Few-shot classifiers trained with contrastive data triplets generated from a custom dataset consisting of racially/culturally exclusive and inclusive speech in both English and French langauges

Blake Preston - 19280019

Final year thesis for:
Bachelor of Science (Honours) Computer Science and Information Technology

Supervisor: **Dr. Bharathi Raja Chakravarthi**

School of Computer Science
University of Galway
Galway City, Ireland

4, 2023

Table of Contents

Acknowledgements

I would like to thank Dr. Bharathi Raja Chakravarthi for his reliable direction throughout my research.
Also, thank you to Prasanna Kumaresan and Rahul Ponnusamy for proofreading and revision.
Special thanks to Jean d'Alès and Nell James for correcting the french dataset.

Abstract

There is an insurmountable amount of negativity on the internet. However, not all negativity is deliberate. This paper focuses on detecting racially and culturally exclusive speech with a neutral sentiment. This is preliminary work for an artificial intelligence that can detect exclusive social media posts and then suggest modifications to the post, so the user can easily amend the exclusive post to be inclusive. The solution will not impose on free speech, rather it is an aid to those who desire to improve their speech to be inclusive of all people.

Promising results have been achieved in detecting exclusivity in French and English language examples using few-shot learning. More specifically, a convolutional neural network classification head was appended to the selected pre-trained NLP classifiers. The head was trained using SetFit(26), which generates contrastive learning triplets to lessen the distance between sentence embeddings of the same label. This makes generalizing exclusivity, regardless of specific race/culture, much easier due to word embeddings being clustered closer together. The best-performing model scores %97 macro F1 score and reliably generalizes exclusivity without racial bias, proven by testing on fictitious fantasy races. You can access this model on HuggingFace[1]

keywords = zero-shot learning, few-shot learning, sentence transformer, k-fold-validation, word embeddings, convolutional neural network, classification head.

[1] `https://huggingface.co/BeToast/xml_xnli__inclusiveORexclusive__binary_classification__frenchANDenglish`

1 Introduction

On the internet, hate speech is rampant and content moderation is expensive. Research done in the field of hate speech detection has indicated the problem is more difficult than we know.(3) Furthermore, many state-of-the-art HSD approaches are brittle(14), most notably Google's Perspective API(15) In research, there is a tunnel vision towards hateful sentiment, disregarding that a statement can be quite hateful with a neutral or positive sentiment. My research uses a novel dataset of neutral sentiment sentences pertaining to race or culture which are labelled exclusive or inclusive. Pretrained NLP models are fine-tuned with contrastive data triplets to classify sentences as inclusive or exclusive in both French and English languages.

Nearly all text classification tasks struggle with insufficient data. The most obvious approach to combat a tiny dataset is to use zero-shot and few-shot learning. Alternatively, there have been attempts to manufacture larger datasets via generative methods in the domain of hatespeech detection.(28) I am weary of amplifying dataset bias(27) when generating heaps of train data from just a few seed training examples so data generation was used sparingly. For these experiments, I created a novel dataset following the European Commission and American Physiological Association guidelines. This dataset went through multiple revisions and corrections due to the importance of dataset trustworthiness. After initial dataset creation, I used GPT3 to generate loads more sentence pairs, however, I quickly discovered the exclusive sentences generated were typically redundant and incorrect. Only 71 sentence pairs generated by GPT3 were acceptable for the dataset. You can view them here[2]. Researchers must be extremely careful about generative approaches in AI data for NLP classification tasks which need to generalize a global concept, such as exclusivity. This paper on facial image generation is a great example of unsatisfactory training data limiting the diversity of ML generation(12)

Due to the dangers of generative approaches, it is best to fine-tune a large pretrained model, like BERT, roBERTa, etc, because they have been trained on a colossal corpus resulting in a broader understanding overall and need fewer train data to converge.

[2] https://docs.google.com/spreadsheets/d/1aEG9GzODGJI29bkND8GgqRWqeoHrMQ6W-1QfFzc46yA/edit?usp=sharing

2 Literature Review

2.1 A call to inclusivity

Tribalism is natural in humans. Research has been done on the quantifiable benefits of aggression for the aggressor. ”According to the recalibration theory of anger, anger is an evolved regulatory program designed to orchestrate the deployment of these tools to cost-effectively bargain for better treatment and resolve conflicts of interest in favor of the angry individual.” (25) Today, aggression and exclusivity on the internet are common in political domains to progress personal agendas. ”Political contests are highly consequential because they determine how society will allocate coveted resources such as wealth, power, and prestige.” (9) Furthermore, there will always be self-centered conflict as it is in our nature. However, we must strive towards peace on the internet and go against our innate inclination to tribalism. ”Peace is not the absence of conflict, but the ability to handle conflict by peaceful means.”(Ronald Reagan). An inclusive internet is a large mountain to climb, but we can foster a peaceful inclusive internet to suppress our naturally exclusive dispositions.

2.2 Evidence for the prevalence of exclusivity

There is ample existing research fueled by awareness of human exclusivity. For example, A study of 8969 job advertisements in Indonesia was conducted to identify illegal discrimination in the recruitment process. This is important because ”recruitment is a critical gateway to economic opportunity” Most notably, 66.27% of job adverts were age discriminatory, and 38.76% were gender-discriminatory (18).

2.3 Discussion on existing hate speech detection approaches

In 2017 Google released Perspective API, an API that gives a toxicity rating to the text given it. A significant amount of research has been done incorporating perspective API for hate speech classification, but it is a mediocre algorithm for detecting hate speech. ”Deceiving Google's Perspective API Built for Detecting Toxic Comments” (15) is a paper which takes high toxicity comments and slightly modifies them to get quite low toxicity scores while maintaining hateful meaning. ”The existence of such adversarial examples is very harmful to toxic detection systems and seriously undermines their usability.” (15)
View these few examples:

- They are liberal idiots who are uneducated (90% toxicity)
- They are liberal i.diots who are un.educated (15% toxicity)
- Climate change is happening and it's not changing in our favor. If you think differently you're an idiot. (84% toxicity)
- Climate change is happening and it's not changing in our favor. If you think differently you're an idiiot. (20% toxicity)

Perspective API is momentum towards internet inclusivity, but it has obvious shortcomings. A more robust solution must be created for hate speech detection and for the border goal of an inclusive internet. We also need to consider the small data size for all subsets of exclusive speech. Hate speech, the largest subset of exclusivity is still a small niche and typically uses small to modest-sized datasets. Towards Hate Speech Detection at Large via Deep Generative Modeling (28) explores hate speech detection with deep learning training on one million examples. These examples were ”produced by a deep generative model” and were able to ”[demonstrate] significant performance improvements across five hate speech datasets”. This approach seems best, but we must be wary of generated examples drifting away from human exclusivity or amplifying bias in the seed dataset. With ample human supervision and a healthy diversity of seeding examples, this approach should be paramount in solving the predicament of small data size. Contrarily, few-shot learning is able to achieve impressive results on a minimal dataset which is proven in the results and discussion section.

2.4 Exclusive speech detection concepts

To begin, we need to understand the classification problem we have. Hate speech cannot be understood merely with diction or grammatical analysis. Conceptually, it is similar to sarcasm detection. Consider this, ”The striking property of satire is that it makes it difficult to bridge the gap between its literal and intended meaning.” (1) Focusing on our scope of racial inclusivity, even if we could infer the intentions of the authors’ hearts, it would still not be enough to solve this classification problem because racially exclusive language is also used ignorantly and accidentally. A 1995 study from Yale (4) discovered, ”Social behavior is ordinarily treated as being under conscious (if not always thoughtful) control.

Considerable evidence now supports the view that social behavior often operates in ”an implicit or unconscious fashion” Furthermore, a lot of exclusive speech in present social media is accidental(with no hateful intention), and typical classification approaches do not give any attention to the accidental case. This research explores the neglected area of neutral sentiment exclusive speech. Train data is presented in exclusive and inclusive pairs so the model can find key differences between the exclusive and inclusive, regardless if the author is hateful or ignorant. Ponder the following cases:

1. Hateful racial exclusivity
   - Detectable by context and textual analysis.(29)
   - Strongly indicated by understanding the author and the target. Explored in this paper by getting extensive data from the targeted racial subgroup for tailored algorithms(16)
2. Non-hateful/accidental racial exclusivity.
   - Solely detectable by context and textual analysis.

These cases are fundamentally different. However, both can be detected by textual analysis. By taking this approach, nothing negative is presupposed about the heart of the user who authored the sentence.

2.5 Approaches to social media moderation

Once we have detected exclusive speech, what next? Research has been done to map commonalities between banned and existing accounts to find active negative users. (20) This approach is very concise for removing hateful people from websites, but it does not solve the issue of hateful thinking. Rather, it creates more division and controversy on the internet, which directly opposes the end goal of unity. Also, companies do not want to remove users from their websites because that will lessen revenue.

I believe the best solution is the following; If a user is hatefully exclusive, suspend their post until they make it inclusive. If a user is accidentally exclusive, send them a notification kindly asking them to reconsider and edit their message to be inclusive. Of course, we cannot infer the intent of a person from a single sentence. Furthermore, we must assume neutral sentiment exclusivity to be accidental. When exclusivity is detected, we should notify the user that their sentence is exclusive and recommend inclusive modifications. An inclusive text generation model will be necessary to suggest the changes and I intend to develop this model in the coming years.

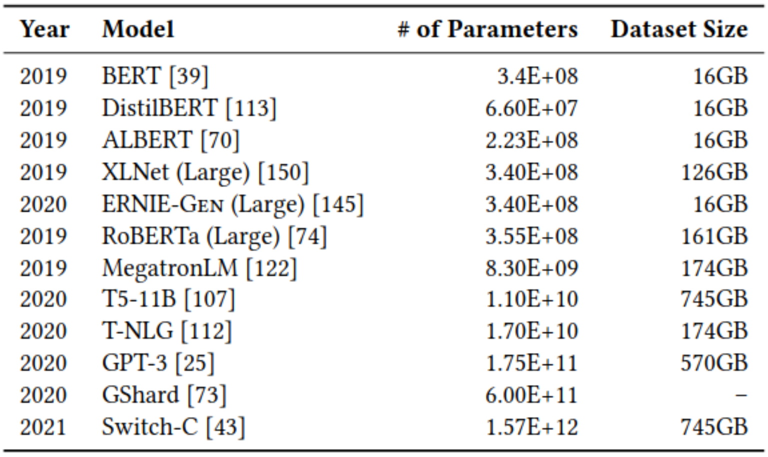
2.6 Satire in NLP

A sentence can have the literal meaning be far from its intended meaning, this is the case for satire and sarcasm. Research on satire finds that Artificial Intelligence struggles when the literal meaning is unlike the intended meaning.(1) For example, hateful sarcasm could be used to degrade an individual online and go completely undetected by typical sentiment analysis methods useless. I suppose to improve sarcasm detection, we would need to have reliable in-context analysis.

2.7 Massive NLP models for zero-shot

GPT-3(Generative Pre-trained Transformer 3) is an impressive accomplishment in AI. It is able to perform very well on various NLP tasks at the zero-shot level. The is an environmental cost for all these colossal DNNs which are exponentially growing with time. Consider this quote and table from *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?* (5) "Training a single BERT base model (without hyperparameter tuning) on GPUs was estimated to require as much energy as a trans-American flight."

To lessen inefficiencies, research has been done to only use a subset of these DNN's for each operation resulting in improved computational efficiency. One solution is *Mixture of Experts*, where the DNN is replaced by thousands of FFNs that are sparsely activated by the *Gating Network* if knowledge is relevant. "[Mixture of Expert] models achieve significantly better results than state-of-the-art at lower computational cost"(23), so why is this method not widely adopted? *Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity*(13) discusses this, "despite several notable successes of MoE, widespread adoption has been hindered by complexity, communication costs, and training instability." This paper outlines various successful approaches to refine the problems initially addressed by MoEs and "obtain[s] up to 7x increases in pre-training speed with the



Year	Model	# of Parameters	Dataset Size
2019	BERT [39]	3.4E+08	16GB
2019	DistilBERT [113]	6.60E+07	16GB
2019	ALBERT [70]	2.23E+08	16GB
2019	XLNet (Large) [150]	3.40E+08	126GB
2020	ERNIE-Gen (Large) [145]	3.40E+08	16GB
2019	RoBERTa (Large) [74]	3.55E+08	161GB
2019	MegatronLM [122]	8.30E+09	174GB
2020	T5-11B [107]	1.10E+10	745GB
2020	T-NLG [112]	1.70E+10	174GB
2020	GPT-3 [25]	1.75E+11	570GB
2020	GShard [73]	6.00E+11	–
2021	Switch-C [43]	1.57E+12	745GB

Figure 1: Overview of recent large language models (5)

same computational resources.”

3 Methodology

This flowchart shows the entirety of the project:

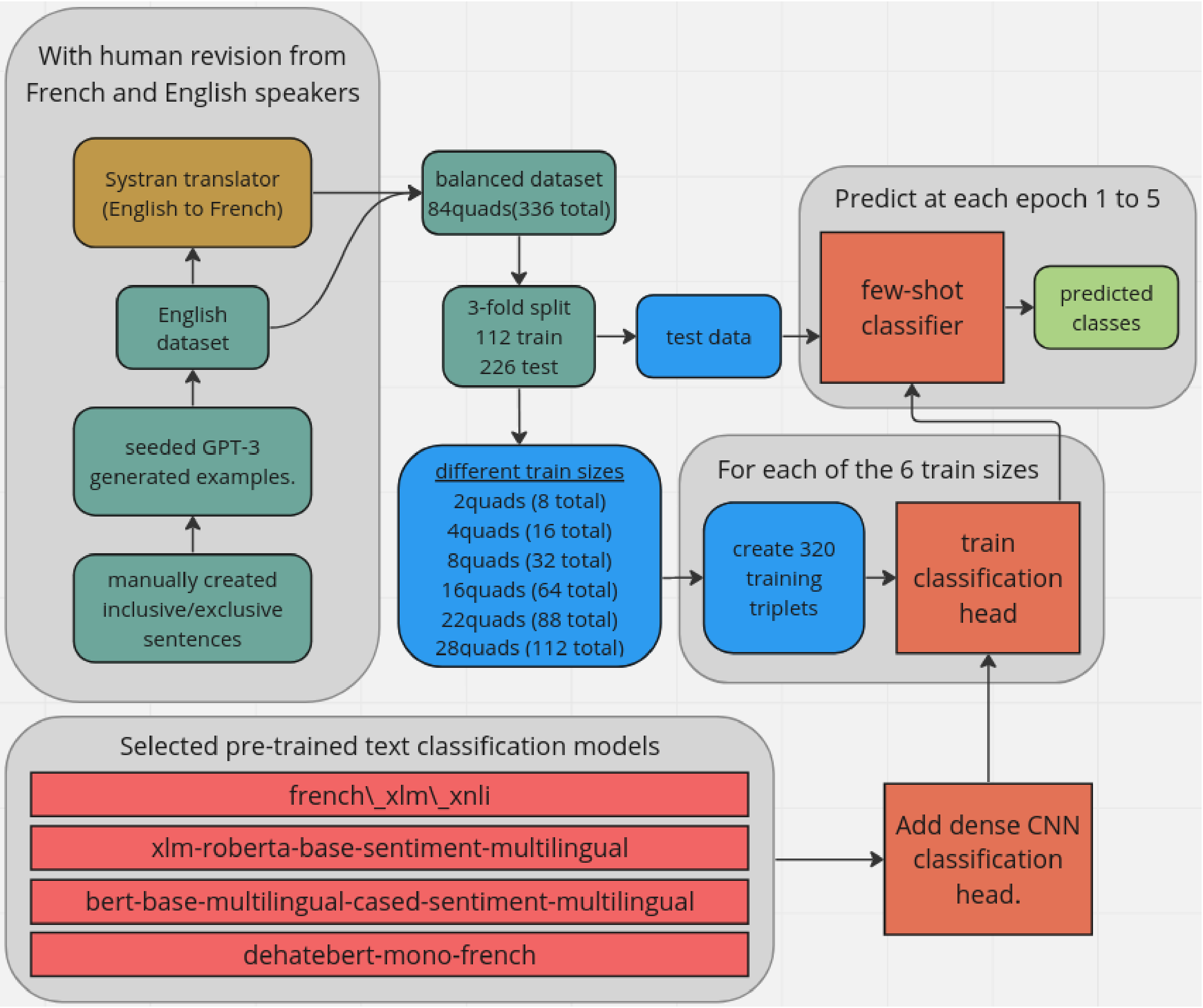

With human revision from French and English speakers
Systran translator (English to French)
English dataset
seeded GPT-3 generated examples.
manually created inclusive/exclusive sentences
balanced dataset 84quads(336 total)
3-fold split 112 train 226 test
test data
Predict at each epoch 1 to 5
few-shot classifier
predicted classes
different train sizes
2quads (8 total)
4quads (16 total)
8quads (32 total)
16quads (64 total)
22quads (88 total)
28quads (112 total)
For each of the 6 train sizes
create 320 training triplets
train classification head
Selected pre-trained text classification models
french_xlm_xnli
xlm-roberta-base-sentiment-multilingual
bert-base-multilingual-cased-sentiment-multilingual
dehatebert-mono-french
Add dense CNN classification head.

Figure 2: Flowchart

3.1 Data

This is an explination of how the data was gathered and structured.

3.1.1 Data Gathering

Following the European Commission and American Physiological Association examples and guidelines, I manually constructed 40 sentence pairs. Each pair is an exclusive sentence with an inclusive alternative sentence because we are going to use SetFit(26). SetFit creates triplets from two training data and a boolean value representing if the selected data pair is the same class. Starting with a pre-trained sentence transformer, these triplets are used to train a convolutional neural network classification head with logistic regression. Focusing again on our data, my original 80 sentences were not all acceptable and after peer review where they were distilled down to 13 good pairs. These 13 pairs were used to seed data generation with GPT3(7), an autoregressive language model. Hundreds of sentence pairs were generated, unfortunately, due to redundancy and incorrectness, there were only 71 acceptable generated sentence pairs. Combining our manually made 26sentences and 142generated sentences produced a balanced English dataset of 168. The English dataset was translated to French with SYSTRAN[3]. Systran was founded in 1968 in La Jolla, California and has proven its reliability by its persistent relevance. It is especially good for French translation because the Systran headquarters is now located in Paris, France. The Systran translations were proofread by two fluent French speakers, Jean d'Al'es and Nell James, who are native to France. Ideally, we could create a model which is able to detect exclusivity in any language which can be implemented on social media to foster healthier cross-cultural communication on the internet. Moving towards this goal, I have chosen French and English to start because they are high-resource languages and sufficient pre-trained French/English sentence transformers are available for few-shot learning. As a result, these experiments are run with a combined dataset of 84 French and 84 English sentence pairs. The total dataset is balanced with 336 sentences.

