

# Multilabel Toxic Comments Classification

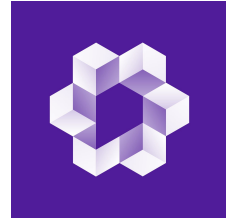
Advanced Machine Learning project - Academic year 2020/2021

Marco Ripamonti - 806785

Christian Bernasconi - 816423

Round 25/01/2021

# Kaggle challenge



## Goal:

- Identification and classification of types of toxicity

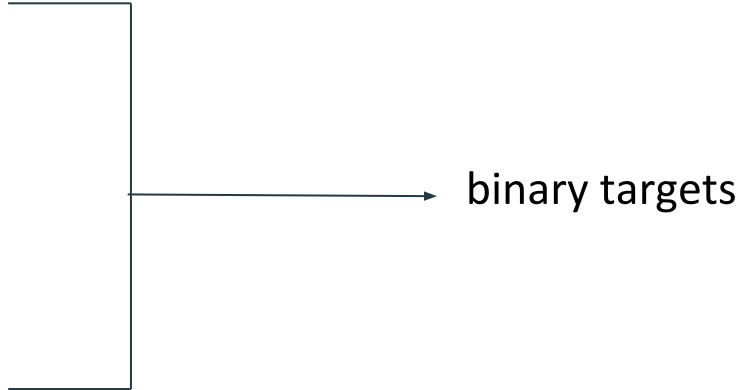
## Context:

- Wikipedia edit pages

## Involving:

- Natural Language Processing
- Multilabel text classification

# Dataset - Features

- **id:** comment identifier → not considered
  - **comment\_text:** raw text in english → main feature
  - **toxic**
  - **severe\_toxic**
  - **obscene**
  - **threat**
  - **insult**
  - **identity\_hate**
- 
- A diagram consisting of a vertical line on the left that connects to a horizontal line, which then points to the text "binary targets". This horizontal line is positioned to the right of the list of toxicity features, effectively grouping them.

