(u) = softmax(h_n W_e^T) \tag{3.0.4}$$

Where W_e is the token embedding and W_p is the position embedding. The supervised learning task uses the pre trained weights and a log likelihood function with respect to label y given as:

$$L_2(C) = \sum_{x,y} logP(y|x^1, ..., x^m) \tag{3.0.5}$$

As the model was already generalized over a large text corpus during the pre training phase, the GPT was able to perform well on supervised or target tasks. When it comes to auxiliary learning tasks such as Position Of Speech tagging, Named Entity Recognition, etc. the pre trained model performed well too as the model had already learned the underlying semantics during the pre training phase. Similar to GPT which uses un supervised pre training and supervised training in target or downstream tasks, BERT also follows the same procedure. However what made this different and novel was the use of task aware inputs for fine tuning and the expected results were achieved with little to no changes to the architecture

for transfer learning. Unlike ELMo and GPT, BERT (Bidirectional Encoder Representation from Transformers) was trained on two different unsupervised tasks namely NSP (Next Sentence Prediction) and MLM (Masked Language Modelling) [9]. This again follows the same trend of training a deep and large transformer model on a huge corpus of text however what makes this different is that unlike the other above-mentioned models which were trained in an unsupervised manner and for generalized tasks, BERT was then trained in a supervised manner but for specific tasks as mentioned above. This made BERT contextually aware of the text it was being trained in and was well able to map the relationship between words and the context or sentence it is being used in. Fine tuning and pre training BERT have yielded SOTA results in most tasks and hence is the choice of text back bone model in CM BERT

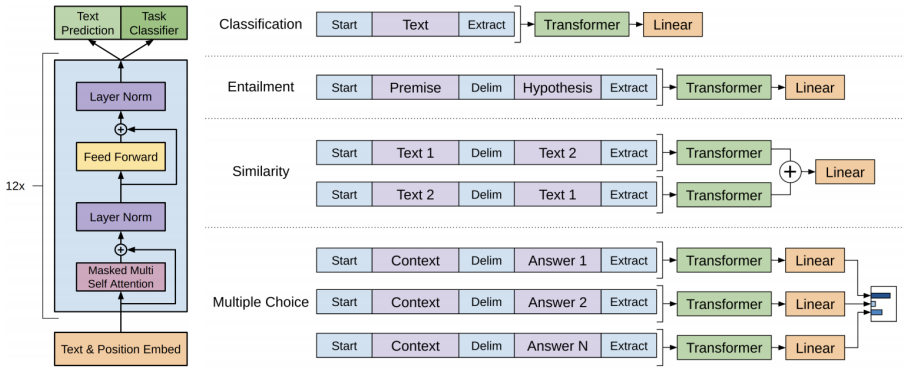

Figure 3.1: GPT architecture

image credits: [24]

3.1 Self-attention layers and transformers

For quite some time LSTM and RNN were achieving state of the art results in language modelling and were able to be stacked to create deep and large language models. The large and deep Recurrent models were used for language modeling as well as machine translation where encoder–decoder models based on LSTM were used [15]. In this technique the input sequences represented as $x = (x_1, x_2...., x_n)$ are mapped by the encoder represented as $z = (z_1, z_2...., z_n)$. This sequence is then given to the decoder which generates the output sequence represented by $y = (y_1, y_2...., y_n)$. LSTM perform their computations by considering the input at time step t and the hidden state at time step t-1. This sequential series of

inputs from the hidden state allows the model to map the relationship between the different inputs while at the same time maintaining the sequential nature of the inputs. For each input sequence for a given time step, since the encoder–decoder is built using LSTM layers consumes the previous output sequence for the current input sequence. However, the shortcoming of such models is that the model cannot map the relationship between inputs over large distances. Another issue with recurrent layers is the very sequential nature that makes it useful. As the input at t will always rely on hidden state t-1 for the output at time-step t, the computation at time step t_n has to wait till previous time-step is processed. Few widely used techniques used to speed up the computation in LSTM networks are:

1. Matix factorization where the matrix is broken down into two smaller matrices and then the calculation is performed [16]

2. Partitioning of the LSTM into inputs and states which allows the calculations to be done independently of each other thus reducing the time dependencies [16]

3. Conditional computation: Where parts of the network are active for some time based on the inputs.[29]

The third technique involves the use of Mixture of Expert (MoE) layers which consist of computational gates. Mixture of Experts embeddings decide which part of the network will be active for a given input. The decision is made by MoE after training the MoE layers which can be as simple as a feed forward network or can be a reinforcement agent.

However, all of these optimizations bring in an added level of complexity to an already bloated LSTM mechanism. To overcome this problem attention mechanism is used. The attention mechanism allows the model to map sequences without regard to distance in the inputs or outputs. However, these layers are usually used in conjunction with recurrent networks for tasks such as neural machine translation and generalized language modelling. In order to push the boundaries of state of the art in large language models stacked self-attention layers are used. Self-attention is a technique in which for a single sequence, the layer learns the relationship between the different entities in different positions of the sequence.

3.1.1 Attention

Attention function is basically a mapping between 2 entities namely a query and a set of key-value pairs with respect to an output. All the entities i.e. query, key, values and outputs are vectors. The weights are assigned to the weighted sum of values by a learned function. The function assigns the weights to each value by taking into consideration the corresponding key.