Table 1: Dataset description for each label in all languages

Languages	**labels**	**counts**
English	Exclusive	84
	Inclusive	84
French	Exclusive	84
	Inclusive	84
Total		336

The data is not labeled with the language. Datasets are commonly You can view the whole dataset on GitHub page[4]. Bias in datasets is a massive issue and undermines the credibility of any AI created with bias. (21) (30) (6) Giving deserved attention to avoiding bias, I was extremely conscious of equality while

[3] `https://www.systran.net/en/translate/`

[4] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/data_unformatted`

creating this dataset. Of course, it cannot be perfect, but I believe this dataset is diverse enough to fairly generalize exclusivity and enable the produced models to detect exclusivity regardless of the race or culture a sentence is pertaining to.

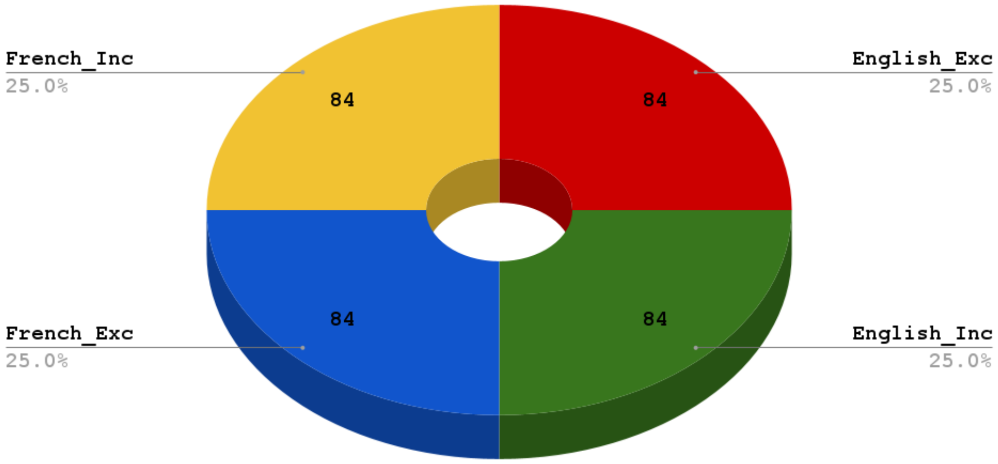


Figure 3: Data distribution

3.1.2 Data setup for experiments

The dataset is split in to 3 folds, each with 224 test. That leaves 112 remaining for training. Each entry in the train data is a *quad*. A *quad* is as follows: (english inclusive, english exclusive, french inclusive translation, french exclusive translation). Here is a breakdown of each train set made in each fold:

- 2quads(8 train)
- 4quads(16 train)
- 8quads(32 train)
- 16quads(64 train)
- 22quads(88train)
- 28quads(112train)

The 2quads train data is created by randomly selecting two sentences from the English inclusive dataset, their corresponding English exclusive, and the french translations. 4quads is made by concatenating 2quads and then retrieving 2 random new quads. Here is an example of 2 quads train dataset shown in **Table 3**

Note: the language of the sentence is not included and the classifier will not be given a language label.

For every larger train dataset, the smaller examples are included in it. For example, 4quads contains all of the 2quads dataset plus an additional randomly selected 2 different quads. 8quads is created by taking the 4quads dataset plus an additional randomly selected 4 different quads. This cycle repeats all the way to 28quads.

Table 2: breakdown of training sentences in each quad

2quads			**16quads**		
English	Exclusive	2	English	Exclusive	16
	Inclusive	2		Inclusive	16
French	Exclusive	2	French	Exclusive	16
	Inclusive	2		Inclusive	16
Total Sentences		8	Total Sentences		64
4quads			**22quads**		
English	Exclusive	4	English	Exclusive	22
	Inclusive	4		Inclusive	22
French	Exclusive	4	French	Exclusive	22
	Inclusive	4		Inclusive	22
Total Sentences		16	Total Sentences		88
8quads			**28quads**		
English	Exclusive	8	English	Exclusive	28
	Inclusive	8		Inclusive	28
French	Exclusive	8	French	Exclusive	28
	Inclusive	8		Inclusive	28
Total Sentences		32	Total Sentences		112

Table 3: Example of 2 quads train dataset

label	Sentence
1	All cultures should be respected and celebrated for their unique contributions and traditions.
0	White culture is superior to all others.
1	Toutes les cultures doivent être respectées et célébrées pour leurs contributions et leurs traditions uniques.
0	La culture blanche est supérieure à toutes les autres.
1	Peruvians hold a variety of jobs.
0	Peruvians are only good for working on farms.
1	Les Péruviens occupent des emplois variés.
0	Les Péruviens ne sont bons que pour travailler dans les fermes.

Table 4: Encoded classes for the dataset

Classes	Encoded labels
Exclusive	0
Inclusive	1

3.2 Sentence Representations

The following subsections are the methods used to construct vectors that represent the sentences. These vectors are passed through the ML algorithms discussed in the Baseline ML algorithms section.

3.2.1 TFidf

Term Frequency by Inverse Document Frequency is a standard metric used in information retrieval used to quantify the importance of a word within a corpus. The term frequency is the count of word x divided by the word count of x a document. TF is multiplied by Idf, the number of documents containing x divided by the number of documents in the whole corpus.

As we know, TFidf is incapable of understanding semantic information. TFidf is trivial with no way to represent attention or meaning and is only used here to make the point that understanding exclusivity, cannot be learned by solely TFidf and sentences need a much more complex representation to understand semantic information. It is important to probe low-resource ML algorithms for performance, before creating a resource-intensive classifier without justification.

3.2.2 Sentence-BERT

Bidirectional Encoder Representations from Transformers, BERT by Google, is "designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers." (11) Bert was a pioneer in attentive encodings and earned a reputation as a reliable word encoder. Two of our few-shot models are built using the Bert base model, which is a testament to its reliability. Notably, Bert has replaced traditional techniques and become a

NLP standard.(10)(24) Sentence-BERT does well on clustering tasks which should enable better performance with our ML algorithms. SBERT produces sentence embeddings mapped in a 512 dimensional dense vector space using Siamese BERT-Networks.(19) This is how SetFet, the fewshot framework we use, creates contrastive training triplets for the fewshot models. The difference here is we take the same sentence embeddings and feed them into a traditional ML models, instead of training a CNN classification head. Theoretically, with a robust enough encoder, which can cluster sentences in a linear manner, we would not need to use a neural network classification head at all to produce the same level of predictions. The trade-offs of these two approaches will be discussed in the results section.

3.3 Baseline ML algorithms

Here is a brief explanation of every classic ML algorithms used for baseline results and the code used to run these algorithms

3.3.1 ML algorithms

Here are the baseline ML models used to evaluate TFidf and SBERT encodings. With baselines results we can gauge the performance of our few-shot models. We choose a diverse set of seven classifiers for our baseline models, including NaiveBayes(gaussian), LinearSVM, LogisticRegression, RandomForestClassifier, DecisionTreeClassifier, and SGDClassifier. I will give a brief explanation of each now if you are unfamiliar with these classic ML algorithms.

Naive Bayes(gaussian): Naive Bayes is a probabilistic algorithm based on Bayes theorem, which assumes that the features are independent of each other. The Gaussian Naive Bayes variant assumes that the continuous features follow a Gaussian distribution. I do not know how our data is clustered in the vector space, however, I hypothesize Gaussian was a good mathematical model as it is malleable and fits to the train data supplied, unlike linear models. The performance will be limited because NaiveBayes is a simple algorithm that works especially well on fewer features.

Linear SVM: Linear Support Vector Machines (SVMs) are widely used for text classification tasks due to their ability to handle high-dimensional and sparse data. Due to the computational efficency of SVMs they are preferable compared to a neural network classification head. SVMs find the hyperplane that maximizes the margin between the two classes. Linear SVMs are able to easily handle large data enabling the larger scope of detecting exclusivity in real time on social media.

Logistic Regression: Logistic Regression is a probabilistic algorithm that models the probability of the output variable given the input variables. LR is also quite good for classification because it will yeild a probability of each class given the input data. With this ML model specifically, I am concerned there might be misleading features in the dataset which are not true indicators of exclusivity, but are being used by LR to classify correctly.

Decision Tree: Decision Trees are simple and interpretable model that consists of a tree of decisions. Data is fed through the tree until it reaches a leaf which is its predicted class. Opposed to other ML algorithms, there is no way to quantify the certainty of the predicted class because the leaf you end on is the only output.

Decision trees are able to handle noisy data as long as there are reliable features to guide classification.

Random Forest: Random Forest is a modification of the Decision Tree algorithm. Each epoch, the tree is pruned where classification is reliable and more branches are made in difficult parts of classification. As a result, Random Forests are flexible and can handle classification tasks well when supplied with reliable features.

SGDClassifier: Stochastic Gradient Descent (SGD) is an optimization algorithm that is widely used in machine learning to minimize computation time with convergence taking more iterations. Stochastic Gradient Descent is used to optimize the logistic loss function of the classifier by estimating the gradient instead of completely calculating the gradient. Because SGD is an estimator, it is especially computationally efficient on high-dimensional data. This will compliment the 512 dimensional SBERT encodings well.

3.3.2 Code for ML models

The Scikit-learn [5] library was used to run these classic ML algorithms. Scikit-learn, aka sklearn, is a popular open-source library for Python built on top of NumPy, SciPy, and Matplotlib that implements a vast variety of pre-processing, machine learning, and evaluation functions. Here is the list of imports used to run the ML baselines. The SBERT encoder was imported from HuggingFace, which is an open-source library of pre-trained models and AI methods for python.

from sklearn import:

- preprocessing.StandardScaler[6]
- feature_extraction.text.TfidfVectorizer[7]
- svm.LinearSVC[8]
- naive_bayes.GaussianNB[9]
- linear_model.LogisticRegression[10]
- linear_model.SGDClassifier[11]
- ensemble.RandomForestClassifier[12]

[5] https://scikit-learn.org/

[6] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[8] https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[9] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[11] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier.html

[12] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

- tree.DecisionTreeClassifier[13]

from huggingface import:

- sentence_transformers.SentenceTransformer[14]

You can view the ML model code on github[15]

3.4 Zero-shot

Here is a short discussion about zero-shot and the coding tools used to run the model.

3.4.1 Short Discussion on zero-shot

Because our dataset size is only 336, is it most reasonable to use a pre-trained model for this classification task. Also, pre-trained models have an understanding of exclusivity due to the massive corpus they are trained on. I would argue few-shot learning is the most rational approach to our problem because we have some data to train with. Therefore, zero-shot learning will be used as another baseline for our few-shot learning results. Out of our four pre-trained models in use, one of them is specifically fine-tuned for zero-shot learning for French. This is the only model used for zero-shot and it performs quite well, but the same model used with few-shot is a notable improvement in F1 score.

3.4.2 Practical information for zero-shot

As for the code, I used the huggingface transformers.ZeroShotClassificationPipeline with french_xlm_xnli [16] zero-shot classification model. The pipeline is just an API interface for our model for ease of use. The zero-shot model was required to label each sentence either exclusive or inclusive. The **only** information given the model was the label array ['exclusive','inclusive']. The entire dataset was used for test as there is no training step for zero-shot learning.

You can also view the zero-shot code on github[17]

[13] https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[14] https://huggingface.co/sentence-transformers

[15] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/classifications/baselines.ipynb

[16] https://huggingface.co/morit/french_xlm_xnli

[17] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/classifications/zero-shot_tests.ipynb

3.5 Few-shot

3.5.1 Reasoning for few-shot

Our data sentences cover exclusivity targeted at a diversity of races and cultures and do not have hateful sentiment. As a result, our model must be able to generalize exclusivity across a plethora of races and cultures. Due to data size, this generalization is near impossible without using a pre-trained model.
Please view this sentence pair to see exclusivity being expressed with a neutral sentiment:

- African immigrants are only able to seek unqualified jobs.
- Structural aspects, such as a lack of equal opportunities, lead to African immigrants being overrepresented in unqualified jobs.

Here, the first sentence is objectively exclusive with a neutral sentiment which is unique to my experiments and should challenges zero-shot learning models due to common intersection between negative sentiment and exclusivity. Furthermore, zero-shot models struggle to classify neutral sentiment sentences as exclusive. I hypothesize the significant intersection between hate speech, negative sentiment, and exclusive speech will mislead classifications on this dataset. Therefore, few-shot learning is rational for our classification problem due to data scarcity and the uniqueness of this classification problem.

3.5.2 Few-shot Model Selection

I have selected four few-shot models which are able to handle English and French data. The models are two Bert and two Roberta. Between the two Bert models, one is fine-tuned for hate speech and the other for sentiment analysis. The Roberta models are also one for hate-speech detection and one for sentiment analysis.

Here are the pretrained models used in few-shot learning:

- french_xlm_xnli [fine-tuned roberta for hatespeech detection][18]
- xlm-roberta-base-sentiment-multilingual(**XRBSM**) [fine-tuned bart for sentiment analysis][19]
- bert-base-multilingual-cased-sentiment-multilingual(**BBMCSM**) [fine-tuned bert for sentiment analysis] [20]
- dehatebert-mono-french [fine-tuned BERT for hate speech detection] [21]

3.5.3 SetFit

SetFit, Sentence Transformer Fine-tuning, is an architecture for fine-tuning transformers that can produce the same accuracy with less data.(26) I have chosen to use this because we have limited train data.

[18] `https://huggingface.co/morit/french_xlm_xnli`
[19] `https://huggingface.co/cardiffnlp/xlm-roberta-base-sentiment-multilingual`
[20] `https://huggingface.co/cardiffnlp/bert-base-multilingual-cased-sentiment-multilingual`
[21] `https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-french`

Table 5: A visual representation of the above two sentences:

	Hate-Speech	Sentiment
BERT	dehatebert-mono-french	BBMCSM
RoBERTa	french_xlm_xnli	XRBSM

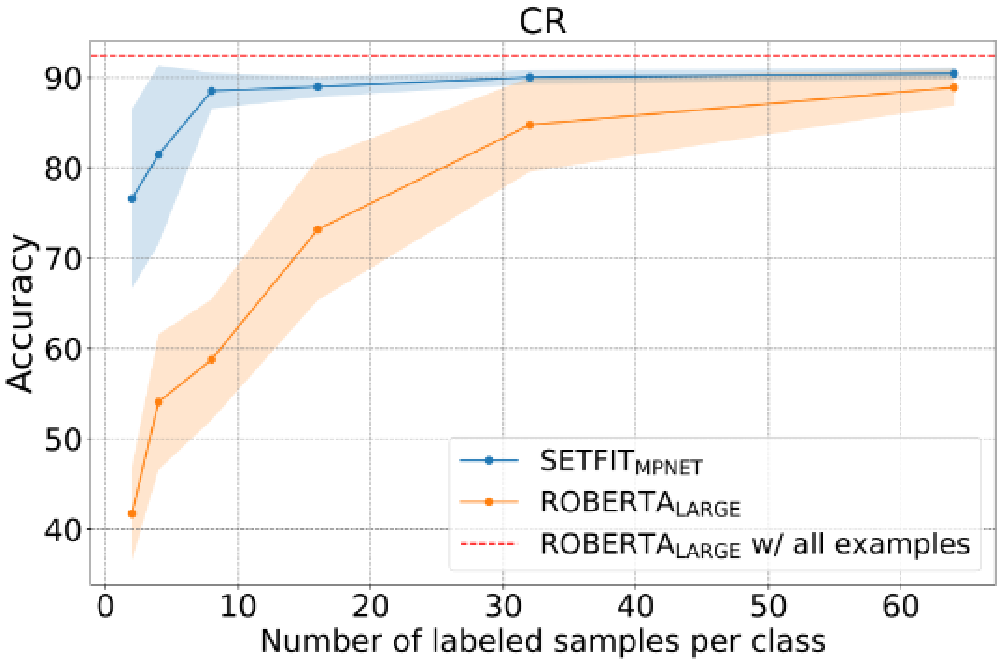


Figure 4: Compared to standard fine-tuning, SETFIT is more sample efficient and exhibits less variability when trained on a small number of labeled examples.(26)

SetFit accomplishes this by creating positive and negative triplets from the training data provided. The triplet is (x_i, y_i, l) where x_i and y_i are sentences and l is the label. $l = 1$ if both x_i and y_i are the same class(i.e. both are inclusive or both are exclusive). $l = 0$ if both x_i and y_i are of different classes(i.e. one inclusive and one exclusive) These triplets are effectively a new set of training data derived from the original data. Equal amounts of negative triples and positive triplets are generated so training data is always balanced. The number of potential training pairs for a binary classification task is $K(K-1)/2$ where K is the number of train data. Furthermore, if all 336 train is used then $336(336-1)/2 = 56280$ train data! We control the amount of triples generated, for each class, by the hyperparameter R. Given number of produced training triples is T and number of classes is C then $T = RC2$.

In our experiments, I change R dynamically to always maintain 320 train triples generated regardless of train set size. We know there is enough data to converge, given data it took in the figure shown above, so our experiments are effectively showing the F1 scores relation to the diversity of the test data. This is an insightful metric to know for the implementation of few-shot models.

