Tali Aharon 034791236
Helit Bauberg 027466002

Natural Language Processing - 99101 – Final Project

Emotion Detection – Artificial Neural Networks Performance Comparison

Project Re-Scoping

Our initial hypothesis suggested that an ANN architecture that imitates the human brain's functionality, such that [Right hemisphere point of view] + [Left hemisphere point of view] are combined into a single balanced point of view, should provide better classification compared to any single ANN.
As per this hypothesis, we started out by researching what type of networks would simulate the Right side and the Left side of the brain, and how best we combine these into one architecture.
Unfortunately, further research as well as test runs yielded that our initial concept does not improve the model, as the models we selected to imitate 'left hemisphere' showed much lower accuracy than the 'right hemisphere' ones. We concluded that there's no benefit in combining the two 'hemispheres' together into one 'bilateral' model.
Moreover, our research suggested that the biggest impact on the overall performance and accuracy is the type of network we use.
We therefore decided to change the scope of our project, and experiment with simulating 'right hemisphere' capabilities using different types of ANNs and configurations.
We evaluated our models performance and accuracy using the GoEmotions dataset classification.

Emotions Text Classification – Overview

Emotions are a key aspect of social interactions, influencing the way people behave and shaping relationships.
This is especially true with language — with only a few words, we're able to express a wide variety of subtle and complex emotions.
As such, it's been a long-term goal among the NLP research community to enable machines to understand context and emotion, which would, in turn, enable a variety of applications, including empathetic chatbots, models to detect harmful online behavior, and improved customer support interactions. (From GoEmotion's site)
'The two critical areas of natural language processing are sentiment analysis and emotion recognition. Even though these two names are sometimes used interchangeably, they differ in a few respects. Sentiment analysis is a means of assessing if data is positive, negative, or neutral.
In contrast, Emotion detection is a means of identifying distinct human emotion types such as furious, cheerful, or depressed. "Emotion detection," "affective computing," "emotion analysis," and "emotion identification" are all phrases that are sometimes used interchangeably.'
(Affect, feeling, emotion, sentiment, and opinion detection in text - Munezero et al. 2014).

Emotions Text Classification - Challenges:

- The need for an emotion dictionary to contain a reasonable number of emotion categories, since limited keywords can greatly affect the performance of the approach among ambiguity of keywords and the lack of linguistic information.

- Irony and sarcasm: In sarcastic text, people express their negative emotions using positive words. This fact allows sarcasm to easily cheat emotion classification models unless they're specifically designed to take its possibility into account.
- Types of negations: The simplest approach for dealing with negation in a sentence, which is used in most state-of-the-art emotion analysis techniques, is marking as negated all the words from a negation cue to the next punctuation token.
- Word ambiguity: The problem of word ambiguity is the impossibility to define polarity in advance because the polarity for some words is strongly dependent on the sentence context.
- The expression of multiple emotions in a single sentence. It is difficult to determine various aspects and their corresponding sentiments or emotions from the multi-opinionated sentence.

(https://www.toptal.com/deep-learning/4-sentiment-analysis-accuracy-traps)

Experiments - framework:

Right Hemisphere Characteristics – mapping to ANN type/configuration:

Right Hemisphere characteristics	Mapping to ANN characteristics
Wide beam attention	BERT – multiple attention heads, long input size; LSTM, bi-LSTM - capable of learning long-term dependencies.
Understands complexity	Pre-trained, large corpus, multiple epochs
Understands context, **implicit meaning**	BERT – multiple attention heads, long input size; LSTM, bi-LSTM - LSTM can derive context from previous words in a sentence. Here bi-LSTM is better since it has context understanding from the following words as well. **multiclass emotion classification**
Wisdom about the whole and world knowledge	Pre-trained, large corpus
Indirect communication (body, intents)	**multiclass emotion classification**

Hypothesis:

Initial hypothesis:

A synthesis of two ANN networks that are trained to process data in different ways will provide better classification accuracy compared to a single network that does the same:

