Generative Pre-trained Transformer (GPT) Models for Irony Detection and Classification

Mustafa Ulvi Aytekin
Department of Computer Engineering
Graduate School of Natural and Applied Sciences
Gazi University
Ankara, Türkiye
mustafaulviaytekin@gmail.com

O. Ayhan Erdem
Department of Computer Engineering
Faculty of Technology
Gazi University
Ankara, Türkiye
ayerdem@gazi.edu.tr

Abstract—The tasks of identifying and classifying ironic texts remain an ongoing challenge, necessitating continued exploration for enhanced solutions in NLP. This study delves into assessing the effectiveness of Generative Pre-trained Transformer (GPT) models, which have emerged in recent years, in handling irony detection and classification tasks through the implementation of zero-shot learning and few-shot learning methods in English texts. Additionally, we compare GPT text embedding models with GloVe, a proven text embedding model, utilizing various machine learning and deep learning approaches. Within this study, we employed the SemEval-2018 Task 3 dataset, curated as part of the Semantic Evaluation 2018 workshop. The most noteworthy achievement in binary classification, namely irony detection, is an F_1 score of 68.9%, attained by the text-davinci-003 model through few-shot learning, with access to forty-two samples for training. In terms of multiclass classification, namely irony classification, the text-embedding-ada-002 text embedding model, in conjunction with the Gaussian Naive Bayes algorithm, attained the best result with an F_1 score of 48.5%. The best results obtained in the study achieved comparable results with previous studies.

Keywords—gpt, text embedding, irony detection, irony classification, chatgpt

I. INTRODUCTION

The increasing use of social media applications and communication devices generates a massive amount of raw text data. Considering the current usage of social media, the number of people using social media will continue to grow in the coming years. As the volume of raw data keeps increasing, new business opportunities based on extracting meaningful information from raw text also emerge [1]. However, working with social media data presents challenges due to misspelled words, unique abbreviations, and unusual speech patterns not commonly used in daily conversations [2]. People can convey facts, opinions, emotions, and emotional intensities on various topics in short texts. Analyzing the expressed emotions in social media content has broad applications in commerce, public health, social welfare, and more [3]. Some areas where social media data can be utilized include detecting offensive language, identifying metaphorical expressions, analyzing and classifying emotions and sentiments, detecting irony, sarcasm, and satire, topic classification, and political sentiment analysis [4, 5, 6].

Humans are social beings. Making jokes, using irony and metaphor while speaking is an action that people can perform in their daily lives without difficulty. Just like speaking, people can understand humor, irony and metaphor without any difficulty, and can easily make sense of an idiomatic expression that they have never heard before, according to the course of the subject or the representations of the words in the idiom. It is certain that the determination of these structures is essential for individuals, institutions or companies to obtain more meaningful information in many areas such as product reviews and service returns, apart from mutual conversation. Although models today perform quite well in certain subtasks of natural language processing, there is a wide range of results that vary according to different datasets in detecting irony, sarcasm, satire and figurative expression. This shows that the search for the model that will give the best results should continue.

There are several different types of irony. In social networks, situational and verbal irony are among the most frequently encountered types of irony. Situational irony arises from incongruous situations involving specific events, as exemplified by statements like "A marriage counselor files for divorce." On the contrary, verbal irony involves expressing a meaning opposite to the literal interpretation, as seen in sentences like "I just love watching the same cartoon five times every afternoon. It's such a delight." Detecting irony in text messages is a challenging and subjective task influenced by various factors. Researchers have identified several relevant features to tackle the irony detection problem, including polarity contrast, common sense knowledge, similes using "about" or "as" structures, affective features, negation, context incongruity, and contextual features.

The detection of irony parallels the development of studies on natural language processing. Initially, the research was based on statistical machine learning models, but later, deep learning models were also employed. With the development of transformer-based models, these models were also utilized, along with various hybrid and custom-developed models. Pre-trained transformer-based models, including GPT (Generative Pre-trained Transformer) models, show significant promise in both general natural language processing and specifically in irony detection.

In this work, first, we examine the success of two GPT-3.5 models, namely text-davinci-003 and gpt-3.5-turbo, in irony detection and classification, in zero-shot and few-shot learning methods. Then, we compare the success of two GPT word embedding models, namely text-embedding-babbage-001 and text-embedding-ada-002 with the success of the GloVe word embedding model using various machine learning and deep learning algorithms. We evaluate the most successful models obtained after various preprocessing methods by comparing them with the results of previous studies using the same dataset.

Main contributions of this paper are:

979-8-3503-1803-6/23/$31.00 ©2023 IEEE

- Comparison of text-davinci-003 and gpt-3.5-turbo GPT models in zero-shot and few-shot learning methods.

- Comparison of text-embedding-babbage-001 and text-embedding-ada-002 GPT text embedding models and the glove.6b.100d GloVe text embedding model using several learning methods.

- Comparison of the results of the studies using the SemEval-2018 Task 3 dataset and the results of the models in this study.

The rest of this paper is orginezed as follows: In Section 2, we discuss several irony detection and classification related works. In Section 3, we describe the dataset used in the study, the models used in irony detection and classification, and the preprocessing of the dataset. In Section 4, we include the comparison of the results obtained with each other. In Section 5 we make comparisons with previous studies using the same dataset. In Section 6, we conclude the work and discuss what can be done in future studies.