3.5.4 Code for few-shot with SetFit

SetFit is open-source on Hugging Face. For these experiments I used: SetFitModel and SetFitTrainer classes. SetFitModel simply instantiates our pre-trained model. SetFitTrainer is used to train the pre-trained model with our parameters, most notably R. The loss function used was simply accuracy which typically could lead to imbalanced prediction, however, our results confusion matrices are well balanced because our data is completely balanced between both classes and languages3. All few-shot tests using setfit are on github[22]

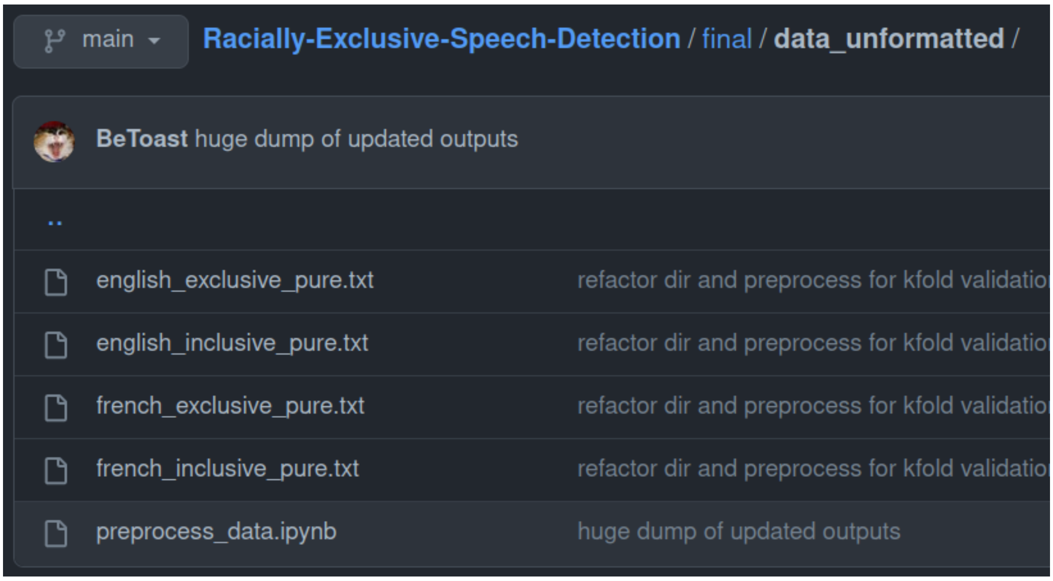
[22] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/classifications/setFit_alltests.ipynb`

4 Experiments

This section is a concise report of exactly the steps taken to run the experiments.

4.1 Dataset preparation

I started with 336 sentences divided into four files: english_exclusive_pure.txt, english_inclusive_pure.txt,
french_exclusive_pure.txt, french_exclusive_pure.txt
These files can be viewed here[23].



main

Racially-Exclusive-Speech-Detection / final / data_unformatted /

BeToast huge dump of updated outputs

..

english_exclusive_pure.txt	refactor dir and preprocess for kfold validatio
english_inclusive_pure.txt	refactor dir and preprocess for kfold validatio
french_exclusive_pure.txt	refactor dir and preprocess for kfold validatio
french_inclusive_pure.txt	refactor dir and preprocess for kfold validatio
preprocess_data.ipynb	huge dump of updated outputs

Figure 5: Unformatted Data files in Git

These files contain 84 sentences each separated by a newline character. Sentences are ordered the same in all files, therefore, getting the sentence at line 5 from each file will be a quad. See a quad example here(**??**)

4.1.1 3fold split

These four files were used to create three folds of train/test data.
View code here[24].
I could not use generic k-fold functions because the quads needed to stick together. Simply put, a list of indexes were randomly selected of appropriate size for each of our 6 train dataset sizes3.1.2. For each fold, one third was train and two thirds for test. You can view the splits used in the experiments here[25]

[23] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/data_unformatted`

[24] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/data_unformatted/preprocess_data.ipynb`

[25] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/data_ready`

4.2 Specifics of testing

The experiments were run on google collab with a standard GPU hardware accelerator and high RAM
All tests, bar zero-shot, were run with identical k-fold split
The tests ran are outlined as follows:

- SBERT and TFIDF sentence representations
  - Naive Bayes
  - Linear SVM
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - Stochastic Gradient Descent
- zero-shot with french_xlm_xnli. Tested on whole dataset.
- few-shot models tested on 1,2,3,4 and 5 epochs.
  - french_xlm_xnli
  - bert-base-multilingual-cased-sentiment-multilingual
  - xlm-roberta-base-sentiment-multilingual
  - dehatebert-mono-french

All model predictions were saved here[26] for later postprocessing. All model training code can be viewed here[27]

4.3 Result calculations

Accuracy, Precision, Recall, macro F1, with standard deviation are calculated on predictions for each model, for each fold, for each of the 6 training sizes
Baseline results code here[28]
Few-shot results code here[29]
All results are available on github here[30]

[26] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/predictions

[27] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/classifications

[28] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/post/evaluate_baselines.ipynb

[29] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/post/evaultate_fewshot.ipynb

[30] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/results

Also, they are reformatted in the Appendix here(9.1.1)
Linegraph code here[31] Confusion matricies code here[32] and can pngs here[33]

[31] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/post/linegraphs.ipynb

[32] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/blob/main/final/post/confusion_matricies.ipynb

[33] https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/confusion_matricies

5 Results and Discussion

This section will review and discuss baseline and few-shot results

5.1 Baseline results

This subsection is divided into ML and zero-shot results

5.1.1 ML models

As a baseline, we performed TFIDF and SBERT with simple ML models. Here are the results for baseline models.

Table 6: TF-IDF F1 scores with machine learning models

Models	2quads	4quads	8quads	16quads	22quads	28quads
SVM	0.70±0.03	0.76±0.03	0.83±0.07	0.87±0.02	**0.89**±0.01	**0.89**±0.00
SGD	0.69±0.03	0.71±0.05	0.81±0.08	0.84±0.05	0.84±0.01	0.86±0.02
LR	0.70±0.03	0.76±0.03	0.83±0.08	0.86±0.02	0.88±0.01	**0.89**±0.01
DT	0.46±0.13	0.55±0.02	0.63±0.07	0.72±0.07	0.76±0.05	0.77±0.05
NB	0.68±0.03	0.73±0.12	0.79±0.13	0.82±0.06	0.83±0.03	0.85±0.03
RF	0.54±0.07	0.54±0.08	0.65±0.06	0.75±0.06	0.80±0.04	0.83±0.03

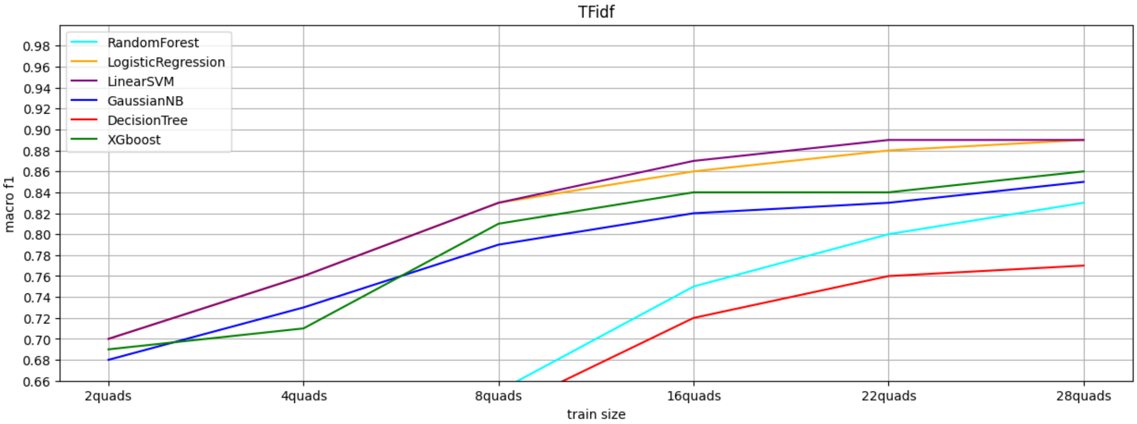


Figure 6: Visualisation of TF-IDF F1 scores by train data diversity with ML models

Table 7: Bert embedding F1 scores with machine learning models

Models	2quads	4quads	8quads	16quads	22quads	28quads
NB	0.58±0.21	0.74±0.12	0.87±0.03	0.91±0.03	0.90±0.03	0.90±0.03
RF	0.71±0.08	0.79±0.07	0.85±0.04	0.89±0.03	0.90±0.03	0.91±0.02
SVM	0.81±0.07	0.81±0.08	0.88±0.03	0.91±0.02	**0.93**±0.02	**0.93**±0.01
LR	0.79±0.05	0.81±0.08	0.88±0.03	0.91±0.02	**0.92**±0.02	**0.93**±0.02
DT	0.66±0.10	0.64±0.15	0.73±0.07	0.73±0.05	0.79±0.07	0.79±0.05
SGD	0.81±0.04	0.83±0.05	0.85±0.04	0.89±0.03	0.91±0.01	**0.92**±0.02

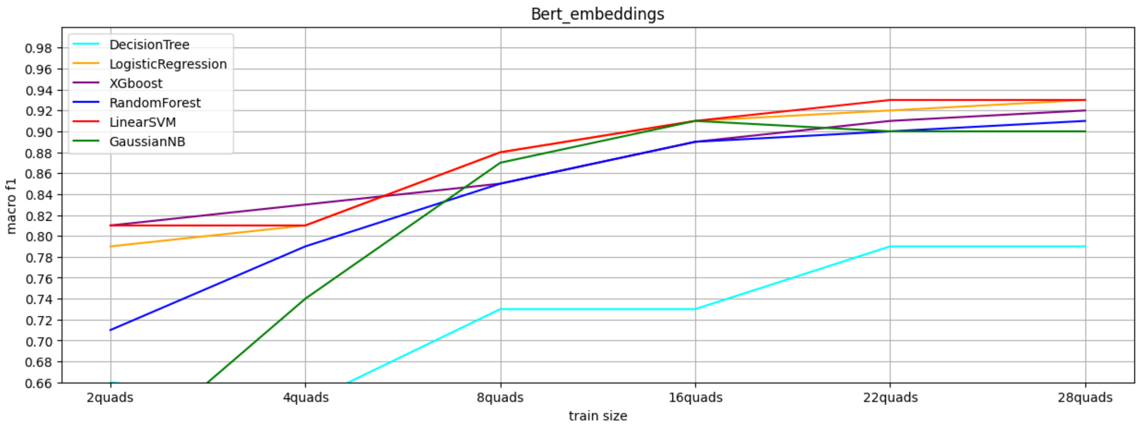


Figure 7: Visualisation of SBERT embedding F1 scores by train data diversity with ML models

Here are the best results followed by a discussion:

- TFIDF6
  - SVM scored 0.89±0
  - LR scored 0.89±0.01
- SBERT7
  - SVM scored 0.93±0.01
  - LR scored 0.93±0.02
  - SGD scored 0.92±0.02

Surprisingly, Classic ML algorithms performed quite well on these tests. There is no way that TFIDF can understand the concept of exclusivity. There are a few features in our dataset that could be enabling these impressive results, most notable is sentence length.

Table 8: Average char length per class

	Inclusive	**Exclusive**
Avg chars	115.11	52.48

Table 9: Outliers of the char count average

	Num Sentences
Inc char count less than exc char count average	8
Exc char count greater than inc char count average	0

This would obviously help traditional ML algorithms classify based on this misleading feature. If an algorithm simply considered sentence length the predictions would have good scores. The dataset contains this misleading feature due to the nature of exclusivity, where comments are rash and inconsiderate. You can generalize a race in a few words, but it takes more explanation to articulate a commonality between people of the same race in a socially acceptable manner.

Also, the standard deviation of the ML scores is very low. It is hard to explain how there can be 0 deviation over the three folds in the SVM with TFIDF. Either, there is not any noise in the dataset or there is an extremely dominant and reliable feature. Most likely it is the latter, and a feature is allowing ML algorithms to predict well. We will not further explore the reasons for well performing ML because the few-shot models perform much better anyways.
How do we overcome imperfect datasets?(In our case and in broader application) A dataset would need to be created where both classes have equal char length to remove this feature. We could apply adversarial machine learning to our classification problem given the reliable ability to generate objectively exclusive and inclusive adversarial examples. I hypothesize sentence generation is quite difficult because humans have trouble creating these sentences, combined with, as acceptable sentences are generated they will become more alike each other as the sentence generation model converges. Due to the likely difficulties of adversarial learning, I suppose it is best to use zero-shot or few-shot learning to avoid the pre-trained model learning to classify based on features which are only available due to an imperfect dataset. Furthermore, if there are unavoidable shortcomings in every dataset, I suspect the best solution is an extremely robust general language model that does not need training or only needs a handful of training examples.
Another plausible solution is explainable artificial intelligence. A model which gives the rationale for each prediction would allow humans to determine if the AI is misaligned. Contrarily, we must acknowledge the possibility that an AI might learn to give rationale that humans approve of whilst making predictions for different reasons. Humans also give false justifications for their actions to maintain the approval of others and ensure their true motives are not exposed. How can we expect to create AI better than ourselves?

5.1.2 Zero-shot

Despite zero-shot being the worst of all approaches it still predicts with %82 F1. %82 does not suffice for deployment, however, %82 F1 proves that pre-trained models understand exclusivity. The confusion matrix is skewed towards inclusive prediction which is expected due to the neutral sentiment of our exclusive examples. This zero-shot model is the french_xlm_xmli[34]. It is based on the XLM-Roberta-base, which was trained on multilingual corpus of twitter data, and finetuned on the french subset of the Cross-lingual Natural Language Inference dataset[35]. This model is fine-tuned for zero-shot classification and performs very well in the few-shot setting. Similarly to humans, a few examples can significantly enhance comprehension when learning a new task.

Our experiments only briefly explore zeroshot classification, but the considerable benefits of zeroshot must not be overlooked. First benefit, with zeroshot classification there is no training to learn false indicators. We do not need to expend resources

[34] https://huggingface.co/morit/french_xlm_xnli

[35] https://paperswithcode.com/dataset/xnli

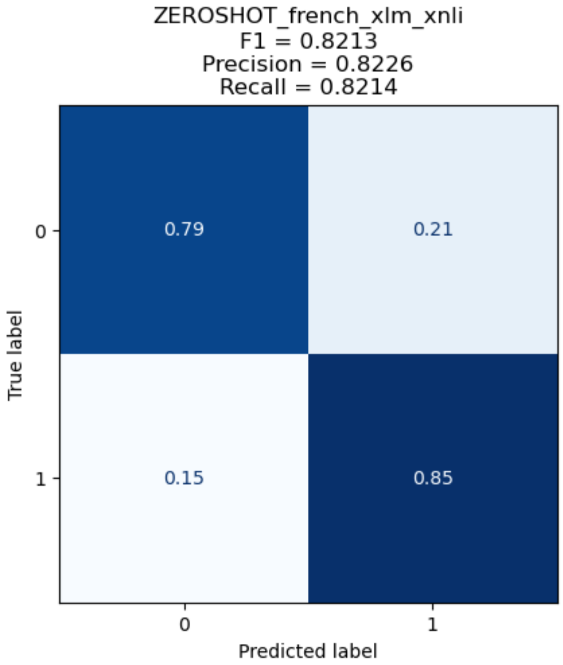


Figure 8: Confusion matrix for zero-shot french_xlm_xnli model

constructing a faultless dataset and can simply focus on developing a robust general language model. The focus on developing an overall comprehensive NLP model will progress research much quicker rather than developing individual models which are used for a niche usecase. Furthermore, with powerful generalized models we can easily handle a plethora of NLP tasks. I suspect practical applications of NLP using multi-task learning, an approach where one model is trained to solve multiple tasks by leveraging commonalities of the domain. For example, an AI implemented on the Twitter site could classify tweets for reccomendation/searching, detecting falsified information, detecting exclusive speech and more all using one model. This paper from Intel is an outstanding read if you are also interested in Multi-objective Learning.(22)

5.1.3 Comparison of baselines

The baseline tests we did encompass ML algorithms and zero-shot learning. Zero-shot scored worse, but we can trust the predictions were not made from falsely drawn conclusions about the dataset. The ML baselines tests have exemplified the need for perfectly crafted datasets. My focus was completely on ensuring the exclusivity was unbiased and objective, consequently, I was not attentive enough to other possible shortcomings of the dataset. The zero-shot baseline is a valuable benchmark to ensure the performance is improved in the few-shot setting and the ML algorithms, scoring %93 F1 score, dictate few-shot models need to score better.