- Right ANN properties - multiclass classification, vs.
- Left ANN properties - binary classification

Reference: https://arxiv.org/abs/2209.06862

Revised hypothesis:

The initial results (See table on “Process” section below) proved that ANNs such as simple ANN/LSTM/Bidirectional LSTM do not provide good enough accuracies and will not contribute anything to the model.
BERT model (which simulates right hemisphere capabilities) has a very high accuracy and would be a great base for our model to grow from.
We now wish to explore different BERT models with or without additional layers and find **one ANN model** that **integrates** between **Context and world knowledge.**
We therefore decided to re-scope our project by trying to simulate right hemisphere capabilities using different types of layers on top of BERT and DistilBert.
Our new hypothesis -
Fine-tuning Pre-trained BERT with additional layers would yield the best accuracy on textual emotion classification.
We used GoEmotions to check our hypothesis.

Datasets:

IMDB Dataset:

- Overall Dataset Size- 50,000 Reviews
- Training Data- 25,000 Reviews labeled with **Binary Sentiments (Pos/ Neg)**
- Test Data- 25,000 Reviews
- Reviews are preprocessed and indexed by overall frequency. Eg: Index 3 indicates the 3rd most frequently used word in that data. As a convention, index 0 is used to encode any unknown/ unidentified word.

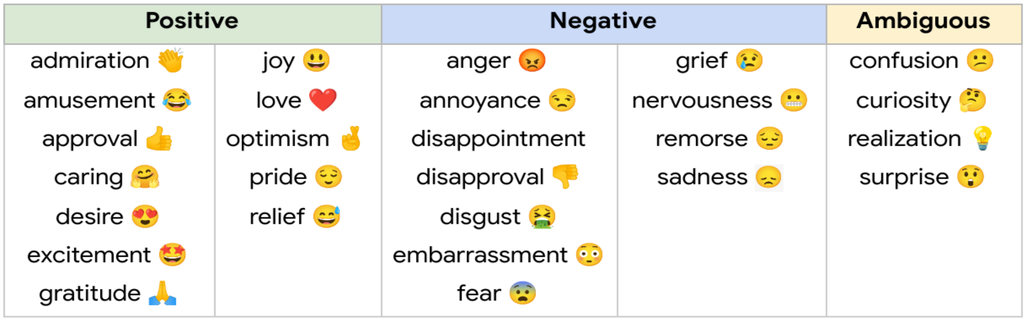
GoEmotions Dataset:

- A corpus of 58k carefully curated comments extracted from Reddit, with human annotations to **27 emotion categories + Neutral** (The Neutral category makes up 26% of all emotion labels)
- Number of examples: 58,009.
- Maximum sequence length in training and evaluation datasets: 30.
- Most of the examples (83%) have a single emotion label and have at least two raters agreeing on a single label (94%).

Labels and Distribution:

*Adopted from GoEmotions official site

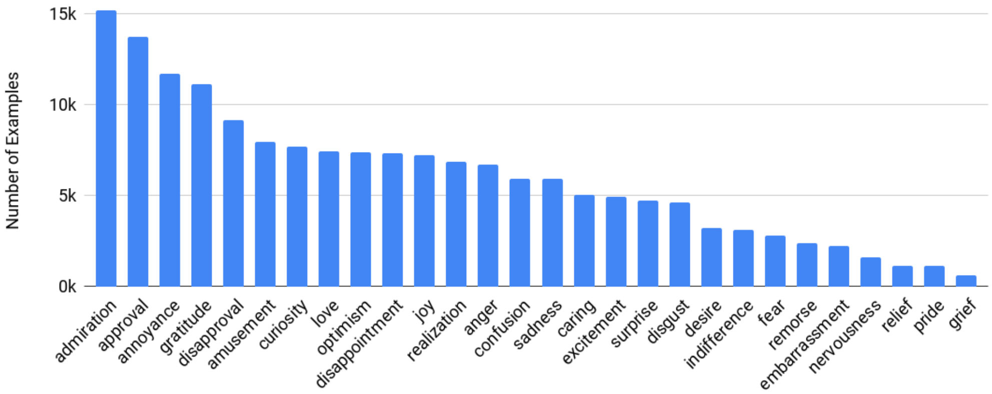
The published GoEmotions dataset includes the taxonomy presented below, and was fully collected through a final round of data labeling where both the taxonomy and rating standards were pre-defined and fixed.



