Affective and Contextual Embedding for Sarcasm Detection

Nastaran Babanejad, Heidar Davoudi, Aijun An, Manos Papagelis
Department of Electrical Engineering and Computer Science,
York University, Toronto, Canada
Faculty of Science, Ontario Tech University, Oshawa, Canada
{nasba, aan, papaggel}@eecs.yorku.ca
{heidar.davoudi}@uoit.ca

Abstract

Automatic sarcasm detection from text is an important classification task that can help identify the actual sentiment in user-generated data, such as reviews or tweets. Despite its usefulness, sarcasm detection remains a challenging task, due to a lack of any vocal intonation or facial gestures in textual data. To date, most of the approaches to addressing the problem have relied on hand-crafted affect features, or pre-trained models of non-contextual word embeddings, such as Word2vec. However, these models inherit limitations that render them inadequate for the task of sarcasm detection. In this paper, we propose two novel deep neural network models for sarcasm detection, namely ACE 1 and ACE 2. Given as input a text passage, the models predict whether it is sarcastic (or not). Our models extend the architecture of BERT by incorporating both affective and contextual features. To the best of our knowledge, this is the first attempt to directly alter BERT's architecture and train it from scratch to build a sarcasm classifier. Extensive experiments on different datasets demonstrate that the proposed models outperform state-of-the-art models for sarcasm detection with significant margins.

1 Introduction

Sarcasm is the use of language in which one conveys implicit information/intention with the opposite meaning of what is said or written. Due to this deliberate ambiguity, sarcasm detection is a challenging task, especially in written expressions where *body gestures*, *tone of voice*, and *facial expression* are not known (Shivhare and Saritha, 2014; Joshi et al., 2015). Sarcasm detection has attracted growing interest over the past decade as it draws a more accurate picture of users' intention on social media (Carvalho et al., 2009), and facilitates accurate sentiment analysis in online comments and reviews (Forslid and Wikén, 2015). It also has useful applications in areas such as healthcare (Channon et al., 2005), hate speech detection (Mozafari et al., 2019; Djuric et al., 2015), disaster management (Forslid and Wikén, 2015).

Early attempts of sarcasm detection from text mainly relied on extracting a set of positive verbs and negative/undesirable situations (e.g. "I love [positive verb] the pain of breakup [negative situation]") (Riloff et al., 2013; González-Ibáñez et al., 2011). Alternatively, one may use lexical features (e.g., capital letters, and excessive usage of exclamatory marks) (Lunando and Purwarianti, 2013) in sarcasm detection. Recently, psychological studies have showed a strong relationship between affect/sentiment features (e.g., sadness, happiness) and sarcasm (Huang et al., 2015; Pickering et al., 2018). However, relying only on affect/sentiment features for sarcasm detection may not be effective, especially when there are no sentiment words in a sentence (Joshi et al., 2016; Riloff et al., 2013). For instance, in the sentence "Is it time for your medication or mine?", the speaker's intention is to mock the person addressed, but there aren't any sentiment words used.

Later attempts for sarcasm detection mostly relied on language models that are based on continuous representation or embeddings of words, such as Word2vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). Use of these general models can eliminate the need for feature engineering or dependence on large emotion labeled datasets. However, due to the mechanism with which word vectors are learned

Proceedings of the 28th International Conference on Computational Linguistics, pages 225–243
Barcelona, Spain (Online), December 8-13, 2020

and embedded to a space, these models have been shown to be inadequate for affective tasks (Amir et al., 2016). For instance, two dissimilar words like "good" and "bad", which often occur in a similar context, will be embedded closer to each other than the words "good" and "happy" that express similar emotions. More recent advances in neural representation models, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) can overcome this limitation by taking into account both the context a word appears in and its importance in that context, using a self-attention mechanism (Vaswani et al., 2017). While these models have been applied successfully to a wide range of Natural Language Processing (NLP) tasks, such as sentiment analysis (Sun et al., 2019) and word similarity computation (Zhang et al., 2019b), they have not been fully exploited for the more challenging problem of sarcasm detection. A few studies that proposed to use these models (Mozafari et al., 2019; Castro et al., 2019), utilize the already pre-trained embeddings, which are not optimized for sarcasm detection, and their performance can be further improved (Potamias et al., 2019; Ghosh et al., 2020).

In this paper, we propose two novel models for sarcasm detection based on Affective and Contextual Embeddings, namely **ACE 1** and **ACE 2**. Given as input a text passage (i.e., a sequence of sentences, which we call a *document* in this paper for brevity), the models predict whether it is sarcastic. The architecture of each model builds upon two components: a) *affective feature embedding*, and b) *contextual feature embedding*. The former utilizes a Bi-LSTM with multi-head attention neural architecture (Vaswani et al., 2017) to obtain representations of *affective features* of a document. The latter is achieved by a BERT model. In ACE 1, the two components are combined by training a new BERT model from scratch by adding affective feature embeddings into the input sequence of BERT so that task-specific embeddings can be obtained. In ACE 2, the two components are combined in a fully connected layer with a softmax to form a classifier that is trained with labeled sarcasm detection data. The main contributions of this work are as follows:

- We present two novel deep neural network language models (ACE 1 and ACE 2) for sarcasm detection. Each model extends the architecture of BERT by incorporating both affective and contextual features of text to build a classifier that can determine whether a document is sarcastic or not. To the best of our knowledge, this is the first attempt to directly alter BERT's architecture and train it from the ground-up (rather than using the already pre-trained BERT embeddings) for sarcasm detection.
- Integral to our proposed models is a novel model that learns the affective representation of a document, using a Bi-LSTM architecture with multi-head attention. The resulting representation takes into account the importance of the affect representations of the sentences in the document.
- We design and evaluate alternatives that materialize each of the two components (affective feature embedding and contextual feature embedding) of the proposed deep neural network architecture model. We systematically evaluate the effectiveness of each alternative architecture.
- We conduct an extensive evaluation of the performance of the proposed models (ACE 1 and ACE 2), which demonstrates that they significantly outperform current state-of-the-art models for sarcasm detection.
- We make *source code* and *data* publicly available to encourage result reproducibility and model re-use[1].

The paper is organized as follows. Section 2 presents the related work and Section 3 presents the proposed models. Section 4 describes the evaluation and results. We conclude the paper in Section 5.

2 Related Work

In this section, we present an overview of the related work on sarcasm detection including models that use *affective features*, *contextual information*, or a combination of the two. We also explain how our work aims to bridge the gap among existing efforts. Due to space limitations, we only provide a short literature

[1] https://github.com/NastaranBa/ACE-for-Sarcasm-Detection

review here. For a more comprehensive coverage, we refer interested readers to the Supplementary Material (Appendix A.1). Other important related work is also cited in context, throughout the manuscript.

Identifying sarcasm in text has evolved from simple lexical-based and syntactic pattern models (Tsur et al., 2010; Davidov et al., 2010; González-Ibáñez et al., 2011) to complex models that consider refined linguistic features, such as positive predicates, interjections, gestural cues (emoticons, quotation marks, etc.) (Carvalho et al., 2011) or behavior modelling (Rajadesingan et al., 2015; Agrawal et al., 2020). With the advent of deep learning, there has been a shift in how prediction models are designed for sarcasm detection. Ghosh et al. (2017) proposed a conditional LSTM network (Hochreiter and Schmidhuber, 1997) for detecting sarcasm in twitter using sentence-level attention mechanisms on hashtags and (Hernández Farías et al., 2018) utilized a knowledge-based model of affective features based on a wide range of lexical resources. While these models mostly relied on manually engineered affective features, Zhang et al. (2019a) proposed two Bi-LSTM models to learn to identify sarcasm in tweets. A drawback of many of these models is that they rely on self-contained content (e.g. twitter hashtags #) and thus do not generalize well, i.e., when such content is not available, the model fails to detect sarcasm.