5.2 Few-Shot results

5.2.1 Few-shot F1 by train data diversity

Table 10: F1 scores for few-shot by train data diversity

Models	2quads	4quads	8quads	16quads	22quads	28quads
BBMCSM	0.83±0.06	0.87±0.07	0.88±0.09	0.94±0.03	0.94±0.04	**0.95**±0.03
dehatebert-mono-french	0.73±0.13	0.82±0.10	0.87±0.03	0.89±0.02	0.91±0.01	0.92±0.00
french_xlm_xnli	0.92±0.04	0.94±0.03	0.95±0.01	**0.97**±0.00	**0.97**±0.01	**0.97**±0.00
XRBSM	0.82±0.06	0.88±0.03	0.88±0.05	0.90±0.01	0.92±0.00	0.92±0.02

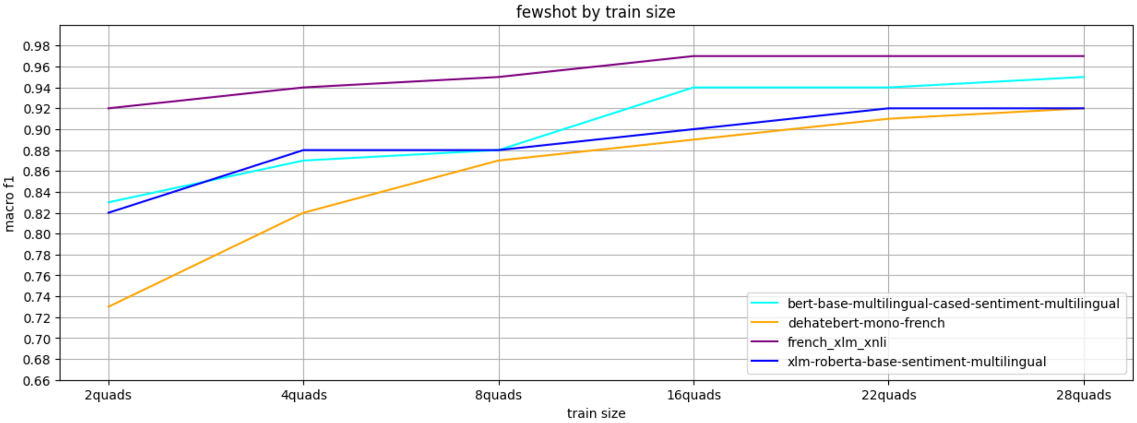


Figure 9: Visualisation of few-shot macro F1 scores by train data diversity

5.2.2 Few-shot best F1 per epoch

Training over epochs taking the best macro F1 score from any train data diversity:

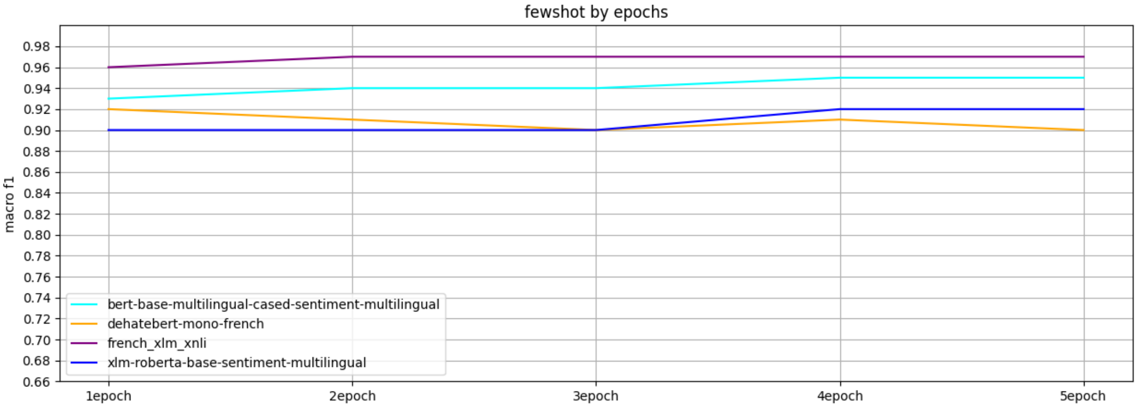


Figure 10: Visualisation of few-shot macro F1 scores by epochs

5.2.3 Best confusion matrices from each tested few-shot model

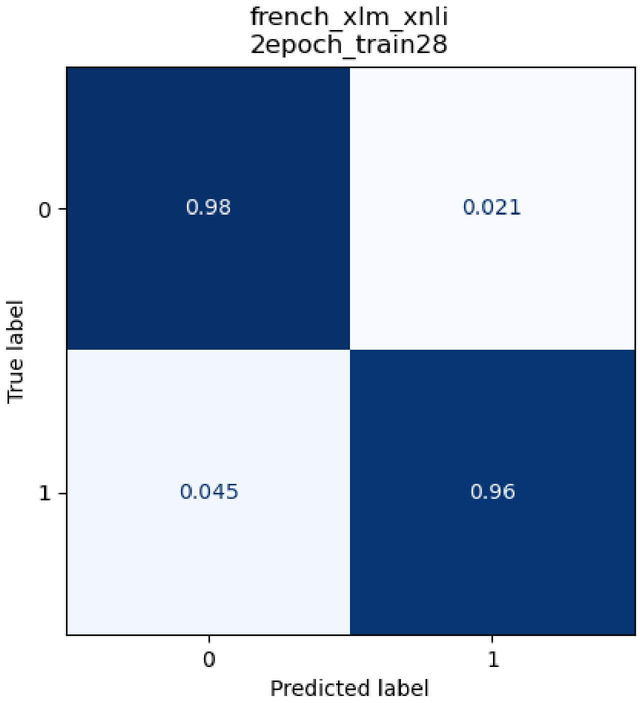


Figure 11: french_xlm_xnli 2epoch 28quads confusion matrix

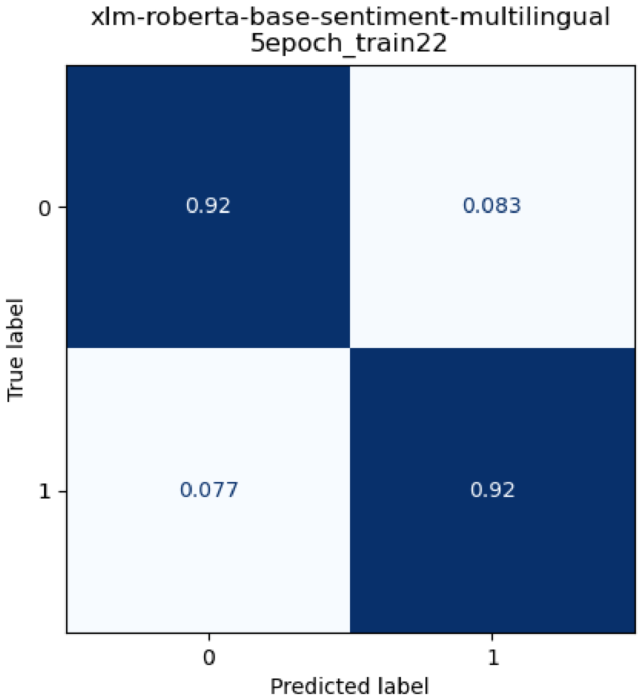


Figure 12: xlm-roberta-base-sentiment-multilingual 5epoch 22quads confusion matrix

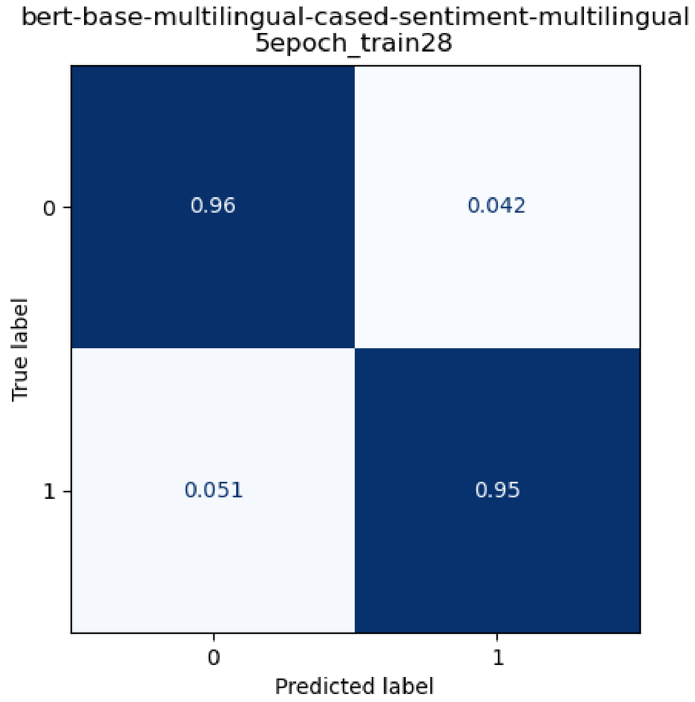


Figure 13: bert-base-multilingual-cased-sentiment-multilingual 5epoch 28quads confusion matrix

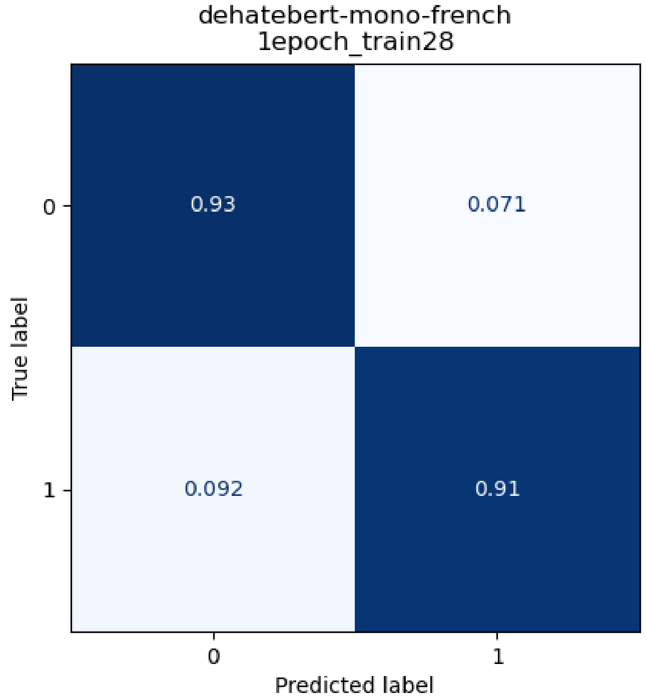


Figure 14: dehatebert-mono-french 1epoch 28quads confusion matrix

5.2.4 Here are the best few-shot results followed by a discussion

- BBMCSM (bert-base-multilingual-cased-sentiment-multilingual)

  - 28quads 0.95±0.03 macroF1

- french_xlm_xnli
  - 16quads 0.97±0.00 macroF1
  - 22quads 0.97±0.01 macroF1
  - 28quads 0.97±0.00 macroF1

• french_xlm_xnli

Few-shot exceeded expectations with quite high scores, especially the french_xlm_xnli model. French_xlm_xnli is designed for zero-shot, making the impressive results unsurprising. Interesting how the difference from the zero-shot setting to just 8 training data, 2quads, was a %10 F1 score improvement with french_xlm_xnli model. Keep in mind, the train size is always 320 regardless of the number of quads. As an example, 2quads creates 320 train data triplets and 28quads also creates 320 train data triplets(3.5.3). French_xlm_xnli model converges with 16quads, 64 train data, and does not improve at all with increased training data. No other model converged in this manner. However, BBMCSM(bert-base-multilingual-cased-sentiment-multilingual) reached %95 macro F1 and probably would have continued increasing if we ran more epochs.
It is shocking how just 8 sentence examples give a %10 increase in macro F1 score when simply appending a CNN classification head. The classification head only sees a binary classification problem which is completely abstracted from the meaning of inclusive and exclusive, furthermore, the models' understanding of inclusive and exclusive is lost; This consequently makes worse generalizations. There is tension between conciseness and generalized knowledge for every AI depending on the purpose. In many AI usecases, I would argue generalized knowledge is much more valuable and will enable higher-quality results in performing niche tasks. Regardless of my opinions, more robust models are being created in pursuit of AGI and XAI; I agree this is the most direct route to improvement.

• BBMCSM and XRBSM

It is difficult to definitively explain why the BBMCSM(bert-base-multilingual-cased-sentiment-multilingual) performs better than XRBSM(xlm-roberta-base-sentiment-multilingual) espically given that french_xlm_xnli performs the best and is based off the XRBSM. Also both BBMCSM and XRBSM models were trained on the cardiffnlp/tweet_sentiment_multilingual[36] dataset. It is reasonable to dissect the differences between BERT and RoBERTa in search of explination for the performance difference. The differences are clearly explained in the RoBERTa paper:

"BERT relies on randomly masking and predicting tokens. The original BERT implementation performed masking once during data preprocessing, resulting in a single static mask. To avoid using the same mask for each training instance in every epoch, training data was duplicated 10 times so that each sequence is masked in 10 different ways over the 40 epochs of training. Thus, each training sequence was seen with the same mask four times during training. We compare this strategy with dynamic masking where we generate the masking pattern every time we feed a sequence to the model. This becomes crucial when pretraining for more steps or

[36] https://huggingface.co/datasets/cardiffnlp/tweet_sentiment_multilingual

with larger datasets.(Liu et al.)”

In other words, RoBERTa copies the train data ten times and applies a different random mask to each copy of train data making training more robust than BERT. The differences in training masks of the base models are unlikely to be the reason for the performance difference and if it was the cause of the performance difference we cannot prove it because are few-shot models are quite abstracted from the original model. Not only are the base models fine-tuned, we also have a two-layer classification head appended to the model.
Disregarding the masking differences between BERT and roBERTa, the variance in F1 score is most likely to be that the BBMCSM model is cased and the XRBSM model is uncased. There is a conceivable gain for a classification problem regarding race and culture, which are proper nouns, to be case sensitive. This cannot be proven from outside the black box of a DNN, but it is the most probable cause of the discrepancy in F1 score between BBMCSM and XRBSM models. These experiments could be repeated with an entirely lowercase dataset to test this hypothesis.

- dehatebert-mono-french

The final model to discuss is dehatebert-mono-french. This model was created for hatespeech detection similar to french_xlm_xnli. Dehatebert-mono-french is a fine-tuned model of bert-base-multilingual. ”The mono in the name refers to the monolingual setting, where the model is trained using only English language data.”[37] The model card on Huggingface must have a typo and this is meant to say *using only French language*. Regardless, all multilingual models which are fine-tuned for a certain language maintain the ability to interpret other languages. This is evident in our top model still being able to classify English at a high accuracy despite being fine-tuned on solely French. Origionally, I expected this model to perform worse because it is fine-tuned for HSD(hate speech detection), but our best performing model is also a HSD model. Dehatebert-mono-french was only fine-tuned on 1220 French tweets.(2) As a result, it is much closer to a bert-base-multilingual model than the other fine-tuned models implemented in these tests. I attribute the poor performance in the few-shot setting to insufficient fine-tuning. Also, this model shows the greatest improvement from increased training data size, which is a byproduct of the original model insufficiency. The poor performance of Dehatebert-mono-french is a result of insufficient fine-tuning data. Of course, we could run it for additional epochs, but overfitting an exclusivity classifier is dangerous and may give preference to certain races and cultures. It is wiser to find a more robust general model which performs better in the few-shot setting.

[37] https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-french

6 Limitations and ethical concerns

6.1 Should a machine say, ”no”?

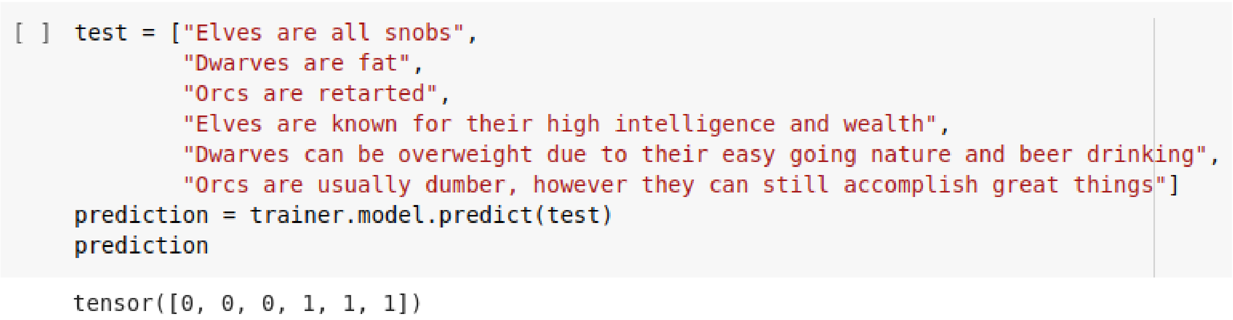
Automatic censorship is a huge concern to free speech. Why should a physical system ever have the power to tell a human their speech is negative or inappropriate? If a person tells you no, you can always ask for an explanation, adversarially, programmed machines cannot have their opinion swayed. This is why these experiments do not have the intention to create any internet censorship, rather, the goal is to offer rather aid to those who desire to improve their speech to be inclusive of all people. I have given the benefit of the doubt that exclusive sentences are not typically ill-intentioned, hence the neutral sentiment dataset. Kind people will benefit from automatic suggestions to improve their accidentally exclusive sentences. Of course, if people are intentionally hateful they will give no heed to the suggestions and continue to outlet their aggression.

6.2 Will this temporal research soon be irrelevant because exclusivity is constantly changing?

A possible limitation of our exclusivity classifier is how exclusive speech changes over time. A statement which is exclusive now is not always exclusive in a few years depending on the subject of the sentence. These concerns are valid, but only an issue if our models have not generalized exclusivity well. A lighthearted example, ”All iPhone users are stupid” is a obviously an incorrect generalization of iPhone users. The same example with a different subject becomes extremely offensive, ”All African Americans are stupid.” Both of these examples are exclusive because exclusive speech does not hinge on factual accuracy at all, rather exclusivity is essentially untrue making it unjust. I suspect an automated exclusivity system would recommend a modification to both sentences, despite the iPhone one being non-harmful. It is okay to classify any derogatory overgeneralizing sentence as exclusive because the purpose is to offer a recommended change which can be dismissed.

6.3 How can a small dataset be enough to understand global exclusivity?

Another concern is the size of the dataset. How could 336 sentences possibly be enough to have an AI understand global exclusivity? I would argue that our dataset is ample to have a NLP model understand that overgeneralizing a subset of the global population is exclusive. I acknowledge that, living in the western world, I cannot give a relevant example of exclusivity of every global race and culture. The best way to defend the model is by results, view these classifications on fantasy races:



```
test = ["Elves are all snobs",
        "Dwarves are fat",
        "Orcs are retarted",
        "Elves are known for their high intelligence and wealth",
        "Dwarves can be overweight due to their easy going nature and beer drinking",
        "Orcs are usually dumber, however they can still accomplish great things"]
prediction = trainer.model.predict(test)
prediction
```

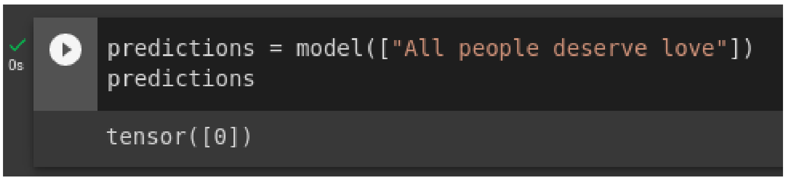
```
tensor([0, 0, 0, 1, 1, 1])
```

Figure 15: Inclusive and Exclusive examples on fantasy races

If you would like to test more sentences view my Huggingface here[38], or run this collab notebook here[39].