II. Related Works

In a study conducted by Ravi and Ravi (2016), it is stated that people struggle to understand figurative language and the existing approaches in this field have limited competence and scope. Therefore, a combined text feature selection method based on text and data mining paradigms is proposed to automatically detect satire, mockery, and irony found in news and customer reviews. The effectiveness of the method is demonstrated in two satirical and one ironic dataset by using three different feature selection methods TAALES, DTM, LIWC2015 in combination with each other. The results of 11 different machine learning and deep learning algorithms are compared. The study considers the corpus's stylometry, semantic measurements, and psychometrics to identify satirical and ironic content. It is suggested that future studies can include metrics related to syntactic and pragmatic information to improve the obtained results [7].

Sulis, Hernández Farías, Rosso, Patti, and Ruffo (2016) aim to highlight the difference between irony and sarcasm in their study. Analyzes are conducted on the tags #irony (irony), #sarcasm (sarcasm), and #not (non-irony), which have not been extensively investigated before, produced within the SemEval-2015 Task11. By performing distribution and correlation analyses on a range of features, including various psycholinguistic and emotional features, arguments are proposed for distinguishing between irony and sarcasm. In the evaluations of binary classification experiments, a new set of emotion, structural, and psycholinguistic features that can be used in the future studies is obtained. It is concluded from the evaluation that users mostly use these tags without making a specific distinction. It is stated that by using the tags for different sociological groups, the distinction between irony and sarcasm can be made more pronounced in future studies, which can contribute more accurately to subsequent sentiment analysis studies [8].

Zhang, Zhang, Chan, and Rosso (2019) discuss in their study that existing works in irony detection have primarily formulated the task as a standard supervised learning text classification task based on explicit expressions to detect contextual incongruity. However, in this study, irony detection is formulated as a transfer learning task enriched with information transferred from the extra resources of sentiment analysis that focus on detecting implicit, covert incongruity without relying on explicit incongruity cues. Three different attention-based LSTM models are proposed. While all models outperform other models, the STBi-LSTM model provides the best results. This model utilizes two different Bi-LSTMs, one for feature extraction and the other for irony detection. These Bi-LSTMs share weights and the features they obtain are merged in another layer before entering the attention mechanism. Compared to LSTM, CNN, Bi-LSTM, and CNN-LSTM hybrid models, it achieves better results in all datasets. The study highlights the lack of sufficient labeled data for irony detection and mentions that some ironies can only be understood through the analysis of mutual conversations by humans, and some physical situations may not be understood by the model, resulting in the inability to detect incongruity in sentence context [9].

Chia, Ptaszynski, Masui, Leliwa, and Wroczynski (2021) conducted experiments aiming to detect irony and sarcasm separately, emphasizing the conceptual difference between irony and sarcasm. They also discuss the relationship between irony, sarcasm, and cyberbullying. Different machine learning methods and data preprocessing techniques were employed for irony and sarcasm detection. SemEval-2018 Task 3 dataset was used for irony detection. In the data preprocessing methods, different treatments were applied to hashtags, including removing hashtags, preserving them as they are, and transforming them into respective tokens. The best result in the hashtag-preserved dataset was achieved by the JRip classifier. The Ghosh dataset was used for sarcasm detection, and different variations of data preprocessing, including the inclusion or exclusion of hashtags, stemming, removing stopwords, PoS tagging, and not applying any labeling, were performed. The best result was obtained with the hashtag-preserved dataset. In a subsequent experiment, it was observed that when the training set was irony and the test set was sarcasm, indicating that these two tags are often used interchangeably. It is suggested that future studies can explore more diverse preprocessing methods [10].

In the study conducted by González, Hurtado, and Pla (2020), a model for irony detection based on the contextualization of pre-trained Twitter word embeddings through the Transformer architecture is proposed. This approach allows the utilization of domain-specific embeddings, different from BERT (Bidirectional Encoder Representations of Transformers), which is based on the same powerful architecture but does not allow the usage of domain embeddings. The proposed model is evaluated on two different datasets, one in English and the other in Spanish. Several strategies are proposed to identify specialized attention heads for irony detection and analyze various features related to irony. By interpreting the values of attention heads, the study analyzes the impact of word polarity, word appropriateness, relationships between individual words, and these aspects on the irony detection task. It is observed that when certain attention heads are masked, the results for the ironic class deteriorate, supporting the hypothesis of the specialization of the identified attention heads [11].

In the study by González, Hurtado, and Pla (2020), it is mentioned that the multilingual version of BERT, mBERT, is not sufficient, and the success shown by the standard BERT model in English cannot be achieved by mBERT and S-BERT on Twitter data in Spanish. To address this, a BERT model named TWilBERT is developed by training it on data obtained from Twitter. The model is developed in three different variations: TW-Base, TW-Large, and TW-Large without reply order prediction (ROP). TW-Base has 6 transformer layers and

6 attention heads, while the other models have 12 layers and 12 heads. TW-Base and TW-Large incorporate reply order prediction (ROP), and another TW-Large model without ROP is used. ROP is used as an alternative to the next sentence prediction mechanism in the BERT model. These models are used in 14 different classification tasks and outperform mBERT and S-BERT in all tasks [12].