To address issues of lack of generalization and manual feature engineering, general word representation learning models have been proposed, such as Word2vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). These models learn embeddings of words that locate them close to one another in the embedded space, if they share common contexts in the corpus. However, this means that even words opposite in meaning to another (*antonyms*), such as *happy* and *sad* that often occur in similar contexts, would be embedded close to each other. As a result, the use of general models can be inadequate for affective tasks (Amir et al., 2016). More recently, transformer-based models such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) have been proposed to advance the state of the art in many NLP tasks by combining word embeddings with context embedding by using attention mechanisms in a bidirectional manner. Potamias et al. (2019) proposed RCNN-RoBERTa for sarcasm detection in social media by leveraging the pre-trained embeddings from RoBERTa combined with a recurrent convolutional neural network. Babanejad et al. (2020) investigated the impact of data pre-processing on word representation learning for affective tasks. The main drawback of using advanced pre-trained models is that they do not incorporate any affect-specific features during the training phase of the model – therefore the accuracy of the model on affective tasks can be lessened.

More sophisticated approaches for sarcasm detection have tried to combine the best of the two worlds by incorporating affective features with general embedding models during training. For example, Felbo et al. (2017) proposed DeepMoji by training a Bi-LSTM model with emojis to learn representations of emotional context, Poria et al. (2016) developed pre-trained sentiment, emotion and personality models for identifying sarcastic text, and (Tay et al., 2018) utilized a multi-dimension intra-attention mechanism to overcome limitations of sequential models and capture words' incongruities for sarcasm detection. Agrawal and An (2018) incorporated affective information into word representations by training a Bi-LSTM using corpora with weak affect labels and used such representations for sarcasm detection. Another important observation for the problem of sarcasm detection is that domain dataset is critical (Rakov and Rosenberg, 2013). For instance, previous studies (Joshi et al., 2015; Zhang et al., 2014) have employed datasets related to comedy to improve on the sarcasm detection task, a literary genre and a type of dramatic work that is often satirical in its tone. Our research is mostly related to this line of work. In particular, we advance the state of the art in sarcasm detection by combining recently proposed transformer-based language models, such as BERT and SBERT, with affective-specific features that leverage domain data of sarcasm-rich corpora.

3 Proposed Models for Sarcasm Detection

We propose two models, ACE 1 and ACE 2, for sarcasm detection, where each model takes a document (i.e., a sequence of sentences) as input and predicts whether the document is sarcastic or not. Figure 1 and 2 depict the architecture of each model that builds upon two components: a) *affective feature embedding (AFE)* (on the right), and b) *contextual feature embedding (CFE)* (on the left). The two models are different in (1) the way the two components are combined and (2) the input to the affective feature

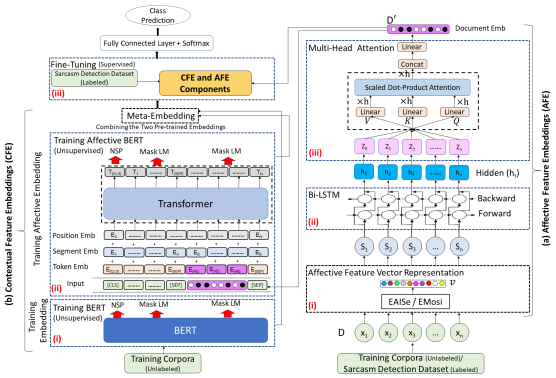

Figure 1: Overview of the proposed model ACE 1

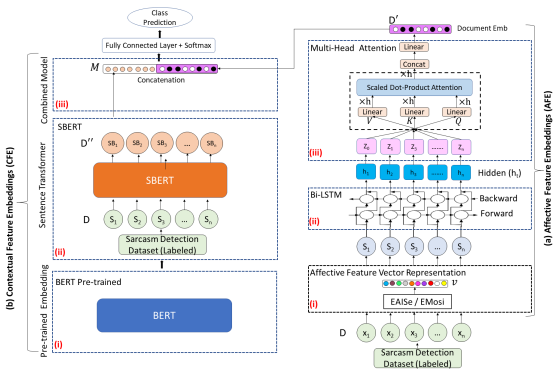

Figure 2: Overview of the proposed model ACE 2

embedding component.

3.1 Affective Feature Embedding (AFE)

The architecture of the AFE component is the same for ACE 1 and ACE 2. The difference is that its input in ACE 1 is an unlabeled training corpus in pre-training and a labeled sarcasm dataset during fine-tuning, while in ACE 2, AFE only takes the labeled sarcasm detection dataset as the input. This component includes three stages: (i) Affective Feature Vector Representation, (ii) Bi-LSTM and (iii) Multi-Head Attention layers.

3.1.1 Affective Feature Vector Representation

In this stage, each input document is first chunked into sentences. Then, the affective features are extracted using one of the following two approaches.

Emotion Affective Intensity with Sentiment Feature (EAISe): We use the NRC Emotion Intensity Lexicon (Mohammad, 2017) to extract the emotion words in a sentence and give each such word 4 intensity scores, one for each of 4 basic emotions: anger, fear, sadness, joy. Each score ranges from 0 to 1, where "1" means that the word conveys the highest degree of the corresponding emotion, and "0" means that the word is not associated with the emotion. Then, we add 2 more binary scores to represent sentiment (positive, negative) of the word based on the NRC Emotion Lexicon (Mohammad and Turney, 2013). To calculate the *affective feature vector of a sentence*, we first average the affective feature vectors of the affect words in the sentence, then multiply (element-wise) it with a vector v' that contains the frequency of words in the sentence in each emotion or sentiment. For instance, assume that we have 3 affect words in a sentence, and the affective feature vectors of 4 emotions and 2 sentiments (anger, fear, sadness, joy, positive, negative) for these words are: $w_{tragedy} = (0, 0.73, 0.61, 0, 0, 1)$, $w_{thanksgiving} = (0, 0, 0, 0.64, 1, 0)$, $w_{happy} = (0, 0, 0, 0.82, 1, 0)$. The element-wise multiplication of the average of these 3 vectors $(0, 0.24, 0.20, 0.48, 0.66, 0.33)$ and the frequency vector $v\prime = (0, 1, 1, 2, 2, 1)$ is $(0, 0.24, 0.20, 0.96, 1.32, 0.33)$.

Emotion Similarity Feature (EMoSi): In this approach, for each word in a sentence, we measure the average semantic similarity score between the words in the sentence and all seed words (20 words) of an emotion in the NRC Emotion Lexicon (Mohammad and Turney, 2013) that has 8 emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust). The semantic similarity score is based on the cosine similarity of pre-trained Word2vec vectors (Mikolov et al., 2013a; Babanejad et al., 2020) of the corresponding words. For instance, in a sentence "I love jogging.", we calculate the cosine similarity between word "I" with a seed word "happy" in emotion "joy". We do this for all the seed words in each emotion, resulting in an emotion intensity vector of 8 scores for each word in the sentence. By averaging the vectors for all the words in the sentence, we obtain an affective feature representation of the sentence.

Given a document D with n sentences $(x_1, x_2, \cdots, x_n)$, x_i is converted into its affective vector representation S_i using one of the above two approaches. Thus, document D can be represented as $(S_1, S_2, .., S_n)$.

3.1.2 Bi-LSTM Layer

Given a document $D = (S_1, S_2, .., S_n)$, we use a Bi-LSTM model (Graves and Schmidhuber, 2005) to capture/encode the affect-changing information of the sentence sequence from both left and right directions[2]. The result is a sequence of hidden state vectors h_t for the sentence sequence. More precisely, the bidirectional LSTM is a concatenation of the forward LSTM: $\overrightarrow{h}_t = \overrightarrow{LSTM}(S_t, \overrightarrow{h}_{t-1})$ and backward LSTM: $\overleftarrow{h}_t = \overleftarrow{LSTM}(S_t, \overleftarrow{h}_{t+1})$ and the concatenation is $h_t = Concat(\overrightarrow{h}_t, \overleftarrow{h}_t)$. The sequence of hidden state vectors h_t ($t = 1, 2, \ldots n$) forms a matrix and serves as the input for the next layer (Multi-head Attention Layer).

