**Selection of pre-trained models for dimensional emotion theory parameter regression tasks**

The purpose of this section is to narrow the scope of potential neural network architectures for assessment for a regression task to generate dimensional emotional theory parameters. This task has been broken down into two phases:

*Phase 1:* Identification of architectures commonly used in relevant literature and critical evaluation of their features with regards to the specific task.

*Phase 2:* Using the most suitable identified architecture from phase 1, assess several pre-trained models which fall into this category for the regression task.

*Evaluation of common architectures in literature for sentiment regression task*

Sentiment is fundamentally linked to context and thus models which successfully capture sentiment are likely to contain characteristics which enable the capture of long-term dependencies. Neural network architectures are widely utilised for such purposes. Their suitability is largely attributable to aspects of their architecture such as their:

*Ability to handle sequential data*: As language is sequential in nature, sentiment is often linked to word order. As neural networks handle data sequentially, their outcomes are often improved compared to more simplistic models.

*Robustness against noise:* Language is inherently noisy and variable due to variance in vernacular across a population in addition to relatively frequent errors in spelling or grammar. Such an effect is known to be more prominent in online content, where such features are sometimes used to add nuance to pragmatics. In the context of a problem set where there is potential for such features to add insight, this feature of such architectures may possess limitations if this information cannot be extracted in another manner.

Several models which would fall under this category are prominently found throughout literature evaluating sentiment analysis methodologies and specifically in the domain of sarcasm detection due to these features. The three primary models identified were as follows:

LSTM: The LSTM mechanism of selective memory is made up of a cell state, hidden state, and gates. The NN models sequential data by propagating over time through the connection of sequential events using the hidden state. This component captures dependencies by considering both the previous step and current output:

However, a feature of such a mechanism result in all previous steps being considered in the current step when implemented in isolation due to the chain rule:

Because:

With the ultimate result being either vanishing or exploding gradients and thus limitations on the abilities to capture long term dependencies in isolation. The cell state mitigates this effect through the filtering of less relevant information from the memory through the forget gate. This information, in addition to the weights from the input gate enable the model to learn which time steps contain important information, resulting in weights for each time step being represented in proportion to their understood importance to the model. Literature evaluating this architecture is not consistent with regards to its assessment of the efficacy of such a mechanism; with some studies citing the model as effective to capture long-term dependencies, and others postulating that the mechanism may ‘dilute’ important information over time. Similar contrasting observations are found in the sentiment across literature with regards to their robustness against noise, with the former asserting that noise information is filtered efficiently during training and the latter arguing the converse and observing amplification of noise. In this context, what is traditionally regarded as noise may provide pragmatic cues with respect to potential sarcasm as discussed above, these cues may be indicative that context of the problem set plays a part in the efficacy of the architecture to model valid patterns in the data, which explains contrasting observations within literature. No works could be identified which assessed this hypothesis, however this may be an area for future research.

GRU: GRU models address vanishing and exploding gradients using a more simplistic mechanism than LSTM models. These models omit the cell state and regulate memory using gates. These architectures utilise an update gate to dictate the information which is retained from the previous step in series with a reset gate which dictates the information which should be eliminated. Compared to LSTMs for natural language processing tasks, due to their more simplistic architecture GRU models seem to perform better where shorter data sequences are used, possibly providing greater potential in the context of the short form content as is used for the problem set in question.

Present state-of-the-art for sentiment analysis also includes significant volume of models which contain an alternative mechanism for memory. Transformer-based models leverage self-attention mechanisms to capture dependencies of long and short ranges.

These models are based upon encoder-decoder architectures, which are capable of processing data in parallel due to their attention mechanism which avoids processing data in parallel in favour of processing the sequences as a whole. The encoder consists of several layers, each containing two sublayers. The first sublayer generates self-attention and the second consists of a fully connected feed-forward network with two linear transformations and Rectified Linear Units activation:

Where each layer uses its own weights and bias parameters. Given there is no inclusion of recurrence, there is no embedded manner to consider the relative position of words. To address this, positional encodings are added to the word embeddings. The decoder consists of several layers, which are each composed of three sublayers: the first decodes the previous input to extract positional information and apply attention. The attention in decoders is distinguished from that found in the encoder cells as they do not consider all words, but rather only words which have occurred before the current. The second layer contains a self-attention mechanism which receives information from the previous sublayers of decoders and the encoders output keys and values. The final decoder sublayer consists of a fully connected feed-forward neural network, like that of the second sublayer in the encoder cells.

The attention layers operate by passing each word in the sequence through the embedding and positional encoding layers to generate their respective vectors. The result is passed into the encoder where it is first processed by the attention module. The sequence is passed through three separate layers which each produce a matrix. These layers consist of query which defines the word for which the attention is to be calculated and key and value are compared to the query with regards to their relevance. These transformations are trainable operations which are adjusted to produce the desired output predictions over the course of training, quantified by the attention score, defined as the dot product between the query matrix and a transpose of the key matrix:

An intermediate matrix is produced, consisting of a multiplication between all combinations of the words in the respective matrices. A second dot product calculation is performed between the intermediate matrix and the value matrix to produce the attention score.

A diagram of a number

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated

A diagram of a graph

Description automatically generated

Given such a mechanism which learns based on similarities and differences in the input words, the architecture is aligned with the previous methodology for word vector transformation which yielded improved results compared to the original word vectors used in the basic theory regression task. Additionally, the lack of necessity for labelled data is advantageous adaptations of the strategy implemented for the basic theory regression could not yield acceptable levels of accuracy. The key characteristics of the training data which may have contributed to this result may be the small size of the training set, and the potential for complexity in the relationships between the emoji and labels, which could not be captured by less complex algorithms, more appropriate for the available training data. While traditional neural networks necessitate large, labelled datasets, the converse is true for transformer-based models which learn based on patterns between elements in the dataset, eliminating the need for such a resource intensive annotation process.

**Optimisation Process for Sentiment-Aware Vector Space Modification Model**

**Architectures of Considered Emoji Sentiment Prediction Models**

**Survey Results**