# Dataset - Features

*"\n\nCongratulations from  
me as well, use the tools  
well. \xa0· talk "*

toxic	severe_toxic	obscene	threat	insult	identity_hate
0	0	0	0	0	0

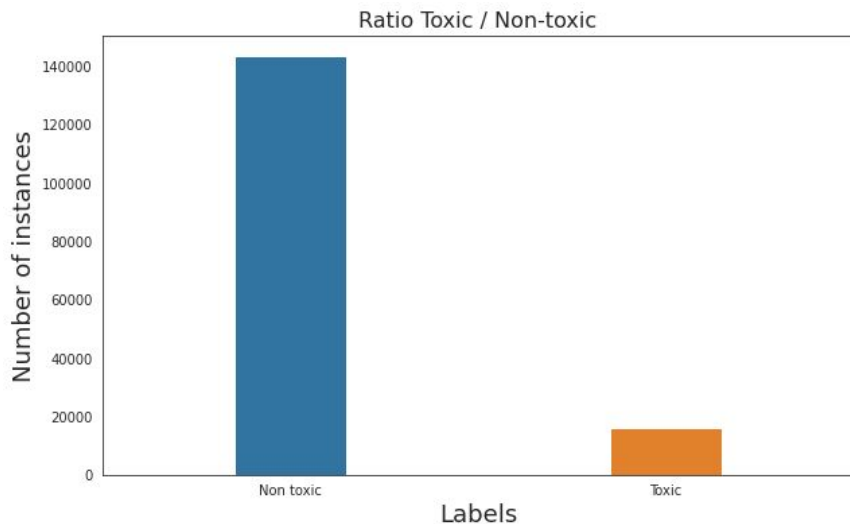


*"C\*\*\*\*\*R BEFORE YOU  
PISS AROUND ON MY  
WORK"*

toxic	severe_toxic	obscene	threat	insult	identity_hate
1	1	1	0	1	0



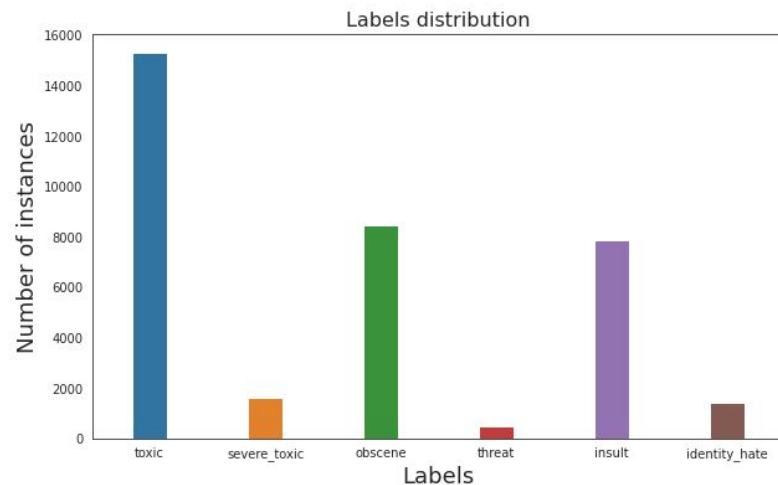
# Dataset - Data distribution



- 160K comments
- 11% toxic comments
- needs balancing
- top words discriminates between toxic / non-toxic

# Dataset - Data distribution

- labels dependencies
- hard to balance
- top words discriminates between toxicities





Preprocessing steps



# Preprocessing - Text cleaning

- lower case
- remove stopwords
- remove punctuation (except “!”)
- remove special chars, numbers, dates, link, etc.
- replace contracted forms
- remove char repetitions (e.g. “hellooo”)