3.1.2 Multi - head attention

A multi head attention mechanism can parallelly calculate the attention over multiple projections of the keys, queries and values over different time steps with different linear projections. These projections are again learned functions. For each of these new projections for keys, values and queries the attention calculation is performed yielding output values. These newly generated multiple outputs are then projected again giving the final output values. This multi head attention mechanism allows the model to attend to information jointly over different positions and sub spaces (sub sets).

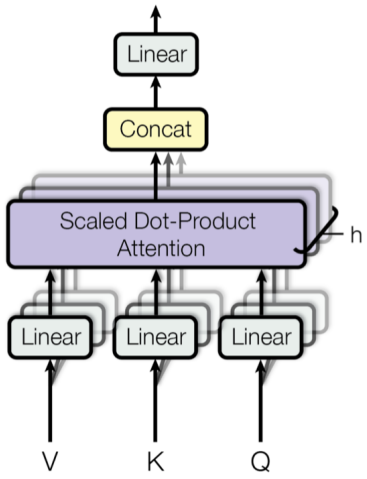

Figure 3.2: Multi head attention

image credits: [35]

3.2 Transformers

Similar to encoder – decoder networks made using LSTM, the transformer also follows the same architecture where two distinct parts of encoder and decoder is used.

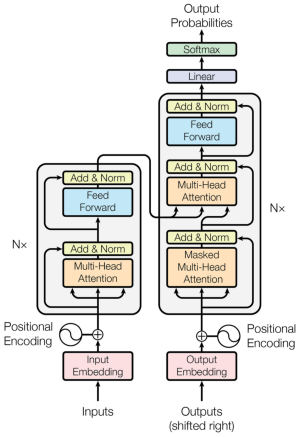

Figure 1: The Transformer - model architecture.

Figure 3.3: Transformer architecture

image credits: [35]

3.2.1 Encoder

The encoder consists of 6 layers in a stack. Each layer further has sub layers. Sub layer one consists of multi head self attention mechanisms and the second sub layer consists of a simple feedforward network with position-wise encoder. A residual connection is used to connect the sub layers which is then followed by a layer normalization [2]. The output of the normalization layer is represented as $LayerNorm(x + Sublayer(x))$ where $Sublayer$ is the function implemented by the sublayer. To maintain and facilitate computations across the residual connections the outputs are maintained in the dimension of $d_{model} = 512$

3.2.2 Decoder

The decoder also consists of a stack of 6 layers. However, unlike an encoder which has 2 sub layers, the decoder consists of three sub layers, the third layer being a multi head attention layer which attends over the output of the encoder stack [35]. Same as in encoder stack, a residual connection is applied over the first two sublayers of the decoder stack followed by a

normalization layer. Another difference in decoder stack is that the self-attention mechanism does not attend over the position embeddings. However, the output embeddings are offset by one position which ensures that predictions at position i can only depend on the outputs of position less than i.

3.2.3 Scaled Dot Product

A scaled dot product is used in the original implementation of transformers. The scaling is done by calculating the dot products of query and keys and then dividing it by $_k$ where d_k is the dimension of the key. This value is then fed into a softmax function to get the weights. In order to improve the compute efficiency the set of Queries, Keys and Values are passed as a matrix where Q is the matrix for the queries, K is the matrix for the keys and V is the matrix for the values. The attention calculation is represented as:

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{3.2.1}$$

Two widely used attention functions are [3] additive attention function and dot product attention function. Dot product attention function is similar to what is used in self attention transformers with the change being that a scaling factor $\frac{1}{\sqrt{d_k}}$ is used. Although both the techniques result in almost the same values, additive dot product performs better for larger values [4]. This is due to the fact that for larger values of d_k the dot products would grow larger and after passing it to the SoftMax activation function the values yielded would be very small gradients. To avoid the issue of vanishing gradients a scaling operation is performed.

3.2.4 Why self attention is better than other techniques

There are three main reasons why self attention is better than recurrent or convolution layers when it comes to mapping an input sequence $X_n = (x_1, \ldots, x_n)$ to another sequence of same length $Z_n = (z_1, \ldots . z_n)$ which is usually the case in typical encoder – decoder mechanism. The first one is parallelism. Second is computational complexity and the third is being able to map long range dependencies. In a self attention operation, all the tokens are sequentially operated upon by a fixed number of operations whereas in a recurrent layer it requires n number of operations i.e the operations increase with an increase in the number of tokens. In terms of computational complexity self attention layers are comparatively more efficient and

faster than recurrent layers if the dense representation dimension is larger than that of the sequence length which is usually the scenario for token representation such as word piece [37] or byte pair encoding [28]. This compute performance can be further optimized by limiting the self attention layer to attend to only limited context which can be achieved by reducing the neighbouring context centered around the current token input. When compared to the convolution layer with kernel size k smaller than sequence size n, it would not be able to map all the inputs and outputs pairs. To achieve this the network would require a stack of (n/k) convolution layers if normal convolution layers are used or $log_k(n)$ layers would be required if dilated convolution layers [4] are used. When compared to recurrent layers, convolution layers are more computationally expensive by a factor of k. Separable convolution layers [7] reduce the computational complexity when compared to conventional convolution layers however when compared to self attention layers they are still computationally expensive since in self attention layers the combination of self attention and feed forward operation would be equal or less than the separable convolution layer. As stated in the paper by Vaswani et al [35] self attention models are more interpretable than convolution networks since the operations in self attention mechanism can be expressed as matrix multiplications however the Query, Key and Value matrices are learned representations which may hinder this assumption.

