Intro to NLP: Assignment 2. Offensive Language Detection

Content warning: this assignment contains an analysis of offensive language examples.

In this assignment, we will work with the OLIDv1 dataset, which contains 13,240 annotated tweets for offensive language detection. The detailed description of the dataset collection and annotation procedures can be found here. This dataset was used in the SemEval 2019 shared task on offensive language detection (OffensEval 2019).

We will focus on **Subtask A** (identify whether a tweet is offensive or not). We preprocessed the dataset so that label '1' corresponds to offensive messages ('OFF' in the dataset description paper) and '0' to non-offensive messages ('NOT' in the dataset description paper).

The training and test partitions of the OLIDv1 dataset (olid-train.csv and olid-test.csv, respectively) can be found here.

You submit a **pdf** of this document, the format should not be changed.
Your analyses should be conducted using **python 3.8**.
You submit a **zip**-file containing all your code.

Each team member needs to be able to explain the details of the submission. By default, all team members will receive the same grade. If this seems unjust to you, provide an extra statement indicating the workload of each team member.

Total points: 20
Structure:

- Part A: Fine-tune BERT for offensive language detection (7 points)
- Part B: Error analysis with checklist (13 points)
- Bonus tasks: options for obtaining a grade > 8

Fill in your details below:
Group number: 61

Student 1
Name: Ekaterina Geraskina
Student id: 2636212

Student 2
Name: Teo Derizzo
Student id: 2749303

Student 3
Name: Melissa de Gast
Student id: 2674749

Part A: Fine-tune BERT for offensive language detection (7 points)

1. **Class distributions (1 point)**

Load the training set (olid-train.csv) and analyze the number of instances for each of the two classification labels.

Class label	Number of instances	Relative label frequency (%)	Example tweet with this label
1	4400	33.23	@USER you are a lying corrupt traitor!!! Nobody wants to hear anymore of your lies!!! #DeepStateCorruption URL
0	8840	66.77	@USER @USER You are correct.

2. **Baselines (1 point)**

Calculate two baselines and evaluate their performance on the test set (olid-test.csv):

- The first baseline is a random baseline that randomly assigns one of the 2 classification labels.
- The second baseline is a majority baseline that always assigns the majority class.

Calculate the results on the test set and fill them into the two tables below. Round the results to two decimals.

Random Baseline			
Class	Precision	Recall	F1
1, ‘offensive’	0.30	0.53	0.38
0, ‘not offensive’	0.74	0.51	0.60
macro-average	0.52	0.52	0.49
weighted average	0.59	0.52	0.53

Majority Baseline			
Class	Precision	Recall	F1
1, “offensive”	0	0	0
0, “not offensive”	0.72	1	0.84
macro-average	0.36	0.5	0.42
weighted average	0.48	0.67	0.56

3. Classification by fine-tuning BERT (2.5 points)

Run your notebook on colab, which has (limited) free access to GPUs.

You need to enable GPUs for the notebook:

- navigate to Edit → Notebook Settings
- select GPU from the Hardware Accelerator drop-down

➢ Install the simpletransformers library: *!pip install simpletransformers*
(you will have to restart your runtime after the installation)

➢ Follow the documentation to load a pre-trained BERT model: ClassificationModel('bert', 'bert-base-cased')

➢ Fine-tune the model on the OLIDv1 training set and make predictions on the OLIDv1 test set (you can use the default hyperparameters). Do not forget to save your model, so that you do not need to fine-tune the model each time you make predictions.
If you cannot fine-tune your own model, contact us to receive a checkpoint.

a. Provide the results in terms of precision, recall and F1-score on the test set and provide a confusion matrix **(2 points)**.

Fine-tuned BERT			
Class	Precision	Recall	F1
1, “offensive”	0.76	0.65	0.70
0, “not offensive”	0.36	0.50	0.42
macro-average	0.56	0.58	0.56
weighted average	0.48	0.54	0.50

Confusion Matrix: Fine-tuned BERT		
	Predicted Class	
Gold Class	1	0
1	156	84
0	48	572

b. Compare your results to the baselines and to the results described in the paper in 2–4 sentences **(0.5 points)**.