3.1.3 Multi-head Attention Layer

In a given document, a specific part could play a more important role in detecting sarcasm (Kumar et al., 2020). Therefore, we use a multiple heads attention mechanism (Vaswani et al., 2017) to capture the importance of hidden affective feature vectors h_t, which have already been learned by the Bi-LSTM layer. This helps us capture long-distance dependencies and form a global representation of the sequence. As shown in Figure1, the output vectors of Bi-LSTM layer ($h_1, h_2, \ldots h_n$) are combined to form a matrix $H = [h_1, h_2, ..h_n]$ which serves as the input matrix for each attention head in the *self-attention* mechanism. In particular, the three matrices Q(query), K(key), and V(value) are created by multiplying H with the weight matrices (W_Q, W_K, W_V) that are trained jointly in the *self-attention* mechanism (Vaswani et al., 2017) as such: $Q = H \times W^Q$, $K = H \times W^K$, and $V = H \times W^V$. Then, this mechanism calculates the context vectors for each self-attention head as $Z = SDPA(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_K}}\right)V$. This is called Scaled Dot-Product Attention (SDPA) in (Vaswani et al., 2017) and $\sqrt{d_K}$ is a scaling factor where d_K is the dimension of queries and keys. Assuming there are N heads, we have N such sets of weight matrices and we compute the output for the $i'th$ head as $Z(head_i) = SDPA(QW_i^Q, KW_i{}^K, VW_i^V)$. The Multi-Head Attention mechanism (MHA) runs through SDPA multiple times in parallel and concatenates the resulting vectors of all the heads and multiply it by an additional weight matrix W^O that was trained jointly with the model as the final representation of the document D as $D' = MHA(Q, K, V) = Concat(Z_1(head_1), ..., Z_n(head_N)W^O$

3.2 Contextual Feature Embedding in ACE 1

We explain how our first model, ACE 1, incorporates the affective feature embedding discussed in Section 3.1 into the BERT model for sarcasm detection. The architecture of the ACE 1 is illustrated in Figure 1, which includes three stages: i) Training BERT, ii) Training Affective BERT and iii) Fine-Tuning.

3.2.1 Training BERT

In this stage, we train a BERT model using the BERT-Large-uncased architecture with the same setting for hyper-parameters as in (Devlin et al., 2018)[3]. The model is trained on an unlabeled text corpus over two unsupervised tasks: *i*) *Masked Language Model* (MLM), in which, some of the tokens in the input sequences are masked and the model is trained to predict these masked tokens, and *ii*) *Next sentence Prediction* (NSP), where the model receives pairs of sentences as input and learns to predict if the second sentence in a pair is the subsequent sentence of the first one in the training corpus. The two tasks are trained together, with the goal of minimizing the combined loss function to generate the final embedding vectors.

More specifically, the training corpus is tokenized with the WordPiece method (Wu et al., 2016)) and then input sequences are generated, where 50% are a pair in which the second sentence is the subsequent sentence in the corpus, and the other 50% contains a random sentence from the corpus as the second sentence. Each input sequence has a [CLS] token at the beginning and a [SEP] token at the end of each sentence. The middle part of Figure 3 illustrates the BERT input representation. BERT has three embedding layers: (i) Token Embedding: it transforms tokens into vector representations of fixed dimension from the WordPiece token vocabulary (ii) Segment Embedding: it discerns between the first

[2]We use post-zero-padding , and using attention masking to distinguish the padding (Vaswani et al., 2017)

[3]More details on used parameters can be found in Appendix A.5 of supplementary file.

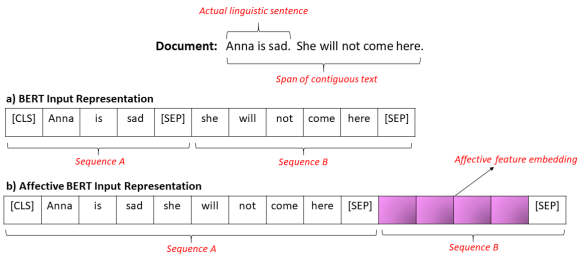

Figure 3: BERT Input Representation Vs Affective BERT Input Representation.

and second sequence to indicate whether the token belongs to the first or the second sentence in the input sequence, and (iii) Position Embedding: it remembers the position of each token in a sequence. These three embeddings are summed up (element-wise) and make up the input to the BERT bidirectional transformer which is a multi-layer bidirectional transformer encoder (Devlin et al., 2018; Vaswani et al., 2017).

3.2.2 Training Affective BERT with Affective Feature Embeddings

The purpose of this stage is to incorporate the affective features into the BERT model so that task-specific embeddings can be obtained. As illustrated in Figure 1, we train a new model (called Affective BERT) from scratch using BERT by adding affective feature embeddings obtained from the AFE component into the input sequence of BERT. Again, we use the BERT-Large-uncased architecture. The unlabeled training corpus is first tokenized using the WordPiece method. The difference between this BERT model and the one trained in the first stage is in the input sequence. The bottom part of Figure 3 illustrates an input sequence in this stage. An input sequence contains two subsequences. The first one (between [CLS] and the first [SEP]) is a document (i.e., a sequence of tokens) and the second subsequence (purpule cells between two [SEP] tokens) is the affective feature embedding of the document, which is the D' vector generated by the AFE component trained earlier using the training corpus. The BERT model is trained using two tasks as usual: *Masked Language Model* (MLM) and *Next Sentence Prediction* (NSP). Since our input sequence contains a document and its affective feature embedding, the NSP task is actually to predict the affective features of a document.

Now, we have two contextual pre-trained embeddings for each token: A from the first stage (Training BERT) and B from the second stage (Training Affective BERT), both with the same dimension. While there are different ways to combine these two embeddings to achieve a meta-embedding (Kiela et al., 2018; Coates and Bollegala, 2018; Peters et al., 2019), we combine these two contextual embeddings by a simple concatenation to obtain the final embedding $C = A \oplus B$ for a token.

3.2.3 Fine-tuning BERT Models

In this stage, the two trained BERT models and the trained AFE component are further combined by adding a fully-connected output layer with a softmax on top of the two BERT models. The [CLS] token representations from the two BERT models are fed into the output layer. This whole ACE 1 model is then fine-tuned with a labeled data set for sarcasm detection, in which all parameters are adjusted. After fine tuning, the two BERT models can be used to perform sarcasm detection given a new document.

3.3 Contextual Feature Embedding in ACE 2

Figure 2 illustrates the architecture of our second model (i.e., ACE 2). This model also contains 2 components: a) *affective feature embedding* (AFE), which is the same as the one in ACE 1 except that the input data are different, and b) *contextual feature embedding* (CFE), to be described in detail below. The purpose of ACE 2 is to avoid the time-consuming embedding-training with very large corpora. For this purpose, we use pre-trained BERT to obtain contextual embeddings, which is called the *feature-based* approach to using BERT (Devlin et al., 2018). For this purpose, we train the AFE component using the texts in the downstream task (i.e., sarcasm detection) dataset, which is much smaller than the usual embedding-training corpus. Below we describe how the CFE component works and how the two components are combined.

3.3.1 Pre-trained Embeddings

In this stage, we use pre-trained BERT contextual embeddings (e.g., the output of the first BERT model in ACE 1, or any other pre-trained contextual embedding model) in the feature-based approach (Devlin et al., 2018) to represent each input token/sentence generated from the hidden layers of the pre-trained model.

3.3.2 Obtaining Sentence Embeddings Using SBERT

The purpose of this stage is to obtain a sentence embedding given an input sentence. The most common approaches to derive a sentence embedding from a pre-trained BERT model is to i) average the outputs of the hidden layers or ii) using the output of the first special token [CLS] (May et al., 2019; Zhang et al., 2019b; Zhao et al., 2019). However, it has been shown that these methods produce poor sentence representations that are not semantically meaningful (Reimers and Gurevych, 2019; Wang and Kuo, 2020). This is because no independent sentence embeddings are computed in the BERT model, which makes it difficult to derive sentence embeddings from pre-trained BERT. Because of this, SBERT (a Sentence Transformer) (Reimers and Gurevych, 2019) was proposed that uses a Siamese or triplet network structure to derive a sentence embedding using a pooling operation by i) computing the mean of all output vectors (MEAN-strategy), or ii) computing a max-over-time of the output vectors (MAX-strategy) from the output of pre-trained BERT.

