Emotion-related Data Analysis by Large Language Model with Contrastive Learning

Yohei Nishimura
University of Wisconsin-Madison
ynishimura@wisc.edu

Amy Koike
University of Wisconsin-Madison
ekoike@wisc.edu

Abstract

Large language models such as ChatGPT have the potential to streamline marketing workflows. However, since these models are trained on public datasets, there is a possibility that they may not perform well in handling marketing-related tasks with zero-shot inference. This research investigates the capability of large language models with zero-shot and fine-tuning for marketing-related tasks using representative models such as T5. Additionally, we introduce a contrastive learning algorithm to enhance the model performance for marketing-oriented binary and multi-labeled text classification. We show that contrastive learning improves performance by training to model the relationship among each label in embedding space.

1 Introduction

Natural Language Processing (NLP) has been a powerful tool for marketing experts and researchers in the marketing field. Because NLP allows them to analyze large amounts of text data, such as customers' feedback, social media posts, and online reviews, they can identify customers' perceptions of products, key topics, sentiments, and trends. For example, sentiment analysis, which is one of the major NLP tasks, can be used for interpreting customers' feedback and opinions about products and services to improve marketing strategies. As another example, chatbots can engage with customers by answering their questions and providing support. Several business-to-business services already offer NLP as a business solution (*e.g.*, Salesforce Marketing Cloud, Amazon Comprehend, Brandwatch, Clarabridge, and MonkeyLearn).

We believe large language models (LLMs), such as ChatGPT (OpenAI, November 2022), accelerate the use of NLP for those marketing-related tasks; they are designed to handle unstructured data in a flexible manner. Unlike traditional rule-based or statistical methods, which relies on handcrafted features or pre-defined rules, LLMs can learn to extract features and relationships directly from the data. Moreover, advances in computer resources and dataset practices make language models more accessible. It should facilitate marketers and researchers to handle enormous amounts of unstructured practical data they have collected.

However, marketers might not apply LLMs trained by public datasets to their tasks; (Kocoń et al., 2023) warned that the capability of zero-shot inference by ChatGPT is limited for human-emotion-related tasks suchas sentiment analysis and emotion recognition. Time-consuming marketing tasks such as design of visual creatives or text copies, which marketers hope to automate, are highly related to customers' insight. Therefore, the zero-shot application of LLMs to marketing tasks could cause undesirable results.

In this research, our questions are 1) "can LLMs be used for 'marketing-related' tasks without fine-tuning?", and 2) "how can we fine-tune to improve the capability of solving 'emotion-related' tasks by LLMs?". Regarding the first question, we will test multiple classification tasks in marketing with zero-shot/finetuned LLMs. Additionally, we will introduce an algorithm with contrastive learning to enhance the trained models with the relationship between each emotion.

Specifically, we will use the T5 (Raffel et al., 2020) as a representative of LLMs since T5 is affordable and designed to perform a wide range of tasks. It can be fine-tuned without cost and small size but efficient results.

In the remainder of this paper, we will outline relevant works and describe our model architecture. Then, we will present the experiment procedure and implementation detail. Finally, we will show the results compared with the benchmark and the next tasks to finalize our project.

2 Related Work

2.1 Marketing with NLP

Several marketing researchers have utilized NLP in their work; NLP has been used for classification tasks and text analysis such as sentiment analysis and empathy classification in marketing tasks in a marketing research field. For example, Pamuksuz et al. (2021) utilized RoBERTa (Liu et al., 2019) for automated measurement of brand personality from social media was proposed. Lin et al. (2020) performed Latent Dirichlet Allocation (LDA) (Blei et al., 2003) for topic analysis to produce candidate words that can be marked as marketing 4C-related characteristic keywords in consumers' comments on social networks. Chakraborty et al. (2022) utilized a Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) network to address the challenge of obtaining attribute-level sentiment ratings from online text reviews.