As described in teh paper, BERT-based model constitueted 8% of Machine Learmning algorithms, taht participated in teh competition, yet first 5 top results from the competition were BERT-based models. The best model scored 82.9% on F1 score, next five ranged between 81.5%-80.6%. The BERT-based model, presented in this report scored only 56% on macro-F1 score, which is considerably lower than all the top participators from the paper.

4. Inspect the tokenization of the OLIDv1 training set using the BERT's tokenizer (2.5 points)

The tokenizer works with subwords. If a token is split into multiple subwords, this is indicated with a special symbol.

a. Calculate how many times a token is split into subwords (hint: use model.tokenizer.tokenize()). **(0.5 points)**

Number of tokens: 374691

Number of tokens that have been split into subwords: 63630

Example: if 'URL' is tokenized by BERT as 'U', '##RL', consider it as one token split into two subwords.

b. What is the average number of subwords per token? **(0.5 points)**

Average number of subwords per token: 1.24

Provide 3 examples of a subword split that is not meaningful from a linguistic perspective. **(1 point)**

Which split would you expect based on a morphological analysis?

Example 1: apologetic
BERT tokenization: 'a', '##pol', '##og', '##etic'
Morphologically expected split: apolog ##etic

Example 2: blatant
BERT tokenization: 'b', '##lat', '##ant'
Morphologically expected split: blatant

Example 3: sentencing
BERT tokenization: 'sent', '##encing',
Morphologically expected split: sentenc ##ing

c. BERT's tokenizer uses a fixed vocabulary for tokenizing any input (model.tokenizer.vocab). How long (in characters) is the longest subword in the BERT's vocabulary? **(0.5 points)**

   Length of the longest subword: 18

   Example of a subword with max. Length: Telecommunications

ASK THIS THOUGH ON THURSDAY

Part B: Error analysis with checklist (13 points)

Often accuracy or other evaluation metrics on held-out test data do not reflect the actual model behavior. To get more insights into the model performance, we will employ three different diagnostic tests, as described in https://github.com/marcotcr/checklist.

Relevant literature:

- https://homes.cs.washington.edu/~marcotcr/acl20_checklist.pdf
- https://arxiv.org/pdf/2012.15606.pdf

Creating examples from existing datasets via perturbations (10.5 points)

Use a subset of the OLIDv1 test set, which contains 100 instances:
(olid-subset-diagnostic-tests.csv, can be found in the same directory) and run the following tests:

5. **Typos (6 points)** Spelling variations are sometimes used adversarially to obfuscate and avoid detection (Vidgen et al., 2019; subsection 2.2), that is, users introduce typos to avoid their messages being detected by automated offensive language/hate speech detection systems. Let us examine how it influences our offensive language detection model.
   Use checklist to add spelling variations (typos) to the subset (olid-subset-diagnostic-tests.csv) and evaluate the model's performance on the

perturbed data. Use a fixed random seed (`np.random.seed(42)`) to facilitate comparison.

Quantitative analysis:

c.

	Fine-tuned BERT for original data			Fine-tuned BERT for typo-ed data		
Class	Precision	Recall	F1	Precision	Recall	F1
1, “offensive”	0.90	0.72	0.80	0.87	0.66	0.75
0, “not offensive”	0.22	0.50	0.31	0.23	0.50	0.31
macro-average	0.56	0.61	0.55	0.55	0.58	0.53
weighted average	0.41	0.56	0.45	0.41	0.54	0.43

Confusion Matrix: Fine-tuned BERT for original data		
	Predicted Class	
Gold Class	1	0
1	36	14
0	4	46

Confusion Matrix: Fine-tuned BERT for typo-ed data		
	Predicted Class	
Gold Class	1	0
1	33	17
0	5	45

- Describe the differences in performance compared to the non-perturbed data (precision, recall, F1-score macro). **(1 point)**

The perfomance of Fine-tuned BERT for original and typo-ed data differs per class. For OFFENSIVE class both precision and recall measures decline considerably for typo-ed data (0.87 and 0.66, respectively), as compared to the original data (0.90 and 0.72). For NOT OFFENSIVE class, the precision measure is even higher for typo-ed data (0.23) as compared to original data (0.22), whereas the recall stays the same for both datasets. Nevertheless, macro-F1 measure still shows superior perfomance of Fine-tuned BERT for original data (0.55) as compared to typo-ed data (0.53)

- How many messages were identified correctly in the original dataset, but erroneously after the perturbation? **(1 point)**
  4

