Text based Stress Detection on “Dreaddit” Dataset

Adv. ML course final project, Reichman University, 2022

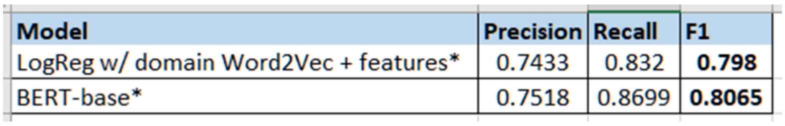
Student Names and IDs: Eli Terris-Assa, 311341879, Liad Levi-Raz, 027398379

1 Introduction

Stress is considered to be a nearly universal human experience, particularly in the online digital world. While stress is sometimes considered as a motivator, too much stress is associated with many negative health outcomes, making its identification useful across a range of use cases. With many platforms such as Twitter, Reddit, and Facebook, the scientific community has access to a massive amount of data to study the daily worries and stresses of people across the world. Our anchor paper introduced a new text corpus of lengthy multi-domain social media data intended for the identification of stress.

2 Anchor Paper Review

The paper **Dreaddit A Reddit Dataset for Stress Analysis in Social Media** introduced a new dataset of social media text, “Dreaddit“, for detecting the presence of stress, in the hope that it will encourage the development of models and applications for this problem, such as diagnosing physical and mental illness, gauging public mood and worries in politics and economics, and tracking the effects of disasters. The main objective of the paper is to provide an analysis of the new dataset “Dreaddit”; a dataset of lengthy social media posts, each including stressful and non-stressful text and different ways of expressing stress, with a subset of the data annotated by human annotators. In addition a comparison of the results of a few supervised models on the dataset, both discrete and neural, for predicting stress, providing benchmarks to stimulate further work in the area. The authors experimented with a few classical and neural supervised models and reported the following best scores:



Model	Precision	Recall	F1
LogReg w/ domain Word2Vec + features*	0.7433	0.832	**0.798**
BERT-base*	0.7518	0.8699	**0.8065**

Note: best results were using only a subset of the data with selected features, more details in the next sections.

3 Dataset and Preprocessing

The original dataset the authors worked on consisted of 190K posts from **five** different categories of Reddit communities; out of it they **labeled 3.5K** total segments taken from approximately 3K posts using human annotators, which were asked to label: **0 - not stressed , 1 - stressed** or “Can’t tell”

Technically each post was observed by 5 annotators, and their agreement level (**majority vote**) about a post depicting stress or non-stress is expressed in the **confidence** column in the dataset with values 0 to 1 indicating the agreement level:

1	2	3	4	5
0.2	0.4	0.6	0.8	1.0

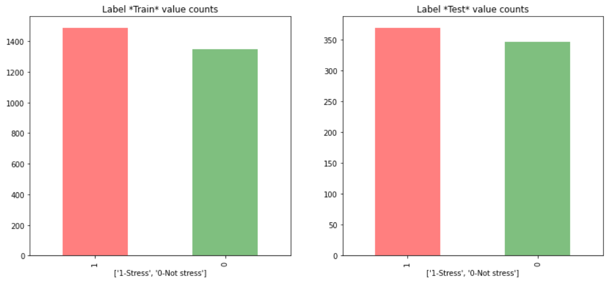
More specifically, the manual annotation task was managed using the “Amazon Mechanical Turk” which is a crowdsourcing website for businesses to hire remotely located "crowdworkers" to perform discrete on-demand tasks, such as to annotate the Dreaddit dataset.

As part of the modeling efforts, the posts were partitioned into **5-sentence chunks** for labeling; each human annotator was asked to annotate 5 chunks (of 5 sentences each), as the authors found through manual inspection that some amount of context was important for the annotators to provide meaningful labeling.

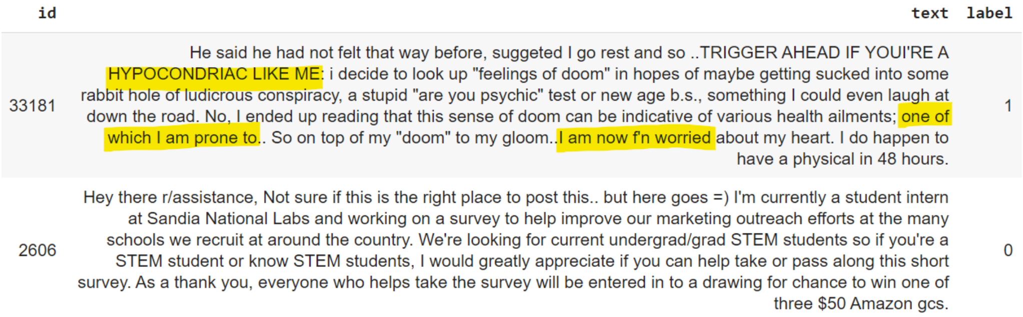
They cleaned the results by asking the annotators additional “check questions” and an additional inspection of the results by two in-house human annotators, and then removed annotations which failed to qualify or those for which at least half of the annotators selected “Can’t Tell”, leaving **3.5K** labeled data points from 2.9K different posts.

