Home assignment questions:

1. What are the accuracy/f1/recall/precision of your model on the Sentiment140 dataset? And on the Dublin dataset? (Provide the metrics measurement for each dataset.)

Sentiment140 dataset	Baseline model	Deep Learning model
accuracy	0.83	0.85
f1	neg: 0.82 / pos: 0.84	neg: 0.84 / pos: 0.84
recall	neg: 0.78 / Pos: 0.88	neg: 0.81 / pos: 0.90
precision	neg: 0.86 / Pos: 0.80	neg: 0.88 / pos: 0.83

Dublin dataset	Baseline model	Deep Learning model
accuracy	0.70	0.66
f1	neg: 0.70 / pos: 0.71	neg: 0.63 / pos: 0.69
recall	neg: 0.69 / pos: 0.72	neg: 0.57 / pos: 0.75
precision	neg: 0.71 / pos: 0.70	neg: 0.70 / pos: 0.64

2. Are they or are they not the same? How large is the difference? Can you think of the cause? How would you corroborate or refute your hypothesis?

The results are different for each test dataset and for the Dublin_test they are noticeably worse. This is probably due to the fact that the tweets in the Dublin_test dataset contain slang and concrete vocabulary from Dublin. These different words and expressions don't appear as frequently in the train dataset, and thus, they are harder for the model to identify.

To see if this hypothesis is correct we could manually test sentences with expressions used in Dublin/Ireland and their corresponding equivalent "translations" to US English, which is more common. Then we could see if the prediction accuracies are lower or not for the first case.

3.What are the main classification challenges? (you can find an idea with our suggested approach here, but feel free to do it at your own discretion).

- Tone of the tweets
- Idioms will be understood by the model as a literal phrase (Eg. "My brother is a couch potato")

- Words like no, not, and never are difficult to handle properly because of double negations (Eg. “I can’t not go to my class reunion”)
- Sarcasm detection

- Code related decisions/explanations

Training the Deep Learning model with the whole dataset was too resource demanding for Google Colab, so I chose to only use a subsample of the 10% of the dataset (160k tweets).

Since my main goal with this code is not to generate the best models with the best scores/hyperparameters, I have decided not to apply too much preprocessing to the data to save time in the development of this assignment. Having said this, preprocessing plays an essential role in the performance of NLP systems and this should be taken into consideration in a real implementation.

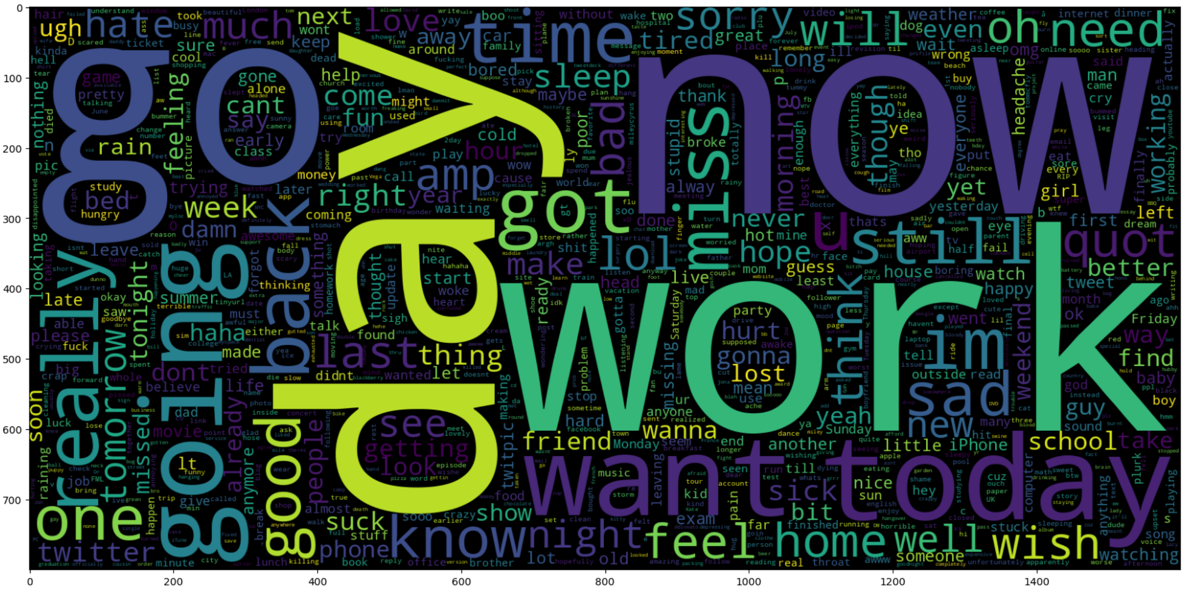
Taking into consideration that both test datasets contain neutral labeled tweets, my approach has been discarding them and using only those with the same labels that appear in the train dataset (negative and positive).