3.3 BERT

BERT which stands for Bidirectional Encoder Representations from Transformers is a ground-breaking deep language model. Two things make BERT stand out from the rest of the models, one is pre training and the second is the architecture. What makes the pre training, especially in Masked Language Modelling is that it is not like the pre training done in GPT where the model can only attend to previous tokens which are unidirectional. This restriction results in sub optimal results for sentence level as well as token level tasks as the information can only be mapped by the model based on previous tokens and hence cannot be applied to very long sentences. BERT applies the masked language modelling technique in which some random words of the sentence are masked and model is able to map the relationship between the different words in the sentence irrespective of the distance as the model is trained to learn the mappings between the unmasked words and masked word. This Masked Language

Modelling task enables the model to fuse the left and right context. The second task in pre training is called Next Sentence Prediction (NSP). Many natural language understanding tasks such as Natural Language Inferences (NLI) or Question Answering (QA) involve the model to learn the mappings or relationships between 2 sentences. In order to achieve this, a binarized sentence is given as input where sentences A B for each training example corresponds to a pair of sentences. The dataset was a balanced one where 50% sentence B was the sentence that follows sentence A and was labelled as IsNext and 50% of the times sentence B was a random sentence labelled as NotNext. These two tasks in turn allowed Jacob et al to train a deep bidirectional Transformer. As BERT is a deep stacked encoder only transformer model it was pre trained on a BooksCorpus [44] which has 800 million words and English Wikipedia which has 2,500 million words. Only text content of the Wikipedia was used i.e. the tables, lists and headers were ignored. The reason for using Wikipedia as a dataset is because document level corpus allows for extraction of long contiguous sentences which may not be available in shuffled sentences datasets such as Billion Word Benchmark [6]. After the pre training of BERT, it can be fine tuned. Fine tuning BERT does not involve

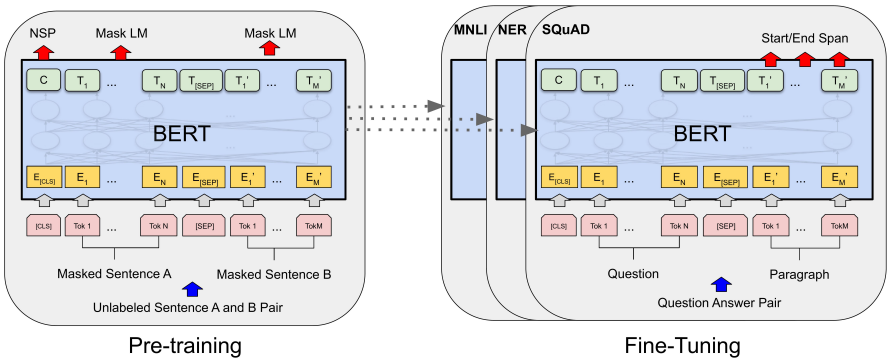

Figure 3.4: BERT architecture in pre training and fine tuning
image credits: [9]

many changes to the architecture as the backbone of BERT is transformer encoders which use self attention mechanisms. This allows BERT to be fine tuned for token level or sentence level or sequence level tasks. Pre training tasks are unsupervised learning tasks. For fine tuning the model is initialized with pre training parameters, and all the parameters are fine tuned or modified during the fine tuning tasks. Fine tuning tasks are usually supervised learning tasks. For non self attention models, text pairs are usually encoded beforehand but BERT can encode and apply self attention together, as concatenated tokens when fed into self attention

layers effectively means applying bidirectional cross attention between the sentences.

The architecture also makes BERT unique in the sense that unlike transformers which uses an encoder – decoder architecture BERT uses only the encoder part of transformer. BERT has many different versions based on the layers and parameters, however $BERT_{BASE}$ is the one used in Fully Cross Modal BERT. $BERT_{BASE}$ has 12 Layers each with a hidden size of 768 and 12 attention heads. $BERT_{BASE}$ has the same model size as that of OpenAI GPT.

3.4 Activation Function

The activation function used in this model is GELU [13] which is defined as

$$GELU(x) = xP(X \leq x) = x\phi(x) = x.\frac{1}{2}1 + erf(x + \sqrt{2}) \tag{3.4.1}$$

GELU which was introduced after RELU to overcome certain drawbacks of RELU such as negative activation is much smaller in GELU than in RELU and the curve of GELU is much smoother than RELU which makes GELU differentiable in a much wider range than that of RELU. In simpler terms this means that RELU which has a scaled linear activation in negative regions can give near zero or zero values if large negative inputs are given. This can cause the neurons to be effectively dead also called the dying RELU problem. GELU overcomes this by having a much smoother negative region graph which enables the GELU activation function to return non zero values if the inputs are large negative values which in turn keeps the weights of neurons active. This means that GELU can more easily approximate functions which may be complicated for other activation functions like RELU and ELUs. GELU is used in almost all Transformer based deep language models such as BERT, GPT 2, etc. GELU is also the default activation function used in this model too, with options to use SWISH too. [26] However Swish has been re proposed as $x.\beta(x)$ where β is either a constant or a trainable parameter. The older version of swish also called as SILU can be used as another option for activation function which has a similar graph to GELU however the difference lies in the speed of convergence as shown in the experimentation performed in the paper [13].