2.2 Language models

Language models, which are founded on their probability distribution of sequences of words or phrases, have been used NLP tasks such as speech recognition, machine translation, text generation, or sentiment analysis (Li et al., 2021; Naseem et al., 2021). Many different types of language models have been introduced over a few decades. An early example of a language model is the n-gram model, which predicts the probability of a word based on the previous words in the sequence. Although n-gram models are simple and effective, they have limitations in capturing long-term dependencies between words.

One key component of many modern language models is attention mechanism, a technique that allows the model to focus on different parts of the input text when generating its output (Hu, 2020). Transformer that utilizes the attention mechanisms achieved significant results in various tasks, and it has since become one of the most widely used NLP models (Vaswani et al., 2017). The transformer architecture has mainly two stacks; one is an encoder that takes a sequence of input tokens and produces a sequence of hidden states, and the other is a decoder that takes the encoder output and produces a sequence of output tokens.

Since the introduction of Transformer, many NLP researchers have used the transformer as a base architecture and attained powerful results. One of the influential examples is BERT (Bidirectional Encoder Representations from Transformers), which is a bidirectional transformer-based architecture pre-trained with a combination of masked language modeling and next-sentence prediction (Devlin et al., 2019). The uniqueness of BERT is its use of a bidirectional transformer encoder. This allows BERT to take into account the context of a word in both directions.

In addition to the advent of BERT, text-to-text language models are also spotlighted by their flexibility in a variety of NLP tasks. ChatGPT, a dialog system based on the GPT-series (Radford et al., 2018a, 2019; Brown et al., 2020), amazed a wide variety of people with its intelligence. The first GPT model (often called "GPT-1") consists of a series of transformer decoder layers trained to predict the next word in a sequence as pre-training (Radford et al., 2018b). As another example of text-to-text language models, T5 is designed as a text-to-text model, making itself a highly flexible model that can be applied to a wide range of NLP tasks (Raffel et al., 2020). T5 model uses a bidirectional encoder-decoder architecture, which allows the model to encode the input sequence and generate the output sequence in a single pass.

2.3 Contrastive learning

In order to further improve the performance of our models, we will introduce a contrastive learning algorithm. Originally proposed as a method for computer vision in (Chen et al., 2020), contrastive learning recognizes argumented images as 'positive' pairs with the original one, and other pairs as 'negative' ones, controlling the distance between positive pairs as close ones, and between negative pairs apart in the embedding space. To apply this method to natural language processing tasks, (Gao et al., 2022) demonstrated that pretraining with contrastive learning using natural language inference (NLI) datasets can improve the classification performance of encoder-based models such as BERT and RoBERTa. While (Gao et al., 2022) proposes both unsupervised and supervised methods, we will employ a supervised learning approach in this paper.

3 Experiments

3.1 Model/Method

T5 In this study, we used T5 (text-to-text Transfer Transformer) to see the potential use for the marketing field (Raffel et al., 2020). T5 is a Transformer-

Dataset	Classes	Average length	Max length	Train/Val samples	Test samples
IMDb	2	292	3,045	25,000	25,000
Empathy dataset	2	170	322	1,002	103
GoEmotions	28	68	703	48,836	5,426

Table 1: Statistics of three text classification datasets.

Tweet	Empathy
@bigbobftworth54 Hello, Robert. We'd like to learn more about your experience. At your convenience, could you please DM us with your best contact information	0
We pride ourselves on making innovative vehicles, and we are thrilled to hear you're enjoying all that your LEAF has to offer! Thanks for kicking gas with us.	1

Table 2: Examples of Empathy dataset.

based architecture that solves various NLP tasks (*i.e.*, translation, summarization, or even classification) by text-to-text approach.

T5 has an encoder-decoder structure that closely follows the original Transformer (Vaswani et al., 2017). First, the input is tokenized into a sequence of tokens and then passed through an embedding layer to create an input representation with positional information. Then, the input representation is provided to the encoder. The encoder is layered with multi-head self-attention, followed by a feedforward network. The encoder encodes the input into a set of hidden representations that capture the relevant information in the input. Next, the representations are given to the decoder. The decoder has similarly structured layers as the encoder, which consists of multi-head self-attention, followed by a feedforward network. After the decoder takes the hidden representation from the encoder, it generates a sequence of output tokens. The output sequence generated is then passed through a final linear layer and softmax activation to produce a probability distribution over the possible output tokens, generating sentences with beam search over the possible sequence. The differences from the original transformer are that T5 removes the Layer Norm bias, places the layer normalization outside the residual path, and uses a different position embedding scheme.