Given an input sentence, we first use the pre-trained BERT-large-uncased model to obtain the token embeddings, which are then passed to SBERT. SBERT computes a sentence embedding using the MEAN-strategy for the pooling operation to compute a sentence embedding, which is the default mode and was also suggested in (Reimers and Gurevych, 2019) for classification tasks. For an input document, we concatenate the embeddings of all the sentences in the document to form a document representation.

3.3.3 Combining the Two Components

In this stage, a fully connected layer with a softmax is added on top of the CFE and AFE components. The input to the fully connected layer is the concatenation of the document embeddings from both CFE and AFE models (that is, the contextual embedding of a document from CFE and its affective feature embedding from AFE). The fully connected layer is trained as a classifier with the labeled dataset for sarcasm detection.

4 Experimental Evaluation

4.1 Corpora for Training Embeddings

The original BERT-Large-uncased was trained on 2.5 billion Wikipedia and 800 million BookCorpus words. Since BookCorpus is no longer publicly available, we used a news corpus along with Wikipedia to train BERT from scratch. For simplicity, through the paper we called it **Wiki**[4], in which the news corpus consists of 142,546 articles from 15 American publications and Wikipedia consists of 23,046,187 articles from Wikipedia.

To test our models with text of less formal writing styles and more sarcasm occurrences, we also created another corpus and call it **WikiSarc** that contains Wiki and the following two datasets: **IMSDB**: an Internet Movie Script Database[5], for which a scraper was used to retrieve comedy movie transcripts, resulting in 11.2 million movie transcripts, and **Riloff**: a dataset consisting of automatically extracted tweets: 60k containing the sarcasm hashtag and 100k random tweets using the method from (Riloff et al., 2013).

4.2 Sarcasm Detection Datasets

We evaluate our models on five labeled sarcasm detection datasets described in Table 1.

[4]**News**:https://www.kaggle.com/snapcrack/all-the-news,**Wikipedia**:https://www.kaggle.com/jkkphys/english-wikipedia-articles-20170820-sqlite.

[5]**IMSDB**:https://www.imsdb.com/, **Scraper**:https://github.com/JoeKarlsson/movie-script-scraper

[6]https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection

Dataset	Genre	# Sarcastic	# Non-Sarcastic	Total	Max # of Sentences
Onion[6]	News Headlines	13,634	14,985	28,619	5
Reddit (Khodak et al., 2017)	Reddit Forum	505,413	505,413	1,010,826	6
Pt′acek (Ptácek et al., 2014)	Tweets	7,000	7,000	14,000	8
SemEval-2018 (Van Hee et al., 2018)	Tweets	2,396	2,396	4,791	7
IAC (Oraby et al., 2016)	Politic Debates	1630	1630	3,260	14

Table 1: Description of sarcasm detection datasets. More details can be found in Appendix A.2

4.3 Experimental Setup

Both ACE 1 and ACE 2 models use a softmax at the output layer and cross-entropy is used as the loss function. The optimizer is Adam (Kingma and Ba, 2014). Parameter settings in the experiments are given in Appendix A.5. All the evaluation datasets are split into training and testing sets with a 80/20 split. The results on test data are reported in F1-score, defined as $2\frac{p.r}{p+r}$, where p and r are precision and recall, respectively.

4.4 Results and Analysis

4.4.1 Comparing Variations of ACE 1 and ACE 2

In this section, we compare ACE 1 and ACE 2 to see which way of fusing the CFE and AFE components is more effective, investigate the performance of the models with and without considering affective features, and also investigate which training corpus (Wiki or WikiSarc) is more effective to train embeddings for sarcasm detection. Table 2 shows the results of ACE 1 and ACE 2 on the 5 sarcasm datasets for different combinations of embedding-training corpus and affective feature representation methods. For example, in ACE 1 (Wiki) we train ACE 1 on Wiki followed by fine-tuning without incorporating affective features, while ACE 1 (Wiki)+(EAISe) means we train ACE 1 on Wiki and also incorporate the affective features of EAISe. For model ACE 2, the comparison is also among different pre-trained embeddings. For instance, ACE 2 (Wiki-BERT) + (EAISe) means the token embeddings inputted into SBERT in ACE 2 were from a BERT model pre-trained on our Wiki corpus and the affective feature representation in the AFE component is EAISe, while in ACE 2 (BERT) the embeddings inputted into SBERT are from the original pre-trained BERT (Large) model and the affective features are not used. Results of 18 variations [7] of the methods are shown in Table 2.

As shown in Table 2, overall ACE 1 outperforms ACE 2, which suggests that incorporating affective features deeply in the embedding phase is more effective than combining the pre-trained contextual embeddings with affective feature embeddings later in the classification model. Also, including affective feature embeddings works better in both models than not including them. Furthermore, using WikiSarc (which contains more sarcastic texts) as embedding-training corpus leads to better results than using Wiki. The best performance of model ACE 1 is achieved by training it on WikiSarc with affective feature EMoSi. We call the resulting BERT model WikiSarcA-BERT (Sarcastic Affective BERT). It is used as one of the pre-trained embedding models for ACE 2. Interestingly, when the models are trained on Wiki or the original BERT training corpus, there is no obvious winner between EAISe and EMoSi affective feature representation methods across the sarcasm datasets. But when WikiSarc is used as the training corpus, EMoSi is a clear winner for ACE 1 and EAISe is a clear winner on ACE 2. Our conjecture is that since WikiSarc has more sarcastic and emotional utterances than the general corpus such as Wikipedia, Book Corpus or News dataset, the affective feature representation method can make a difference in this case.

4.4.2 Evaluating Proposed Models against State-of-the-art Baselines

In this section we compare the performance of our proposed models against those of various state-of-the-art models in four different categories: (i) Only Affective, (ii) Only Contextual with Fine-Tune, (iv) Only Contextual with Pre-trained, and (iv) Affective Contextual. Some of the baseline results were taken from their original publication when the data split is the same as ours[8]. In case we cannot find the result of a baseline for a data set that we use, the baseline was properly re-implemented using the available codes and guidelines. The model in (Zhang et al., 2019a) in the "Only Affective" category was only

[7] Results of more variations are shown in Appendix A.4.

[8] The details of each baseline can be found in Appendix A.3 of the supplementary file.

Model	Corpus+Affective Feature	Onion	Reddit	Pt′acek	SemEval-2018	IAC
ACE 1	(Wiki)	83.88	80.25	71.06	77.38	85.10
	(Wiki) + (EAISe)	89.40	85.11	**75.38**	**78.10**	87.20
	(Wiki) + (EMoSi)	**90.07**	**86.20**	75.18	77.91	**88.40**
	(WikiSarc)	90.61	86.59	75.15	80.45	89.20
	(WikiSarc) + (EAISe)	90.70	87.19	77.82	84.25	89.74
	(WikiSarc) + (EMoSi)	**92.21**	**89.22**	**80.71**	**84.57**	**93.14**
Model	**Pre-trained Model+Affective Feature**	**Onion**	**Reddit**	**Pt′acek**	**SemEval-2018**	**IAC**
ACE 2	(BERT)	82.45	70.20	70.49	72.19	78.19
	(BERT) + (EAISe)	84.11	**78.79**	**72.24**	76.19	**80.36**
	(BERT) + (EMoSi)	**84.19**	77.45	71.85	**79.00**	80.00
	(Wiki-BERT)	82.31	70.20	70.04	72.10	77.48
	(Wiki-BERT) + (EAISe)	**84.33**	**79.22**	71.44	**79.61**	**80.25**
	(Wiki-BERT) + (EMoSi)	84.00	77.04	**72.10**	79.15	79.88
	(WikiSarc-BERT)	84.19	79.41	75.29	75.41	80.07
	(WikiSarc-BERT) + (EAISe)	**87.46**	82.28	**79.09**	**80.46**	**85.29**
	(WikiSarc-BERT) + (EMoSi)	86.17	**83.37**	78.19	80.11	85.10
	(WikiSarcA-BERT)	87.09	82.25	76.84	78.18	84.79
	(WikiSarcA-BERT) + (EAISe)	**90.31**	**86.50**	**80.39**	**84.33**	**88.19**
	(WikiSarcA-BERT) + (EMoSi)	88.25	86.11	79.00	83.60	86.47

Table 2: F1-scores of two models with different settings. Best results are in **bold** and 2nd best are underlined.

designed for tweets with hashtags. Thus, its results are reported only on two datasets that were from Twitter. Also, the model in (Hazarika et al., 2018) in the "Affective Contextual" category was not possible to be re-implemented for two of our datasets because the user history information embedding used in their models was not available in these datasets.