GELU was able to converge faster i.e. in less number of epochs as compared to other non linear activation functions. Hence because of these reason GELU is a preferred choice for deep transformer networks which requires an activation function to map the complicated non linearities and also does not cause vanishing gradient problems.

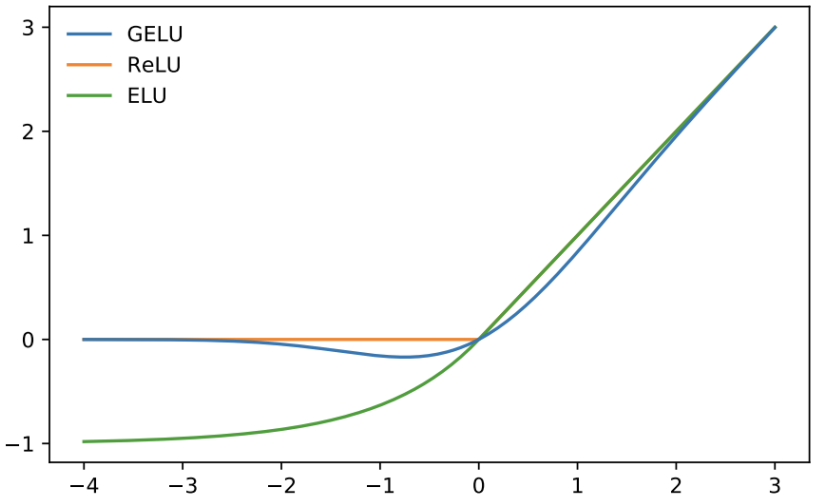

Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

Figure 3.5: graph comparing GELU, RELU and ELU

image credits: [13]

CHAPTER 4

Methodology

In this paper we propose a modification on top of CM BERT. In CM BERT two modalities are used i.e. text and audio modalities. These two modalities are used to fine tune the pre trained BERT model. In this proposed method instead of using only two modalities, all three modalities i.e. text, video and audio modalities will be used. Video modality will be following the same flow as that of audio modality with the difference being in the shape of the fused attention mask output. The shape of the fused attention mask changes in order to take into account all three modalities as the input shapes of each of modalities are different.

4.1 Problem Definition

In this case the goal is to learn the tightly coupled semantic relationship between the different modalities and utilize the learned interaction so as to better fine tune the BERT model and improve the sentiment predictions. [38] Given the input computational sequences of Xt, Xv, Xa the goal is to train the weights of 1D Convolution layer to get the fused masked attention output which will then be used to fine the BERT model for final sentiment prediction. As this is built on top of CM BERT the pre processing steps for text and audio will be the same. For a sequence of word piece tokenized text where n is the number of sequence length due to BERT pre processing steps which adds a special [CLS] token before the input sequence, the output will be of length n+1 for text sequence. To keep the shapes as close as possible for all the modalities a zero vector padding is performed before word level alignment for the

audio features where A_{CLS} is zero vector.

4.2 Fully Cross Modal BERT

The input for this proposed model consists of three parts: acoustic, visual and textual data. After word level alignment the audio and visual features will not be of the same size. In order to melt these features into the same shape 1D temporal convolution is used as:

$$\hat{X}_t, \hat{X}_v, \hat{X}_a = Conv1D(\{\hat{X}_t, \hat{X}_a, \hat{X}_v\}, k_{\{t,v,a\}}) \tag{4.2.1}$$

where $k\{t, v, a\}$ represents the size of convolution kernels for each modality i.e. k_t represents the size of convolution kernels for text, k_a for audio and k_v for video. Because the dimensionality of X_t and X_v is higher than X_a, the model will be unstable due to learning higher weights for $\hat{X}_t$ and $\hat{X}_v$ due to higher input dimensions and the value of $\hat{X}_t$ and $\hat{X}_v$ will higher that $\hat{X}_v$. In order to make sure the model learns weights which are not highly biased towards high dimensional inputs we scale the audio, text, and visual features. Another reason to do this is to prevent the issue of vanishing gradient as we are using SoftMax function. Due to the high dimension input the dot products will be extremely large which will cause the SoftMax function to give extremely small gradients and as this is back propagated through the network the effect becomes more pronounced as the network is also deep and the issue of decaying and exploding gradients [27] cause the neurons to be effectively dead and the model will not learn. In order to avoid this X_t, X_a and X_v are scaled to $\hat{X_t}'$, $\hat{X_a}'$ and $\hat{X_v}'$ respectively. This will not only make reduce the dimensionality of the vectors which would make it computationally more efficient but also avoid the problem of exploding gradients during the initial stages of training.