*"\n\nCongratulations from  
me as well, use the tools  
well. \xa0 talk "*

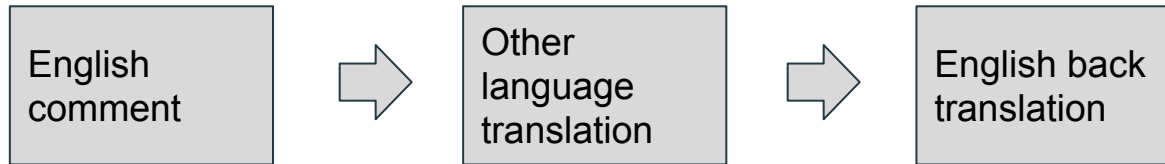


*“congratulations well use  
tools well talk”*

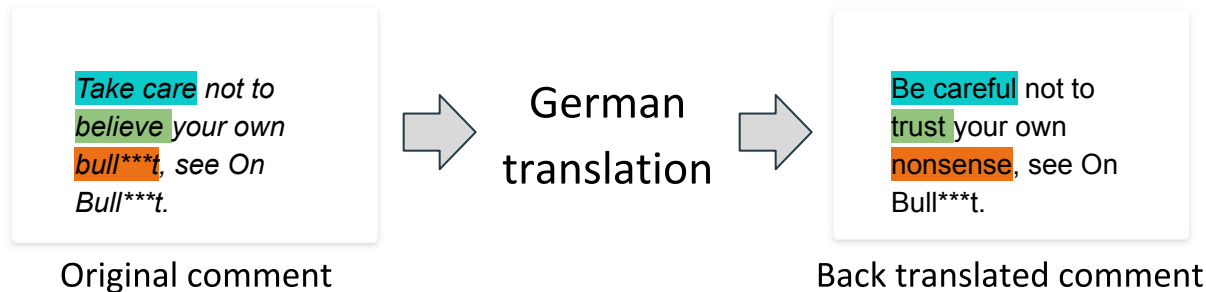


# Preprocessing - Data Augmentation

## Back translations strategy:

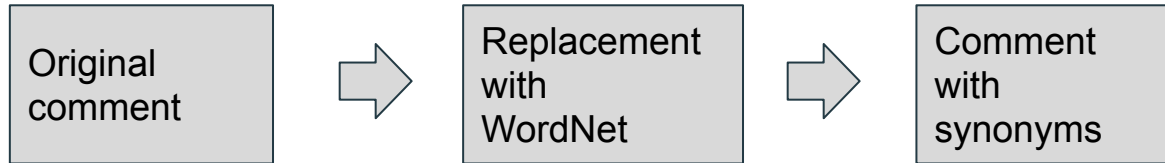


## Example:

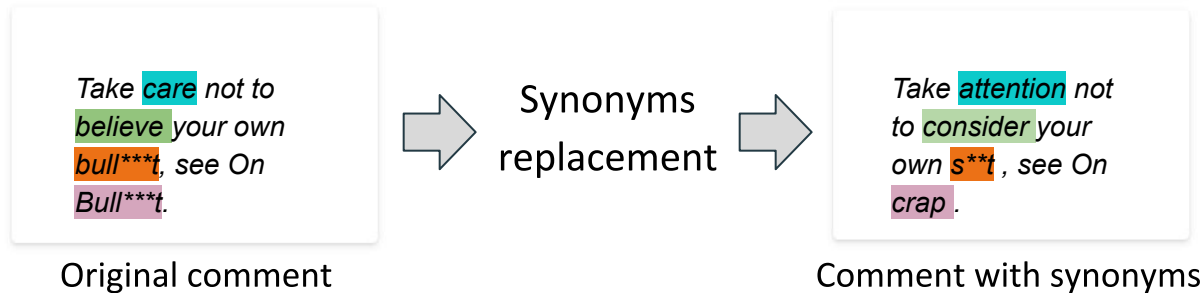


# Preprocessing - Data Augmentation

## Synonyms strategy:



## Example:



# Preprocessing - Word Embeddings

Vocabulary top 20k words

stupid	1
...	...
peace	50
war	51
...	...
hello	137
...	..



*"Stupid peace of s\*\*t stop deleting my stuff"*

1	50	...	0	0
---	----	-----	---	---

**150 tokens**



Word embedding

0.17	0.52	0.34	...	0.01
0.25	0.59	-0.34	...	-0.99
0.44	-0.58	0.34	...	1.23
0.87	-0.52	-1.23	...	0.3
0.17	0.52	0.34	...	0.04
...	...	...	...	...
0.02	0.63	-0.07	...	-0.78

# Word Embeddings - Word2Vec and BERT

- **fastText:**

- CBOW model
- 2 million word vectors trained on Common Crawl
- 300 features

- **GloVe:**

- co-occurrences of word statistics on a large corpus
- 1 million word vectors trained on tweets
- 200 features