Regarding tokenizer, T5 utilizes SentencePiece (Kudo and Richardson, 2018) to encode text as WordPiece tokens (Sennrich et al., 2016), using a vocabulary of 32,000-word pieces during pre-training.

To investigate the capability of LLMs in marketing-related classification tasks with zero-shot and fine-tuning, we conducted three types of experiments using T5: 1) binary sentiment classification based on a public dataset for sentiment analysis, 2) binary empathy classification based on our collected data, and 3) multi-labeled classification based on a public dataset. Since the classification tasks for the same public datasets were conducted with BERT(Alaparthi and Mishra, 2021), (Demszky et al., 2020), we evaluate the performance compared with BERT's results as benchmarks for all experiments.

Contrastive learning We used two types of what is a 'positive' pair in a contrastive learning algorithm. First of all, based on (Khosla et al., 2021), texts with the same labels are all positive samples, and those with different labels are negative ones. Second, we expanded 'positive' samples from the same label to similar labels; the similarity stems from between emotions. For instance, a pair of joy and amusement can be considered similar to each other, but sadness should be different from joy. We leveraged the emotional relationship in our algorithm to formulate a more optimized embedding space. We call the similar relationship between emotions 'soft positive' or 'continuous positive' dependent on the weights for each label.

Additionally, we adopted two approaches to introduce a contrastive learning algorithm. The first was to calculate a contrastive loss and train the encoder "during fine-tuning of T5 simultaneously". The second was to install this algorithm "as an additional pre-training step" for an encoder prior to fine-tuning. We referred the former approach as Model 1, 2, 3, and 7, and the latter as Model 4, 5,

6, and 8 in Table 5.

3.2 Dataset

We evaluated our approaches on three datasets: IMDb dataset (Maas et al., 2011), private empathy dataset of individuals' tweets, called 'empathy dataset,' and GoEmotions dataset (Demszky et al., 2020). We show the summary for each dataset in Table 1.

IMDb dataset The IMDb dataset contains 50K movie reviews along with binary sentiment labels: positive and negative. The overall distribution of labels is balanced (25K positive and 25K negative). The 50K reviews are split evenly into a set of 25K reviews for training and a set of 25K reviews for testing.

Empathy dataset This dataset was collected by randomly collecting 1,113 tweets from brand and company accounts. The data was annotated by crowd workers who assessed each tweet and assigned a value of 1 if they think it contained empathy and 0 if not. The ratio of positive to negative annotations was 432 to 681. To maintain this ratio, the data set was randomly divided into training, validation, and test data sets in an 8:1:1 proportion. Examples are shown in Table 2.

GoEmotions This dataset is composed of 58K English comments curated from Reddit, labeled for one or more of 27 emotion(s) plus 'neutral.' For example, the emotion categories include happiness, sadness, anger, fear, and surprise. Each comment was manually labeled by three raters, and most of them (83%) have a single emotion label.

The GoEmotion dataset includes the top-level concepts of 'positive,' 'negative,' and 'ambiguous' for each emotional label except for 'neutral.' Table 3 and 4 shows all emotional labels and their top-level concepts, and basic statistics of the dataset. We utilized these three concepts of 'positive,' 'negative,' and 'ambiguous' to measure the similarity between emotional labels that will be described later. Since 'neutral' does not have a top-level concept, we assumed that there are no emotional labels similar to 'neutral' except for 'neutral' itself.

3.3 Evaluation metrics

We evaluated our experimental performances by the precision, recall, F1 score, and accuracy (Forman et al., 2003). We used F1 as the most important score since some of our dataset have biases in the numebr of each label.