After getting $\hat{X_t}'$, $\hat{X_a}'$ and $\hat{X_v}'$ and X_t we can calculate the masked multimodal attention. This masked multimodal attention vector is a trained parameter which learns to adjust the attention (or weight) given to each aligned sequence of each modality. After calculating the masked multimodal attention vector, in order to maintain the shape of the input data as a pre trained BERT model is used, a residual connection for X_{att} and X_t is used [35]. It is then passed into a linear and normalization layer. Finally, the output of the final linear layer $\hat{Y_l} = L_{CLS}, L_1, L_2, L_3...L_n$ is used to predict the sentiment of the input computational sequence.

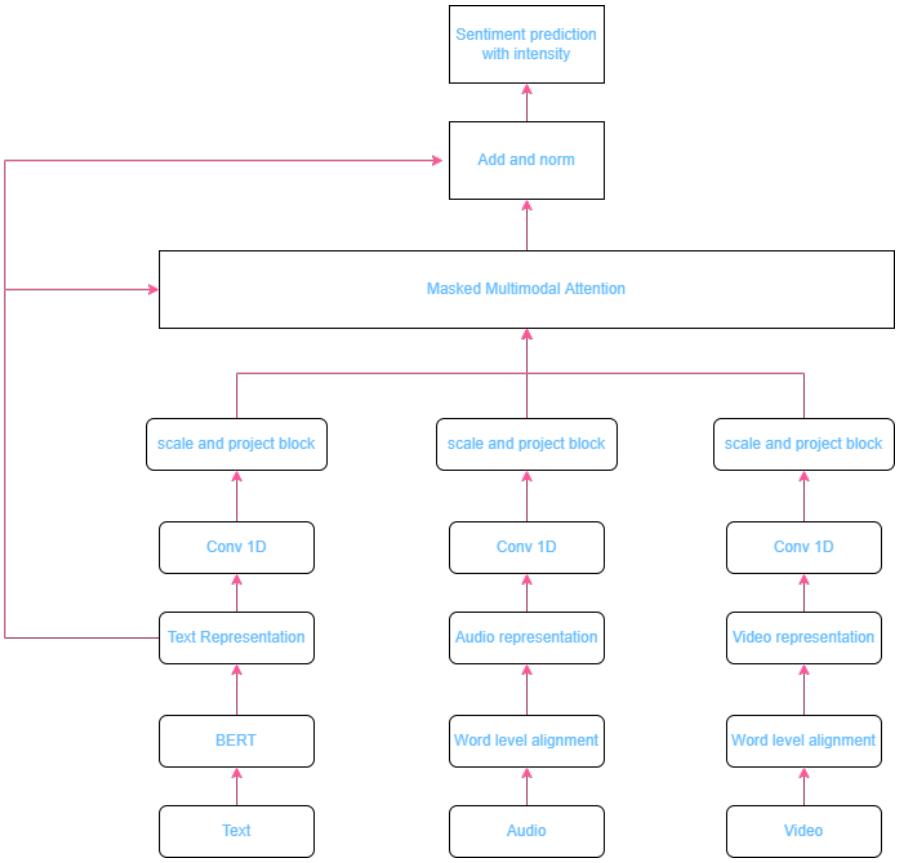

Figure 4.1: FCM-BERT block representation

Pre trained BERT model used is uncased $BERT_{BASE}$ version. The learning rate of encoder layers is 0.01 and learning rate of remaining layers to $2e^{-5}$. For training the proposed model the batch size is 20 and max sequence length is 50 with 5 epochs is used. The optimizer used is Adam with MSE (Mean Squared Error) loss function.

4.3 Masked Multimodal Attention

As stated in the [38] the masked multimodal attention (MMA) is the vector that learns, adjusts and fuses the information from multimodal sequences. It is used by the model to adjust the attention weights given to the words in the sequence and eventually fine tune the BERT model. The attention weight for each sequence in each modality is learned during the training. Similar to BERT where the text attention weights are learned using a Query and Key matrix a similar approach is used here too; for each modality a corresponding Query and Key matrix is used to learn the attention weights. The Query Q_t and the key K_t for text modality is defined as $Q_t = K_t = \hat{X}_t'$ where $\hat{X}_t'$ is the scaled text vector. Similarly the Query Q_a and the

key K_a for audio modality is defined as $Q_a = K_a = \hat{X}'_a$ where $\hat{X}'_a$ is the scaled audio vector and Query Q_v and the key K_v for visual modality is defined as $Q_v = K_v = \hat{X}'_v$ where $\hat{X}'_v$ is the scaled visual vector. The attention matrix for text, audio and video features are calculated as follows:

$$\alpha_t = Relu(Q_t K_t^T) \tag{4.3.1}$$

$$\beta_t = Relu(Q_a K_a^T) \tag{4.3.2}$$

$$\gamma_t = Relu(Q_v K_v^T) \tag{4.3.3}$$

For the model to learn and adjust the attention weights dynamically during training for the combined multimodal features the weighted fusion attention matrix is given as:

$$W_f = \omega_t * \alpha_t + \omega_a * \beta_a + \omega_v * \gamma_v + b \tag{4.3.4}$$