- **BERT:**

- Contextual word embedding
- DistilBERT

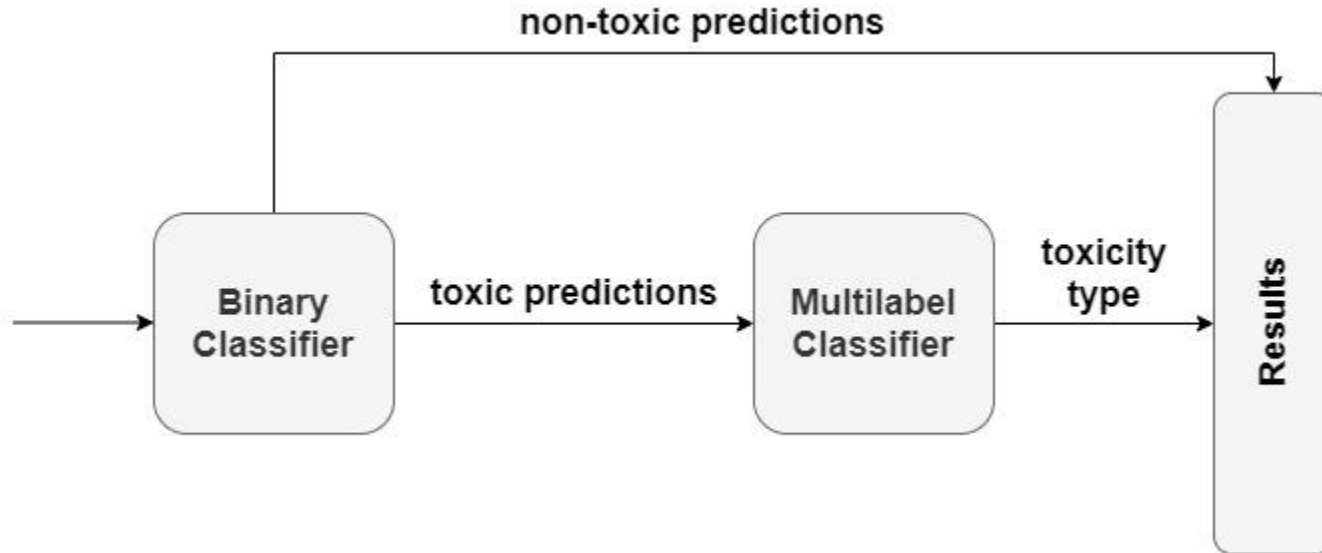


# Methodological Approach



# Approach - Pipeline

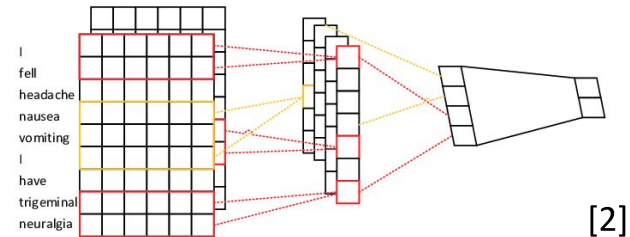
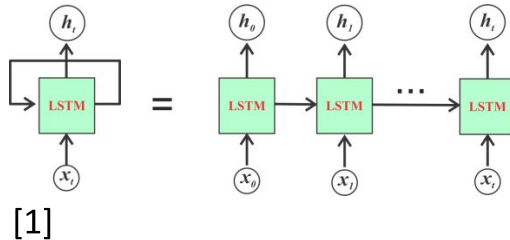
Split the classification into two phases:



# Approach - Models Architectures

For both the two phase have been considered the following architectures:

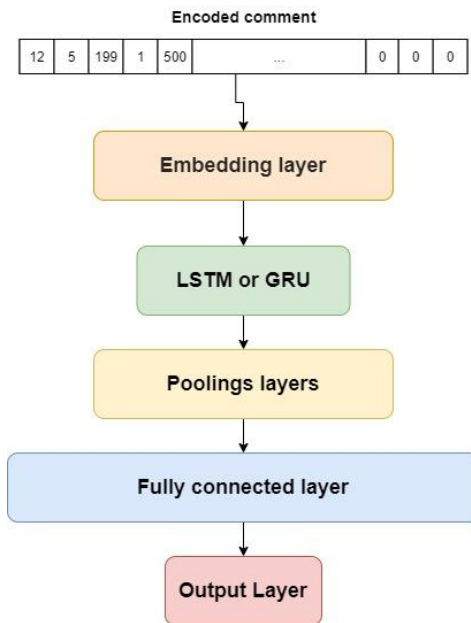
- LSTM
- GRU
- CNN



[1] <https://www.codeproject.com/Articles/5165357/In-depth-LSTM-Implementation-using-CNTK-on-NET-pla>

[2] [https://www.researchgate.net/figure/Architecture-of-CNN-for-Text-Classification\\_fig4\\_305695642](https://www.researchgate.net/figure/Architecture-of-CNN-for-Text-Classification_fig4_305695642)