Label / Category	Train/ Val	Test	Total	%
Positive	19,170	2,301	21,471	37.4
Negative	11,985	1,518	13,503	23.5
Ambiguous	5,729	723	6,452	11.2
Neutral	14,219	1,787	16,006	27.9
Total	**51,103**	**6,329**	**57,432**	**100.0**

Table 3: Basic statistics of GoEmotion dataset. Text data categorized in positive are larger than the rest of the two. Breakdowns for each emotion are found in Table 4

Accuracy It is the total number of samples classified as target classes, calculated by $\frac{TP+FP}{N(total\ samples)}$, where TP represents "true positive," $N(\hat{y}=1|y=1)$, whereas FP represents "false positive," $N(\hat{y}=0|y=0)$.

F1, precision, and recall F1 score is the harmonic mean of the precision (P) and recall (R), which is a popular performance measure for classification. In the third experiment, which encompasses multi-classes, we use a macro-averaged F1 score to compare with benchmarks. The metrics are calculated by the equations 1. Here, FN represents "false negative," $N(\hat{y}=1|y=0)$

$$P=\frac{TP}{TP+FP},\quad R=\frac{TP}{TP+FN}$$
$$F1=\frac{2}{P^{-1}+R^{-1}} \tag{1}$$

3.4 Contrastive learning

We applied contrastive learning to multi-labeled GoEmotion dataset, the most challenging dataset among our three datasets. While NLI datasets including two text ('text' and 'hypothesis') were used to compute cosine similarities based on the positive pair in the original paper (Gao et al., 2022), GoEmotion dataset includes of 1) 'Text' (sentence), and 2) 'Emotion' (label). Therefore, we built our loss functions based on (Khosla et al., 2021), as shown in Equation 2 in both an additional pre-training and a simultaneous fine-tuning.

$$\mathcal{L}_{cl}^{sup}=-\sum_{i\in I}\frac{1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp\left(z_i\cdot z_p/\tau\right)}{\sum_{a\in A(i)}\exp\left(z_i\cdot z_a/\tau\right)} \tag{2}$$

Label / Category	Train/ Val	Test	Total	%
Positive				
admiration	4,130	504	4,634	8.1
amusement	2,328	264	2,592	4.5
approval	2,939	351	3,290	5.7
caring	1,087	135	1,222	2.1
desire	641	83	724	1.3
excitement	853	103	956	1.7
gratitude	2,662	352	3,014	5.2
joy	1,452	161	1,613	2.8
love	2,086	238	2,324	4.0
optimism	1,581	186	1,767	3.1
pride	111	16	127	0.2
relief	153	11	164	0.3
Negative				
anger	1,567	198	1,765	3.1
annoyance	2,470	320	2,790	4.9
disappointment	1,269	151	1,420	2.5
disapproval	2,022	267	2,289	4.0
disgust	793	123	916	1.6
embarrassment	303	37	340	0.6
fear	596	78	674	1.2
grief	77	6	83	0.1
nervousness	164	23	187	0.3
remorse	545	56	601	1.0
sadness	1,326	156	1,482	2.6
Ambiguous				
confusion	1,368	153	1,521	2.6
curiosity	2,191	284	2,475	4.3
realization	1,110	145	1,255	2.2
surprise	1,060	141	1,201	2.1
Neutral				
	14,219	1,787	16,006	27.9
Total	**51,103**	**6,329**	**57,432**	**100.0**

Table 4: Breakdowns for GoEmotion dataset. Regarding each emotion, 'admiration' is the largest excluding 'neutral', and 'grief' is the smallest.

Here, $i \in \{1, \ldots I\}$, $P(i)$ means the set of the same or similar samples as text i, and $A(i)$ does all the text data. This loss was used in Model 1 and 4 in fine-tuning.

For our second type of supervised contrastive learning algorithms introducing 'soft positive', we used a 'threshold' $\in (0, 1)$ such as 0.5, one of key hyperparameters, and calculated the cosine similarity of the similar emotions in the numerator in the loss function weighted by the threshold. The loss function is shown in Equation 3.

$$\mathcal{L}_{spos}^{sup} = -\sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i),\ s \in S(i)} \log \frac{\exp\left(z_i \cdot z_p / \tau\right) + \omega \exp\left(z_i \cdot z_s / \tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a / \tau\right)} \tag{3}$$

Here, ω means a threshold for soft positive data, $S(i)$ includes soft positive data of data i. This loss was used in Model 2 and 5 in fine-tuning and pre-training.