(i) **Only Affective**: We compare the AFE component of our models with those that only used hand-crafted or automatic features for sarcasm detection. To make a fair comparison, our AFE component is trained on the sarcasm dataset (not the Wiki or WikiSarc corpus which would produce better results), to be the same as the baseline methods. Table 3 shows AFE with EmoSi performs the best on 4 of 5 datasets, while the second best is AFE with EAISe. Note that the AFE results in this experiment are not as good as the ACE 2 results in Table 2, indicating combining contextual and affective features is better than using the affective features alone.

(ii) **Only Contextual with Fine-Tune**: We compare the CFE component of ACE 1 without using affective features (i.e., the stage 1 model) with those that initialize the model by pre-trained embeddings followed by fine-tuning. Among the baselines, (Potamias et al., 2019) is a transformer-based model for sarcasm detection. The other baselines are benchmark models used for a wide range of NLP tasks. Table 3 shows that ACE 1 trained on WikiSarc significantly outperforms all baselines on all five datasets, indicating that training embeddings with a corpus that contains more emotional or sarsastic untarrences is better for sarcasm detection.

(iii) **Only Contextual with Pre-trained without Fine-Tune**: We compare the CFE component of ACE 2 without incorporating affective features with those that used either contextual pre-traind embeddings (i.e., transformer-based models) or other pre-trained embeddings (e.g. GloVe used in (Zhang et al., 2016)) without fine-tuning the embeddings. Table 3 shows that ACE 2 with the pre-trained embedding model WikiSarcA-BERT consistently outperforms all the baselines. This finding supports our intuition that incorporating the affective feature information into the contextual word embeddings in the training phase (ACE 1) improves the performance in sarcasm detection, as WikiSarcA-BERT was trained in stage 2 of ACE 1.

(iv) **Affective-Contextual**: We compare our ACE 1 and ACE 2 models that combine the AFE and CFE components with those that used both affective features and pre-trained embeddings in a single architecture for sarcasm detection. The results showed that ACE 1 with WikiSarc and the EMoSi affective feature represention outperforms all the baselines, while the second best is our ACE 2 (WikiSarcA-BERT)+(EAISe) on all 5 datasets. Note that none of the existing baselines in this category uses contextual embeddings. Thus, the results suggest that using transformer-based models to generate contextual embeddings leads to better performance.

Components	Models	Onion	Reddit	Pt′acek	SemEval-2018	IAC
Only Affective	(Rajadesingan et al., 2015)	67.25	64.21	75.13	70.12	68.33
	(Ghosh et al., 2017)	69.23	68.41	74.11	72.45	64.38
	(Hernández Farías et al., 2018)	68.00	69.34	75.10	71.70	70.39
	(Zhang et al., 2019a) (a)	-	-	69.15	64.28	-
	(Zhang et al., 2019a) (b)	-	-	72.39	65.33	-
	(Zhang et al., 2019a) (c)	-	-	72.47	67.55	-
	AFE with EAISe	70.49	71.87	**76.90**	72.51	72.40
	AFE with EMoSi	**74.20**	**74.04**	76.40	**72.60**	**73.01**
Only Contextual with Fine-Tune	(Potamias et al., 2019)	84.39	78.00	71.01	70.00	85.21
	RoBERTa	80.23	76.04	67.25	68.00	82.44
	XLNet-Large	79.66	76.48	69.33	68.25	70.06
	BERT-Base	80.04	76.14	67.13	69.03	82.27
	BERT-Large	83.49	78.21	70.33	76.19	84.25
	ACE 1 (WikiSarc)	**90.61**	**86.59**	**75.15**	**80.45**	**89.20**
Only Contextual with Pre-trained without Fine-Tune	(Zhang et al., 2016)	67.08	69.20	70.49	70.66	69.38
	(Ilić et al., 2018)	70.12	76.05	75.46	68.90	72.00
	RoBERTa	76.51	66.00	62.51	66.37	75.10
	XLNet-Large	79.23	70.25	60.13	66.45	72.41
	BERT-Base	78.13	66.27	63.12	68.14	74.90
	BERT-Large	79.11	65.27	62.39	69.47	75.48
	ACE 2 (WikiSarcA-BERT)	**87.09**	**82.25**	**76.84**	**78.18**	**84.79**
Affective-Contextual	(Poria et al., 2016)	70.00	64.27	67.00	69.45	60.25
	(Amir et al., 2016)	67.79	65.14	69.25	71.59	68.51
	(Yang et al., 2016)	63.25	64.83	71.16	67.45	70.14
	(Felbo et al., 2017)	69.47	53.08	63.51	69.27	71.00
	(Wu et al., 2018)	70.00	69.20	68.50	71.20	65.23
	(Tay et al., 2018) (a)	70.68	67.25	71.52	70.01	72.00
	(Tay et al., 2018) (b)	70.13	68.23	70.13	69.46	71.85
	(Hazarika et al., 2018)	70.90	75.16	70.24	-	-
	(Kumar et al., 2020)	68.36	77.01	70.27	75.44	69.33
	ACE 1 (WikiSarc) + (EMoSi)	**92.21**	**89.22**	**80.71**	**84.57**	**93.14**
	ACE 2 (WikiSarcA-BERT) + (EAISe)	90.31	86.50	80.39	84.33	88.19

Table 3: F1-scores for comparing our models against state-of-the-art models. The best scores are in **bold**, and 2nd best are underlined, while the 3rd best are double underlined.

5 Conclusions

We proposed two novel models (ACE 1 and ACE 2) that incorporate contextual and affective features in a deep neural network architecture for sarcasm detection. Each model extends BERT's architecture by incorporating into it affective features. ACE 1 uses them to adjust the contextual embeddings and also fine-tune the model, while ACE 2 uses them along with SBERT for performing the final classification. Our evaluation results showed that combining the two types of features greatly improves the sarcasm detection accuracy. In particular, deeply incorporating the affective features in the embedding training process (as in ACE 1) is more beneficial than simply concatenating the two types of features (as in ACE 2). We also observed that training embeddings with corpora containing rich sarcastic or emotional utterances greatly benefits the sarcasm detection tasks. Our findings suggest that transformer-based models like BERT can be trained to incorporate task-specific features to improve downstream task performance. As future work, we plan to investigate whether affective and contextual embeddings, e.g., WikiSarcA-BERT trained in ACE 1, can improve the performance of other tasks, such as emotion detection.

Acknowledgements

We thank the anonymous reviewers for their insightful comments. This work is funded by Natural Sciences and Engineering Research Council of Canada (NSERC) and the Big Data Research, Analytics, and Information Network (BRAIN) Alliance established by the Ontario Research Fund Research Excellence Program (ORF-RE). We also would like to thank Majid Taghdimi from Questrade to provide us with the computing resources and help in the parallelization algorithm, and Dr. Ameeta Agrawal for collaborations and discussions on affect and sarcasm detection.

References

Ameeta Agrawal and Aijun An. 2018. Affective Representations for Sarcasm Detection. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1029–1032, Ann Arbor MI USA, June. ACM.

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2018. Learning emotion-enriched word representations. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 950–961.

Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. Leveraging transitions of emotions for sarcasm detection. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1505–1508.

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany, August. Association for Computational Linguistics.