In the study by Karoui, Zitoune, and Moriceau (2017), a supervised learning method is proposed for irony detection in Arabic tweets. Four feature groups that have experimentally proven effectiveness in other languages such as French, English, Italian, Dutch, and Japanese are used in classification algorithms SVM, NB, LR, RT (Random Tree), and RF. Using all features except speech verbs, the RF classifier achieves F1 scores of 73% for the ironic class and 72.5% for the non-ironic class in the classification task. The results are lower than expected due to the presence of different Arabic dialects in the created dataset and the lack of suitable social media feature extraction tools for the Arabic language. Recent studies indicate that the use of frequent and rare synonyms, discussion threads, and external sources of information are highly effective in understanding irony in other languages, and it is suggested that future work can be done for the Arabic language in this regard [13].

Charalampakis, Spathis, Kouslis, and Kermanidis (2016) propose a scheme for irony detection in Greek political tweets, claiming that election results can be predicted through humorous tweets. The proposed model is based on a semi-supervised learning approach that takes into account both labeled and unlabeled training data by considering algorithms for collective learning. The obtained results are compared with the results from another study they conducted, which used supervised learning. The relationship between the number of ironic tweets received by parties on Twitter and the difference between the last two election results is explored. It is mentioned that Twitter was more popular among a specific segment of the population at the time of the study, being used by only 3.7% of the population, and therefore, the model's performance can be further improved in the future [14].

Shrivastava and Kumar (2020) propose a model based on BERT, a pretrained transformer model. In this model, the BERT-base model with capital letters is used, which has 12 transformer layers and 110 million parameters. The pretrained model is fine-tuned with training data and used for classification with test data. The study focuses on binary classification tasks using the SemEval-2018 Task 3 dataset [15].

Vu et al. (2018) propose a model based on a multilayer perceptron, a simple neural network architecture. Various input types containing lexical, syntactic, semantic, and polarity features are used in the model. The tweets in the dataset are normalized using a word-based approach, and URL and mention expressions are transformed into corresponding tokens. The proposed model consists of an input layer, two hidden layers with ReLU activation function, and a softmax output layer. The input layer receives a combined vector representing the lexical, syntactic, semantic, and polarity features of the respective tweet. The two hidden layers with ReLU activation function are used to select the most important features in this vector. The selected features are then passed to the softmax output layer for irony detection and classification [16].

Rangwani, Kulshreshtha, and Singh (2018) propose a model that utilizes linguistic features and pretrained neural network activations. A CNN model is trained on the task of emoji prediction, and two feature vectors are combined and fed into an XGBoost (eXtreme Gradient Boosting) classifier for classification. To address the imbalance in the dataset for the multiclass classification task, the dataset is balanced using the SMOTE (Synthetic Minority Oversampling Technique) algorithm [17].

Baziotis et al. (2018) propose a model based on siamese neural networks for irony detection. The model consists of two independent subnetworks based on bidirectional LSTM (Bi-LSTM) layers, which operate at the word and character levels to capture semantic and syntactic information in tweets. The models are enriched with an attention mechanism to identify informative words. In the siamese network, two inputs are expected, and a comparison is made between the generated weights. Thus, each tweet is split into two parts, and different combinations of these tweets are recombined and included in the training set [18].

Wu et al. (2018) propose a model based on densely connected LSTM networks and multitask learning strategy. The dense LSTM model takes inputs from all the previous layers at each layer. The output of the last LSTM layer represents the hidden representation of the text, which is used in three different classification tasks. Additionally, several different feature types are used to improve the model's performance [19].

Rohanian, Taslimipoor, Evans, and Mitkov (2018) propose a model that utilizes LR and SVM component models of supervised systems along with an ensemble soft voting classifier, combining feature sets based on sentiment, semantic, and surface features. These handcrafted features are used in conjunction with dense vector representations [20].

Ghosh and Veale (2018) propose a siamese neural network model for irony detection. The model consists of two subnetworks based on LSTM and GloVe text embeddings. The siamese network takes two inputs and compares the generated weights. Therefore, each tweet is split into two parts. The model incorporates an attention mechanism to identify the most informative words [21].

III. Dataset, Evaluation Metrics, Models, and Preprocessing Techniques

This section provides an overview of the dataset utilized in the study, the evaluation metrics employed for irony detection and classification tasks, details about GPT models and the preprocessing methods applied to the dataset.

A. SemEval-2018 Task 3

Semantic Evaluation is the name of an international natural language processing research workshop established to advance the state of semantic analysis in languages and assist in creating high-quality annotated datasets for various challenging problems in natural language understanding. Each year, different teams release various datasets prepared for different tasks during the workshop [22]. The Task 3 dataset released for the 2018 workshop is designed for irony detection and classification in the English language. This dataset is compiled from Twitter messages written by various individuals at different times on the social media platform Twitter. The dataset is prepared in two different forms, with separate training and test sets. The dataset referred to as

Subtask A is a binary dataset. In the training set, there are a total of 3,834 labeled tweets, with 1,911 labeled as '1' (indicating ironic) and 1,923 labeled as '0' (indicating non-ironic). In the test set, there are 784 tweets, with 311 labeled as '1' and 473 labeled as '0'. On the other hand, Subtask B is a multiclass dataset. In the training set, there are a total of 3,834 labeled tweets, with 1,390 labeled as '1' (indicating verbal irony by polarity contrast), 316 labeled as '2' (indicating situational irony), 205 labeled as '3' (indicating other verbal irony), and 1,923 labeled as '0' (indicating non-ironic). In the test set, there are 784 tweets, with 164 labeled as '1', 85 labeled as '2', 62 labeled as '3', and 473 labeled as '0'.