6.4 Shortcomings of the classifiers

As mentioned earlier(8), the difference in average character between the classes leads to a reliable feature for classification. Due to this flaw in the dataset, the models perform well on our dataset but struggle to classify very short sentences as inclusive. Additionally, there is no *hope speech* in the dataset. ”Hope speech is any message or content that is positive, encouraging, reassuring, inclusive and supportive that inspires and engenders optimism in the minds of people(8)”. Because of these two reasons, short sentences that generalize positively are always classified as exclusive. See ”All people deserve love” classified by both zero-shot and few-shot french_xml_xnli:



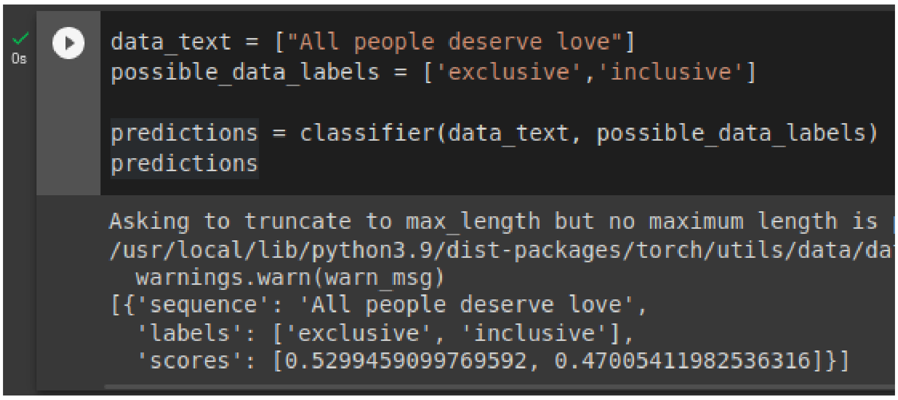
```
predictions = model(["All people deserve love"])
predictions
```

```
tensor([0])
```

Figure 16: ”All people deserve love” classified by best performing few-shot model, french_xml_xnli

[38] https://huggingface.co/BeToast/xml_xnli__inclusiveORexclusive__binary_classification__frenchANDenglish

[39] https://colab.research.google.com/drive/1EABmyXsjQihRS1W9If-weg-lR8i4xn_4?usp=sharing



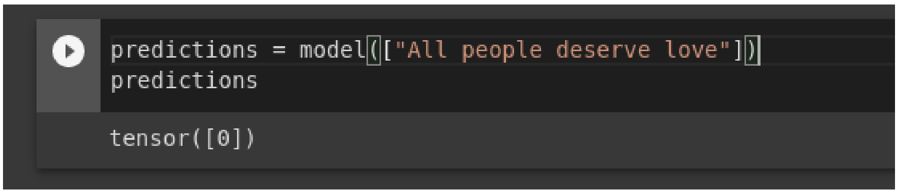
```
data_text = ["All people deserve love"]
possible_data_labels = ['exclusive','inclusive']

predictions = classifier(data_text, possible_data_labels)
predictions
```

```
Asking to truncate to max_length but no maximum length is
/usr/local/lib/python3.9/dist-packages/torch/utils/data/da
  warnings.warn(warn_msg)
[{'sequence': 'All people deserve love',
  'labels': ['exclusive', 'inclusive'],
  'scores': [0.5299459099769592, 0.47005411982536316]}]
```

Figure 17: "All people deserve love" classified by zero-shot, french_xml_xnli

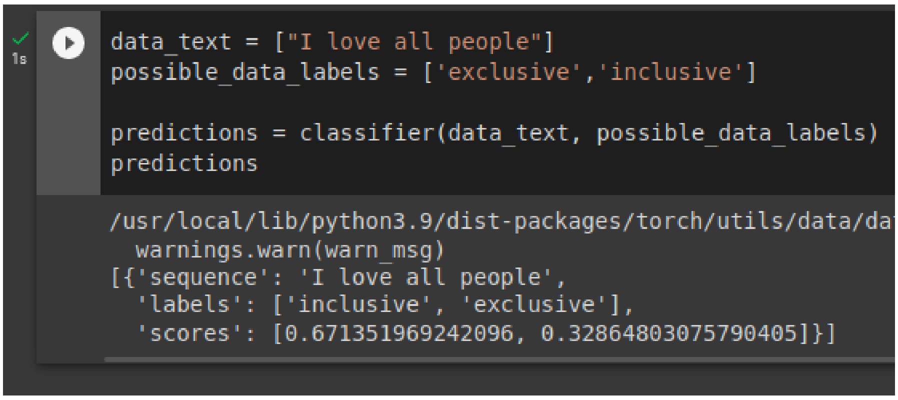
The few-shot model classifies this sentence as exclusive despite it being objectively positive and hopeful speech.
Note how the model in zero-shot context also leaned more towards exclusivity.
It seems like an over-generalization is too heavily correlated with exclusivity in the french_xml_xnli model.
See "I love all people" classified by both zero-shot and few-shot french_xml_xnli:



```
predictions = model(["All people deserve love"])
predictions
```

```
tensor([0])
```

Figure 18: "I love all people" classified by best performing few-shot model, french_xml_xnli



```
data_text = ["I love all people"]
possible_data_labels = ['exclusive','inclusive']

predictions = classifier(data_text, possible_data_labels)
predictions
```

```
/usr/local/lib/python3.9/dist-packages/torch/utils/data/da
  warnings.warn(warn_msg)
[{'sequence': 'I love all people',
  'labels': ['inclusive', 'exclusive'],
  'scores': [0.671351969242096, 0.32864803075790405]}]
```

Figure 19: "I love all people" classified by zero-shot, french_xml_xnli

Our few-shot model still fails to classify "I love all people" as inclusive. This is most likely due to the short sentence length.
The same model in zero-shot is able to classify "I love all people" as inclusive when it failed on "All people deserve love". The key difference here is one is generalizing a large population and the other is not.
These models should not be trusted for their classifications and must be retrained with an expanded dataset for real-world application.

6.5 Can this classifier be expanded to more languages?

As of now, multilingual BERT is trained on 104 languages and multilingual roBERTa is trained on 100. Theoretically, we could create a classifier with these multilingual models that can understand all languages. The contrastive learning triplets would be able to cluster the different language sentence embeddings as inclusive or exclusive regardless of variance of encodings across different languages.
The original English sentences were simply translated to French and proofread. This approach could be done with all 100 BERT languages easily. Given we are reusing the same translated sentence so many times, overfitting could be possible at very low epochs. I think this would be pleasurable research to do in the future.

7 Conclusions

In conclusion, our baseline models showed room for improvement so few-shot classifiers were developed. Four pre-trained models(3.5.2) were chosen to be trained and tested using 3-fold validation on a binary classification problem with a custom dataset. These pre-trained models were an even balance of BERT and roBERTa, trained for Sentiment Analysis or Hatespeech Detection. The dataset(1) consisted of equal parts French and English and was completely balanced between exclusive and inclusive classes. A convolutional neural network classification head was appended to the chosen pre-trained models and were trained on contrastive learning triplets were created using SetFit(26) Impressive results were achieved, most notably french_xlm_xnli model with %97 macro F1 score in the few-shot setting. This model reliably generalizes exclusivity well and does not show racial bias, proven by testing on fictitious fantasy races. However, due to a high character average of inclusive train data, the model fails to classify very short statements as inclusive regardless of their content. Further research must be undertaken to develop a distributable exclusivity detection model for implementation on the internet.

8 Future Work

The original intention of this work was preliminary for an AI that can detect exclusive social media posts and then suggest modifications to the post, so the user can easily amend the exclusive post to be inclusive. The dataset is structured in pairs, each exclusive sentence has a corresponding inclusive improvement, to also be used for training the generative model necessary to accomplish this larger goal. I desire to continue research and create a generative improvement model to help people amend unintentionally exclusive sentences. However, the current best exclusivity detection model is unsatisfactory and I will not put the cart before the horse. In the fall, I intend to continue research on this project and in the broader field of NLP in postgraduate research at the University of Galway.

9 Appendices

Extra goodies!

9.1 Few-shot : accuracy, precision, recall, f1

9.1.1 french_xlm_xnli

A table for accuracy, precision, recall, and f1 scores from french_xlm_xnli model

Table 11: french_xlm_xnli **accuracy**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.89±0.03	0.92±0.04	0.91±0.05	0.95±0.0	0.96±0.01	0.96±0.01
2epoch	0.9±0.04	0.94±0.03	0.93±0.02	0.96±0.0	0.97±0.01	0.97±0.0
3epoch	0.9±0.04	0.93±0.03	0.95±0.01	0.96±0.01	0.97±0.0	0.97±0.01
4epoch	0.91±0.04	0.94±0.04	0.94±0.01	0.97±0.0	0.97±0.01	0.96±0.0
5epoch	0.92±0.04	0.93±0.03	0.94±0.01	0.97±0.0	0.97±0.0	0.96±0.0

Table 12: french_xlm_xnli **precision**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.89±0.03	0.92±0.04	0.91±0.04	0.95±0.0	0.96±0.0	0.96±0.01
2epoch	0.9±0.04	0.94±0.03	0.93±0.02	0.96±0.0	0.97±0.01	0.97±0.0
3epoch	0.91±0.05	0.94±0.03	0.95±0.01	0.96±0.01	0.97±0.0	0.97±0.01
4epoch	0.91±0.05	0.94±0.03	0.95±0.01	0.97±0.0	0.97±0.01	0.96±0.0
5epoch	0.92±0.04	0.94±0.03	0.95±0.01	0.97±0.0	0.97±0.0	0.96±0.0

Table 13: french_xlm_xnli **recall**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.89±0.03	0.92±0.04	0.91±0.05	0.95±0.0	0.96±0.01	0.96±0.01
2epoch	0.9±0.04	0.94±0.03	0.93±0.02	0.96±0.0	0.97±0.01	0.97±0.0
3epoch	0.9±0.04	0.93±0.03	0.95±0.01	0.96±0.01	0.97±0.0	0.97±0.01
4epoch	0.91±0.04	0.94±0.04	0.94±0.01	0.97±0.0	0.97±0.01	0.96±0.0
5epoch	0.92±0.04	0.93±0.03	0.94±0.01	0.97±0.0	0.97±0.0	0.96±0.0

Table 14: french_xlm_xnli **f1**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.88±0.03	0.92±0.04	0.91±0.05	0.95±0.0	0.96±0.01	0.96±0.01
2epoch	0.9±0.04	0.94±0.03	0.93±0.02	0.96±0.0	0.97±0.01	0.97±0.0
3epoch	0.9±0.04	0.93±0.03	0.95±0.01	0.96±0.01	0.97±0.0	0.97±0.01
4epoch	0.91±0.04	0.94±0.04	0.94±0.01	0.97±0.0	0.97±0.01	0.96±0.0
5epoch	0.92±0.04	0.93±0.03	0.94±0.01	0.97±0.0	0.97±0.0	0.96±0.0

9.1.2 xlm-roberta-base-sentiment-multilingual

A table for accuracy, precision, recall, and f1 scores from xlm-roberta-base-sentiment-multilingual model

Table 15: xlm-roberta-base-sentiment-multilingual **accuracy**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.74±0.09	0.85±0.01	0.84±0.03	0.86±0.03	0.86±0.01	0.9±0.04
2epoch	0.78±0.1	0.85±0.02	0.86±0.05	0.89±0.01	0.9±0.01	0.9±0.01
3epoch	0.79±0.09	0.88±0.03	0.88±0.04	0.89±0.01	0.9±0.02	0.9±0.01
4epoch	0.81±0.07	0.88±0.03	0.87±0.05	0.89±0.0	0.92±0.0	0.92±0.02
5epoch	0.83±0.05	0.87±0.04	0.88±0.05	0.9±0.01	0.92±0.01	0.92±0.04

Table 16: xlm-roberta-base-sentiment-multilingual **precision**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.8±0.05	0.85±0.01	0.85±0.02	0.87±0.03	0.87±0.01	0.9±0.04
2epoch	0.82±0.04	0.86±0.01	0.88±0.03	0.89±0.01	0.91±0.01	0.9±0.01
3epoch	0.84±0.04	0.88±0.03	0.89±0.03	0.89±0.01	0.91±0.02	0.91±0.01
4epoch	0.83±0.04	0.89±0.02	0.88±0.04	0.89±0.0	0.92±0.0	0.92±0.02
5epoch	0.85±0.03	0.88±0.03	0.89±0.04	0.9±0.01	0.92±0.01	0.92±0.04

Table 17: xlm-roberta-base-sentiment-multilingual **recall**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.74±0.09	0.85±0.01	0.84±0.03	0.86±0.03	0.86±0.01	0.9±0.04
2epoch	0.78±0.1	0.85±0.02	0.86±0.05	0.89±0.01	0.9±0.01	0.9±0.01
3epoch	0.79±0.09	0.88±0.03	0.88±0.04	0.89±0.01	0.9±0.02	0.9±0.01
4epoch	0.81±0.07	0.88±0.03	0.87±0.05	0.89±0.0	0.92±0.0	0.92±0.02
5epoch	0.83±0.05	0.87±0.04	0.88±0.05	0.9±0.01	0.92±0.01	0.92±0.04

Table 18: xlm-roberta-base-sentiment-multilingual **f1**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.73±0.11	0.85±0.01	0.84±0.04	0.86±0.03	0.86±0.01	0.9±0.04
2epoch	0.76±0.12	0.85±0.02	0.86±0.05	0.89±0.01	0.9±0.01	0.9±0.01
3epoch	0.78±0.11	0.88±0.03	0.88±0.05	0.89±0.01	0.9±0.02	0.9±0.01
4epoch	0.8±0.08	0.88±0.03	0.87±0.05	0.89±0.0	0.92±0.0	0.92±0.02
5epoch	0.82±0.06	0.87±0.04	0.88±0.06	0.9±0.01	0.92±0.01	0.92±0.04

9.1.3 bert-base-multilingual-cased-sentiment-multilingual

A table for accuracy, precision, recall, and f1 scores from bert-base-multilingual-cased-sentiment-multilingual model

Table 19: bert-base-multilingual-cased-sentiment-multilingual **accuracy**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.81±0.06	0.85±0.07	0.86±0.07	0.92±0.06	0.93±0.02	0.92±0.02
2epoch	0.81±0.04	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.04	0.92±0.02
3epoch	0.83±0.06	0.87±0.07	0.87±0.08	0.94±0.04	0.94±0.03	0.94±0.03
4epoch	0.83±0.05	0.87±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.03
5epoch	0.84±0.06	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.02

Table 20: bert-base-multilingual-cased-sentiment-multilingual **precision**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.84±0.03	0.87±0.04	0.86±0.07	0.92±0.06	0.93±0.02	0.92±0.02
2epoch	0.86±0.03	0.88±0.04	0.88±0.08	0.94±0.03	0.94±0.04	0.92±0.02
3epoch	0.86±0.04	0.88±0.05	0.88±0.07	0.94±0.04	0.94±0.03	0.94±0.03
4epoch	0.86±0.04	0.88±0.05	0.89±0.08	0.94±0.03	0.94±0.03	0.95±0.03
5epoch	0.86±0.05	0.88±0.05	0.89±0.08	0.95±0.03	0.94±0.03	0.95±0.02

Table 21: bert-base-multilingual-cased-sentiment-multilingual **recall**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.81±0.06	0.85±0.07	0.86±0.07	0.92±0.06	0.93±0.02	0.92±0.02
2epoch	0.81±0.04	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.04	0.92±0.02
3epoch	0.83±0.06	0.87±0.07	0.87±0.08	0.94±0.04	0.94±0.03	0.94±0.03
4epoch	0.83±0.05	0.87±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.03
5epoch	0.84±0.06	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.02

Table 22: bert-base-multilingual-cased-sentiment-multilingual **f1**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.81±0.07	0.85±0.07	0.86±0.08	0.92±0.06	0.93±0.02	0.92±0.02
2epoch	0.8±0.05	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.04	0.92±0.02
3epoch	0.82±0.07	0.87±0.07	0.87±0.08	0.94±0.04	0.94±0.03	0.94±0.03
4epoch	0.83±0.06	0.87±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.03
5epoch	0.83±0.06	0.86±0.07	0.88±0.09	0.94±0.03	0.94±0.03	0.95±0.02

9.1.4 dehatebert-mono-french

A table for accuracy, precision, recall, and f1 scores from dehatebert-mono-french model

Table 23: dehatebert-mono-french **accuracy**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.71±0.08	0.81±0.04	0.86±0.05	0.89±0.02	0.91±0.01	0.92±0.0
2epoch	0.71±0.11	0.82±0.1	0.87±0.03	0.88±0.01	0.9±0.02	0.91±0.03
3epoch	0.74±0.11	0.79±0.07	0.87±0.02	0.88±0.01	0.88±0.02	0.9±0.03
4epoch	0.74±0.11	0.81±0.06	0.85±0.03	0.87±0.01	0.88±0.03	0.91±0.02
5epoch	0.75±0.1	0.81±0.06	0.85±0.03	0.89±0.02	0.89±0.02	0.9±0.01

Table 24: dehatebert-mono-french **precision**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.77±0.02	0.82±0.06	0.87±0.05	0.89±0.02	0.91±0.01	0.92±0.0
2epoch	0.78±0.0	0.83±0.1	0.87±0.03	0.88±0.01	0.9±0.02	0.91±0.03
3epoch	0.8±0.02	0.81±0.06	0.87±0.02	0.88±0.01	0.89±0.02	0.91±0.03
4epoch	0.8±0.01	0.82±0.06	0.86±0.03	0.89±0.01	0.89±0.03	0.91±0.02
5epoch	0.8±0.01	0.82±0.06	0.86±0.03	0.9±0.02	0.89±0.02	0.9±0.01

Table 25: dehatebert-mono-french **recall**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.71±0.08	0.81±0.04	0.86±0.05	0.89±0.02	0.91±0.01	0.92±0.0
2epoch	0.71±0.11	0.82±0.1	0.87±0.03	0.88±0.01	0.9±0.02	0.91±0.03
3epoch	0.74±0.11	0.79±0.07	0.87±0.02	0.88±0.01	0.88±0.02	0.9±0.03
4epoch	0.74±0.11	0.81±0.06	0.85±0.03	0.87±0.01	0.88±0.03	0.91±0.02
5epoch	0.75±0.1	0.81±0.06	0.85±0.03	0.89±0.02	0.89±0.02	0.9±0.01