Nastaran Babanejad, Ameeta Agrawal, Aijun An, and Manos Papagelis. 2020. A comprehensive analysis of preprocessing for word representation learning in affective tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5799–5810, Online, July. Association for Computational Linguistics.

Paula Carvalho, Luís Sarmento, Mário J. Silva, and Eugénio de Oliveira. 2009. Clues for detecting irony in user-generated contents: Oh...!! it's "so easy"FIX ME!!!!;-). In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion*, TSA '09, page 53–56, New York, NY, USA. Association for Computing Machinery.

Paula Carvalho, Luís Sarmento, Jorge Teixeira, and Mário J. Silva. 2011. Liars and saviors in a sentiment annotated corpus of comments to political debates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 564–568, Portland, Oregon, USA, June. Association for Computational Linguistics.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an-obviously-perfect paper). *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Shelley Channon, Asa Pellijeff, and Andrea Rule. 2005. Social cognition after head injury: Sarcasm and theory of mind. *Brain and Language*, 93(2):123–134, May.

Joshua Coates and Danushka Bollegala. 2018. Frustratingly easy meta-embedding – computing meta-embeddings by averaging source word embeddings. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden, July. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 29–30, New York, NY, USA. Association for Computing Machinery.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 International Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Erik Forslid and Niklas Wikén. 2015. *Automatic irony-and sarcasm detection in Social media*. Ph.D. thesis, UPPSALA UNIVERSITET.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The Role of Conversation Context for Sarcasm Detection in Online Interactions. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 186–196, Saarbrücken, Germany. Association for Computational Linguistics.

Debanjan Ghosh, Avijit Vajpayee, and Smaranda Muresan. 2020. A report on the 2020 sarcasm detection shared task. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 1–11, Online, July. Association for Computational Linguistics.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA, June. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *NEURAL NETWORKS*, pages 5–6.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Delia Irazú Hernández Farías, Manuel Montes-y Gómez, Hugo Jair Escalante, Paolo Rosso, and Viviana Patti. 2018. A knowledge-based weighted KNN for detecting Irony in Twitter. In *Proceedings17th Mexican International Conference on Artificial Intelligence, MICAI 2018*, volume 11289, pages 194–206. Springer Verlag. Accepted: 2019-04-14T18:17:25Z.

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

Li Huang, Francesca Gino, and Adam D. Galinsky. 2015. The highest form of intelligence: Sarcasm increases creativity for both expressers and recipients. *Organizational Behavior and Human Decision Processes*, 131(C):162–177.

Suzana Ilić, Edison Marrese-Taylor, Jorge Balazs, and Yutaka Matsuo. 2018. Deep contextualized word representations for detecting sarcasm and irony. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–7, Brussels, Belgium, October. Association for Computational Linguistics.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China, July. Association for Computational Linguistics.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark J. Carman. 2016. Harnessing sequence labeling for sarcasm detection in dialogue from TV series 'Friends'. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 146–155, Berlin, Germany, August. Association for Computational Linguistics.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2017. A large self-annotated corpus for sarcasm. *arXiv preprint arXiv:1704.05579*.

Douwe Kiela, Changhan Wang, and Kyunghyun Cho. 2018. Dynamic meta-embeddings for improved sentence representations. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Avinash Kumar, Vishnu Teja Narapareddy, Veerubhotla Aditya Srikanth, Aruna Malapati, and Lalita Bhanu Murthy Neti. 2020. Sarcasm Detection Using Multi-Head Attention Based Bidirectional LSTM. *IEEE Access*, 8:6388–6397. Conference Name: IEEE Access.

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.

Edwin Lunando and Ayu Purwarianti. 2013. Indonesian social media sentiment analysis with sarcasm detection. *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, September.

Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. 2019. On measuring social biases in sentence encoders. *Proceedings of the 2019 Conference of the North*.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed Representations of Words and Phrases and their Compositionality. *arXiv:1310.4546 [cs, stat]*, October. arXiv: 1310.4546.

Rishabh Misra and Prahal Arora. 2019. Sarcasm detection using hybrid neural network. *arXiv preprint arXiv:1908.07414*.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a Word–Emotion Association Lexicon. *Computational Intelligence*, 29(3):436–465, August.

Saif M. Mohammad. 2017. Word affect intensities. *ArXiv*, abs/1704.08798.

Marzieh Mozafari, Reza Farahbakhsh, and Noël Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media. *Studies in Computational Intelligence*, page 928–940, Nov.

Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. Creating and characterizing a diverse corpus of sarcasm in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles, September. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Bethany Pickering, Dominic Thompson, and Ruth Filik. 2018. Examining the emotional impact of sarcasm using a virtual environment. *Metaphor and Symbol*, 33(3):185–197, July. Publisher: Routledge _eprint: https://doi.org/10.1080/10926488.2018.1481261.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, and Prateek Vij. 2016. A deeper look into sarcastic tweets using deep convolutional neural networks. *ArXiv*, abs/1610.08815.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas Georgios Stafylopatis. 2019. A transformer-based approach to irony and sarcasm detection.

Tomás Ptácek, Ivan Habernal, and Jun Hong. 2014. Sarcasm detection on czech and english twitter. In *COLING*.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm detection on twitter: A behavioral modeling approach. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, WSDM '15, page 97–106, New York, NY, USA. Association for Computing Machinery.

Rachel Rakov and Andrew Rosenberg. 2013. "sure, i did the right thing": a system for sarcasm detection in speech. In *INTERSPEECH*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA, October. Association for Computational Linguistics.

Shiv Naresh Shivhare and Sri Khetwat Saritha. 2014. Emotion Detection From Text Documents. *International Journal of Data Mining & Knowledge Management Process*, 4(6):51–57, November.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Yi Tay, Anh Tuan Luu, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. *Proceedings of 56th Annual Meeting of the Association for Computational Linguistics*.

Oren Tsur, Dmitry Davidov, and Ari Rappoport. 2010. ICWSM-A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews. In *In Fourth International AAAI Conference on Weblogs and Social Media*.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 task 3: Irony detection in English tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 39–50, New Orleans, Louisiana, June. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Bin Wang and C. C. Jay Kuo. 2020. Sbert-wk: A sentence embedding method by dissecting bert-based word models.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Chuhan Wu, Fangzhao Wu, Sixing Wu, Junxin Liu, Zhigang Yuan, and Yongfeng Huang. 2018. 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*, pages 51–56, New Orleans, Louisiana, June. Association for Computational Linguistics.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding.

Zhengchen Zhang, Minghui Dong, and Shuzhi Sam Ge. 2014. Emotion analysis of children's stories with context information. *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific*, pages 1–7.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Tweet sarcasm detection using deep neural network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2449–2460, Osaka, Japan, December. The COLING 2016 Organizing Committee.

Shiwei Zhang, Xiuzhen Zhang, Jeffrey Chan, and Paolo Rosso. 2019a. Irony detection via sentiment-based transfer learning. *Information Processing & Management*, 56(5):1633–1644, September.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019b. Bertscore: Evaluating text generation with bert.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. 2019. Moverscore: Text generation evaluating with contextualized embeddings and earth mover distance. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Appendix

A Extended Related Work, Baselines, Hyper-parameters

A.1 Related Work (Extended version)

In this section, we present an extended version of the related work on sarcasm detection including models that use *affective features*, *contextual information*, or a combination of both *affective features* and *contextual information*. We also explain how our work aims to bridge the gap among existing efforts.

A.1.1 Affective Features

Identifying sarcasm in text has evolved from simple lexical-based and syntactic pattern models (Tsur et al., 2010; Davidov et al., 2010; González-Ibáñez et al., 2011) to complex models that consider refined linguistic features, such as positive predicates, interjections, gestural cues (emoticons, quotation marks, etc.) (Carvalho et al., 2011) or behavior modelling (Rajadesingan et al., 2015; Agrawal et al., 2020). With the advent of deep learning, there has been a shift in how prediction models are designed and engineered. For example, Ghosh et al. (2017) proposed a conditional LSTM network (Hochreiter and Schmidhuber, 1997) for detecting sarcasm in twitter using sentence level attention mechanisms on hashtags and (Hernández Farías et al., 2018) learned and utilized a knowledge-based model of affective features based on a wide range of lexical resources. These models mostly relied on manually engineered patterns and features. However, more recently Zhang et al. (2019a) proposed multiple deep learning models, including a sentiment augmented/supervised with attention Bi-LSTM model and a sentiment transferred Bi-LSTM model to identify sarcasm in twitter datasets.