B. Evaluation Metrics

The evaluation metrics used in the work are accuracy, precision, recall and F_1 score. Accuracy is calculated by dividing the number of correct predictions made by the total number of instances. Precision indicates the success rate in positive predictions. Recall shows the success rate among those that should be predicted positively. The F_1 score is calculated by taking the harmonic average of the precision and recall values.

$$accuracy = \frac{true\ positive + true\ negative}{total\ number\ of\ instances} \quad (1)$$

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (2)$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (3)$$

$$F_1 = \frac{2\ .\ precision\ .\ recall}{precision + recall} \quad (4)$$

While accuracy gives an overall view of the system's performance across all classes, the remaining three metrics were specifically computed for the positive class in binary classification task or were macro-averaged across four class labels in multiclass classification task. Macro-averaging the F_1 score means that all class labels carry equal weight in the final score.

C. Model Details

GPT is a transformer based pre-trained language model developed by OpenAI. The first GPT model, GPT-1, was released in 2018 and has 120 million parameters. It is trained with the data of 7000 unpublished books named Book Corpus in 4.5 GB size [23]. GPT-2, published in 2019, preserves its architectural structure, but the number of parameters reaches 1.5 billion [24]. In the training, WebText data containing 40 GB of text was used. With the GPT-3 announced in 2020, the model has made a huge leap forward. Although the number of parameters varies in different models, it has reached up to 175 billion and has been trained with 570 GB of data from many different sources such as filtered Common Crawl, Books1, Books2, WebText and Wikipedia Internet Encyclopedia. With the number of parameters it has and the amount of data it trains, it has reached a very different level from previous GPT models [25]. The GPT-3 models, which continue to be developed, have been replaced by the GPT-3.5 models and the GPT-4 models.

Two different types of GPT models used in this study. The first type of models are complete GPT models. These are the models of GPT-3.5 i.e. text-davinci-003 and gpt-3.5-turbo-0301. These models is the most successful models of the GPT-3.5 when the study conducted. Throughout the article, gpt-3.5-turbo-0301 model will be referred to as gpt-3.5-turbo. Gpt-3.5-turbo model is the model which runs famous AI chat bot ChatGPT. The second type of models are GPT embedding models i.e. text-embedding-babbage-001 and text-embedding-ada-002. These models uses GPT2/GPT3 and cl100k-base tokenizers respectively. These embedding models used with different learning algorithms. Same learning algorithms also used with GloVe embedding model of glove.6b.100d which includes hundred-sized vectors for each of six billion words.

D. Preprocessing Methods

In the dataset used in the study, emojis are expressed with Unicode Common Locale Data Repository (CLDR) annotations. Spaces, if any, in these annotations are indicated by underscores. Annotations have colons at the beginning and at the end. Expressions such as mentions, hashtags, smileys, urls, numbers and converted emoji annotations which can be found in tweets, were evaluated according to various preprocessing conditions. For this purpose, tweet-preprocessor library is used. Expressions were evaluated as one of the processes of transforming them into representative tokens, removing them from the tweet text or keeping them as they are. If GPT models are trained specifically for classification, the '->' marker can be added to the end of the text in training data preparation phase. Based on this, adding this marker to the end of the text is also considered as another preprocessing technique.

In cases where text embedding models are used, in addition to the previously mentioned preprocessing techniques, the removal of structures such as stopwords and punctuation marks from the text has been examined. In addition, since the GloVe model expects a certain length, various lengths were determined for tweet texts, and for those who could not reach this length, adding padding tokens to the front or end of the sentence was evaluated.

IV. Methodology

This study consists of two parts. In the first part, the success of the GPT models, namely text-davinci-003 and gpt-3.5-turbo, in irony detection and classification tasks was evaluated in both zero-shot learning and few-shot learning methods. In the second part, the word embedding models used by the GPT models, namely text-embedding-babbage-001 and text-embedding-ada-002, are compared with the GloVe word embedding model glove.6b.100d using various machine learning methods. All the results obtained were compared within themselves. The best results obtained were compared with the results of previous studies.

A. GPT models on binary and multiclass classification

The models expect the user to define the tasks to be performed and to give one or more examples if necessary, depending on the learning method. In this way, it is aimed to first understand the task it needs to do, and then increase its success in the task it will do through examples, if provided. For the models used in the study in zero-shot learning method, the task to be performed for each classification process was defined first, and then, in accordance with this definition, each data expected to be classified in the test set was provided to the model and it was expected to do the classification. When the few-shot learning method was applied, the samples provided to the model were randomly selected from the training set of the dataset. The number has been determined as

42. The examples provided to the model in the few-shot learning method are the same in binary and multiclass classification. While there are 21 ironic and 21 non-ironic examples in binary classification, there are 21 examples of non-ironic, 16 verbal irony, 3 situational irony, and 2 other irony in multiclass classification.