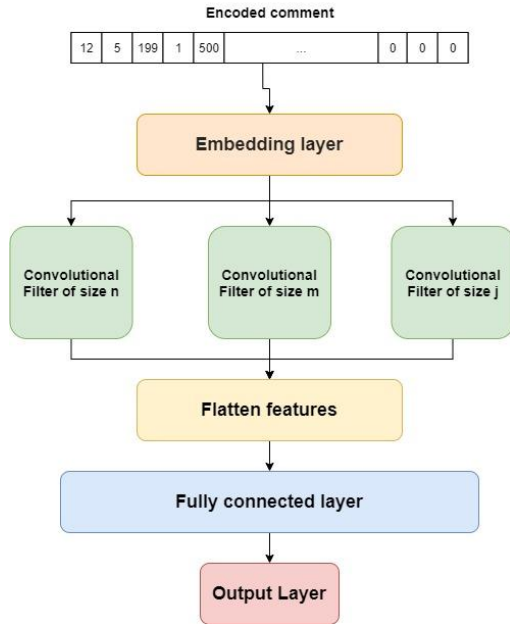
# Approach - Models Architectures



- fastText/BERT embedding
- bidirectional LSTM/GRU
- max/avg poolings combination
- dropout as regularization



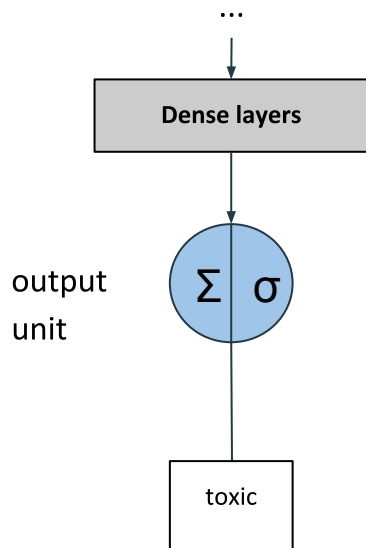
# Approach - Models Architectures



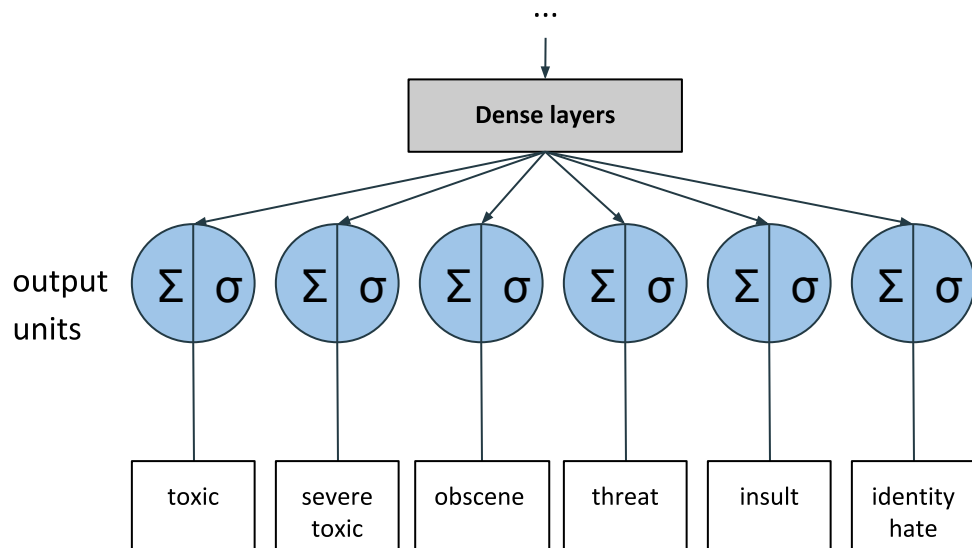
- fastText embedding
- parallel convolutions of different sizes [3]
- dropout as regularization