Where ω_t, ω_a, ω_v represents the weights of text, audio and visual modalities respectively and b is the bias. An Attention Mask M is used which represents the padded positions with -∞ (or -10000.0 in code) and non padded i.e. token positions with 0. This is done so as to remove the effect of vector padding used to get the features in shape. By using a huge value like –10000 the value will be effectively zero after passing through SoftMax activation. The multimodal attention matrix after attention masking will be $W_m = Softmax(W_f + M)$. After calculating the multimodal attention matrix, it is multiplied with a masked attention matrix V_m from Pre Trained BERT i.e. the final encoder layer representation of text modality from BERT. Therefore, the masked multimodal attention matrix X_{Att} is given as:

$$X_{Att} = W_m V_m \tag{4.3.5}$$

where $V_m = BERT final Encoder Layer(X_t)$

CHAPTER 5

Experimental Methodology

This section discusses the hardware and compute resources required, dataset used, the preprocessing done on the dataset with respect to word level alignment, then the evaluation metrics used

5.1 Experimental setup

The warmup proportion was set to 0.1, batch size to 20, learning rate to $2e^{-}5$ and the experiment was run 5 times each having a randomized seed and an epoch size of 5. The entire experiment was run on a system with i7 – 7700 HQ clocked at 2.8 GHz with a single core turbo boost frequency of 3.3 GHz and 24 GB RAM. The GPU was a GTX 1050ti with 640 CUDA cores and 4GB VRAM. There were significant issues with batch size due to the lack of adequate VRAM and hence the batch size was limited to a maximum of 20. Further experiments need to be carried out with a larger batch sizes as bigger batch sizes usually result in better generalization performance up to a point where larger batch sizes may decrease the performance [20]. However in this case batch sizes need to be increased to expect better performance.

5.2 Dataset

In this experiment the dataset used is CMU MOSI (CMU Multimodal Opinion-level Sentiment Intensity) [41]. It is a dataset mainly made to deal with opinion mining and summarization. Unlike other multi modal datasets, CMU MOSI tackles three things head on

1. Opinion volatility in videos where a speaker may switch between different topics and opinions.

2. Labelling not only the sentiment but also the intensity of the sentiment i.e. the labels not only contain where a certain sequence is positive or negative sentiment but it also contains a number which shows how strong or weak a particular sentiment is.

3. It is made specifically for multi modal analysis. It tries to answer the question of using more than one modality for the prediction of the sentiment. Every day human communication occurs in a multimodal aspect as speech carries the words which hold the explicit sentiments however the facial expressions also carry the implicit sentiments too.

The data-set has the following aspects:

1. multi modal observations i.e. transcribed speech and visual features

2. opinion level sequences i.e. each sequence is segmented to express only one sentiment along with the intensity of the sentiment

3. alignment between the textual, acoustic and visual features.

The dataset contains 93 movie review videos from YouTube. The videos are segmented into 2199 sequences where each sequence is annotated by 5 different humans. Each annotation is a continuous number between –3 to +3 where –3 is highly negative and +3 is highly positive. The dataset is split into train, test and validation set keeping in mind that the same speaker does not appear in the different splits and the balance of positive and negative reviews are kept in balance. The train, test and validation set consists of 52, 31 and 10 respectively with 1284, 686 and 229 sequences in each split.

5.3 Alignment of Features

COVAREP [8] which is an open-source collection of reproducible speech feature extraction tool is used to extract the audio features. Each audio feature is a 74-dimension vector which includes the following:

1. 12 Mel-frequency cepstral coefficients (MFCCs)
2. pitch and segmenting features
3. glottal source parameters
4. peak slope parameters
5. maxima dispersion quotient

To align the words with acoustic features P2FA (Penn Phonetics Lab Forced Aligner) is used which is a forced alignment toolkit. This requires audio to be sampled at either 8, 11.25 or 16 KHz along with a transcript text file. The P2FA then aligns the words with the phonemes using a word – phonemes dictionary provided in P2FA. After the alignment in order to keep the sequence length of text and audio modalities consistent zero vectors are padded to audio sequences. The video and audio alignment are straightforward as the interval for each word is obtained using P2FA and it can be directly aligned with corresponding video frames. Thus audio – word then word interval – video frame alignment leads to audio – video – word level alignment. This process is applied to the dataset and the post processed dataset is hosted by Carnegie Mellon University at this link.

5.4 Evaluation Metrics

To compare with model developed by Tsai et al [33], the same evaluation metrics are used, they are:

1. 7 class accuracy (Acc7)
2. 2 class accuracy (Acc2)
3. F1 score

4. MAE (Mean Absolute Error)

5. Corr (Correlation)

7 class accuracy is used for sentiment classification task. F1 score and 2 class accuracy is used for binary sentiment classification. Mean Absolute Error and Correlation values are used in regression task.

A higher value of metrics implies a better performing model. To reduce the chances of by chance or non-reproducible experiments, the experiment is run five times with five random seeds and the average of 5 results is taken into consideration for the final score.

5.5 Baselines

Fully cross modal BERT (FCM-BERT) is compared with previous models which deal with multi modal sentiment analysis tasks.

EF-LSTM A single LSTM is used by Early Fusion LSTM (EF-LSTM) to combine multimodal inputs to acquire contextual information.