The definition and data provided to the models in binary classification and zero-shot learning methods are as follows:

"Decide if a tweet is ironic or not. If it is ironic write '1', else write '0'.

Tweet: {Here each tweet in the test set is put.}

Irony:"

The definition and data provided to the models in binary classification and few-shot learning methods are as follows:

"Decide if a tweet is ironic or not. If it is ironic write '1', else write '0'.

Tweet: {Tweet 1}

Irony:{Label 1}

...

Tweet: {Tweet 42}

Irony:{Label 42}

Tweet: {Here each tweet in the test set is put.}

Irony:"

The definition and data provided to the models in multiclass classification and zero-shot learning methods are as follows:

"You are a data expert working for Twitter.

You are detecting and analysing the irony in tweets and classifying them into one of four categories.

The four categories are Not Ironic, Verbal Irony, Situational Irony and Other Irony.

Tweet: {Here each tweet in the test set is put.}

The Classification is:"

The definition and data provided to the models in multiclass classification and few-shot learning methods are as follows:

"You are a data expert working for Twitter.

You are detecting and analysing the irony in tweets and classifying them into one of four categories.

The four categories are Not Ironic, Verbal Irony, Situational Irony and Other Irony.

Tweet: {Tweet 1}

The Classification is: {Label 1}

...

Tweet: {Tweet 42}

The Classification is: {Label 42}

Tweet: {Here each tweet in the test set is put.}

The Classification is:"

The best results obtained by the text-davinci-003 and gpt-3.5-turbo models are shown in the Table I for irony detection and Table II for irony classification. For text-davinci-003 model in irony detection task, the most successful results obtained in preprocessing setting where url addresses were converted to 'URL' tokens, smileys were removed, the '#' sign at the beginning of hashtags was removed, the ':' signs at the beginning and end of the emojis that were converted to text were removed, mentions, numbers and reserved words were not touched. For gpt-3.5-turbo model in irony detection task, the most successfull results obtained in preprocessing setting where url and smileys were deleted from the tweet, the mentions were converted to the $MENTION$ token, the numbers, reserved words, hashtags and emojis were kept as they were, and the '->' sign was added to the end of the tweets. For irony classification task, both GPT models obtained best result when no preprocessing setting was employed.

B. Embedding models with different learning algorithms

GPT embedding models, namely text-embedding-babbage-001, text-embedding-ada-002 models, were compared among themselves and with the glove.6b.100d model, which is one of the GloVe embedding models, using various machine learning and deep learning algorithms in various preprocessing settings. Learning algorithms used for this purpose are Random Forest (RF), K Nearest Neighbor (KNN), Decision Tree (DT), Gaussian Naive Bayes (GNB), Adaptive Booster (AdaBoost), Multilayer Perceptron (MLP), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU).

Scikit-learn library is used for RF, KNN, DT, GNB, AdaBoost and MLP algorithms. The parameters have been kept at their default values and no fine-tuning has been made. In CNN, LSTM and GRU, models that give better results with various parameters were sought. The parameter settings of the CNN model were determined as 96 filters, kernel size 3, activation function tanh, output activation function in the hidden layer sigmoid, batch size 10, epoch number 6. The LSTM model has an LSTM layer with 32 LSTM units, followed by a 24-unit hidden layer with activation function sigmoid and a hidden layer with output activation function sigmoid, batch size is 10, epoch number is 6. The GRU model has a GRU layer with 32 GRU units, followed by a 24-unit hidden layer with activation function relu and a hidden layer with output activation function sigmoid, batch size is 10, epoch number is 6.

TABLE I. BEST RESULTS OBTAINED IN IRONY DETECTION

Model	**Evaluation Metrics (%)**			
	Accuracy	***Precision***	***Recall***	***F_1 Score***
Text-davinci-003[a]	69.9	58.4	83.9	68.9
Gpt-3.5-turbo[b]	69.0	58.7	73.6	65.3

a. Few-shot learning.
b. Zero-shot learning.

TABLE II. BEST RESULTS OBTAINED IN IRONY CLASSIFICATION

Model	**Evaluation Metrics (%)**			
	Accuracy	***Precision***	***Recall***	***F_1 Score***
Text-davinci-003[a]	61.4	44.5	46.5	45.3
Gpt-3.5-turbo[a]	49.5	40.3	40.1	36.3

a. Few-shot learning.

In the GloVe text embedding model glove.6b.100d, while the vector was created according to the length of the sentence, this length was shortened in various preprocessing and padding was made at the front or the end of the sentence. A 100-dimensional vector was created for each word. As the number of words in the text increases, the number of vectors created also increases. In the text-embedding-babbage-001 and text-embedding-ada-002 models, one vector of 2048 and 1536 lengths, respectively, was created for each text.

The best results obtained by each model for each learning algorithm are shown in the Table III for irony detection and Table IV for irony classification.

CNN, LSTM and GRU models, when used in natural language processing tasks, classify each word based on the relationship it establishes with the surrounding words. These models work well when GloVe text embedding is used, as each word is represented by a vector of a certain length. However, when text-embedding-babbage-001 and text-embedding-ada-002 text embeddings are used, only one vector is formed for the entire sentence, so the effect of words on each other cannot be adequately understood by the model, therefore consistent and successful results cannot be obtained, thus no results could be entered.