Additionally, we implemented the third type of contrastive learning algorithms with the positive labels named 'continuous positive.'

$$\mathcal{L}_{cpos}^{sup} = -\sum_{i \in I} \frac{1}{|P(i)|} \sum_{p \in P(i),\ s \in S(i)} \log \frac{\exp\left(z_i \cdot z_p / \tau\right) + \omega_{i,s} \exp\left(z_i \cdot z_s / \tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a / \tau\right)}$$
$$\Omega = \{\omega_{i,j} | i, j \in L\} \tag{4}$$

where, L means the set of labels $\{0, 1, \ldots, 28\}$, and $\omega_{i,j}$ means that the weights representative the similarity between labels l_i and l_j, calculated by embedding vectors from the RoBERTa fine-tuned by GoEmotion dataset. This loss was used in Model 3 and 6, 7 and 8 in fine-tuning.

We experimented eight models with contrastive learning algorithms including soft positive and continuous positive. Our models are summarized in Table 5.

	Calc for pos pair		Approximation loss for Contrastive learning
	Pre-training	Fine-tuning	
Model 1	-	Equation 2	Equation 5
Model 2		Equation 3	
Model 3		Equation 4	
Model 4	Equation 3	Equation 2	
Model 5		Equation 3	
Model 6		Equation 4	
Model 7	-	Equation 4	Equation 6
Model 8	Equation 3		

Table 5: Comparisons among eight models in our experiments.

3.5 Implementation

We used T5 with zero-shot and fine-tuning to solve three-dataset-usage text classification tasks. However, T5 is a text-to-text language model and does not have a downstream head for classification. Therefore, we implemented special codes to solve classification tasks.

3.5.1 Binary classification

Our binary classification code using IMDb dataset is mainly based on the code in https://colab.research.google.com/github/patil-suraj/exploring-T5/blob/master/t5_fine_tuning.ipynb. We conducted an additional implementation of the code for 1) zero-shot inference for binary classification, 2) zero-shot inference and fine-tuning the model for multi-label classification tasks, and 3) computation of metrics for binary classification and multi-label test data for the reimplementation's requirement; since we used the scikit-learn package to compute the metrics, we processed the data accordingly to fit this implementation.

Additionally, we implemented the code to fine-tune BERT as our benchmark for IMDb and Empathy datasets. In both cases, the base BERT models are uncased with a training epoch of 10, a max length of 256, and a batch size of 8. Both codes are included in our repository.

Fine-tuning We converted target data (numerical values) to the corresponding text (e.g., 'positive' in a binary classification or 'amusement' in a multi-labeled classification) to train/test the model. This implementation method is inspired by the training method for SST2 (Stanford Sentiment Treebank v2) introduced in Appendix D.7 (Raffel et al., 2020).

Furthermore, we inserted a specific task name (e.g., EmpathyClassification) as a prefix, followed by a comma and a space. Then we concatenated the text data to be classified and the word corresponding to the target of the data (e.g., "negative" for 0), wrapped up by "</s>".

During fine-tuning, we use the prefix and input text for training and train the model with respect to the target text converted from class numbers. The model was trained for 10 epochs.

3.5.2 Multi-classification

Some of the labels in GoEmotions are represented by multiple token IDs for a single word; for example, 'disapproval' corresponds to $[1028, 12497, 2165]$. Therefore, the original code cannot handle these labels. In our implementation, we converted the numerical targets in the dataset to their corresponding words (e.g. target 10 into 'disapproval'), then mapped them to their token IDs with "<EOS>(</s>)" token by T5 tokenizer. Then the model reads them up to the "</s>". We determined the maximum length of the target words based on the maximum length of words in the batch, and masked any tokens shorter than this length with "<PAD>". This implementation resolved the issue of emotion labels having various token lengths.