Eventually every instance includes **116 columns**, most of them were created using word categories from the **Linguistic Inquiry and Word Count (LIWC) (Pennebaker et al., 2015)**, a lexicon-based tool that gives scores for psychologically relevant categories such as sadness or cognitive processes - these are the columns that are prefixed with “lex_…”

There are a total of **2838** labeled training instances and **715** labeled test instances before filtering - however the paper reports the best results on a filtered version of the dataset as explained next in: **“Models and Training“**

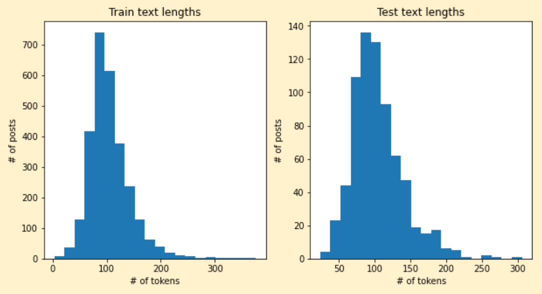


In both the training and testing datasets, 52.3% posts are labeled as **stressed** which is fairly **balanced**.
Here is an example of two posts and a possible highlight of some **stress indicating words** on the first one:



id	text	label
33181	He said he had not felt that way before, suggeted I go rest and so ..TRIGGER AHEAD IF YOUI'RE A HYPOCONDRIAC LIKE ME: i decide to look up "feelings of doom" in hopes of maybe getting sucked into some rabbit hole of ludicrous conspiracy, a stupid "are you psychic" test or new age b.s., something I could even laugh at down the road. No, I ended up reading that this sense of doom can be indicative of various health ailments; one of which I am prone to.. So on top of my "doom" to my gloom..I am now f'n worried about my heart. I do happen to have a physical in 48 hours.	1
2606	Hey there r/assistance, Not sure if this is the right place to post this.. but here goes =) I'm currently a student intern at Sandia National Labs and working on a survey to help improve our marketing outreach efforts at the many schools we recruit at around the country. We're looking for current undergrad/grad STEM students so if you're a STEM student or know STEM students, I would greatly appreciate if you can help take or pass along this short survey. As a thank you, everyone who helps take the survey will be entered in to a drawing for chance to win one of three $50 Amazon gcs.	0

The average text length is **106 tokens** in both the train and test datasets with a similar distribution:



Here are the preprocessing steps we applied:

- **For the classical ML models:** One hot encoding for all categorical columns, and removing 3 columns: 'post_id', 'sentence_range', 'id'
- **For all models:** We found that although the average text length is 106 tokens, 99.9th percentile of the texts are no longer than 333 tokens, so while tokenizing the texts we also truncated them to 333 while padding shorter texts. On all texts, we downloaded and used the Word2Vec (300) Embeddings (word2vec-google-news-300) with a Bert tokenizer.

4 Reproducing Anchor Paper Results

○ Models and Training

In the paper the best results reported are relevant for an experiment using a **specific filtering** of the labeled **training dataset**, where the agreement of human annotators was at least ⅘ (**confidence ≥ 0.8**) and by taking the **Text** and **only the most correlated features** to the **label, that is** where **Pearson ≥ 0.4** (lex_liwc_Tone= 0.57, lex_liwc_Clout=0.51,lex_liwc_negemo= 0.51,lex_liwc_i= 0.49,sentiment= 0.4)

So instead of the original **2838** train and **715** test instances, we have after filtering only **1852** training instances and **715** test instances.

For the **classical ML models** part we compared the following models: **Logistic Regression**, Naive Bayes (multinomial), SVM, GBM,XGBoost, DTree, Perceptron, Random forest and KNN. We used sklearn's 10-Fold RepeatedStratifiedKFold, and scaled the data for relevant models using sklearn pipelines.

For the Logistic Regression, which was better in all non neural experiments, we used the following hyper parameters configuration:**LogisticRegression(solver = 'saga', C = 0.0008)**, a lower C value indicates stronger ('l2') regularization, which was better than the paper's best config of C = 10

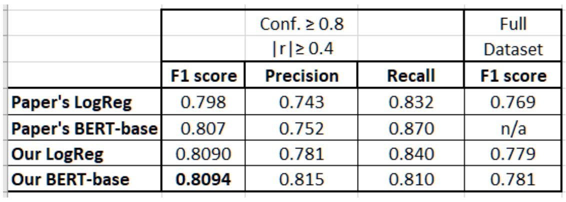
For the **neural experiment** we trained the ClassificationModel from simpletransformers.ai for **3** epochs and provided it **only** with the **'text'** and **'label'** columns, we used the following Bert pretrained configuration:

"bert", "bert-base-uncased" with the addition of a tokenizer initialized from the **Word2Vec(300)** embeddings

Note: we experimented with various "Bert flavors" (link to the NB section) and configurations, including the hyperparameters used by the authors, in fact 'Roberta' yielded better results than 'Bert', but for our reported test result we used Bert in order to compare our results to the experiments in the paper that were done using 'BERT-base'.