TABLE III. BEST RESULTS OBTAINED BY LEARNING ALGORITHMS IN IRONY DETECTION

Algorithm	**Evaluation Metrics (%)**					
	Glove.6b.100d		***Text-embedding-babbage-001***		***Text-embedding-ada-002***	
	Acc.	***F_1***	***Acc.***	***F_1***	***Acc.***	***F_1***
MLP	48.1	51.9	**72.1**	65.2	69.3	**66.3**
CNN	**64.2**	**62.3**	-	-	-	-
LSTM	**66.8**	**61.2**	-	-	-	-
GRU	**64.4**	**62.7**	-	-	-	-
RF	62.6	57.7	**70.7**	**64.5**	69.6	63.4
KNN	49.5	53.3	**67.3**	**62.5**	65.8	62.1
DT	59.6	**60.4**	**62.0**	55.4	60.3	56.6
AdaBoost	61.1	59.3	**67.2**	**60.9**	65.6	59.8
GNB	45.0	57.3	**67.9**	61.1	65.6	**63.6**

TABLE IV. BEST RESULTS OBTAINED BY LEARNING ALGORITHMS IN IRONY CLASSIFICATION

Algorithm	**Evaluation Metrics (%)**					
	Glove.6b.100d		***Text-embedding-babbage-001***		***Text-embedding-ada-002***	
	Acc.	***F_1***	***Acc.***	***F_1***	***Acc.***	***F_1***
MLP	43.1	27.9	**68.4**	**36.3**	67.1	35.8
CNN	**56.0**	**38.9**	-	-	-	-
LSTM	**61.0**	**39.3**	-	-	-	-
GRU	**60.7**	**38.2**	-	-	-	-
RF	56.6	27.6	**68.9**	**36.3**	68.1	35.5
KNN	44.4	24.9	**63.8**	**40.7**	58.5	37.6
DT	58.7	28.5	**59.8**	**29.7**	59.7	29.4
AdaBoost	**58.0**	31.4	57.4	**39.6**	57.1	39.0
GNB	30.9	19.5	51.5	45.0	**62.5**	**48.5**

Apart from CNN, LSTM and GRU algorithms, GloVe text embedding model was only able to achieve a more successful F_1 score in the DT algorithm of the binary classification task. Moreover, in any model where GloVe text embedding is used, neither binary classification nor multiclass classification has been able to give successful results from text-embedding-babbage-003 and text-embedding-ada-002 models.

The most successful model in binary classification task is MLP model, which uses the text-embedding-ada-002 model as the text embedding model with preprocessing of the tweet text as following. Url and smileys have been deleted from the tweet. Mentions have been converted to $MENTION$ token. Numbers, reserved words, hashtags, and emojis in text format are kept as they were. The '->' sign has been added to tweets after each text. On the other hand, the most successful model in the multiclass classification task is the GNB model, which uses the text-embedding-ada-002 as the text embedding model for texts without any preprocessing.

V. DISCUSSION AND RESULTS

In this section, the best results of all models used in the study were compared with each other and their superiority to each other was discussed. Then, GPT models with the best results and the learning algorithms using GPT text embedding models were compared with the models with the best results in SemEval-2018 Task 3 competition and then some successful models using this dataset.

Table V shows the best results obtained in irony detection task in each specific category. GRU model, which achieved the best results by using GloVe text embedding model in detecting irony, did not achieve as much success as MLP model, which had better results using GPT text embedding models. Text-davinci-003, on the other hand, achieved the most successful F_1 score, and gpt-3.5-turbo model, which works much more cost-efficient than the former, also achieved comparable results.

Table VI shows the best results obtained in irony classification task in each specific category. LSTM model, which achieved the best results by using GloVe text embedding model in classification according to irony types, did not achieve as much success as GNB model, which got better results using GPT text embedding models. However, it managed to get better results than the gpt-3.5-turbo model. Text-davinci-003 model was more successful in classifying irony types, just as it was in the task of irony detection. GNB model, which uses text-embedding-ada-002 text embedding model, had the highest F_1 score obtained within the scope of this study in the classification of irony.

TABLE V. BEST RESULTS IN IRONY DETECTION

Model	**Evaluation Metrics (%)**	
	Accuracy	***F_1 Score***
Text-davinci-003[a]	69,9	**68,9**
text-embedding-ada-002 - MLP	69,3	66,3
gpt-3,5-turbo[b]	69,0	65,3
text-embedding-babbage-001 - MLP	**72,1**	65,2
GloVe - GRU	64,4	62,7

a. Few-shot learning.
b. Zero-shot learning.

TABLE VI. BEST RESULTS IN IRONY CLASSSIFICATION

Model	Evaluation Metrics (%)	
	Accuracy	*F_1 Score*
text-embedding-ada-002 - GNB	**62,5**	**48,5**
Text-davinci-003[a]	61,4	45,3
text-embedding-babbage-001 – GNB	51,5	45,0
GloVe - LSTM	61,0	39,3
gpt-3,5-turbo[a]	49,5	36,3

a. Few-shot learning.

According to Table VII, in binary classification task, the text-davinci-003 model which achieved the best result in this study, achieved a higher F_1 score than most of the models used in the official competition and models suggested after the competition, while achieving comparable success with the best results. The recall value of this model is the best of all results.