Table 26: dehatebert-mono-french **f1**

	2quads	4quads	8quads	16quads	22quads	28quads
1epoch	0.69±0.11	0.8±0.04	0.86±0.05	0.89±0.02	0.91±0.01	0.92±0.0
2epoch	0.68±0.15	0.82±0.1	0.87±0.03	0.88±0.01	0.9±0.02	0.91±0.03
3epoch	0.72±0.15	0.78±0.08	0.87±0.02	0.87±0.01	0.88±0.02	0.9±0.03
4epoch	0.72±0.14	0.81±0.06	0.85±0.03	0.87±0.01	0.88±0.03	0.91±0.02
5epoch	0.73±0.13	0.81±0.06	0.85±0.03	0.89±0.02	0.89±0.02	0.9±0.01

9.2 Few-shot : confusion matricies

In this appendix, I have only shown the best performing train size for each epoch. You can view every confusion matrix for all experiments here[40]

9.2.1 french_xlm_xnli

Best performing confusion matrices from each epoch of french_xlm_xnli model

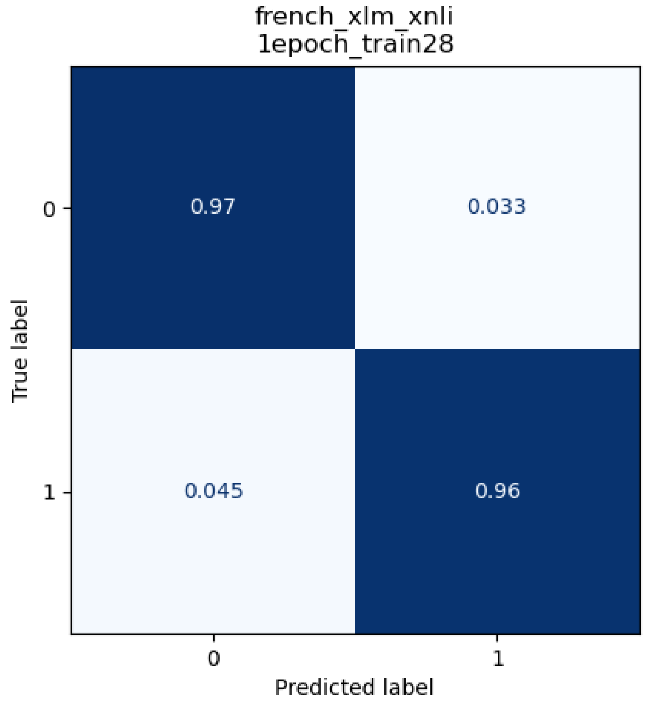


Figure 20: french_xlm_xnli 1epoch 28quads

[40] `https://github.com/BeToast/Racially-Exclusive-Speech-Detection/tree/main/final/confusion_matricies`

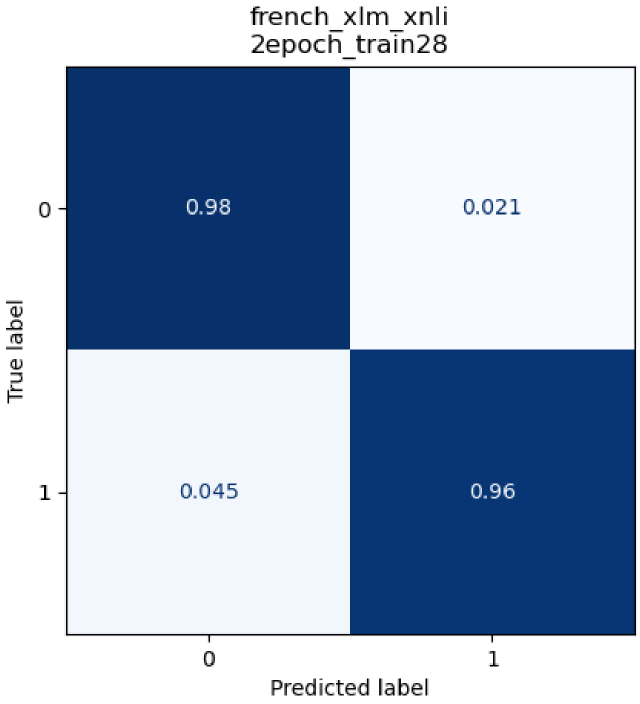


Figure 21: french_xlm_xnli 2epoch 28quads

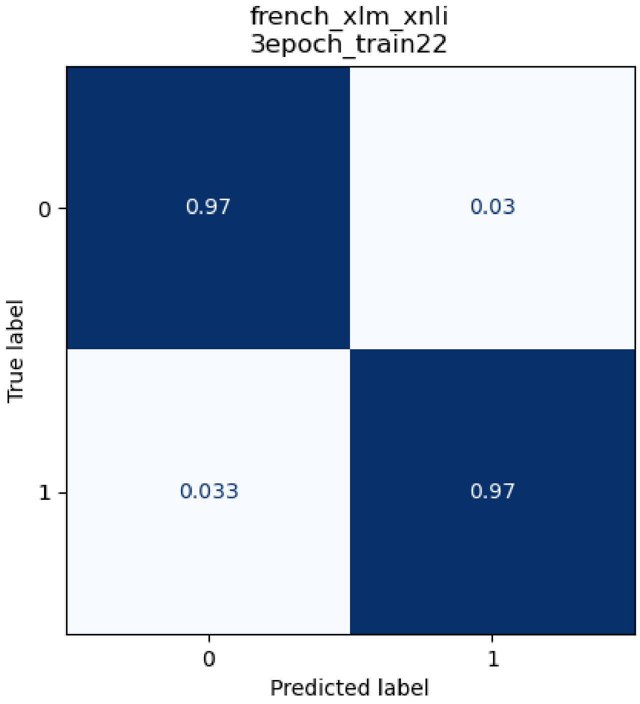


Figure 22: french_xlm_xnli 3epoch 22quads

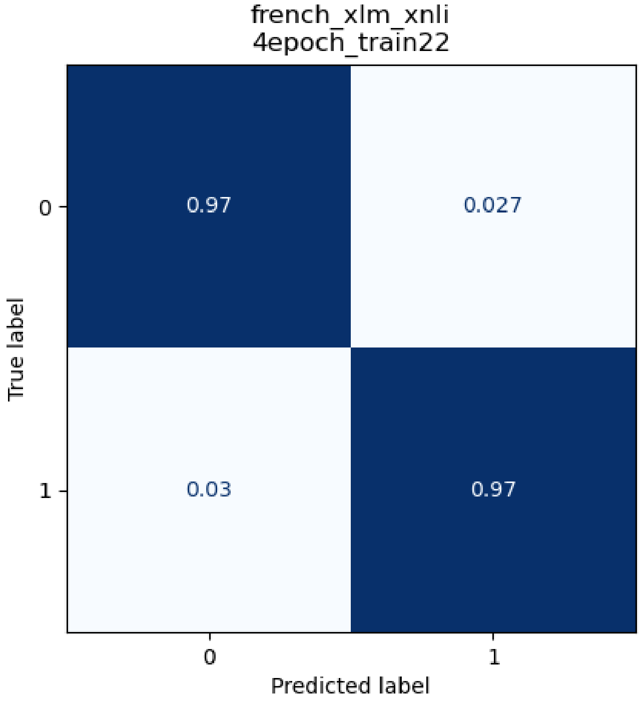


Figure 23: french_xlm_xnli 4epoch 22quads

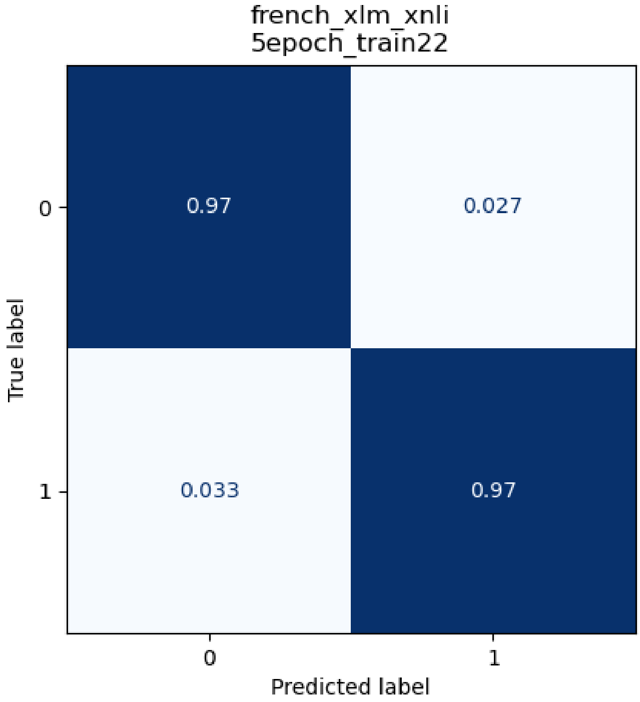


Figure 24: french_xlm_xnli 5epoch 22quads

9.2.2 xlm-roberta-base-sentiment-multilingual

Best performing confusion matrices from each epoch of xlm-roberta-base-sentiment-multilingual model

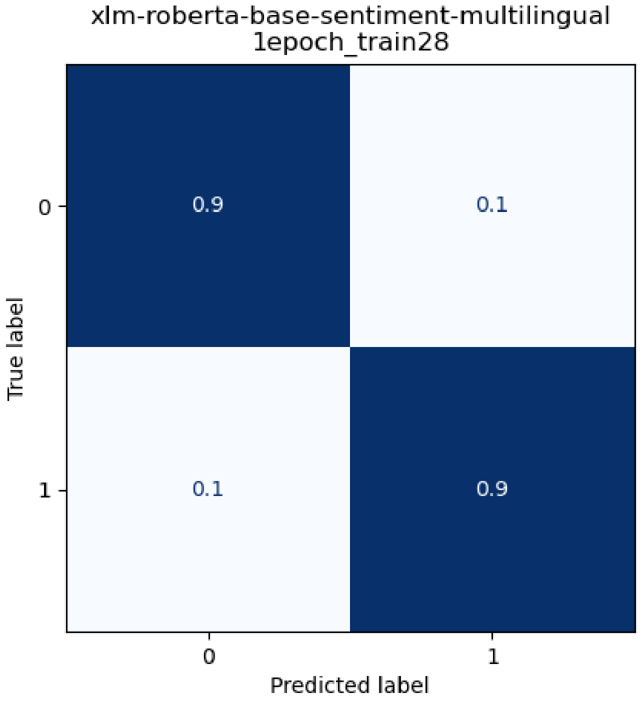


Figure 25: xlm-roberta-base-sentiment-multilingual 1epoch 28quads

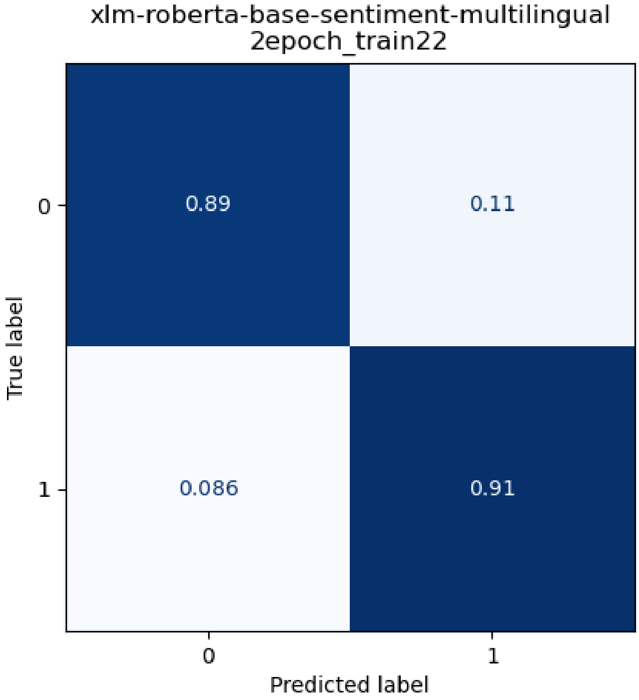


Figure 26: xlm-roberta-base-sentiment-multilingual 2epoch 22quads

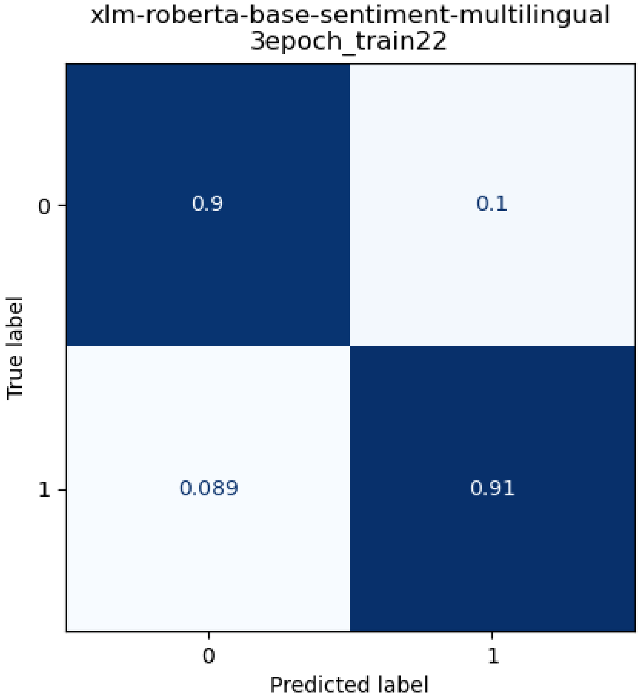


Figure 27: xlm-roberta-base-sentiment-multilingual 3epoch 22quads

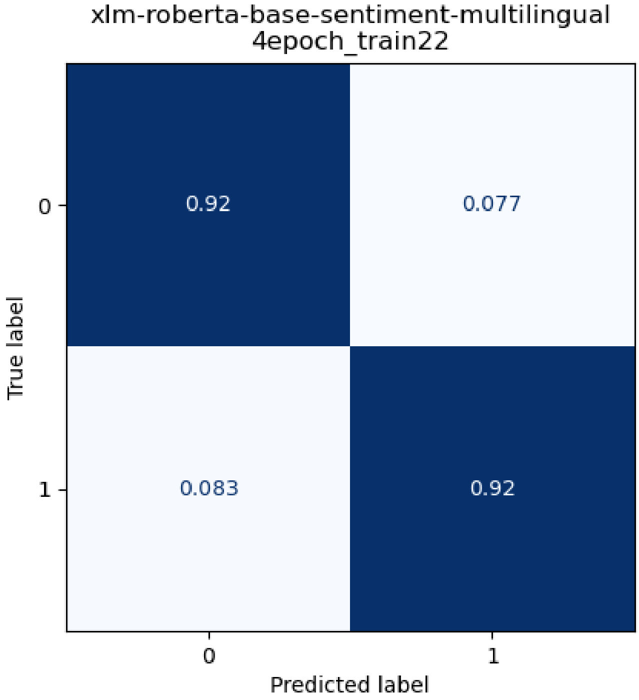


Figure 28: xlm-roberta-base-sentiment-multilingual 4epoch 22quads

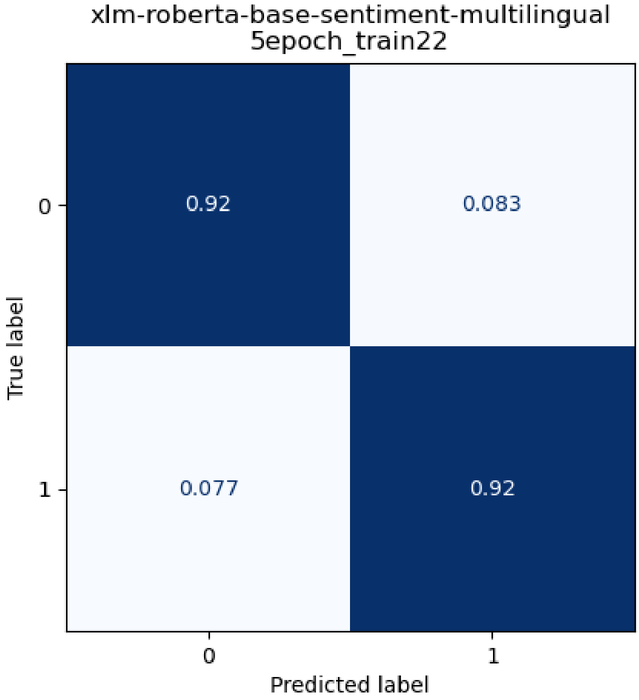


Figure 29: xlm-roberta-base-sentiment-multilingual 5epoch 22quads

9.2.3 bert-base-multilingual-cased-sentiment-multilingual

Best performing confusion matrices from each epoch of bert-base-multilingual-cased-sentiment-multilingual model