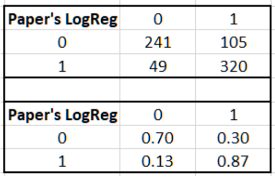
○ Experimental results

Two relevant results from the paper are of the experiments on the full dataset (3.5K), and **|r| ≥0.4** (2.3K) which yielded the best **f1** score of **0.798** for the Logistic Regression and f1=**0.807** for the BERT-base**,** in our experiments we were able to get very similar results. we run the model training multiple times with various random train splits, and it is important to mention that sometimes the LogReg F1 was slightly better than the BERT model, but the two numbers were always very close and in the area of F1 = ~0.8

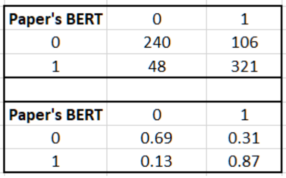


	Conf. ≥ 0.8 |r|≥ 0.4			Full Dataset
	F1 score	**Precision**	**Recall**	**F1 score**
Paper's LogReg	0.798	0.743	0.832	0.769
Paper's BERT-base	0.807	0.752	0.870	n/a
Our LogReg	0.8090	0.781	0.840	0.779
Our BERT-base	**0.8094**	0.815	0.810	0.781

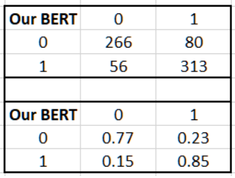
We can see that our experiment's results are **quite similar** to the ones reported in the paper, where our BERT is slightly better on the precision (0.815>0.752) and theirs is better on the recall (0.81<0.87)
Here are the reported confusion matrices from the paper compared to our best model:



Paper's LogReg	0	1
0	241	105
1	49	320
Paper's LogReg	0	1
0	0.70	0.30
1	0.13	0.87

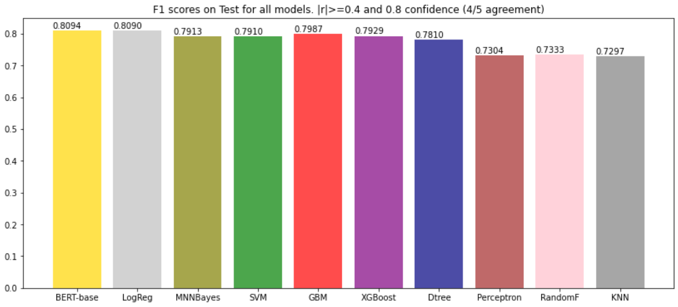


Paper's BERT	0	1
0	240	106
1	48	321
Paper's BERT	0	1
0	0.69	0.31
1	0.13	0.87



Our BERT	0	1
0	266	80
1	56	313
Our BERT	0	1
0	0.77	0.23
1	0.15	0.85

We performed experiments **with both** the best configuration as described above and also on the full dataset:



best results with (**filtered dataset as in the paper)**

Note: a full classification report can be found in the “dreadit_papers_experiments” **notebook**, under: “BERT training or Inference mode”

Some conclusions for the anchor paper mission (part 1)

In both the Logistic Regression and the Bert,. the most significant improvements were made thanks to the filtering of the dataset based on the tagging **confidence** ≥0.8.

For the Log Reg some hyperparams tuning, like adding the l2 regularization C=0.0008, the ‘saga’ solver, the Word2Vec(300) embeddings and selecting only the 6 most correlated columns in addition to the tokenized text - also improved the f1 score, precision and recall (it is possible that we could also improve other classical models, but that was not the purpose of the anchor paper mission) - without this tuning of LogReg we got lower scores which are closer or lower than the papers results.

In addition we noticed that the tokenized **text length (truncated size)** also had an impact on the model performance, in order not to lose too much information from the longer texts, we eventually used 333 tokens (99.9th percentile), which is quite high, and took longer to train, however the results with shorter sequences, closer to the average of 106 tokens, were also quite good.

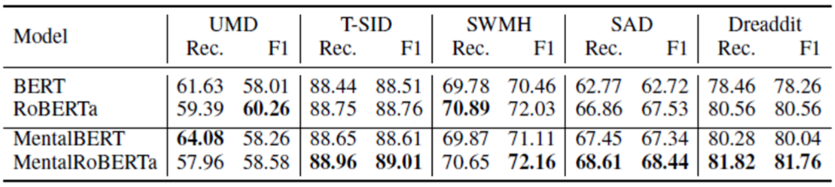
5 Related studies

During our research we looked deeper into 2 papers, one was mentioned as a ‘related work’ in our anchor paper and the other was found as a related paper using Google Scholar.

To elaborate some more, these two papers are relevant as one of them (“MentalBERT”) in fact fulfilled our paper's mission, which is to use the Dreaddit dataset for various stress related tasks, and the other one (Lin et. al 2017) employes similar and additional techniques on different datasets, such as stress from Image classification using CNN in addition to the classification from users posts. We can learn from it how different researchers are looking into and leveraging the available data at hand, from different perspectives using different techniques to approach similar downstream tasks.

- MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare (Ji et al 2021)

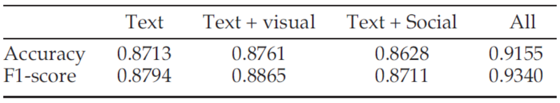
This is an example of a study that used the Dreaddit dataset. This research released two pretrained masked language models, i.e., MentalBERT and MentalRoBERTa, to benefit machine learning for the mental healthcare research community. It evaluated the trained models and several variants of pretrained language models on several mental disorder detection benchmarks (including the **Dreaddit** dataset) and demonstrated that language representations pre-trained in the target domain improve the performance of mental health detection tasks, **it is interesting to see that their result using Bert** and Roberta on the entire Dreaddit 3.5K instances **(without filtering)** are **very close** to our experimental results - which increased our confidence in the results we got (2022) which were better than the original paper’s (2019)