Qualitative analysis:

- Check the add_typos function in checklist pertub.py. How were the typos introduced? **(1 point)**
  Random characters inside a string were swapped with their neighbors
- Provide an example of a typo that cannot be produced by this function, but would play a role in offensive language detection. **(0.5 points)**
  A neighborhood typo. When one of more characters in a world are replaced by their neighbor on the keyboard. (hello = gello).
- Provide 3 examples when the model failed to assign the correct label after perturbation. **(1 point)**
  ['#Christian #America – If we go by #Trmup's example, where liberals support open borders, I guess conservatives support school shotoings. Please explain how this makes America great agai.n']
  ['#Antifa: Take note of how to protest with civility. This si why socialism (aka communims) will never win. It is inherently evil adn unsustainable. URL']
  ['@USER ho fukc fof 😂😂']
- What is the main source of the erroneous predictions produced by the model (main source of errors caused by typos)? **(1 point)**
  Noize and inability to place typoed words to pos-tags.
- How can the model be improved? **(0.5 points)**
  One can train the model on data with typos directly.

6. **Negation (4.5 points)** Offensive language detection models have been shown to struggle with correctly classifying negated phrases such as "I don't hate trans people" (Rottger et al., 2021; subsection 2.2).
   Add negations to the subset and evaluate the model's performance on the perturbed data.
   Qualitative analysis:

- Check the add_negation function in checklist pertub.py. What kind of negations does it produce? **(1 point)**
  It produces negated verbs and auxillaries in positive statements of exclamations (skipping questions)
- Look at the created negated sentences, are they linguistically correct? Provide 2–5 examples of linguistically incorrect sentences. **(1 point)**
  Not all of them are grammatically correct. For example:
  ["@USER @USER Who the hell doesn't he think he is?"]
  ```
  ['#BCOG @USER she is not cool, confident, cocky, and she kicks ass!!! @USER @USER @USER URL']
  ["@USER I don't have three accounts. Two quasi-anonymous, and one completely anon. And if twitter ever bans one of the quasi-anon, I'll make two anon accounts and tweet the most insane shit. I will drown twitter in bullshit. If one of those is banned, I'll make even more."]
  ```
- Check the first 10 negated messages. For which of these negated messages should the label be flipped, in your opinion? **(1 point)**

1.["@USER @USER Who the hell doesn't he think he is?"]
Original label: '1' Proposed new label: '1'
2. ["#BREAKING. #Greece: Molotov cocktails don't fly after protest honouring killed antifa arti... URL via @USER URL"]
Original label: '1' Proposed new label: '1'
3. ['#OrrinHatch I can't believe this sexist , clueless, old fart gets to weigh in on another woman's charges against a Supreme Court nominee. And he is not spouting the same old nasty shit he spewed 20+ years ago against Anita Hill. His time's up! Good riddance Neanderthal!']
Original label: '1' Proposed new label: '1'
4. ["@USER @USER I'll use that one the next time im in a gun control debate or in a debate about free speech or taxes. Yes you can't choose to be irresponsible or choose not to be. I argue responsible. Whats wrong with that? Don't justify murder by saying it was never alive or its my right."]
Original label: '1' Proposed new label: '1'
5. ["0-1 didn't lose my acca on the first fucking fight cba"]
Original label: '1' Proposed new label: '1'
6. ['#Bakersfield is not why we need gun control! Sorry for the victims other than the gunman himself I hope he rots in hell!']
Original label: '1' Proposed new label: '1'
7. ["#Christian #America – If we go by #Trump's example, where liberals support open borders, I guess conservatives support school shootings. Please don't explain how this makes America great again."]
Original label: '1' Proposed new label: '1'
8. ['@USER @USER @USER She is not the most disingenuous person in the Senate. If she was my Senator I would hide in shame after her despicable behavior in the SCOTUS hearings.']
Original label: '1' Proposed new label: '1'

9. ["#Democrats #Liberals you are being #threatened by #Armed #Nazis and you talk about #disarmament of yourselves and of #Disconnection from the #Military #Lifestyle ? You shouldn't #broadcast your #Armory and #killSkills #RightFuckingNOW URL"]
Original label: '1' Proposed new label: '1'
10. ["699. Just didn't want to tell you you should Hang Out With Me More. i think it'll be fun! Also, lmao dick for a tongue? So people can easily suck me off. HAHA kidding aside, i aint want a tongue as my dick that looks hella weird.. but my choice was weird to but WHATEVER"]
Original label: '1' Proposed new label: '1'