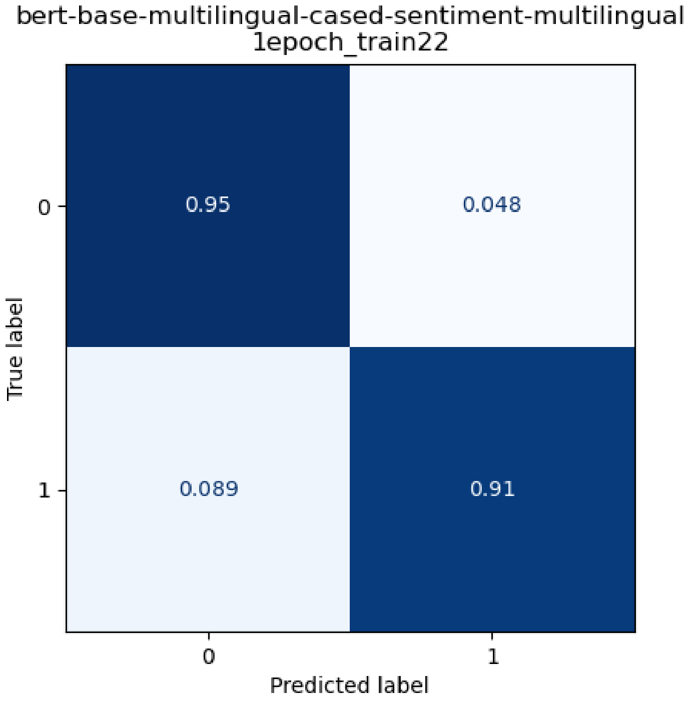


Figure 30: bert-base-multilingual-cased-sentiment-multilingual 1epoch 22quads

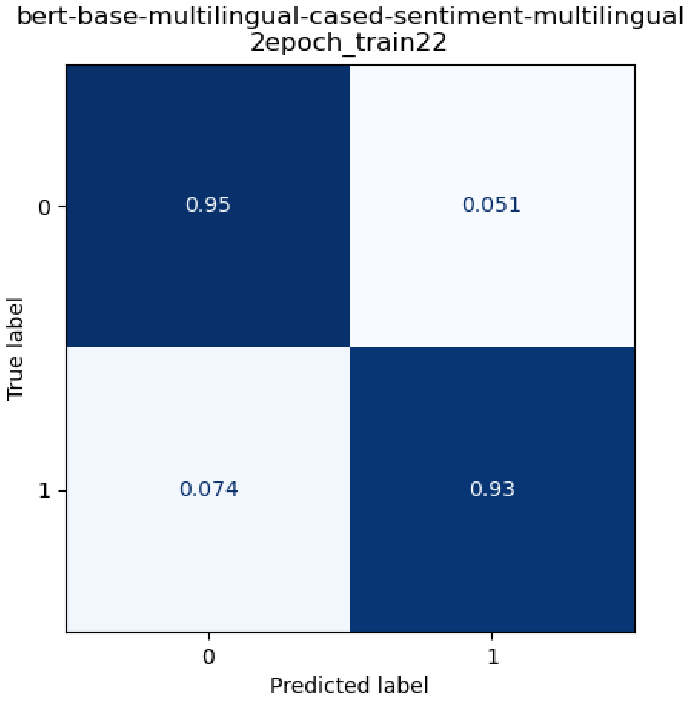


Figure 31: bert-base-multilingual-cased-sentiment-multilingual 2epoch 22quads

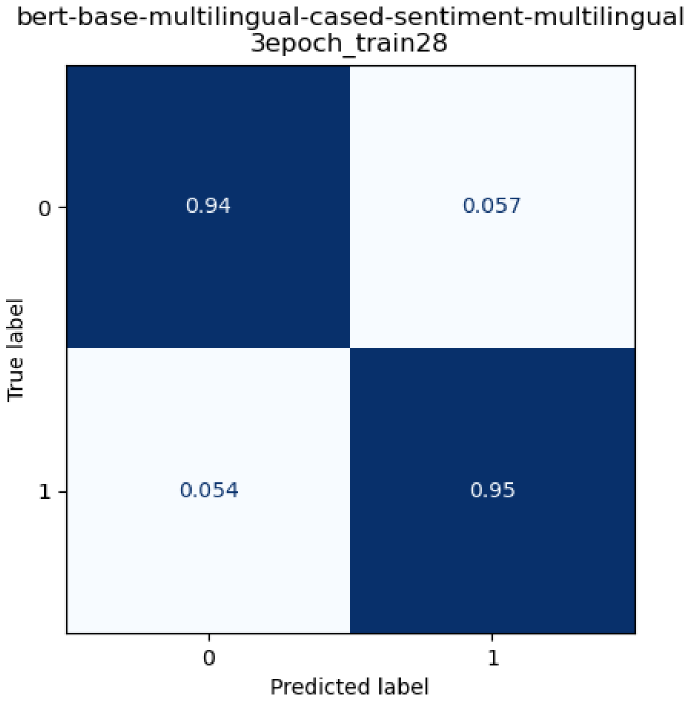


Figure 32: bert-base-multilingual-cased-sentiment-multilingual 3epoch 28quads

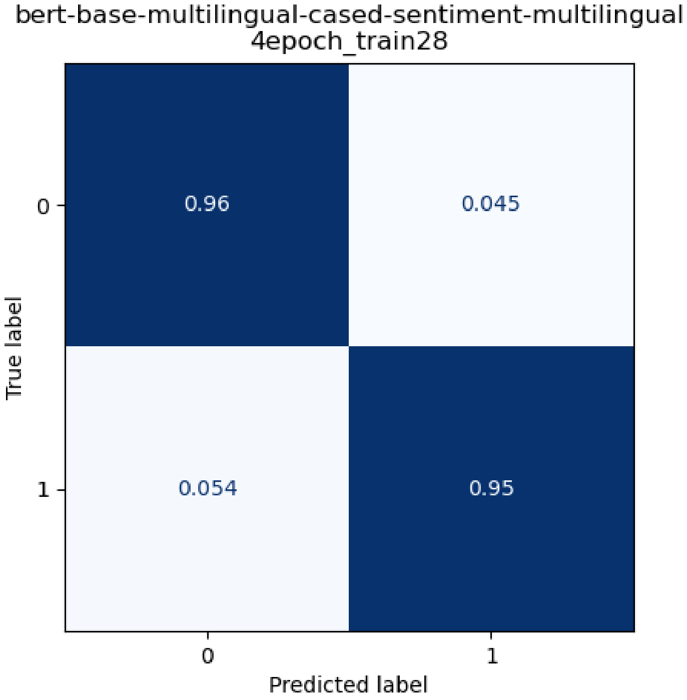


Figure 33: bert-base-multilingual-cased-sentiment-multilingual 4epoch 28quads

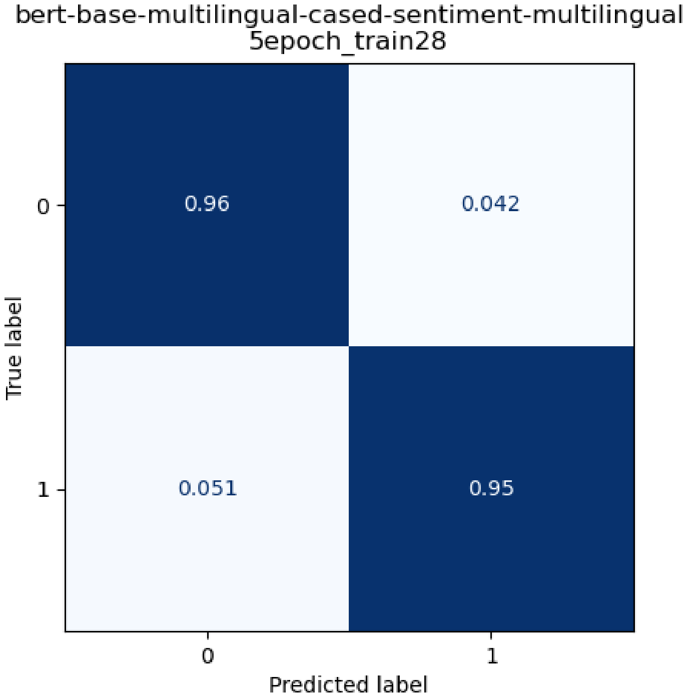


Figure 34: bert-base-multilingual-cased-sentiment-multilingual 5epoch 28quads

9.2.4 dehatebert-mono-french

Best performing confusion matrices from each epoch of dehatebert-mono-french model