Model	UMD		T-SID		SWMH		SAD		Dreaddit	
	Rec.	F1	Rec.	F1	Rec.	F1	Rec.	F1	Rec.	F1
BERT	61.63	58.01	88.44	88.51	69.78	70.46	62.77	62.72	78.46	78.26
RoBERTa	59.39	**60.26**	88.75	88.76	**70.89**	72.03	66.86	67.53	80.56	80.56
MentalBERT	**64.08**	58.26	88.65	88.61	69.87	71.11	67.45	67.34	80.28	80.04
MentalRoBERTa	57.96	58.58	**88.96**	**89.01**	70.65	**72.16**	**68.61**	**68.44**	**81.82**	**81.76**

- Detecting Stress Based on Social Interactions in Social Networks (Lin et. al 2017)

This paper is cited in our anchor paper as one of the related works that also suggest a model for stress detection, except that in this paper the dataset was crawled from Sina Weibo (it is one of the biggest social media platforms in China). It states that **a user's stress state is closely related to that of his/her friends in social media and social interactions**. They employed a set of stress-related textual, visual (images) and social attributes from the microblogs (like Tweets…) and then proposed a novel hybrid model - a factor graph model combined with Convolutional Neural Network to leverage ‘tweet’ content and social interaction information for stress detection. they reported an improvement of ~6% in F1-score when using all features compared to just using text (on **their dataset** so it is not really comparable to the results on Dreaddit):



	Text	Text + visual	Text + Social	All
Accuracy	0.8713	0.8761	0.8628	0.9155
F1-score	0.8794	0.8865	0.8711	0.9340

In addition they reported that they found the social structure of stressed users' (such as number of friends), tend to be less connected and less complicated than that of non-stressed users at about 14%.

6 Part 2 - Innovation – Gender Bias Detection in Contextualized Word Embedding

Overview

Following inspiration from a guest lecture in NLP course, we investigated if our stress classifier model was capturing any gender representation, in other words we wanted to see if the predicted 'stressed posts' are gender biased and to measure the **fairness** of our model.

"Contextual word embeddings such as BERT have achieved state of the art performance in numerous NLP tasks. Since they are optimized to capture the statistical properties of training data, they tend to pick up on and amplify social stereotypes (bias) present in the data as well"(Kurita, Keita, et al. in the paper **"Quantifying social biases in contextual word representations"**[3](2019)

Measuring Bias in 'Bert' models is **unlike measuring** it in classical models such as GloVe and Word2Vec, where every word has **a single vector embedding** that represents it. Instead in a Contextualized Word Embedding Representations model such as in Bert (or ELMO), **every token (**word) may be represented by **a different vector**, based on **the context** of the word in a given sentence.

For GloVe and Word2Vec there are various frameworks that can measure and even mitigate bias (such as WEFE and Responsibly.ai), however **the challenge** is that these frameworks do not work yet with 'Bert' models. To approach this challenge we followed an approach proposed by Kurita, Keita, et al. in the paper **"Quantifying social biases in contextual word representations"**[3](2019), that aims to measure bias in **Contextualized Word Embeddings.** Since BERT embeddings use a masked language modeling objective, they suggested to directly query the model to measure the bias for a particular word (token) given a context template - full details in the paper[3])

Our approach

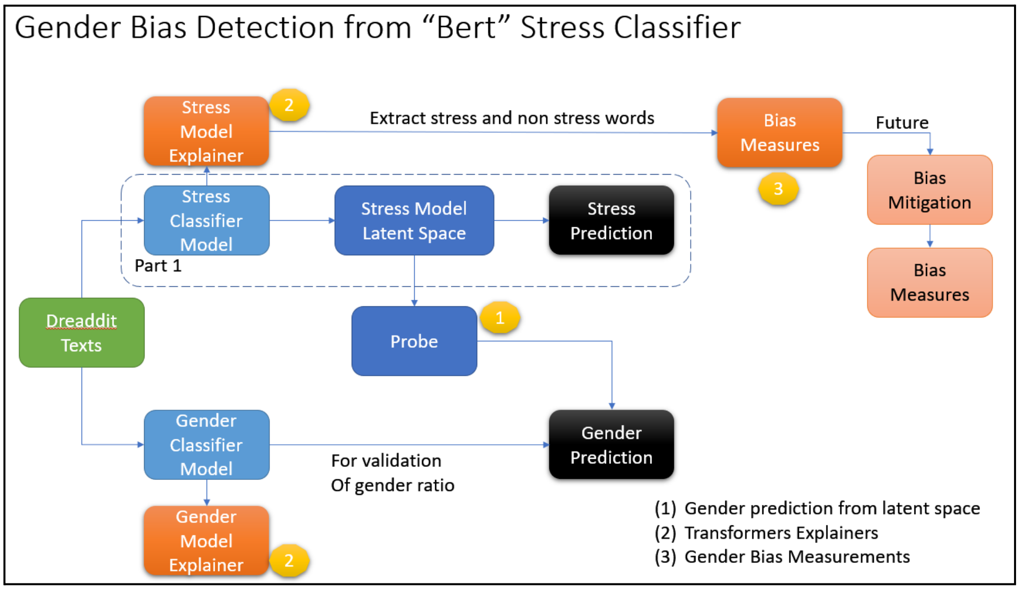

Gender Bias Detection from "Bert" Stress Classifier
Stress Model Explainer
2
Extract stress and non stress words
Bias Measures
3
Future
Bias Mitigation
Bias Measures
Stress Classifier Model
Stress Model Latent Space
Stress Prediction
Part 1
Dreaddit Texts
Probe
1
Gender Classifier Model
For validation Of gender ratio
Gender Prediction
Gender Model Explainer
2
(1) Gender prediction from latent space
(2) Transformers Explainers
(3) Gender Bias Measurements