Positive		Negative		Ambiguous
admiration 👏	joy 😃	anger 😡	grief 😢	confusion 😕
amusement 😂	love ❤️	annoyance 😒	nervousness 😬	curiosity 🤔
approval 👍	optimism 🤞	disappointment	remorse 😔	realization 💡
caring 🤗	pride 😌	disapproval 👎	sadness 😞	surprise 😲
desire 😍	relief 😅	disgust 🤮		
excitement 🤩		embarrassment 😳		
gratitude 🙏		fear 😨		

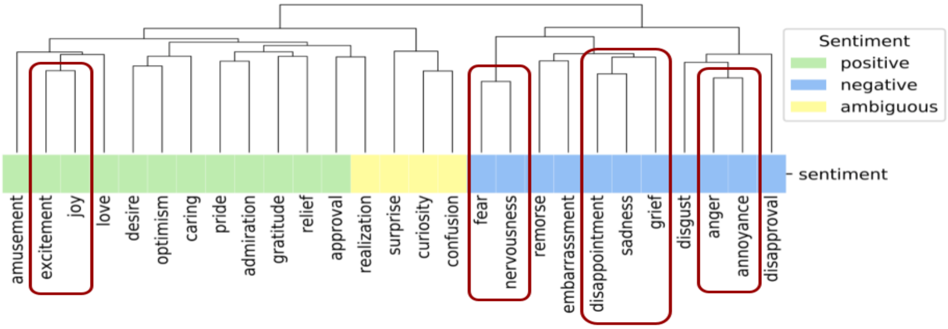
GoEmotions taxonomy: Includes 27 emotion categories, plus "neutral".

Emotions are not distributed uniformly in the GoEmotions dataset.
Importantly, the high frequency of positive emotions reinforces the motivation for a more diverse emotion taxonomy than that offered by the canonical six basic emotions.