3.5.3 Zero-shot inference

We implemented the code to explore the T5's estimation capability of empathy or emotion classification with zero-shot. In order to make the T5 base model perform the tasks of binary and multivalued classification with zero-shot, it is necessary to limit the logits in the model's output; since T5 does not have a head for classification, it may output answers other than 'binary' or 'multivalued' if the user generates 'meaningless' answers without any constraints.

To avoid meaningless generation, we had preselected the positions of the logits that correspond to the answers of the classification; for instance, for positive 1465 and for negative 2841 for binary classification. Within this limited number of answers, the scores are compared, and the word that takes the maximum value is taken as the answer.

Additionally, as written above, some labels in GoEmotions are composed of a few token ids, as the vector of $[3, 60, 2528, 7, 15]$ represents 'remorse.' The calculation of the probability of 'remorse' results from 1) taking a softmax of logits for 256 words, 2) taking the numbers representing specific words, and 3) multiplying them to provide the probability of the words.

3.5.4 Contrastive learning

To keep our implementation efficient, we utilized the approximation methods in calculating Equation 2 or 3, which enables us to conduct a matrix-based implementation in PyTorch.

$$\mathcal{L}^{sup} \approx -(S \odot S^* \odot M - S \odot (\mathbb{1} - S^*) \odot M) \tag{5}$$

Here, $\{s_{i,j}\}$ in S means the cosine similarity between ith and jth data, and $\{s^*_{i,j}\}$ in S^* is 1 if the label of ith and jth data are the same, the value of

Dataset	Architecture	Model	F1	Precision	Recall	Accuracy
IMDb	T5	Zero-Shot	0.42	0.42	0.43	0.43
		Fine-tuned	**0.95**	0.95	0.95	**0.93**
	BERT	Benchmark	**0.95**	0.95	0.95	**0.93**
Empathy	T5	Zero-Shot	0.37	0.30	0.49	0.59
		Fine-tuned	**0.84**	0.86	0.83	**0.78**
	BERT	Benchmark	**0.84**	0.85	0.83	**0.78**
GoEmotion	T5	Zero-Shot	0.02	0.91	0.04	0.28
		Fine-tuned	0.49	0.59	0.44	**0.64**
	BERT	Benchmark	0.49	0.59	0.44	**0.64**
	T5	Model 1	0.50	0.62	0.45	0.53
		Model 2	**0.52**	0.58	0.48	0.51
		Model 3	**0.52**	0.57	0.48	0.51
		Model 4	0.48	0.57	0.44	0.52
		Model 5	0.49	0.57	0.46	0.52
		Model 6	0.49	0.60	0.45	0.52
		Model 7	0.50	0.57	0.46	0.51
		Model 8	0.51	0.59	0.47	0.53

Table 6: Results from all models and dataset. Bold means the best score(s) in each dataset.

the threshold if the label of ith and jth data are in the same group (they are soft positive each other), and 0 otherwise. M denotes the mask, which is a upper triangular matrix where all values are 1 in the upper side and others are zero including diagonal components. If we calculate the loss with soft positive, S^* has the threshold values on $\{s^*_{i,j}\}$ if ith and jth data are in the same group. This approximated loss calculation is used in from Model 1 to Model 6.

Additionally, for continuous positive pairs, we introduced L1 loss defined by Equation 6. This approximated loss calculation is implemented in Model 7 and 8.

$$\mathcal{L}^{sup}_{L1} \approx |S \odot S^* \odot M - S \odot (\mathbb{1} - S^*) \odot M)| \quad (6)$$

3.6 Hyperparameters

We use the T5 model (Raffel et al., 2020) with a hidden size of 768, 12 multi-head self-attention, and 12 Transformer blocks (Vaswani et al., 2017) in both the encoder and the decoder.

To train the model, we used one RTX3090 and set the batch size to 8 for fine-tuning with a max sequence length of 256 and training epoch of the range from 7 to 10, whereas the batch size is 32 for validation and testing in each dataset. The threshold for soft positive contrastive loss is 0.5.