Regarding the chosen models, for the baseline one, I also tried the Random Forest algorithm, but got worse results than with SVC and Linear Regression, so I have not included it in the final code/report. For the deep learning approach, I used transfer learning from BERT, applied dropout and passed it through a fully connected layer, then a Cross Entropy loss function that has incorporated a softmax.

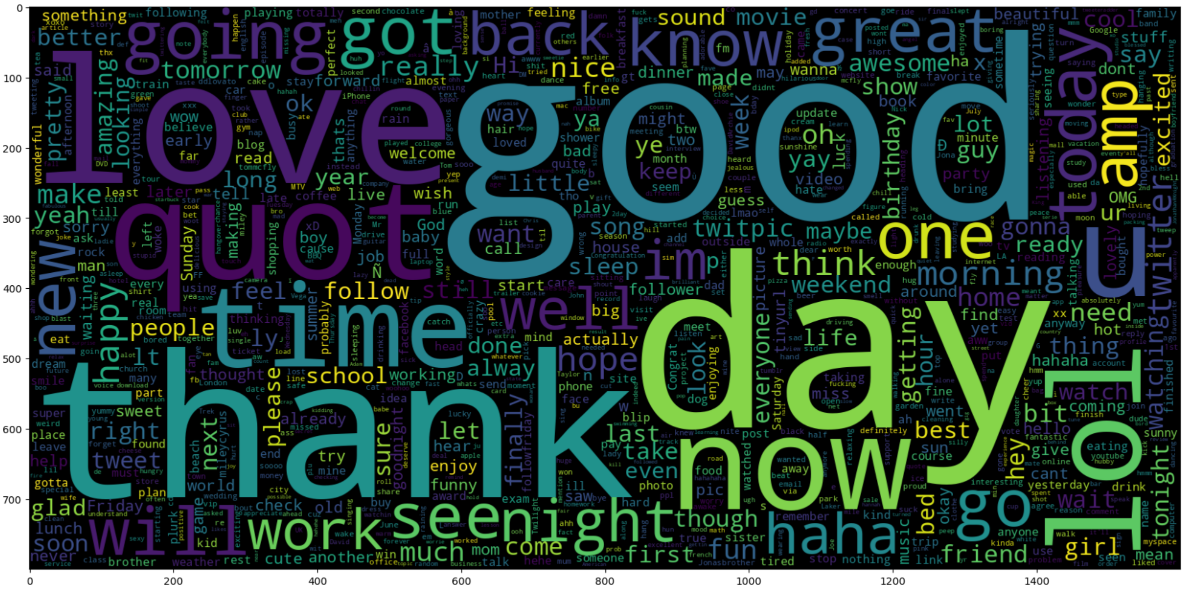
- <u>Error analysis</u>

  Before analyzing some of the incorrect predictions, let’s first take a look at the word clouds for the most common words in positive and negative tweets:

  Negative tweets:



  Positive tweets:



Incorrect pred. example #1:

Row: 39

Dataset: S140_test

Text: @spinuzzi: Has been a bit crazy, with steep learning curve, but LyX is really good for long docs. For anything shorter, it would be insane.

Label: Positive

Prediction: Negative

INCORRECT PREDICTION

I wouldn’t know how to label this tweet. It looks very neutral to me, but the dataset labels it as positive. Since it contains words like crazy and insane, the model interprets it as negative.

Incorrect pred. example #2:

Row: 143

Dataset: S140_test

Text: NOOOOOOO my DVR just died and I was only half way through the EA presser. Hate you Time Warner

Label: Negative

Prediction: Positive

INCORRECT PREDICTION

“time” is one of the top positive words in the model.

Incorrect pred. example #3:

Row: 132

Dataset: S140_test

Text: Night at the Museum, Wolverine and junk food - perfect monday!

Label: Positive

Prediction: Negative

INCORRECT PREDICTION

Probably interpreted as negative because of “junk” and maybe “monday”.

Incorrect pred. example #4:

Row: 622

Dataset: Dublin_test

Text: RT @greenparty_ie: â œThe Taoiseach said the climate emergency declaration was symbolic â “ these licences are symbolic of Fine Gaelâ ™s failureâ ¦

Label: Positive

Prediction: Negative

INCORRECT PREDICTION

Lots of missing data + the word failure

Incorrect pred. example #5:

Row: 1495

Dataset: Dublin_test

Text: RT @TinaMurnotbot: Yep...that's how it works. When Ms Jordan put the 'phone down on a sick mother with three children who had just been madâ ¦

Label: Negative

Prediction: Positive

INCORRECT PREDICTION

Sick is a difficult word since it can have both positive and negative meanings.