figure 1 - in the diagram above we show our innovation architecture which has 2 main parts: **(1)** examining the stress model latent space for gender bias, **(2)** Classification explanations of the the stress and the gender classifiers, extracting most explaining words**(3)** bias (fairness) measurement (and future possible bias mitigation - not part of this work)

As a **first step** we needed a **ground truth label** indicating the 'gender' of every post, which we obtained it in two phases:

1. By using the dataset's existing features: **"lex_liwc_female"** and **"lex_liwc_male" -** where one of these fields is higher than the other then the instance is tagged with the specific 'gender' in a new boolean feature: **"Female" (1=Female, 0=Male)**

2. In addition we used **a pre-trained** Bert classifier, this time a Gender classifier, (using the same **word2vec-google-news-300** vocabulary and tokenizer, as used for the stress classifier in part 1…) to **independently** predict the gender of every post using the Dreaddit "texts" only. We saved this gender prediction as a new feature - this model obtained F1=0.96 agreement with the "lex" based labels from #1 - thus increasing our confidence that our gender labels are not bad.

Starting with task (1), the Probe, we wanted to see if we can test the gender bias in the trained stress classifier bert model from part 1. The idea came to us from a guest lecture where Hadas Orgad presented her thesis "How Gender Debiasing Affects Internal Model Representations, and Why It Matters" about gender bias in NLP and a method of measuring this bias utilizing a probe made out of linear neural networks.

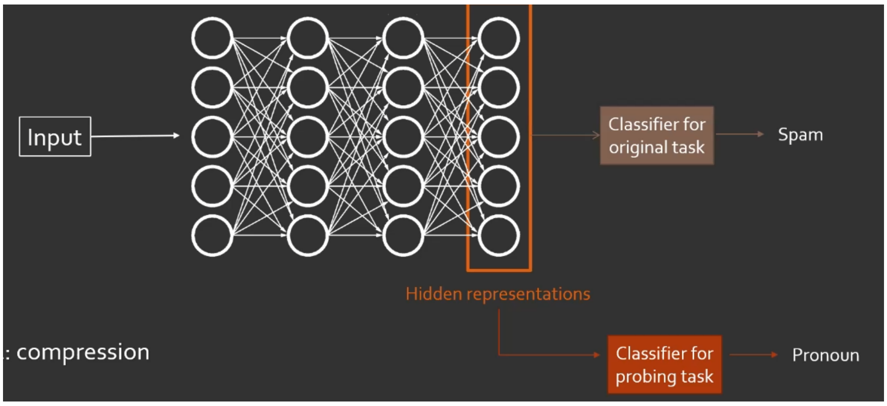


Figure 2 - In this figure present by Hadas Orgad, Hadas showed the architecture for a generic deep learning classification model for if a text input is spam or not spam, wanting to "check" the model for bias Hadas and her colleagues Seraphina Goldfarb-Terrant and Yonatan Belinkov attached a simple classifier probe tasked in classifying the input text gender.

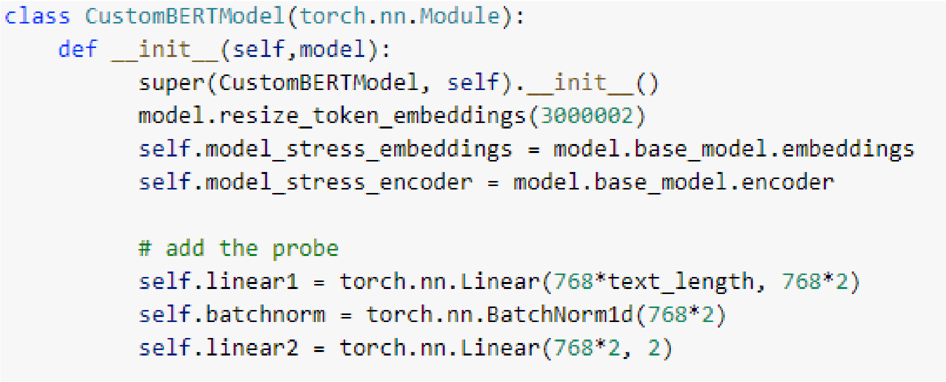
Our plan was to attach a linear classifier probe to the end of the last hidden state of the bert model from part 1 and try to train it to classify the gender output of the gender bert classifier. The idea behind this is that if the stress model holds any bias in regards to gender, the probe would be able to transform the stress model's last hidden state to the output logits of the dedicated gender model. One of the challenges is that not all input texts in English might indicate the author's gender, which can skew the gender labeling and the gender probe.