As can be seen from Table VIII, in multiclass classification task, GNB algorithm using text-embedding-ada-002 embedding model achieved the fourth best F_1 score which is slightly lower than the better models and got third best recall value after the best resulting model and text-embedding-babbage-001 model with GNB algorithm which received second best recall result.

TABLE VII. COMPARISONS OF THE RESULTS IN IRONY DETECTION

Model	Evaluation Metrics (%)			
	Accuracy	*Precision*	*Recall*	*F_1 Score*
[21]	**79.7**	**78.8**	66.9	**72.4**
[19]	73.5	63.0	80.1	70.5
[15]	68.9	-	-	69.6
[11] TE-NoPos	67.0	-	-	69.5
[9] STBi-LSTM	-	-	-	69.0
text-davinci-003[a]	69.9	58.4	**83.9**	68.9
[11] Att-LSTM	68.8	-	-	68.2
[9] AABi-LSTM	-	-	-	67.9
[18]	73.2	65.3	69.1	67.2
text-embedding-ada-002 - MLP	69.3	58.7	76.2	663
[10]	-	-	-	66.0
gpt-3.5-turbo[b]	69.0	58.7	73.6	65.3
[9] SABi-LSTM	-	-	-	65.3
text-embedding-babbage-001 - MLP	72.1	64.5	65.9	65.2
[20]	64.3	53.2	83.6	65.0
[17]	66.1	55.1	78.8	64.8
[16]	70.2	60.9	69.1	64.8
[22] Unigram SVM BL	63.5	53.2	65.9	58.9
[22] Random BL	50.3	37.3	37.3	37.3

a. Few-shot learning.
b. Zero-shot learning.

TABLE VIII. COMPARISONS OF THE RESULTS IN IRONY CLASSIFICATION

Model	Evaluation Metrics (%)			
	Accuracy	*Precision*	*Recall*	*F_1 Score*
[21]	**73.2**	**57.7**	50.4	**50,7**
[18]	65.2	49.6	51.2	49,6
[19]	60.5	48.6	**54.1**	49,5
text-embedding-ada-002 - GNB	62.5	47.7	51.6	48,5
[17]	60.3	46.6	50.6	47,4
text-davinci-003[a]	61.4	44.5	46.5	45,3
Text-embedding-babbage-001 - GNB	51.5	45.2	52.9	45,0
[16]	65.9	54.5	44.8	44,4
[20]	67.1	43.1	41.5	41,5
gpt-3,5-turbo[a]	49.5	40.3	40.1	36,3
[22] Unigram SVM BL	56.9	41.6	36.4	34,1
[22] Random BL	41.6	24.1	24.1	24,1

a. Few-shot learning.

VI. CONCLUSION

In this study, the success of GPT models in irony detection and classification tasks was examined. For this purpose, this was observed in two different parts.

In the first part, text-davinci-003 and gpt-3.5-turbo models, which are among the most successful and advanced GPT models available, were used with both zero-shot learning and few-shot learning methods. In this part of the study, there was no use of the training set of the dataset, except for 42 samples that were used only in the few-shot learning method.

In the other part, text-embedding-babbage-001 and text-embedding-ada-002 text embedding models, which are GPT text embedding models, were used with various machine learning and deep learning methods. For comparison, the glove.6b.100d embedding model of the GloVe embedding model, which consists of 6 billion words and 100 dimensional vectors for each word, was used.

Based on the data obtained in the study, GPT-3.5 models can achieve comparable results with the best models in a difficult task such as irony detection and classification, even if they do the whole classification process or are used in hybridization with other models. It is noteworthy that the text-davinci-003 and gpt-3.5-turbo models, although they are models based on text generation, achieve very successful results in classification tasks as well. This shows that language models pre-trained with big datasets such as GPT can be used as a single model for many different tasks, needing little or no training data. For such reasons, it stands out from the task-oriented models. Such pre-trained models, whose biggest problem is the power required for computation and the cost associated with it, expect the ever-evolving technology to reduce the computational cost in order to become widespread in a wide and diverse range of uses.

For the second part of the study, GPT text embedding models, namely text-embedding-babbage-001 and text-embedding-ada-002, proved their capabilities with different learning algorithms against GloVe text embedding model glove-6b-100d. Although these learning algorithms with GPT

text embedding models requires training data unlike plain GPT-3.5 models, some of them achieved better score than plain GPT-3.5 models.

It has been observed that the task definition given to the model can significantly affect the classification success in tasks where only GPT models are used. Further studies can be conducted on the job descriptions given to the GPT models, and research on how to improve the evaluation results can be continued and thus contribute to the literature on finding the most appropriate definition form. Since the main focus in the second part of the study is comparison between GPT text embedding models and GloVe text embedding model, no hyperparameter tuning made in the algorithms from Scikit-learn library. So in future studies with GPT embedding models instead of using default parameters, different hyperparameter settings can be studied.

ACKNOWLEDGMENT

This study was carried out within the scope of the master's thesis.

REFERENCES

[1] Toçoğlu, M. A., Alpkocak, A. (2019). Lexicon-based emotion analysis in Turkish. Turkish Journal of Electrical Engineering and Computer Sciences, 27(2), 1213-1227.