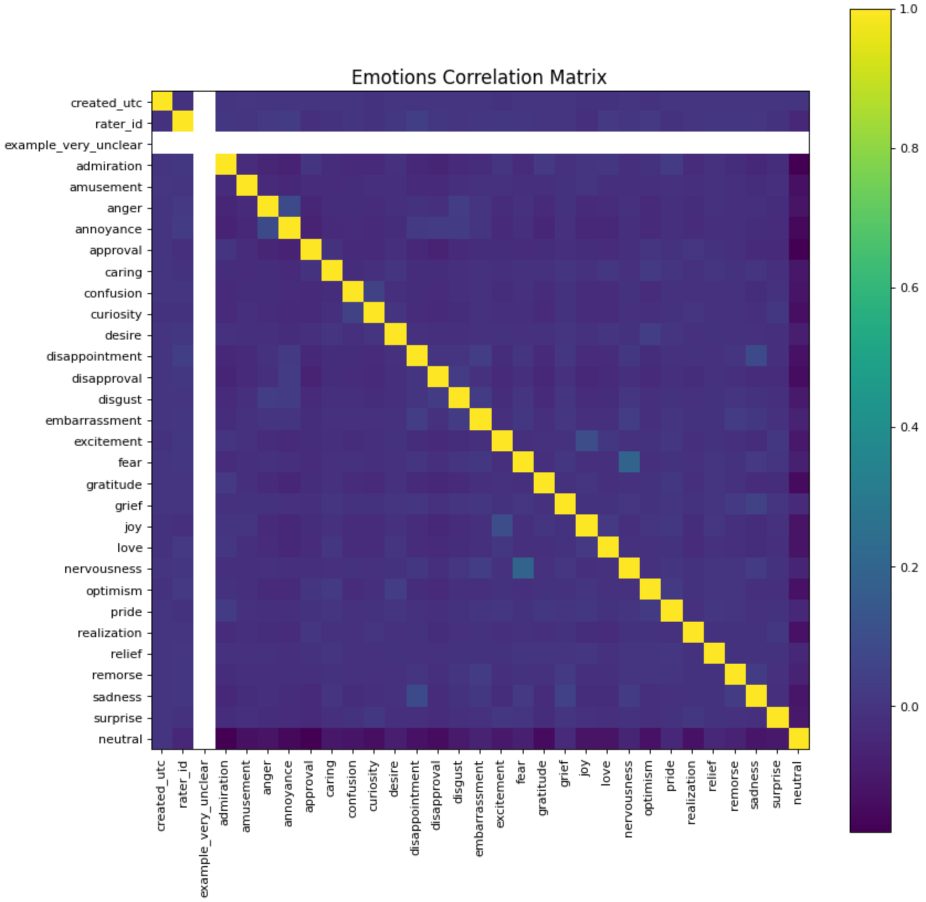


Correlations between Emotions:

Related emotions (Eg. joy and excitement) are closely correlated:



In our work, we saw the same emotions correlate strongly on a full HeatMap:



GoEmotions correlation between 28 emotion categories, including "neutral"

(please Zoom in to view properly)

Methods:

ANN Type Selection:

There are numerous architectures for Artificial Neural Networks.
We chose to experiment with two basic architectures:
1. (bi) Long short-term memory (LSTM) Recurrent Neural Network (RNN)
2. Large Language Model (LLM) - Bidirectional Encoder Representations from Transformers (BERT)

The reasoning behind selecting those architectures is that both are well known for NLP tasks, and both handle sequential data within context – i.e., possess memory capabilities.
Since we would like to experiment with classification of emotions from text, we require wide context, complex and implicit meaning extraction, i.e. we require an architecture that remembers.

Recurrent Neural Network (RNN)

RNN is a type of artificial neural network which uses sequential data or time series data. These deep learning algorithms are commonly used for ordinal or temporal problems, such as language translation, natural language processing (NLP), speech recognition, and image captioning; they are incorporated into popular applications such as Siri, voice search, and Google Translate. Like feedforward and convolutional neural networks (CNNs), recurrent neural networks utilize training data to learn. They are distinguished by their "memory" as they take information from prior inputs to influence the current input and output. While traditional deep neural networks assume that inputs and outputs are independent of each other, the output of recurrent neural networks depend on the prior elements within the sequence. While future events would also be helpful in determining the output of a given sequence, unidirectional recurrent neural networks cannot account for these events in their predictions.

LSTM RNN works to address the problem of long-term dependencies. That is, if the previous state that is influencing the current prediction is not in the recent past, the RNN model may not be able to accurately predict the current state. As an example, let's say we wanted to predict the italicized words in the following, "Alice is allergic to nuts. She can't eat peanut butter." The context of a nut allergy can help us anticipate that the food that cannot be eaten contains nuts. However, if that context was a few sentences prior, then it would make it difficult, or even impossible, for the RNN to connect the information. To remedy this, LSTMs have "cells" in the hidden layers of the neural network, which have three gates–an input gate, an output gate, and a forget gate. These gates control the flow of information which is needed to predict the output in the network.

Bidirectional RNNs pull in future data to improve the accuracy of it. If we return to the example of "feeling under the weather" earlier in this article, the model can better predict that the second word in that phrase is "under" if it knew that the last word in the sequence is "weather."

https://www.ibm.com/topics/recurrent-neural-networks

For example, in the sentence "Apple is something that …", the word Apple might be about the apple as fruit or about the company Apple. The traditional LSTM won't be able to know what Apple means, since it doesn't know the context from the future.