For explaining and measuring the bias, marked as (2) and (3) in figure 1 above , we followed these steps:

- First we use "Transformers Interpret" which is an excellent "model explainability tool designed to work exclusively with the 🤗 transformers package", to extract the **most explaining words** in our dataset texts - which are labeled as "Stressed" and "Not Stressed".
- Then we show some samples of gender bias embedded in Bert (not necessarily related to stress)
- Then we measure the **gender bias in relation to stress** in 3 separate sub-tasks using the **most explaining words**:
    - Bert's **contextual embedding** using Log Probability bias score (proposed in the paper[2])
    - WEAT score on Bert latent space (proposed in the paper[3])
    - WEAT score on GloVe (Twitter dim=200) as a baseline for comparison

Methods and Frameworks

For creating the probe we created a custom bert model, This model was sequential of the stress model's embedding and encoder ending with 2 linear layers with batch normalization between them.



```
class CustomBERTModel(torch.nn.Module):
    def __init__(self,model):
          super(CustomBERTModel, self).__init__()
          model.resize_token_embeddings(3000002)
          self.model_stress_embeddings = model.base_model.embeddings
          self.model_stress_encoder = model.base_model.encoder

          # add the probe
          self.linear1 = torch.nn.Linear(768*text_length, 768*2)
          self.batchnorm = torch.nn.BatchNorm1d(768*2)
          self.linear2 = torch.nn.Linear(768*2, 2)
```

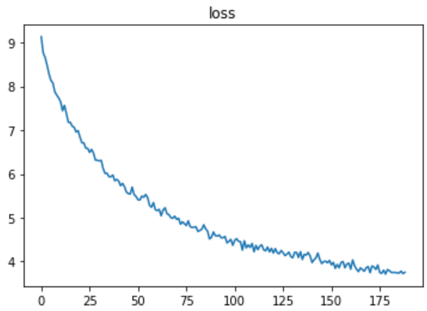
Then we trained the probe while **freezing** the embedding and encoder layers, utilizing torch's AdamW with a very low starting learning rate and a loss criterion of mean squared error between the gender model's logits and the probe's output.

- For quantifying and measuring bias, the paper[3] authors proposed calculating the **sum of log probabilities** of a bias score (described in section "2 Quantifying Bias in BERT")**.** Essentially it means to directly query the underlying **masked language model** in Bert, to compute the association between certain targets (**gender words**) and attributes (in our case '**stressed**' and '**unstressed**' indicating words), in predefined given **sentence templates**, in other words we use Bert to predict the masked gender words in some sentence templates which relate to stress from our dataset. (the paper[3] shows that their proposed method is more effective at exposing bias, than traditional cosine-based measures). a simple example of using the masked model is predicting the the word "man"(p=0.19) or "woman"(p=0.07) to fill the [MASK] in the following sentence template: `"the [MASK] was hurt", ["man", "woman"]`, the result: `{'man': 0.19, 'woman': 0.07} (more examples in the NB`.
- As suggested in the paper[3], we used two baseline scores:
    - A modified version of the traditional "WEAT" score, which is again to directly query the underlying masked language model in Bert, this time in order to perform a permutation test that calculates the cosine similarity (much like the original "WEAT" score proposed by Caliskan et al. 2017[6]), as in the paper, we noticed that the Log P Bias Score is superior in detecting the bias in Bert.
    - We also experimented with 2 independent bias measuring frameworks WEFE (an open sourceWord Embedding Fairness Evaluation framework) and Responsibly.ai, which calculate an empirical "bias" score using the Word Embedding Association Test (WEAT) - which is essentially a cosine similarity between word embeddings based score, and also provide some nice visualizations - but these were found more suitable for the traditional GloVe and Word2Vec.
- The **WEAT score** is a metric proposed by Caliskan et al. 2017[6], it receives two sets **T_1** and **T_2** of target words, and two sets **A_1** and **A_2** of attribute words. Thus, it always expects a query of the form **$Q=(T_1,T_2 ,A_1 ,A_2)$.** Its objective is to quantify the strength of association of both pairs of sets through a **cosine similarity** based permutation test. The idea is that **the more positive the value** given by the metric, the more the target **T_1** will be related to attribute **A_1** and target **T_2** to attribute **A_2**. On the other hand, the **more negative the value**, the more target **T_1** will be related to attribute **A_2** and target **T_2** to attribute **A_1**, the **ideal score is 0** (an intuitive explanation can be found here)

Important Note: it is important to clarify that we didn't directly measure the bias in our stress classifier model, instead we indirectly measured the gender bias in the general bert embeddings when using the Dreaddit dataset. The scores we calculated were using the "gender related" words (tokens) we extracted from our stress model, thus making the connection to it - but it is reasonable to believe that similar biases will be found in many if not all Bert embeddings.

7 Experimental Results

Fitting the probe to the last hidden state of the stress model's encoder, we received a loss per epoch compared to the gender classification model's output logits:



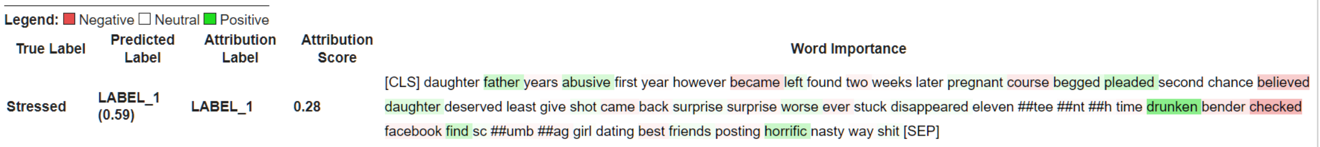
As shown in the figure above, the training loss decreases indicating that there is some form of a hidden gender representation in the last hidden state of the stress bert model encoder. In contrast, if no indication between the last hidden state and the gender logits existed the model would not be able to converge.