Incorrect pred. example #6:

Row: 1346

Dataset: Dublin_test

Text: If you have a profile on @ThePracticalDev, link it below as a comment to this post and I'll follow you! I'm looking to expand my following base to increase my chances of learning more from the community! Please RT #DEVcommunity https://t.co/rL9b7qEjhI

Label: Negative

Prediction: Positive

INCORRECT PREDICTION

This tweet is labeled as negative ¿probably because it’s spam?, but the model does not understand this and predicts it as positive since the message is happy.

<u>Incorrect pred. example #7:</u>

Row: 1155

Text: Thank you to the hundreds of people who have registered to become a volunteer at this years #Christmas Day Dinner for the poor &amp; homeless of #Dublin City ðŸ”´Registration is now closed The Order of the knights of St. Columbanus #InstaurareOmniaInChristo #RestoreAllThingsInChrist https://t.co/oh8kw8asXz

Label: Negative

Prediction: Positive

INCORRECT PREDICTION

Again this tweet is labeled as negative maybe because of spam, but it’s talking about volunteering and Christmas so that’s probably why it is predicted as positive.

<u>Incorrect pred. example #8:</u>

Row: 1834

Text: @MurrayImfurst The Spire was a test run, it's all about making the city easier to navigate for tourists via the medium of obnoxiously oversized shit.

Label: Negative

Prediction: Positive

INCORRECT PREDICTION

I’m not sure if this one contains sarcasm, but since it’s labeled as negative, it looks like the author is complaining about something. This part: “...it's all about making the city easier to navigate for tourists…” is probably what makes the model predict it as positive.

<u>Incorrect pred. example #10:</u>

Row: 73

Text: Back when I worked for Nike we had one fav word : JUST DO IT! :)

Label: Positive

Prediction: Negative

INCORRECT PREDICTION

The model doesn’t like the word work. It’s one of the most common words in “Negative” tweets.

Incorrect pred. example #9:

Row: 708

Text: @CllrEoinOBroin @DiarmuidOM @ProfJohnCrown @sdublincoco Sadly consensus is "pie in the sky" lets stick with gridlock and invent "bus connect". Its why many Irish people leave. The comical paucity of ambition in official Ireland

Label: Positive

Prediction: Negative

INCORRECT PREDICTION

This one contains idioms and I personally would not be able to label it. The dataset states it’s positive, but the model predicts it as negative.

- Proposal for improvement

To address the incorrect predictions, I would try to decrease the weight of the tokens that appear more frequently in incorrectly predicted tweets.

Also, I’d try to make it learn to detect spam, because some incorrect predictions seem to happen with tweets with spam.

Besides from preprocessing the text, to improve the performance, I would perform a hyperparameter tuning to find the best values.

Since BERT does the majority of the work, I haven’t tried using a complex network after it, but if I had time I’d like to test the results using a CNN or LSTM network.

Also, since this is a binary classification problem I’d also try using BCELoss instead of CrossEntropyLoss for the loss function.

4. Can you think of a way to measure whether the accuracy is higher for some of the project categories instead of others, e.g. whether sentiment classification accuracy is higher for the Public Spaces category than for Community and Culture? (see the dataset schema for details on the categories.) (1-2 paragraphs for each idea is enough; examples and/or pseudocode would be highly appreciated, too :)

I'd test the tweets category by category.

Eg., I would filter the rows where Category = 'Culture' and then select the important columns, therefore, creating a test dataset for each category. After it, I would evaluate them at the same time and create a function to compare the metrics obtained by each one of them.

5. Can you think of any way to measure whether the metrics are higher for documents containing specific types of entities, e.g. Person vs. Location? (Same as above.)

I'd try to obtain entity related words and link them under an identity through an entity linking task. Then, same as above, I would separate them into different test datasets, evaluate and compare the results.

6. Can you make a proposal to address any/all the issues you have encountered in the evaluation? (Same as above.)

7. Can you bring any idea of new feature to develop at Citibeats for Social Understanding after working on this home assignment ?

One idea of a feature to develop could be a language decoder for text with emojis.

Two same sentences can have a different tone or even meaning just by adding an emoji at the end of one of them.

An emoji accompanying some text can indicate you're being sarcastic, or simply they can add more sentiment to an emotion that was already expressed in a sentence.

Also emoji use in social networks could be analyzed to track trends in diverse groups of the population.