In contrast, most likely in the following two sentences:

"Apple is something that competitors simply cannot reproduce."

and

"Apple is something that I like to eat." ,

Bidirectional LSTM can do a great job distinguishing apple the fruit from Apple the tech company, utilizing the information from its future context.

https://dagshub.com/blog/rnn-lstm-bidirectional-lstm/

BERT

Bidirectional Encoder Representations from Transformers is a family of language models introduced in 2018 by researchers at Google.

Transformers have increasingly become the model of choice for natural language processing. Many modern large language models such as ChatGPT, GPT-4, and BERT use it.

The most obvious difference between ChatGPT and BERT is their architecture. ChatGPT is an autoregressive model, while BERT is **bidirectional**. While ChatGPT only considers the left context when making predictions, BERT considers both left and right context. This makes BERT better suited for tasks such as sentiment analysis or NLU, where understanding the full context of a sentence or phrase is essential.

Process:

Our Notebooks:

(we are adding a link to the drive as these are huge even when zipped):
Notebooks on IMDB:
https://drive.google.com/drive/folders/1-7guEDuPGgccpVdIxFzlkkqSlfvV-FjA?usp=sharing
Notebooks on GoEmotions:
https://drive.google.com/drive/folders/1dTE9zbd05k_ezkHI_mznggYQXQHUhN-U?usp=sharing

(A) Initial Hypothesis:

Network Type	Dataset	Classification	Best Accuracy
*simple ANN	imdb	binary	86.57%
LSTM	imdb	binary	82.27%
*LSTM	goEmotions	3 – positive, negative, neutral	62.29%
*biLSTM	goEmotions	3 – positive, negative, neutral	66.19%
BERT	imdb	binary	94.2%
***** BERT**	**goEmotions**	**multiclass-28**	**96.21%**

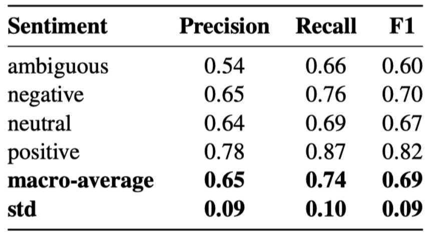
*Disclaimer: accuracy taken from https://www.rakshitraj.com/binary-classification-of-imdb-movie-reviews/
** Disclaimer: Notebooks for this initial investigation of LSTM/BiLSTM standalone performance were taken from https://github.com/keya-desai/GoEmotions-classification
*** Running FULL (non-distilled version) Pretrained BERT on goEmotions yielded the best results yet the model is huge and we ran out of GPU after one training epoch on only 0.1% of the data. We then resulted in various distilBERT pre-trained models.
DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

(B) Revised Hypothesis:

Our Base model: distilBERT pre-trained
Dataset: GoEmotions
Classification: Multiclass (27 + 1 neutral emotions)

Benchmark - https://ai.googleblog.com/2021/10/goemotions-dataset-for-fine-grained.html

Best model (: Pretrained BERT + 1 x dense layer + 1 x sigmoid activation layer.
Best results:



Sentiment	Precision	Recall	F1
ambiguous	0.54	0.66	0.60
negative	0.65	0.76	0.70
neutral	0.64	0.69	0.67
positive	0.78	0.87	0.82
macro-average	**0.65**	**0.74**	**0.69**
std	**0.09**	**0.10**	**0.09**

Table 5: Results based on sentiment-grouped data.