LMF [19] A technique called low-rank multimodal fusion (LMF) uses low-rank weight tensors to improve the performance of multimodal fusion without sacrificing performance. Efficiency is greatly enhanced while computational complexity is drastically reduced.

MARN [40] To identify the interactions between various modalities, the Multi-attention Recurrent Network (MARN) makes use of the Multi-attention Block and the Long-Short Term Hybrid Memory.

RMFN [18] Recurrent Multistage Fusion Network (RMFN) models temporal and intra-modal interactions by fusing the multistage fusion process with recurrent neural networks.

MFM [34] Multimodal Factorization Model can explicitly break the multimodal embeddings into multimodal discriminative and generative factors which can help each factor to learn from the subset of fused information across the multimodal data

MCTN [22] Multimodal Cyclic Translation Network (MCTN) is developed to learn fused representations by taking into account the multi modal data. However, it only uses text modality in the testing process but is still able to give state of the art results

MulT [33] The previous state-of-the-art approach on the MOSI dataset is called Multimodal Transformer (MulT), which employs directional cross modal attention to map the

relationships between multimodal sequences across different time steps.

T-BERT [9] Bidirectional Encoder Representations from Transformers (BERT), which is fine-tuned only using text modality information.

CM-BERT [38] The current State-of-the-art multimodal sentiment analysis model which uses two modalities i.e audio and text.

Fully Cross Modal BERT (FCM-BERT) The proposed model which takes in all three modalities.

CHAPTER 6

Comparisons with other models

FCM BERT is evaluated on CMU MOSI data-set and when compared with other models which take in all three modalities, the proposed model does perform better in most metrics of evaluation. The lowest scoring and most scoring fully multi modal model scores are taken into consideration for percentage comparison. In binary sentiment classification Acc_2^h FCM-BERT achieves 83.1% accuracy which is 6.7% to 0.1% better than previous baseline models. Similar to Acc_2^h the model achieves 7.3% to 0.2% improvement in F1 scores. In sentiment score classification task FCM-BERT improvements are evident as it achieves a score of 44.2% which is 11.4% to 4.2% better than previous models. However, in regression tasks the improvements are not observed. For example, FCM-BERT reduces about 0.16 to 0.119 on MAE_l scores but improves on $Corr_h$ scores by 0.108 to 0.078. The experimental results demonstrate that the MulT model performs significantly better than the other baselines. The MulT takes in the multimodal data and uses the transformer to dynamically learn and assign weights across modalities via attention mechanism. But when comparing the T-BERT model with the MulT model, the latter can achieve superior results through better learned representations of multi modal data by fine tuning the BERT model. The same is true for FCM BERT as BERT has been extended to use audio, visual and textual modalities.

Experimental results achieved on CMU-MOSI dataset. This model results are highlighted in bold.$_h$ means higher score is better and $_l$ means lower score is better. T:text,A:audio,V:video.

Model	Modality	Acc_7^h	Acc_2^h	F_1^h	MAE^l	$Corr^h$
LMF	T+A+V	32.8	76.4	75.7	0.912	0.668
EF-LSTM	T+A+V	33.7	75.3	75.2	1.023	0.608
MARN	T+A+V	34.7	77.1	77.0	0.968	0.625
MCTN	T+A+V	35.6	79.3	79.1	0.909	0.676
MFM	T+A+V	36.2	78.1	78.1	0.951	0.662
RMFN	T+A+V	38.3	78.4	78.0	0.922	0.681
MulT	T+A+V	40.0	83.0	82.8	0.871	0.698
T-BERT	T	41.5	83.2	83.2	0.784	0.774
CM-BERT	T+A	44.9	84.5	84.5	0.729	0.791
FCM-BERT	**T+A+V**	**44.2**	**83.1**	**83.0**	**0.752**	**0.776**

Table 6.1: Tables showing experimental results of FCM-BERT. Previous results taken from [38]

CHAPTER 7

Future Work and improvements

Although this model gives near state-of-the-art results in fully multi modal inputs there is still scope for further improvements. First and foremost, would be to increase the performance of the model predictions in both classification and regression tasks. This can be done in multiple ways but the widely used approach will be hyper-parameter tuning. Then the next approach would be to use Vision Transformers or BERT like deep vision models to get the vision embeddings and then follow the same process as that done for text modality including skip connections, scaling and projection operations.

The second task would be to look into memory reduction using smaller back bone models which would not only decrease the memory footprint but also improve the overall performance of the model. This would make such multi modal models ideal for real time applications in surveillance. Reduced compute requirements also mean higher cost savings for businesses running such models in the real world. Not only does this increase the availability of model but also makes it accessible to smaller businesses with limited budgets. The third task would be to make the model run on real time data streams which would require a high level of performance and memory optimisations to enable it to run on edge devices. This would further help in achieving the second point.

CHAPTER 8

Conclusions

In this paper an improvement over CM BERT is implemented which shows promising results on fully multi modal inputs which takes into account all three modalities i.e. textual, visual and acoustic data. One can also say that this is an extension of CM BERT to take in all three modalities of input. The masked multi modal attention is now able to utilize all three modalities and dynamically adjust the attention weights for the words by learning the tightly coupled mappings between the text, audio and video inputs. The experiment results show that FCM-BERT performs much better than previous baseline models which take in all three modalities and has shown to improve performance on CMU MOSI dataset.