The main drawback of these models is that they are based on self-contained content (e.g. twitter hashtags #) and when such content is not available (which is the case for learning a more general model), the model fails to detect sarcasm. In addition, these models assume availability of the complete conversational context to detect sarcasm, which in most cases is not available; as a result they can not generalize properly. Distinct from this line of work, our proposed models minimize the need for manual feature engineering by utilizing an architecture with Bi-LSTM and attention mechanism that can accurately learn the affective representation of an input, even by utilizing a single affective feature.

A.1.2 Contextual Features

To address issues of lack of generalization and manual feature engineering, general word representation learning models have been proposed, such as Word2vec (Mikolov et al., 2013b) and GloVe (Pennington et al., 2014). These models learn embeddings of words that would locate them close to one another in the embedded space, if they share common contexts in the corpus. However, this means that even words opposite in meaning to another (*antonyms*), such as *happy* and *sad* that often occur in similar contexts, would be embedded close to each other. As a result, the use of general models can be inadequate for affective tasks (Amir et al., 2016; Agrawal et al., 2018). To this end, Zhang et al. (2016) proposed a bi-directional gated recurrent neural network (GRNN) to capture syntactic and semantic information locally, and a pooling neural network to extract contextual features automatically for sarcasm detection. While, Iliċ et al. (2018) proposed a deep learning model based on character-level word representations obtained from ELMo (Peters et al., 2018) using a learned representation that models features derived from morpho-syntactic cues to solve the issue of dissimilarity in context for sarcasm detection. In (Wu et al., 2018) a system based on a densely connected LSTM network with multi-task learning strategy using POS tag features was proposed. More recently, transformer-based models (i.e., using encoder and decoder methods), such as BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019) and XLNet (Yang et al., 2019) have been proposed to advance the state of the art in many NLP tasks by combining word embeddings with context embedding by using attention mechanisms in a bidirectional manner. However, little work has considered using these models for sarcasm detection. A multi-modal sarcasm detection method including text, speech and video features was proposed, where pre-trained BERT-based model was used for representing the sentences (Castro et al., 2019). Moreover, Potamias et al. (2019) proposed RCNN-RoBERTa for sarcasm detection in social media by leveraging the pre-trained embeddings from RoBERTa combined with a recurrent convolutional neural network.

The main drawback of using advanced pre-trained models is that they do not incorporate any affective-specific features during the training phase of the model – therefore the accuracy of the model on affective tasks can be lessened. Our proposed models incorporate affective features along with contextual information in one architecture for sarcasm detection.

A.1.3 Affective-Contextual Features

More sophisticated approaches for sarcasm detection have tried to combine the best of the two worlds by incorporating affective features with general embedding models during training. For example, Felbo et al.

(2017) proposed DeepMoji by training a Bi-LSTM model with emojis to learn representations of emotional context. Through a series of extensive experiments, particularly those related to incorporating affective features with pre-trained embeddings for sarcasm detection, the authors demonstrated the need to consider affective features in word embedding models for sarcasm detection. Poria et al. (2016) also developed pre-trained sentiment, emotion and personality models for identifying sarcastic text using Convolutional Neural Networks (CNN-SVM). More recently, (Tay et al., 2018) utilized a multi-dimension intra-attention mechanism to overcome limitations of sequential neural network models and capture words' incongruities for sarcasm detection. Moreover, a model that uses the semantic, sentiment and punctuation based hand-crafted features for sarcasm detection was proposed using multi-head attention based Bidirectional Long-Short Term Memory (MHA-BiLSTM) with Glove pre-trained embeddings (Kumar et al., 2020). Finally, Hazarika et al. (2018) proposed a ContextuAl SarCasm DEtector (CASCADE), by adopting a hybrid approach of both content and context-driven modeling for sarcasm detection where they utilized a user's personality features and style of writing to detect sarcasm.

Our research is mostly related to this line of work. In particular, we advance the state of the art in sarcasm detection by combining recently proposed transformer-based language models, such as BERT and SBERT, with affective-specific features.

A.1.4 Task-specific Corpora for Sarcasm Detection

Another important observation for the problem of sarcasm detection is that using task-specific corpora can be beneficial (Rakov and Rosenberg, 2013). Previous studies (Joshi et al., 2015; Zhang et al., 2014) have employed datasets related to comedy (movie transcripts, novels, etc.) to improve on the sarcasm detection task. This is because the utterance of humor/emotion and sarcasm is more expressed in the comedy genre than in other genres. Comedy is a literary genre and a type of dramatic work that is often satirical in its tone. For instance, they used corpus of children's stories (e.g., Harry Potter Books), transcripts of a MTV show (e.g., Big Bang Theory) and transcripts of comedy TV series (e.g., Friends) to train models for emotion and sarcasm detection. Following this intuition and to adhere to best practices, our proposed models leverage domain knowledge of two sarcasm-rich corpora for training embeddings (IMSDB and Riloff; see descriptions in the manuscript) during training to improve the sarcasm detection accuracy.

A.2 Sarcasm Detection Datasets

We evaluate the effectiveness of our proposed models on five sarcasm detection datasets as follows:

- **Onion**: This news headlines dataset[9] collected sarcastic versions of current events from The Onion[10] and non-sarcastic news headlines from HuffPost (Misra and Arora, 2019). The dataset contains 28,619 headlines, with 13,634 labeled as sarcastic, and 14,985 as non-sarcastic.

- **IAC**: This is a subset of the Internet Argument Corpus (Oraby et al., 2016). The dataset contains response utterances annotated for the sarcasm detection task. We extract 3260 instances from the general sarcasm type, with 1630 as sarcastic and 1630 as non-sarcastic[11].

- **Reddit**: Self-Annotated Reddit Corpus (SARC)[12] is a collection of Reddit posts where sarcasm instances are labeled by authors (in contrast to other datasets where the data is typically labeled by independent annotators) (Khodak et al., 2017). This results in 1,010,826 posts, with 505,413 as sarcastic and 505,413 as non-sarcastic.

- **Pt′acek**: The dataset consists of manually annotated sarcastic tweets. Authors annotated 7k tweets as sarcastic and 7k ones as non-sarcastic (Ptácek et al., 2014).

- **SemEval-2018**: the dataset was provided in SemEval-2018 Task 3 (Irony detection in English tweets), with total of 4,792 tweets including 2,396 ironic and 2,396 non-ironic(Van Hee et al., 2018).

[9]https://github.com/rishabhmisra/News-Headlines-Dataset-For-Sarcasm-Detection
[10]https://www.theonion.com
[11]https://nlds.soe.ucsc.edu/sarcasm2
[12]https://nlp.cs.princeton.edu/SARC/2.0/pol/

A.3 State-of-the-art Baselines

The state-of-the-art baselines used in our paper are as follows:

A.3.1 Only Affective

- **(Rajadesingan et al., 2015)**: Authors proposed a behavioral modeling framework for sarcasm detection. They discussed different forms of sarcasm: *i*) a contrast of sentiments, *ii*) a complex form of expression, *iii*) a means of conveying emotion, *iv*) a possible function of familiarity, and *v*) a form of written expression. They constructed relevant features to detect these forms on the Twitter dataset.

- **(Ghosh et al., 2017)**: They investigated several Long Short-Term Memory (LSTM) networks variations that can model both the conversation context and the sarcastic response. They showed that the conditional LSTM and LSTM networks, with sentence-level attention on context and response, outperform the LSTM model that only reads the response.

- **(Hernández Farías et al., 2018)**: They proposed a model exploring the utility of *affective features* based on a wide range of lexical resources (available for English) in the sarcasm detection task.