Our optimization method for fine-tuning was AdamW (Loshchilov and Hutter, 2019) with a learning rate of the range from 1e-4 to 5e-5 corresponding to the models and weight decay of 0.0.

4 Results

We show our results in Table 6 and 7. The benchmark scores of fine-tuning BERT for IMDb and Empathy are conducted by our code in the repository. The scores for GoEmotions by fine-tuned BERT are introduced in the paper by Demszky et al.. The results of our implementation perform as well as or better than benchmark scores.

Regarding furture improvement of GoEmotion-related models, as shown in Table 7, the model after fine-tuning improved the baseline model by 3.0% on each dataset. The fine-tuning algorithm with

simple contrastive learning (Model 1) improved the results of the baseline model using the GoEmotion dataset by 4%, while the results of the plain fine-tuning model by 3%.

Additionally, the algorithm using contrastive learning with soft positive (Model 2) and with continuous positive (Model 3) improved the results of the model trained with simple Pos/Neg Contrastive learning by 2%.

On the other hand, the performance of the models with pretraining (Model 4, 5, 6) by contrastive learning exceeded baseline, but they did not enhance the results of simultaneous training by contrastive learning. The L1 loss improved the models by 1-2% compared to the simple fine-tuned model.

Regarding individual emotion labels, on average, the model with simultaneous contrastive learning fine-tuning of soft positive worked best; other models performed better in some emotions such as 'amusement' and 'surprise' by the model with simultaneous contrastive learning fine-tuning of simple pos/neg, or the additional pre-trained model estimated 'admiration,' 'excitement,' or 'gratitude' better.

Interestingly, the models with simultaneous contrastive learning fine-tuning were able to estimate the label of 'grief,' whie benchimark, simple fine-tuned, and pretrained models did not. All models almost precisely estimated the label of 'gratitude,' which is one of the explicit emotion in the sentence.

5 Conclusion

In this paper, we addressed the questions of 1) "can LLMs be used for 'marketing-related' tasks without fine-tuning?", and 2) "how can we fine-tune to improve the capability of solving 'emotion-related' tasks by LLMs?" using emotion-related datasets. By adding contrastive learning to train the encoder, we were able to transfer the relationship between emotions well into the embedding space and create a model that performs by around 3-5 % better than simple fine-tuning.

6 Contribution of each member

6.1 Yohei

Whole architecture design; implementation of all codes; writing Experiments, Implementation, Results, and Conclusion and tables.

6.2 Amy

Implemented the test codes; wrote Introduction, Related Work, Model.

References

Shivaji Alaparthi and Manit Mishra. 2021. BERT: a sentiment analysis odyssey. *Journal of Marketing Analytics*, 9(2):118–126.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Ishita Chakraborty, Minkyung Kim, and K Sudhir. 2022. Attribute sentiment scoring with online text reviews: Accounting for language structure and missing attributes. *Journal of Marketing Research*, 59(3):600–622.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

George Forman et al. 2003. An extensive empirical study of feature selection metrics for text classification. *J. Mach. Learn. Res.*, 3(Mar):1289–1305.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Dichao Hu. 2020. An introductory survey on attention mechanisms in nlp problems. In *Intelligent Systems and Applications: Proceedings of the 2019 Intelligent Systems Conference (IntelliSys) Volume 2*, pages 432–448. Springer.