- Provide 2 examples when the model correctly assigned the opposite label after perturbation and 2 examples when the model failed to identify negation. Fill in the table below **(1 point)**

Examples correct	Tweet ID	Original label	Expected label after negation	Model prediction	Discussion: what is the potential reason for this behavior?
1["@USER @USER Who the hell doesn't he think he is?"]	89200	1	1	1	The model still identified this tweet as offensive most likely due to the fcat, that the negation made it grammatically incorrect, yet it still contains an idiomatic offensive turn "who the hell … think he is?"
2['#OrrinHatch I can't believe this sexist , clueless, old fart gets to weigh in on another woman's charges against a Supreme Court nominee. And he is not spouting the same old nasty shit he spewed 20+ years ago against Anita Hill. His time's up! Good riddance Neanderthal!']	55633	1	1	1	In this statement, there are several sentences, and not in all of them the verbs are negated. Moreover, it contains plenty offensive adjectives.
Examples wrong	Tweet ID	Original label	Expected label after	Model prediction	Discussion: what is the potential reason for this behavior?

			negation		
1["#Christian #America – If we go by #Trump's example, where liberals support open borders, I guess conservatives support school shootings. Please don't explain how this makes America great again."]	60466	1	1	0	Probably because the tweet does not contain any obviously offensive languge to begin with.
2["#Democrats #Liberals you are being #threatened by #Armed #Nazis and you talk about #disarmament of yourselves and of #Disconnection from the #Military #Lifestyle ? You shouldn't #broadcast your #Armory and #killSkills #RightFuckingNOW URL"]	16333	1	1	0	This statement does not necessarily make much sense after the negation. Nevertheless, since the call to violence was negated, it could be the reason the model decided its not offensive anymore.

- How can the model be improved? **(0.5 points)**
  It can be trained on the model with grammatically and semantically correct negations, as in, for at least 30% of offensive statements one can create negated examples which are not offensive anymore. Unfortunately, this is very time-consuming and can be sometimes unfeasible (it is hard to turn a statement with a lot of offensive adjectives into something not offensive while still leaving the similarity, for example).

Creating examples from scratch with checklist (2.5 points)

7. **Creating negated examples**
   Let us further explore the impact of negations on our offensive language detection model.
   Consider the following templates: '*I hate …*' and '*I don't hate…*', and fill in the templates below:
   - Use masked language model suggestions: 'I hate {mask}' and 'I don't hate {mask}' .
   - Offensive language is often directed towards minority groups. Use the built-in lexicon and explore: 'I hate {nationality}', 'I don't hate {nationality}', 'I hate {religion}', 'I don't hate {religion}'

   Run the model on the created examples.
   - Provide 3 examples when the model assigns the correct label (correct label according to you) and 3 examples when the model fails to assign the correct label (choose both from masking and lexicon suggestions) **(1 point)**
     Correct:
     I hate gays. is 1
     I hate computers. is 0
     I hate Republicans. is 1
     Wrong:
     I hate cats. is 1
     I hate robots. is 1
     I don't hate dogs is 1
   - Analyze the examples. Can you think of a reason why some examples are classified as offensive while others are not? **(1 point)**
     It could be the case, that it often assignes 'i hate' statements for minorities, political parties and religions as offensive. It can also be assumed, that it assignes 'i hate + [animal]" statements as offensive, since it is a widespread practice to use animal names as an insult (pigs = cops, etc)
   - How can the model be improved? **(0.5 points)**
     The model in this assignment was trained on olid-train dataset, which, as described in the respective paper, is mostly targeted at political content, since it usually contains more offensive stetements. Yet, most statements produced by the CheckList do not neccessarily contain many political statements. Moreover, not-offensive statements in the train dataset are less likely political. Therefore, model can sometimes fail to identify offensive language outside of political statements, which leads to the need to train it on neutral offensive and political not offensive data as well.

BONUS:
Develop 2 new diagnostic tests (you can use checklist): describe what they test, explain why they are relevant and implement them. Run the tests and describe your observations. Provide

examples of difficult cases, that is, when the model fails to assign the correct label. Discuss potential sources of errors and propose improvements to the model.