- **(Zhang et al., 2019a) (a)**: In their 1'st model, authors proposed a Sentiment-Augmented Attention Bi-LSTM model employing an attention-based neural network to identify context incongruity in the irony detection task. They used sentiment word corpora as an external features, and designed a soft attention mechanism focusing on context incongruity of tweets.

- **(Zhang et al., 2019a)(b)**: In their 2'nd model, authors improved the attention mechanism in a supervised manner to capture the context of incongruity in Twitter data. In particular, they incorporated two different types of sentiment resources (sentiment word lexicon and sentiment tweets copra) into the irony detection task.

- **(Zhang et al., 2019a)(c)**: In their 3'rd model, authors proposed a model to transfer deep features from sentiment analysis into the irony detection task for learning both explicit and implicit context incongruity in Twitter data. Their model consists of two Bi-LSTMs: one Bi-LSTM model serves as the sentiment feature extractor while the other one acts as the irony detector.

A.3.2 Only Contextual with Fine-Tune

- **(Potamias et al., 2019)**: They proposed a model (i.e., RCNN-RoBERTa) leveraging the pre-trained RoBERTa model and a recurrent convolutional neural network to tackle figurative language in sarcasm/irony detection in social media (Potamias et al., 2019).

- **RoBERTa**: A Robustly Optimized BERT Pre-training approach proposed by (Liu et al., 2019). The model uses dynamic masking instead of static masking (that was used in BERT), and are optimized to improve the accuracy of different NLP tasks (e.g., question answering) on GLUE, RACE and SQuAD detests.

- **XLNet-Large**: A generalized auto-regressive pre-training method that uses a permutation language modeling objective to combine the advantages of AR (Auto-Regressive) and AE (Auto-Encoding) methods. The model fixes BERT's negligence on dependency between the masked positions, causing a pre-train/fine-tune discrepancy (Yang et al., 2019).

- **BERT-Base/Large**: A Bidirectional Encoder Representations from Transformers. The model is trained on an unlabeled text corpus over two unsupervised tasks: *i*) *Masked Language Model* (MLM), in which, some of the tokens in the input sequences are masked and the model is trained to predict these masked tokens, and *ii*) *Next sentence Prediction* (NSP), where the model receives pairs of sentences as input and learns to predict if the second sentence in a pair is the subsequent sentence of the first one in the training corpus (Devlin et al., 2018).

A.3.3 Only Contextual with Pre-trained without Fine-Tune

- **(Zhang et al., 2016)** : Authors proposed a deep neural network using a gated recurrent neural network (GRNN) to induce semantic features for sarcasm detection. In particular, they modeled the tweet content with a GRNN, and used a gated pooling function to extract features, then predicted sarcastic tweets.
- **(Ilić et al., 2018)**: They proposed a deep learning model based on character-level word representations obtained from ELMo (Peters et al., 2018). The model used a learned representation features derived from morpho-syntactic cues.

A.3.4 Affective-Contextual

- **(Poria et al., 2016)**: They developed pre-trained sentiment, emotion and personality models for identifying sarcastic text using Convolutional Neural Networks (CNN-SVM).
- **(Amir et al., 2016)**: Authors proposed a deep neural network model (called CUE-CNN) that learns embeddings of content with lexical signals to recognize sarcasm in text documents.
- **(Yang et al., 2016)**: They proposed an attention-based neural model that learns an intra-attentive representation of the sentence, enabling it to identify contrasting sentiment, situations and incongruity for sarcasm detection.
- **DeepMoji**: These word embeddings were trained using BiLSTM on 1.2 billion tweets with emojis (Felbo et al., 2017).

  The model learns representations of emotional content in texts.
- **(Wu et al., 2018)** : Authors proposed a system based on a densely connected LSTM network (every LSTM layer will take all outputs of previous layers as inputs) with a multi-task learning strategy to combine the information in different tasks. The model improves the performance using POS tags and sentiment features.
- **(Hazarika et al., 2018)**: They proposed a ContextuAl SarCasm DEtector (CASCADE) by adapting a hybrid approach of both content-based and context-driven modeling for sarcasm detection. They used user profiling along with discourse modeling from comments in discussion threads. Then, the information is used jointly to learn a CNN-based model.
- **(Tay et al., 2018)(a)**: Authors proposed a model called "MIARN" that utilizes a multi-dimension intra-attention mechanism to overcome limitations of sequential neural networks in capturing words' incongruities in sarcasm detection.
- **(Tay et al., 2018)(b)**: In another model, they proposed a model called "SIARN" which employs a single-dimension intra-attention network for irony detection.
- **(Kumar et al., 2020)**: Authors proposed a model that uses the semantic, sentiment and punctuation based hand-crafted features for sarcasm detection. They utilized multi-head attention based Bidirectional Long-Short Term Memory (MHA-BiLSTM) combined with Glove Pre-trained embeddings for this purpose.

A.4 Evaluating the Performance of Each Proposed Models

Results of 7 more variations are shown in Table 4. For example, in ACE 1 (Wiki)+(EAISe) we train ACE 1 on Wiki and also incorporate the affective feature of EAISe (only stage 2 of the model ACE 1) followed by fine-tuning. Note that, only stage 2 of the model ACE 1 means without concatenation the two pre-trained embeddings and CFE component of model ACE 1 starts from stage 2 in this experiment. Another example, for model ACE 2 (WikiSarcA-BERT) means the token embeddings inputted into SBERT in ACE 2 were from a BERT model pre-trained on our WikiSarc corpus in model ACE 1 (where the CFE component of the model ACE 1 starts from stage 2 and incorporate the affective feature of EMoSi) without incorporating affective features.

Models	Combos	Onion	Reddit	Pt′acek	SemEval-2018	IAC
ACE 1	(Wiki) + (EAISe)	89.41	85.61	**75.16**	**78.21**	87.00
	(Wiki) + (EMoSi)	**89.84**	**85.70**	74.34	77.92	**88.15**
	(WikiSarc) + (EAISe)	90.33	86.71	77.00	84.63	89.21
	(WikiSarc) + (EMoSi)	**92.00**	**89.01**	**80.54**	**84.31**	**93.37**
ACE 2	(WikiSarcA-BERT)	86.73	82.17	76.90	78.29	84.00
	(WikiSarcA-BERT) + (EAISe)	**90.00**	**85.88**	**81.11**	**84.15**	**87.27**
	(WikiSarcA-BERT) + (EMoSi)	88.02	85.47	79.30	83.19	86.73

Table 4: F1-score results of comparing different pre-trained embeddings with different affective embeddings for each model. The best score is highlighted in **bold**, and the second best result is underlined.

A.5 Hyper-parameters for BERT

We train BERT from scratch on our datasets using Microsoft Azure ML[13] cluster of 8xND40-v2 nodes (64 NVidia V100 GPUs total) using Microsoft CNTK parallelization algorithm[14], 16 TPUs (64 TPU chips), Tensorflow 1.15, 1TB memory on Google Cloud and two 32 GPUs cluster of V100/RTX 2080 Ti, 1TB memory for up to 5 days.

Training: We train BERT-Large-uncased architecture for both stages of ACE 1 from scratch (24-layer, 1024-hidden, 16-heads, 340M parameters) , where the dimension is 768 and the max length for each sequence is 512. We experimented with different learning rates of $(1e-4, \beta 1 = 0.9, \beta 2 = 0.999)$, L2 weight decay of 0.01, and dropout probability of 0.1 for all layers as suggested for BERT-Large-Uncased (Devlin et al., 2018).

Fine-Tune It takes much less time to fine-tune our model than training it from scratch. In fact, the authors (Devlin et al., 2018) recommend only 2-4 epochs of training for BERT fine-tuning on a specific NLP task (sarcasm detection in our case) with the learning rate Adam Optimizer: 5e-5, 3e-5, 2e-5, and batch size of 16, 32. Note that, all datasets are split into training/testing using 80%/20%.

[13]https://github.com/microsoft/AzureML-BERT

[14]https://docs.microsoft.com/en-us/cognitive-toolkit/multiple-gpus-and-machines