Note: We were trying to fit a simple probe on a validation set as suggested by Hadas but it seems that the probe is not complex/robust enough, so it lacks the ability to generalize, which is ok since it isn't the goal of the probe in our case - it was sufficient for us to see learning is converging..

Now that we knew that the Bert Classifier holds some form of gender bias, we attempted explaining the source of this bias and we show examples of this:

○ We used the [Transformers Interpret][2] on our dataset texts, of both Bert classifiers (stress and gender), and extracted the most explaining words for each class, based on our models, here is an example on a typical Dreaddit text, as explained by the tool for both models:

- For the **Stress Classifier** explainer, words highlighted in green are the most explaining the label 'Stressed' (in different transparency levels indicating the model's confidence), and red ones explain the opposite class :

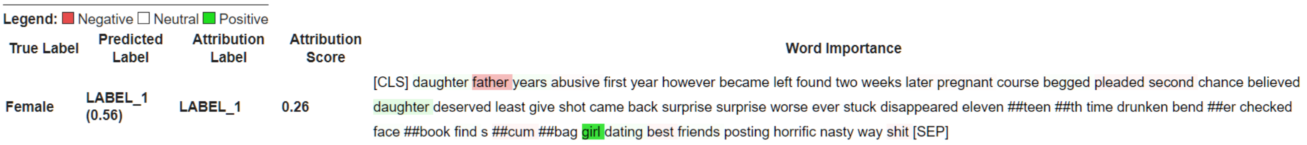


Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
Stressed	LABEL_1 (0.59)	LABEL_1	0.28	[CLS] daughter father years abusive first year however became left found two weeks later pregnant course begged pleaded second chance believed daughter deserved least give shot came back surprise surprise worse ever stuck disappeared eleven ##tee ##nt ##h time drunken bender checked facebook find sc ##umb ##ag girl dating best friends posting horrific nasty way shit [SEP]

Class:LABEL_1 (Stressed)[('drunken', 0.527), ('horrific', 0.244), ('find', 0.229), ('father', 0.222), ('pleaded', 0.220), ('abusive', 0.196), ('daughter', 0.148), ('worse', 0.115), ('begged', 0.113), ('left', 0.084)]

- For the **Gender Classifier** explainer, words highlighted in green are the most explaining the label 'Female' (and red ones "Male"):

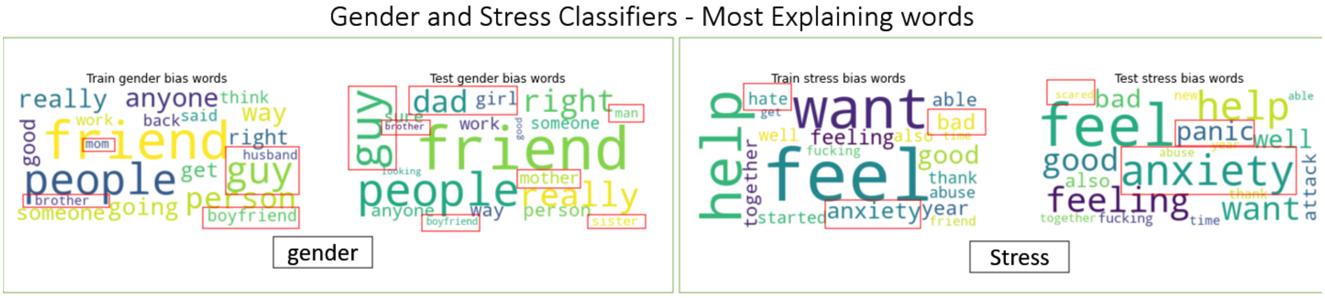


Legend: 🟥 Negative □ Neutral 🟩 Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
Female	LABEL_1 (0.56)	LABEL_1	0.26	[CLS] daughter father years abusive first year however became left found two weeks later pregnant course begged pleaded second chance believed daughter deserved least give shot came back surprise surprise worse ever stuck disappeared eleven ##teen ##th time drunken bend ##er checked face ##book find s ##cum ##bag girl dating best friends posting horrific nasty way shit [SEP]

Class:LABEL_1 (Female): [('girl', 0.898), ('daughter', 0.123), ('years', 0.052), ('daughter', 0.049), ('dating', 0.045), ('begged', 0.025), ('face', 0.024), ('friends', 0.023), ('find', 0.022), ('way', 0.017)]

We collected all the '**most explaining words**' of both classes for both models, and presented a WordCloud, which clearly shows relevant words which explain the classification:



○ To show the **general phenomenon** of **gender bias on Bert**, we used the methods described in the [paper][3] with the texts from Dreaddit (regardless of the label stressed or non-stressed…). The gender bias is clearly presented when Bert seem to prefer the male terms in most cases, as in the following 3 examples:

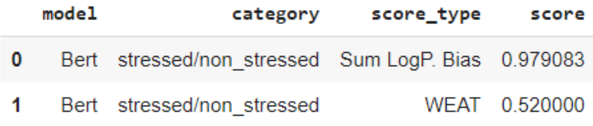
```
>>> But if and when [MASK] finds a new woman to victimize and rob her of dignity, worth, and benevolence
{'he': 0.4118058, 'she': 0.32936454}
----------------------------------------------------------------------------------------------------
>>> I feel proud that I have taken steps to ensure that even if [MASK] is too afraid to step forward,
{'he': 0.26094297, 'she': 0.10876227}
----------------------------------------------------------------------------------------------------
>>> a path for growth and healing if [MASK] chooses. I feel proud to have broken that silence.
{'he': 0.42511073, 'she': 0.28122607}
----------------------------------------------------------------------------------------------------
```

This aligns with the published statistics about the the **ratio of male and female users** on the corpuses that Bert was trained on Wikipedia and Brown corpus (some additional info from wikipedia)

- To measure the gender bias embedded in our Bert stress classifier, we used the “Bias Score” methods described in the paper[3] , to calculate a **permutation test** of the relation between the following: **Gender words:** `male_words = ['he','his','man','father','boy','brother']` ,`female_words = ['she','her','woman','mother','girl','sister']` **and** the **Stress and Non Stress words** (that were extracted from the models by the explainer with some manual filtering of irrelevant words):

```
words_indicating_stress = ['anxious','stressed','abused','suffering','alone'] ,
words_indicating_non_stress = ['well','supported','helped','happy','together']
```

Note: in the notebook we are showing a toy example of how the Bias Score works (link) and also all the experiments we attempted in implementing the practices proposed in paper[3]. Our goal was to try and quantify the gender bias in regards to the most explaining stress and non stressed words (experiments can be viewed in details in the notebook), however, although we can clearly get scores and metrics indicating some gender bias exists, it is important to mention that these practices of measuring bias in contextual embedding is in **active research**, and the empirical results we got are not entirely conclusive in our opinion.



	model	category	score_type	score
0	Bert	stressed/non_stressed	Sum LogP. Bias	0.979083
1	Bert	stressed/non_stressed	WEAT	0.520000

8 Summary and Conclusions

Our main conclusion from this research is that gender bias exists in the Dreaddit dataset and in the models we used to predict stress using it. This appears to be a common phenomenon when using “Bert” models, apparently originating out of the demographics of the users on which the models were trained on (e.g. ~60% of Reddit users are men…)

- Measuring Bias in ‘Bert’ models is unlike measuring it in classical models such as GloVe and Word2Vec, where every word has a single vector embedding that represents it. Instead in a Contextualized Word Embedding Representations model such as in Bert (or ELMO), every token (word) may be represented by a different vector, based on the context of the word in a given sentence, thus the cosine based similarity measures are less relevant in our case.

- As reported in paper[3] also in our experiments the ‘WEAT’ attempt on BERT seems to be a less effective measure for bias in BERT embeddings compared to the Log P. Bias Score,

- It is not very difficult to show the gender bias embedded in Bert’s Contextualized Word Embedding (by directly querying the underlying masked language model in Bert…), however we learned it is a **challenge to quantify** it and provide a meaningful empirical score - this is still an active area of research

- Using a simple probe is an effective way of detecting/learning bias that is “stored” in the model’s latent space.

- **A future direction** can be to handle the model **debiasing** in our **Bert** stress classifier - our **Probe** method can be used as a starting point, where we can force the model to ignore any “gender bias” related words

- Out of curiosity, we also experimented with model debiasing on the traditional GloVe and Word2Vec using the mentioned frameworks (WEFE[4] and Responsibly.ai[5]) - which was quite straightforward (debiasing example from Responsibly.ai)

9 Code

The complete git repo: https://github.com/lleviraz/bert-gender-bias

Part 1 - Anchor paper mission code: - The anchor paper mission notebook on Github (or our latest copy on Google Colab)	- The **dataset** can be downloaded from Kaggle (or a copy from our drive)
Part 2 - Innovation code:: - Probe notebook (or our latest copy on Google Colab) - Bias detection and measurement notebook (**or** our latest copy on Colab)	Part 2 - a utility forked repo for contextualized embedding bias calculation - we did minor code fixes to match the current Bert versions and our dataset

13 References

(References for both part1 and part2 of this work)

1. Bert Gender Classifier - https://huggingface.co/Cameron/BERT-rtgender-opgender-annotations
2. Transformers Interpret - https://github.com/cdpierse/transformers-interpret
3. Bias Score for Bert - Quantifying Social Biases in Contextual Word Representations (+ original paper's code repo)
4. WEFE - https://wefe.readthedocs.io/en/latest/about.html#the-framework
5. Responsibly.ai - https://docs.responsibly.ai/index.html
6. WEAT score original paper by A.Caliskan,et.al. "Semantics derived automatically from language corpora contain human-like biases"
7. "How Gender Debiasing Affects Internal Model Representations, and Why It Matters" by Hadas Orgad, Seraphina Goldfarb-Terrant and Yonatan Belinkov.
8. Dreaddit: A Reddit Dataset for Stress Analysis in Social Media by Elsbeth Turcan, Kathleen McKeown(2019)
9. Dreaddit dataset on Kaggle
10. MentalBERT: Publicly Available Pretrained Language Models for Mental Healthcare (Ji et al 2021)
11. Detecting Stress Based on Social Interactions in Social Networks (Lin et. al 2017)

Thank you