[2] Tocoglu, M. A., Ozturkmenoglu, O., Alpkocak, A. (2019). Emotion analysis from Turkish tweets using deep neural networks. IEEE Access, 7, 183061-183069.

[3] Jabreel, M., Moreno, A. (2019). A deep learning-based approach for multi-label emotion classification in tweets. Applied Sciences, 9(6), 1123.

[4] El-Alami, F. Z., El Alaoui, S. O., Nahnahi, N. E. (2022). A multilingual offensive language detection method based on transfer learning from transformer fine-tuning model. Journal of King Saud University-Computer and Information Sciences, 34(8), 6048-6056.

[5] Sailunaz, K., Alhajj, R. (2019). Emotion and sentiment analysis from Twitter text. Journal of Computational Science, 36, 101003.

[6] Singla, C., Al-Wesabi, F. N., Pathania, Y. S., Alfurhood, B. S., Hilal, A. M., Rizwanullah, M., Hamza, M. A., Mahzari, M. (2022). An Optimized Deep Learning Model for Emotion Classification in Tweets. Computers, Materials & Continua, 70(3), 6365-6380.

[7] Ravi, K., Ravi, V. (2017). A novel automatic satire and irony detection using ensembled feature selection and data mining. Knowledge-based systems, 120, 15-33.

[8] Sulis, E., Farías, D. I. H., Rosso, P., Patti, V., Ruffo, G. (2016). Figurative messages and affect in Twitter: Differences between# irony,# sarcasm and# not. Knowledge-Based Systems, 108, 132-143.

[9] Zhang, S., Zhang, X., Chan, J., Rosso, P. (2019). Irony detection via sentiment-based transfer learning. Information Processing & Management, 56(5), 1633-1644.

[10] Chia, Z. L., Ptaszynski, M., Masui, F., Leliwa, G., Wroczynski, M. (2021). Machine Learning and feature engineering-based study into sarcasm and irony classification with application to cyberbullying detection. Information Processing & Management, 58(4), 102600.

[11] González, J. Á., Hurtado, L. F., Pla, F. (2020). Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. Information Processing & Management, 57(4), 102262.

[12] Gonzalez, J. A., Hurtado, L. F., Pla, F. (2021). TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter. Neurocomputing, 426, 58-69.

[13] Karoui, J., Zitoune, F. B., Moriceau, V. (2017). Soukhria: Towards an irony detection system for arabic in social media. Procedia Computer Science, 117, 161-168.

[14] Charalampakis, B., Spathis, D., Kouslis, E., Kermanidis, K. (2016). A Comparison Between Semi-supervised and Supervised Text Mining Techniques on Detecting Irony in Greek Political Tweets. Engineering Applications of Artificial Intelligence, 51, 50-57.

[15] Shirivastava, M., Kumar, S. (2021). A Pragmatic and Intelligent Model for Sarcasm Detection in Social Media Text. Technology in Society, 64, 101489.

[16] Vu, T., Nguyen, D. Q., Vu, X. S., Nguyen, D. Q., Catt, M., Trenell, M. (2018). NIHRIO at SemEval-2018 Task 3: A Simple and Accurate Neural Network Model for Irony Detection in Twitter. In Proceedings of the 12th international workshop on semantic evaluation (pp. 525-530).

[17] Rangwani, H., Kulshreshtha, D., Singh, A. K. (2018, June). NLPRL-IITBHU at SemEval-2018 Task 3 Combining Linguistic Features and Emoji pre-trained CNN for Irony Detection in Tweets. In Proceedings of the 12th international workshop on semantic evaluation (pp. 638-642).

[18] Baziotis, C., Athanasiou, N., Papalampidi, P., Kolovou, A., Paraskevopoulos, G., Ellinas, N., Potamianos, A. (2018). NTUA-SLP at SemEval-2018 Task 3: Tracking Ironic Tweets using Ensembles of Word and Character Level Attentive RNNs. In Proceedings of the 12th international workshop on semantic evaluation (pp. 613-621).

[19] Wu, C., Wu, F., Wu, S., Liu, J., Yuan, Z., Huang, Y. (2018, June). THU_NGN at SemEval-2018 Task 3: Tweet Irony Detection with Densely Connected LSTM and Multi-task Learning. In Proceedings of The 12th International Workshop on Semantic Evaluation (pp. 51-56).

[20] Rohanian, O., Taslimipoor, S., Evans, R., Mitkov, R. (2018, June). WLV at SemEval-2018 Task 3: Dissecting Tweets in Search of Irony. Association for Computational Linguistics. In Proceedings of the 12th International Workshop on Semantic Evaluation (pp. 553-559).

[21] Ghosh, A., Veale, T. (2018, June). IronyMagnet at SemEval-2018 Task 3: A Siamese Network for Irony Detection in Social Media. In Proceedings of the 12th International Workshop on Semantic Evaluation (pp. 570-575).

[22] Van Hee, C., Lefever, E., Hoste, V. (2018, June). SemEval-2018 Task 3: Irony Detection in English Tweets. In Proceedings of the 12th International Workshop on Semantic Evaluation (pp. 39-50).

[23] Radford, A., Narasimhan, K., Salimans, T., Sutskever, I. (2018). Improving language understanding by generative pre-training.

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

[25] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al., (2020). Language models are few-shot learners. Advances in neural information processing systems, 33, 1877-1901.