Transfer Learning - BERT Architecture for Pre Trained Model and Fine tuning:

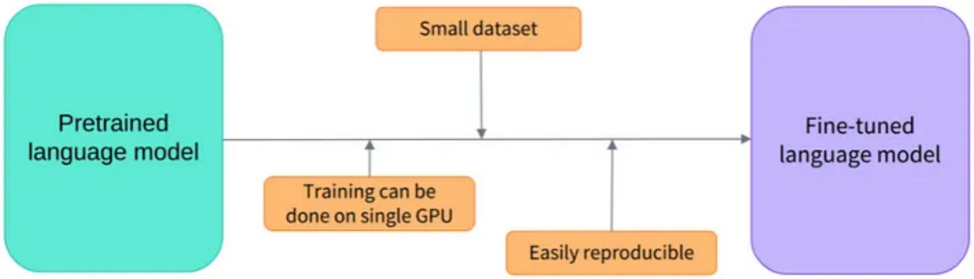

Pretrained language model
Small dataset
Training can be done on single GPU
Easily reproducible
Fine-tuned language model

**Image taken from: https://medium.com/nerd-for-tech/what-are-transformers-models-part-1-cf7ec6e8b3e8*

Fine-tuning of pre-trained models (i.e., models that have already been trained on large amounts of data) for tasks related to the original task the model was trained for, significantly reduces required training resources and has proven to produce good accuracy.

Results:

model	additional layers/activation layers	epochs	data size	Accuracy Pre training	Accuracy Post training
distilbert-base-uncased	None	3	1%	50.05%	96.19%
distilbert-base-uncased	None	3	100%	51.05%	96.16%
*distilbert-base-uncased	1 x bidirectional LSTM layer	12	1%	57.66%	95.94%
*distilbert-base-uncased	1 x bidirectional LSTM layer	12	100%	47.1%	95.93%
distilbert-base-uncased	1 x dense (fully connected) layer 1 x RELU	3	1%	49.37%	96.20%

distilbert-base-uncased	1 x dense (fully connected) layer 1 x Sigmoid	3	1%	40.85%	96.20%
distilbert-base-uncased	**1 x dense (fully connected) layer** **1 x tanh**	**3**	**1%**	**60.12%**	**96.22%**
** distilbert-base-uncased	1 x dense (fully connected) layer 1 x tanh	10	100%	52.79%	96.20%
distilbert-base-uncased	1 x dense (fully connected) layer 1 x tanh	3	100%	56.98%	96.19%

* hyperparameters tweaks: Learning rate changed from 2e-5 to 0.001 as it was too slow. Results were not promising. We then increased data size, knowing that LSTM networks require a lot of training data, but the results were still not improving so we decided to neglect this approach and focus on BERT only models.

** After 3 epochs we notice degradation in the model's accuracy. We therefore limited training on all models to 3 epochs only.

Overall conclusions:

- In our process, we followed the evolution of Artificial Networks for NLP tasks.
- We have seen that there's no benefit in combining two ANNs (simulating two hemispheres) together into one 'bilateral' model.
- The theoretical advantages of the newer bi-directional transformer architectures over the sequential processing of the LSTM network architectures were empirically established through a set of comparative experiments.
- biLSTM showed somewhat better results compared to LSTM model. This is due to its ability to derive bi-directional context (from sentence beginning to end, and from sentence end to beginning).
- Data size – we saw that LSTM and biLSTM require massive amount of training data to provide accurate classification. BERT, on the other hand, achieved superior results with far less training data. This is probably due to both factors – multi attention architecture as well as the use of pre-trained models for our experiments.
- There is a big variety of open source pre-trained models for various NLP tasks freely available online. One of our key observations is how significant *Transfer Learning* is to the process: whether it is in the short training time, or the small amount of data that is required to train a pre-trained model.
- We have managed to produce a model with slightly better classification accuracy: distilBERT + dense layer + tanh activation layer showed 96.22% compared to our benchmark model (BERT + dense layer + sigmoid) that produced 96.20%

Future Work:

- Alternatives to Human-Labeling of datasets for ML.
- Automating Social Media emoji tags (See https://www.researchgate.net/publication/321057905_Emoji_as_Emotion_Tags_for_Tweets)
- Application to identify and recognize emotions for children with autism (See https://www.frontiersin.org/articles/10.3389/frcha.2023.1118665/full)