	F1									
	Benchmark	Fine-tuned	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7	Model 8
admiration	0.65	0.68	0.67	0.66	0.67	0.69	0.67	0.66	0.66	0.67
amusement	0.80	0.80	0.82	0.79	0.77	0.78	0.79	0.78	0.78	0.79
anger	0.47	0.47	0.46	0.48	0.52	0.48	0.50	0.51	0.48	0.48
annoyance	0.34	0.26	0.26	0.34	0.33	0.27	0.24	0.27	0.31	0.29
approval	0.36	0.42	0.39	0.37	0.42	0.40	0.42	0.42	0.41	0.41
caring	0.39	0.48	0.43	0.39	0.41	0.47	0.45	0.42	0.39	0.45
confusion	0.37	0.44	0.42	0.43	0.43	0.38	0.43	0.44	0.41	0.43
curiosity	0.54	0.50	0.49	0.47	0.49	0.43	0.54	0.49	0.47	0.51
desire	0.49	0.44	0.48	0.49	0.49	0.53	0.55	0.53	0.50	0.58
disappointment	0.28	0.31	0.28	0.32	0.29	0.32	0.31	0.32	0.33	0.37
disapproval	0.39	0.41	0.37	0.40	0.42	0.43	0.40	0.42	0.43	0.42
disgust	0.45	0.47	0.46	0.48	0.45	0.45	0.47	0.50	0.44	0.52
embarrassment	0.43	0.54	0.53	0.47	0.48	0.54	0.51	0.51	0.48	0.44
excitement	0.34	0.40	0.39	0.38	0.40	0.40	0.40	0.38	0.40	0.41
fear	0.60	0.68	0.69	0.67	0.68	0.69	0.71	0.68	0.69	0.69
gratitude	**0.86**	**0.91**	**0.88**	**0.89**	**0.89**	**0.91**	**0.91**	**0.91**	**0.89**	**0.92**
grief	0.00	0.00	0.22	0.53	0.55	0.00	0.00	0.00	0.33	0.25
joy	0.51	0.58	0.59	0.57	0.58	0.59	0.58	0.58	0.56	0.59
love	0.78	0.79	0.81	0.80	0.81	0.78	0.81	0.80	0.79	0.81
nervousness	0.35	0.34	0.41	0.39	0.43	0.46	0.39	0.36	0.45	0.45
optimism	0.68	0.55	0.48	0.56	0.55	0.56	0.53	0.52	0.54	0.52
pride	0.51	0.30	0.45	0.45	0.56	0.11	0.19	0.20	0.11	0.30
realization	0.36	0.18	0.24	0.25	0.22	0.20	0.19	0.20	0.22	0.19
relief	0.21	0.50	0.40	0.50	0.35	0.44	0.44	0.40	0.50	0.44
remorse	0.15	0.58	0.59	0.67	0.61	0.57	0.66	0.63	0.61	0.59
sadness	0.66	0.53	0.52	0.56	0.51	0.54	0.52	0.54	0.57	0.54
surprise	0.49	0.51	0.59	0.56	0.54	0.52	0.54	0.52	0.56	0.53
neutral	0.50	0.67	0.67	0.63	0.64	0.66	0.65	0.66	0.64	0.66
Average (macro)	0.46	0.49	0.50	0.52	0.52	0.48	0.49	0.49	0.50	0.51

Table 7: Results of each emotion from all models in GoEmotion dataset. Bold means the highest score per row, and underline means the highest per column.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised contrastive learning.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, Anna Kocoń, Bartłomiej Koptyra, Wiktoria Mieleszczenko-Kowszewicz, Piotr Miłkowski, Marcin Oleksy, Maciej Piasecki, Łukasz Radliński, Konrad Wojtasik, Stanisław Woźniak, and Przemysław Kazienko. 2023. Chatgpt: Jack of all trades, master of none.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing.

Junyi Li, Tianyi Tang, Wayne Xin Zhao, and Ji-Rong Wen. 2021. Pretrained language models for text generation: A survey.

Hao-Chiang Koong Lin, Tao-Hua Wang, Guo-Chung Lin, Shu-Chen Cheng, Hong-Ren Chen, and Yueh-Min Huang. 2020. Applying sentiment analysis to automatically classify consumer comments concerning marketing 4cs aspects. *Applied Soft Computing*, 97:106755.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. 2021. A comprehensive survey on word representation models: From classical to state-of-the-art word representation language models. *Transactions on Asian and Low-Resource Language Information Processing*, 20(5):1–35.

Utku Pamuksuz, Joseph T. Yun, and Ashlee Humphreys. 2021. A brand-new look at you: Predicting brand personality in social media networks with machine

learning. *Journal of Interactive Marketing*, 56:55–69.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018a. Improving language understanding by generative pre-training.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018b. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.