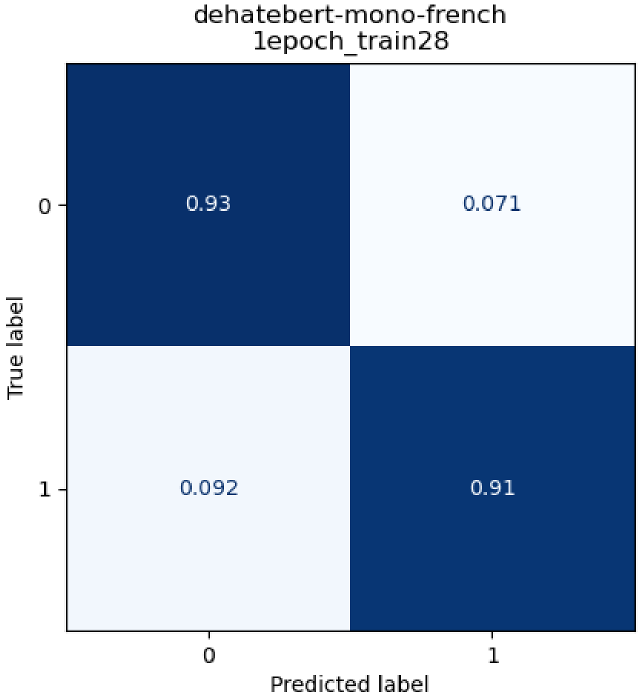


Figure 35: dehatebert-mono-french 1epoch 28quads

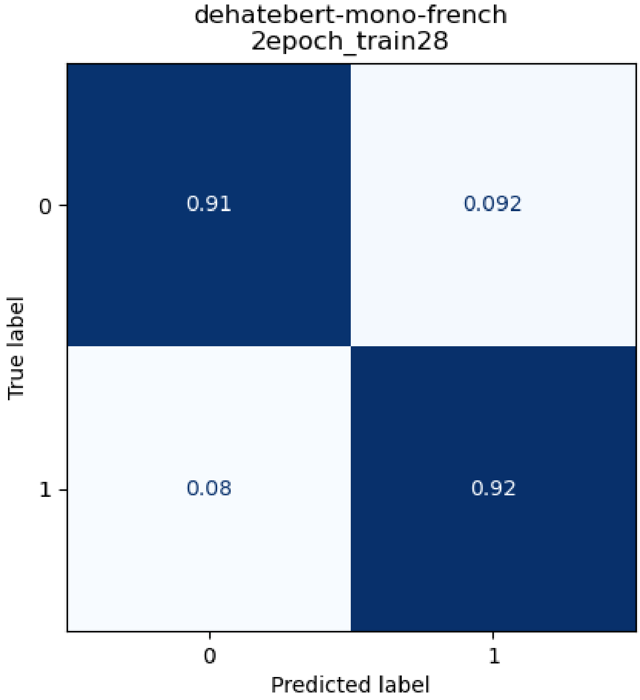


Figure 36: dehatebert-mono-french 2epoch 28quads

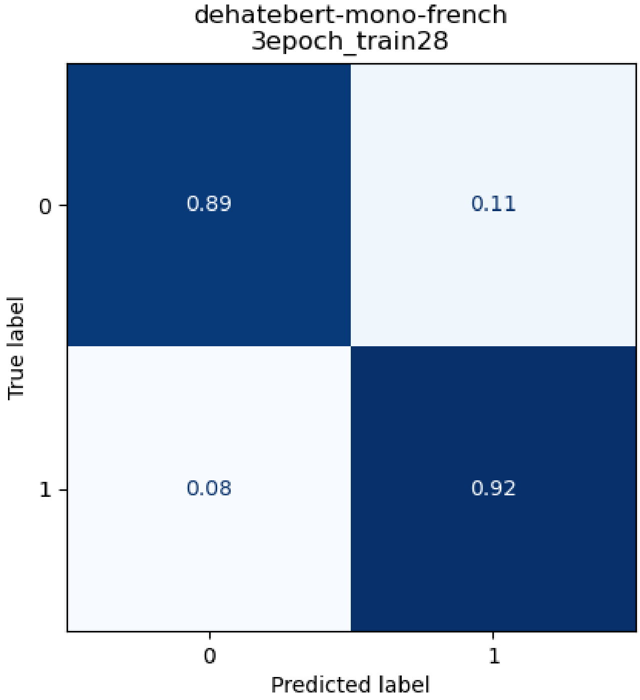


Figure 37: dehatebert-mono-french 3epoch 28quads

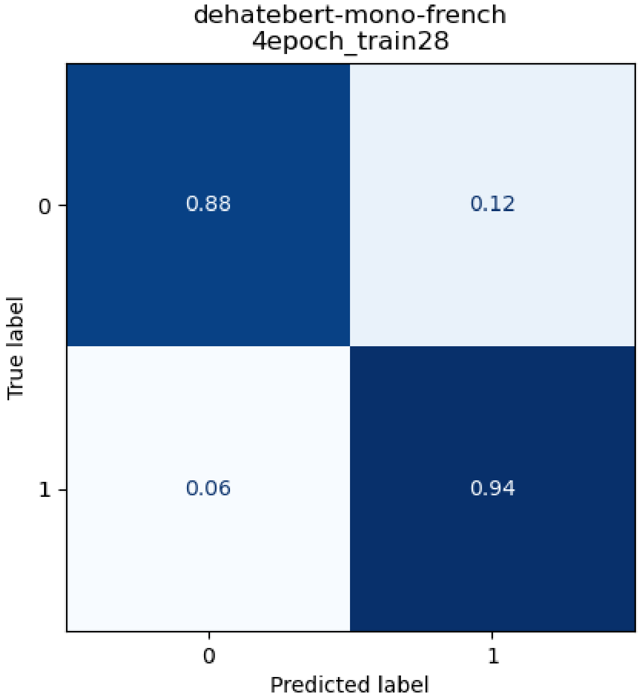


Figure 38: dehatebert-mono-french 4epoch 28quads

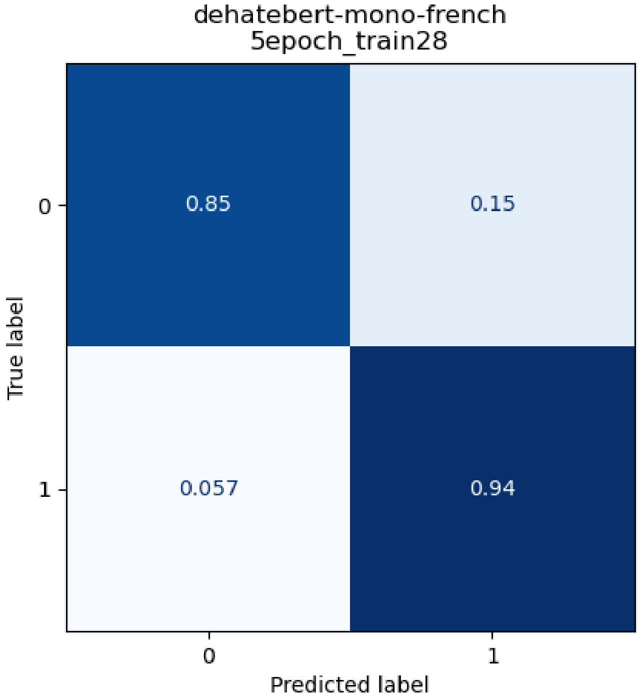


Figure 39: dehatebert-mono-french 5epoch 28quads

9.3 ML with SBERT : accuracy, precision, recall, f1

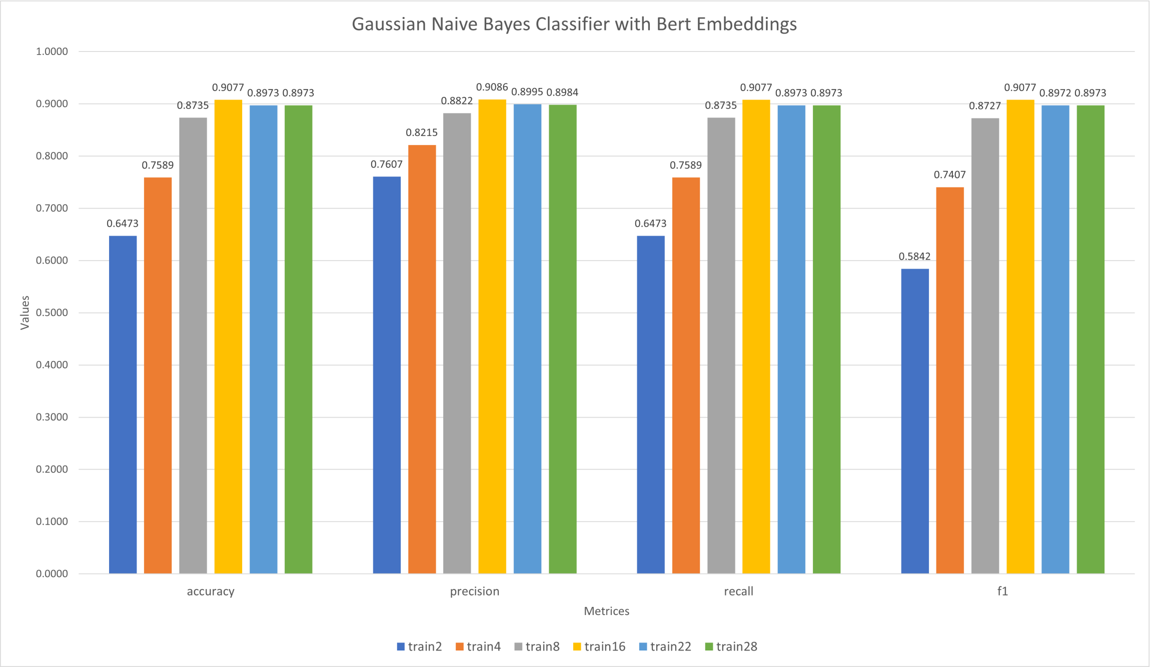


Figure 40: Gaussian NaiveBayes on SBERT

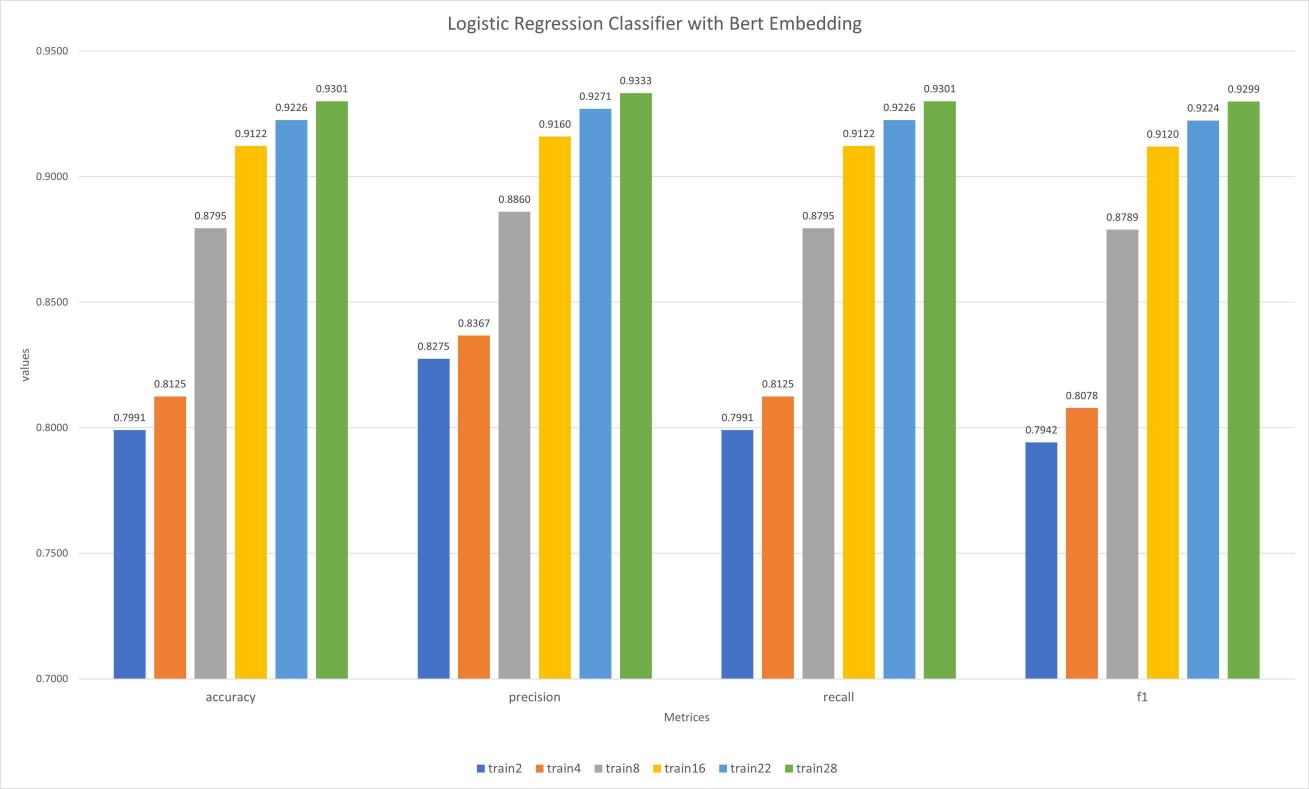


Figure 41: Linear Regression on SBERT

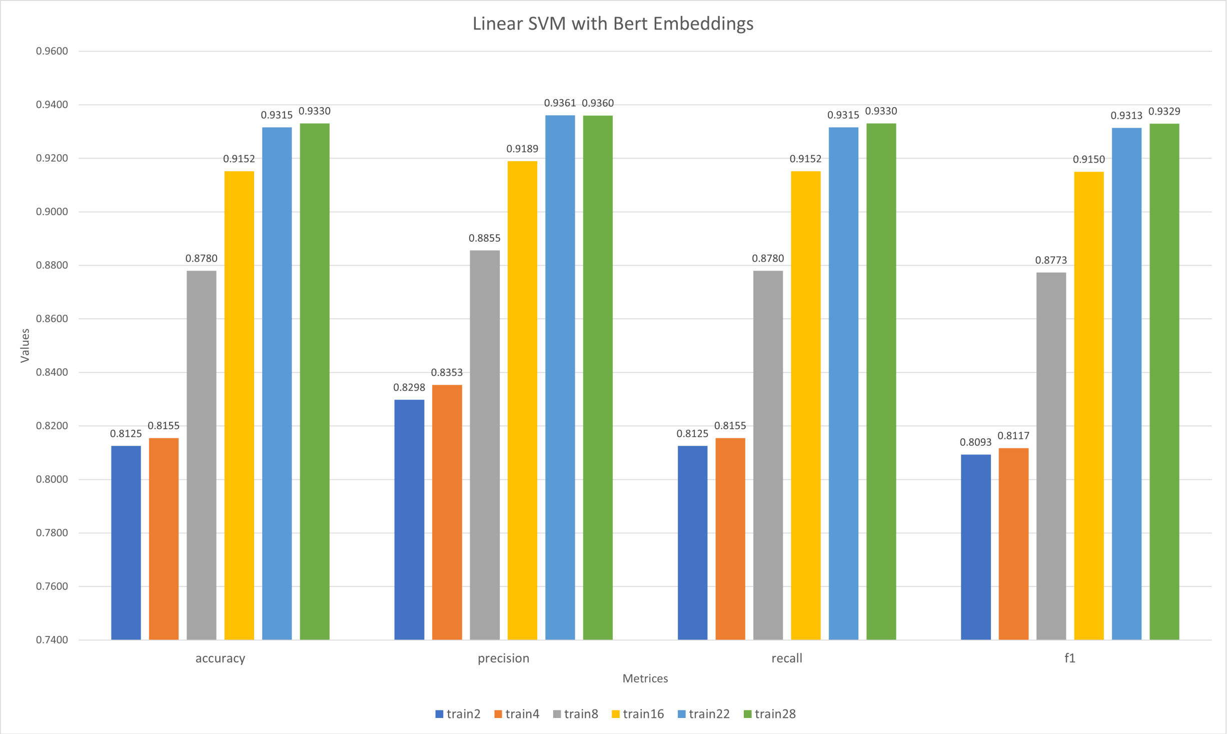


Figure 42: Linear Support Vector Machine on SBERT

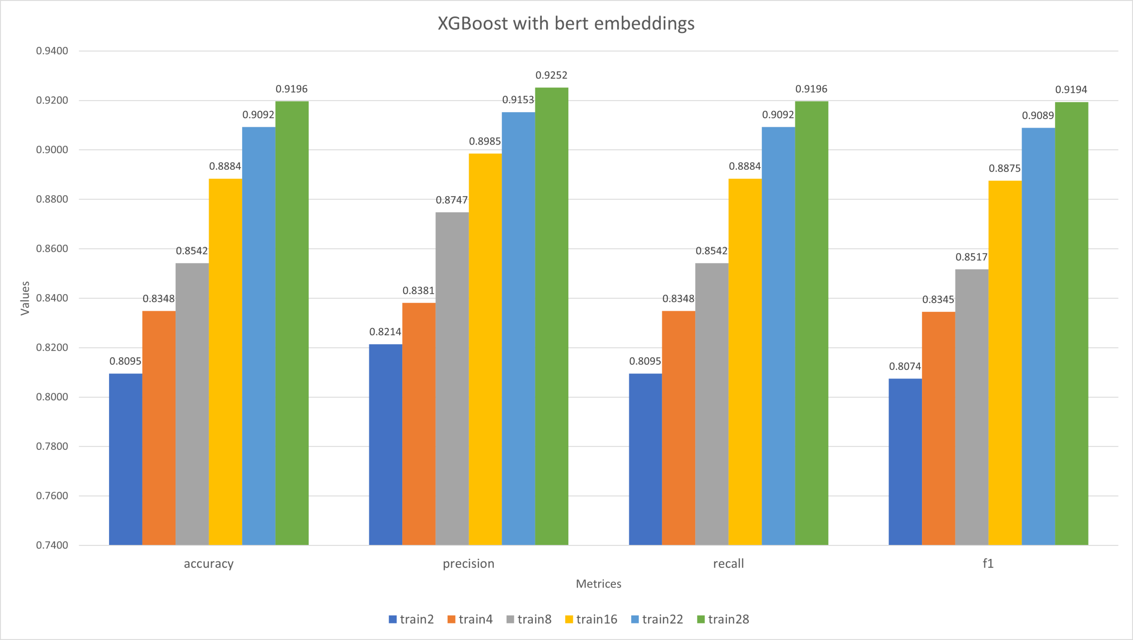


Figure 43: Extreme Gradient Boosting on SBERT

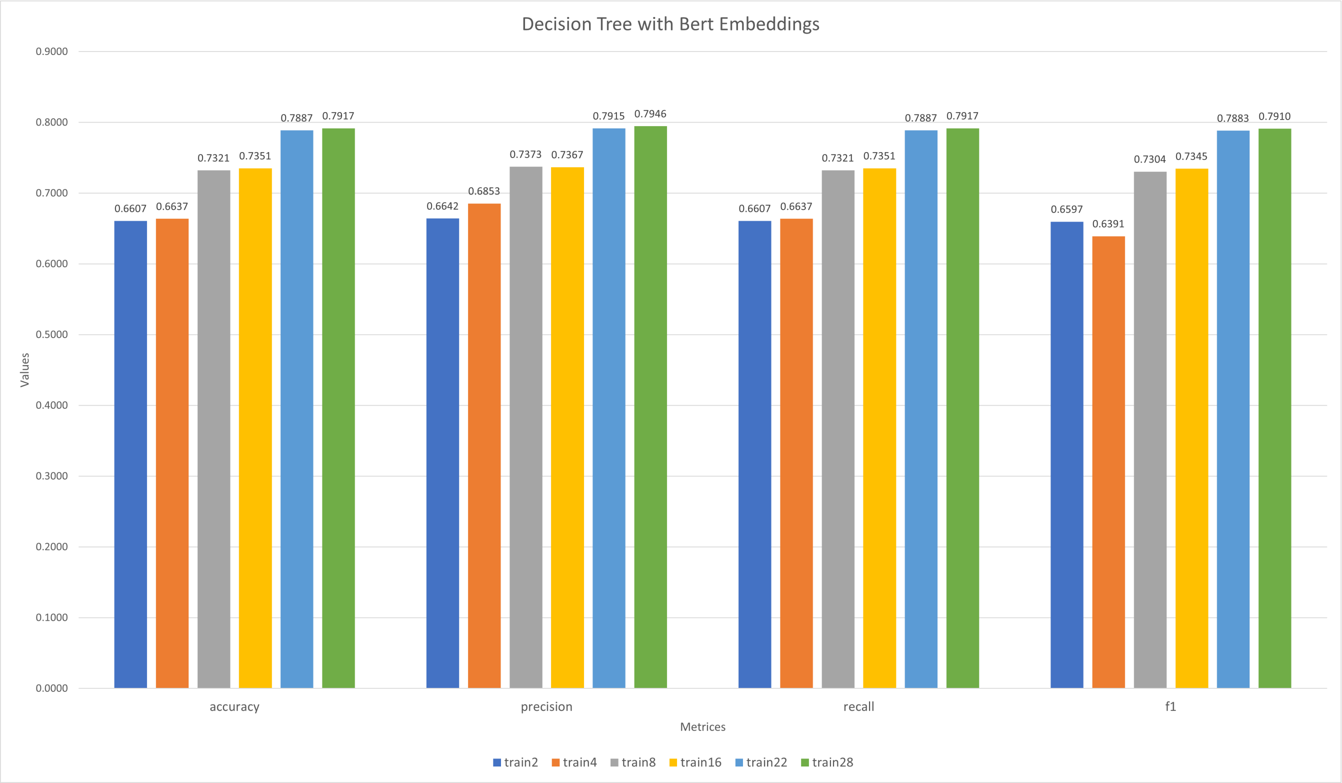


Figure 44: Dicision Tree on SBERT

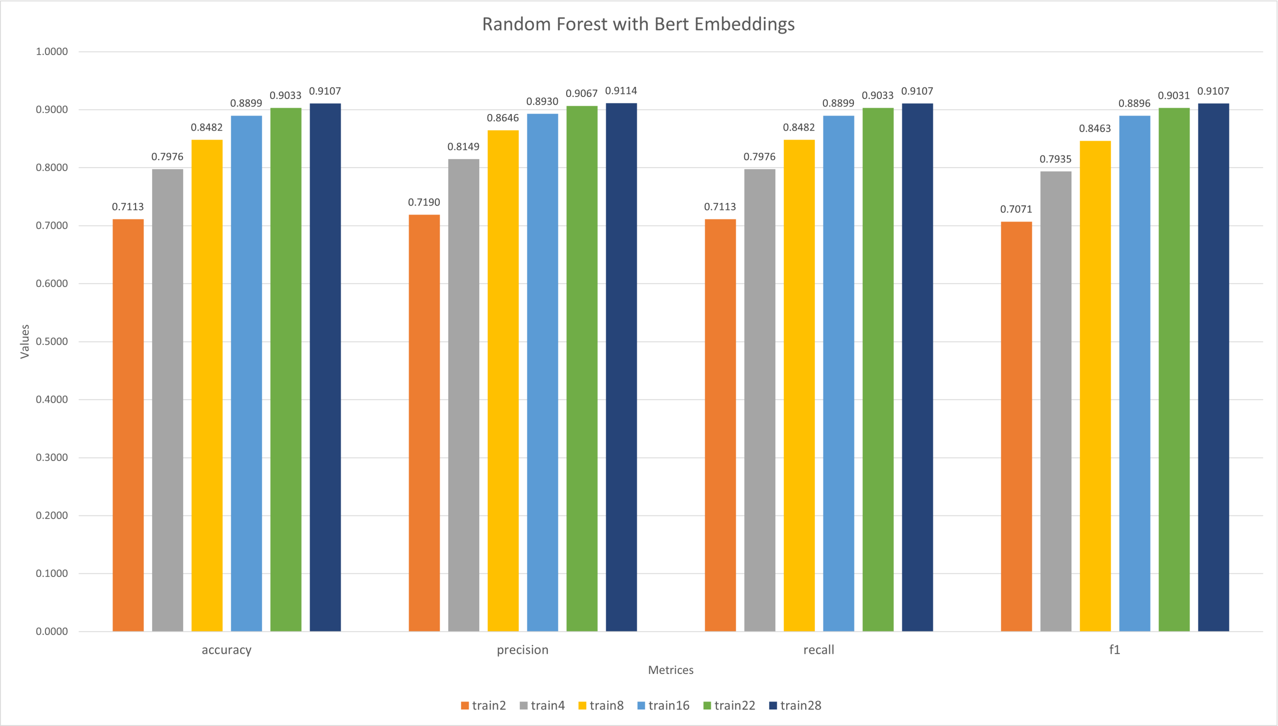


Figure 45: Random Forest on SBERT

Bibliography

[1] A, A., G, S., Shruthi, Upadhyaya, M., Ray, A. P., and T C, M. (2021). Sarcasm detection in natural language processing. *Materials Today: Proceedings*, 37:3324–3331. International Conference on Newer Trends and Innovation in Mechanical Engineering: Materials Science.

[2] Aluru, S. S., Mathew, B., Saha, P., and Mukherjee, A. (2020). Deep learning models for multilingual hate speech detection.

[3] Arango, A., Pérez, J., and Poblete, B. (2022). Hate speech detection is not as easy as you may think: A closer look at model validation (extended version). *Information Systems*, 105:101584.

[4] Banaji MR, A. G. G. (1995). Implicit social cognition: attitudes, self-esteem, and stereotypes. *American Psychological Association*, pages 4–27.

[5] Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? page 610–623.

[6] Bolukbasi, T., Chang, K., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker debiasing word embeddings. *Advances in neural information processing systems*, 29.

[7] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. 33:1877–1901.

[8] Chakravarthi, B. R., Muralidaran, V., Priyadharshini, R., Cn, S., McCrae, J., García, M. Á., Jiménez-Zafra, S. M., Valencia-García, R., Kumaresan, P., Ponnusamy, R., García-Baena, D., and García-Díaz, J. (2022). Overview of the shared task on hope speech detection for equality, diversity, and inclusion. In *Proceedings of the Second Workshop on Language Technology for Equality, Diversity and Inclusion*, pages 378–388, Dublin, Ireland. Association for Computational Linguistics.

[9] Clark, C., Liu, B., Winegard, B., and Ditto, P. (2019). Tribalism is human nature. *Current Directions in Psychological Science.*

[10] Devlin, J., andKenton Lee, M. C., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

[11] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

[12] Esser, P., Rombach, R., and Ommer, B. (2020). A note on data biases in generative models.

[13] Fedus, W., Zoph, B., and Shazeer, N. (2021). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961.

[14] Grondahl, T., Pajola, L., Juuti, M., Conti, M., and Asokan, N. (2018). All you need is 'love': Evading hate speech detection. In *Proceedings of the 11th ACM Workshop on Artificial Intelligence and Security*, AISec '18, page 2–12, New York, NY, USA. Association for Computing Machinery.

[15] Hosseini, H., Kannan, S., Zhang, B., and Poovendran, R. (2017). Deceiving google's perspective api built for detecting toxic comments. *CoRR*, abs/1702.08138.

[16] Kocoń, J., Figas, A., Gruza, M., Puchalska, D., Kajdanowicz, T., and Kazienko, P. (2021). Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach. *Information Processing & Management*, 58(5):102643.

[Liu et al.] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta:.

[18] Ningrum, P. K., Pansombut, T., and Ueranantasun, A. (2020). Text mining of online job advertisements to identify direct discrimination during job hunting process: A case study in indonesia. *PLOS ONE*, 15(6):1–29.

[19] Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks.

[20] Ribeiro, M., Calais, P., Santos, Y., Almeida, V., and Meira Jr., W. (2018). Characterizing and detecting hateful users on twitter. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

[21] Sap, M., Card, D., Gabriel, S., Choi, Y., and Smith, N. A. (2019). The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

[22] Sener, O. and Koltun, V. (2018). Multi-task learning as multi-objective optimization. *CoRR*, abs/1810.04650.

[23] Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. (2017). Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *CoRR*, abs/1701.06538.

[24] Tenney, I., Das, D., and Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline.

[25] Tooby, J. and Cosmides, L. (2010). Groups in mind : the coalitional roots of war and morality.

[26] Tunstall, L., Reimers, N., Jo, U. E. S., Bates, L., Korat, D., Wasserblat, M., and Pereg, O. (2022). Efficient few-shot learning without prompts.

[27] Wiegand, M., Ruppenhofer, J., and Kleinbauer, T. (2019). Detection of Abusive Language: the Problem of Biased Datasets. 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 602–608, Minneapolis, Minnesota. Association for Computational Linguistics.

[28] Wullach, T., Adler, A., and Minkov, E. (2021). Towards hate speech detection at large via deep generative modeling. *IEEE Internet Computing*, 25(2):48–57.

[29] Wullach, T., Adler, A., and Minkov, E. (2022). Character-level hypernetworks for hate speech detection. *Expert Systems with Applications*, 205:117571.

[30] Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. (2018). Gender bias in coreference resolution: Evaluation and debiasing methods. *CoRR*, abs/1804.06876.