CHAPTER 9

Appendix

The code for this project can be found on the following github repo here.

Bibliography

[1] M. Arjmand, M. J. Dousti, and H. Moradi. Teasel: A transformer-based speech-prefixed language model. *arXiv preprint arXiv:2109.05522*, 2021.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] D. Britz, A. Goldie, M.-T. Luong, and Q. Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[5] L. Cai, Y. Hu, J. Dong, and S. Zhou. Audio-textual emotion recognition based on improved neural networks. *Mathematical Problems in Engineering*, 2019, 2019.

[6] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, and T. Robinson. One billion word benchmark for measuring progress in statistical language modeling. *arXiv preprint arXiv:1312.3005*, 2013.

[7] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[8] G. Degottex, J. Kane, T. Drugman, T. Raitio, and S. Scherer. Covarep—a collaborative voice analysis repository for speech technologies. In *2014 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 960–964. IEEE, 2014.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] S. Dobrišek, R. Gajšek, F. Mihelič, N. Pavešić, and V. Štruc. Towards efficient multi-modal emotion recognition. *International Journal of Advanced Robotic Systems*, 10(1):53, 2013.

[11] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[12] S. Gao, X. Chen, C. Liu, L. Liu, D. Zhao, and R. Yan. Learning to respond with stickers: A framework of unifying multi-modality in multi-turn dialog. In *Proceedings of the Web Conference 2020*, pages 1138–1148, 2020.

[13] D. Hendrycks and K. Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

[14] T. Henighan, J. Kaplan, M. Katz, M. Chen, C. Hesse, J. Jackson, H. Jun, T. B. Brown, P. Dhariwal, S. Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[16] O. Kuchaiev and B. Ginsburg. Factorization tricks for lstm networks. *arXiv preprint arXiv:1703.10722*, 2017.

[17] Z. Lian, J. Tao, B. Liu, and J. Huang. Domain adversarial learning for emotion recognition. *arXiv preprint arXiv:1910.13807*, 2019.

[18] P. P. Liang, Z. Liu, A. Zadeh, and L.-P. Morency. Multimodal language analysis with recurrent multistage fusion. *arXiv preprint arXiv:1808.03920*, 2018.

[19] Z. Liu, Y. Shen, V. B. Lakshminarasimhan, P. P. Liang, A. Zadeh, and L.-P. Morency. Efficient low-rank multimodal fusion with modality-specific factors. *arXiv preprint arXiv:1806.00064*, 2018.

[20] S. McCandlish, J. Kaplan, D. Amodei, and O. D. Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

[21] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[22] H. Pham, P. P. Liang, T. Manzini, L. Morency, and B. Póczos. Found in translation: Learning robust joint representations by cyclic translations between modalities. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6892–6899, 2019.

[23] S. Poria, E. Cambria, D. Hazarika, N. Majumder, A. Zadeh, and L.-P. Morency. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 873–883, 2017.

[24] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[26] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[27] M. Roodschild, J. Gotay Sardiñas, and A. Will. A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, 9(4):351–360, 2020.

[28] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[29] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. Le, G. Hinton, and J. Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017.

[30] C. Sun, L. Huang, and X. Qiu. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.

[31] Z. Sun, P. Sarma, W. Sethares, and Y. Liang. Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8992–8999, 2020.

[32] E. Tong, A. Zadeh, C. Jones, and L.-P. Morency. Combating human trafficking with deep multimodal models. *arXiv preprint arXiv:1705.02735*, 2017.

[33] Y.-H. H. Tsai, S. Bai, P. P. Liang, J. Z. Kolter, L.-P. Morency, and R. Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2019, page 6558. NIH Public Access, 2019.

[34] Y.-H. H. Tsai, P. P. Liang, A. Zadeh, L.-P. Morency, and R. Salakhutdinov. Learning factorized multimodal representations. *arXiv preprint arXiv:1806.06176*, 2018.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[36] H. Wang, A. Meghawat, L.-P. Morency, and E. P. Xing. Select-additive learning: Improving cross-individual generalization in multimodal sentiment analysis. *arXiv preprint arXiv:1609.05244*, 2016.

[37] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[38] K. Yang, H. Xu, and K. Gao. Cm-bert: Cross-modal bert for text-audio sentiment analysis. In *Proceedings of the 28th ACM international conference on multimedia*, pages 521–528, 2020.

[39] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L.-P. Morency. Memory fusion network for multi-view sequential learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[40] A. Zadeh, P. P. Liang, S. Poria, P. Vij, E. Cambria, and L.-P. Morency. Multi-attention recurrent network for human communication comprehension. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[41] A. Zadeh, R. Zellers, E. Pincus, and L. Morency. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems*, 31(6):82–88, 2016.

[42] R. Zhong, D. Ghosh, D. Klein, and J. Steinhardt. Are larger pretrained language models uniformly better? comparing performance at the instance level. *arXiv preprint arXiv:2105.06020*, 2021.

[43] S. Zhou, J. Jia, Q. Wang, Y. Dong, Y. Yin, and K. Lei. Inferring emotion from conversational voice data: A semi-supervised multi-path generative neural network approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[44] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.