# Approach - Models Architectures

## Binary model:



## Multilabel model:



# Approach - Training configuration

- adam optimizer
- learning rate 0.001
- binary cross-entropy loss
- early stopping
- 75% training set - 25% validation set

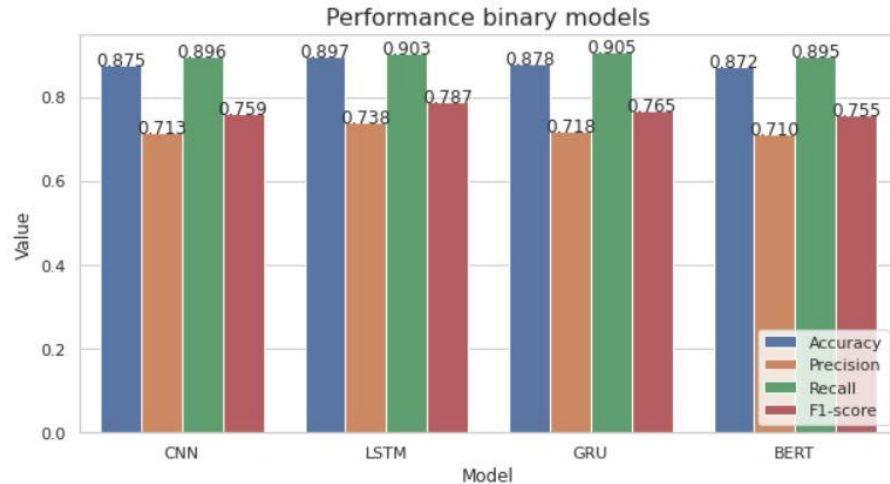
Model	Phase	# epochs	Time	Validation Loss
LSTM	binary	13	9m	0.0940
GRU	binary	13	20m	0.1032
CNN	binary	19	8m	0.1001
BERT	binary	20	2h	0.1115
LSTM	multilabel	50	36m	0.0713
GRU	multilabel	42	28m	0.0846
CNN	multilabel	34	4m	0.0720
BERT	multilabel	43	3h	0.1131



# Evaluation



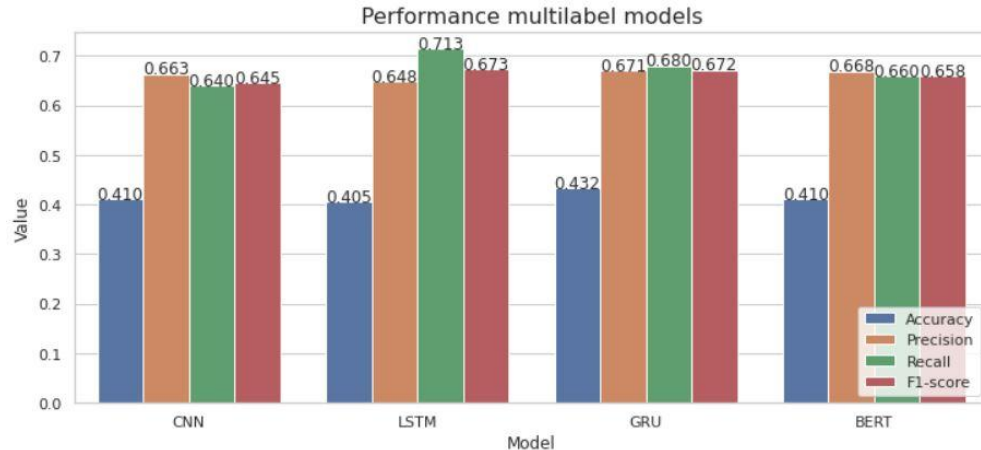
# Evaluation - Binary classification



## Best models:

- precision: LSTM
- recall: GRU
- F1-score: LSTM

# Evaluation - Multilabel classification

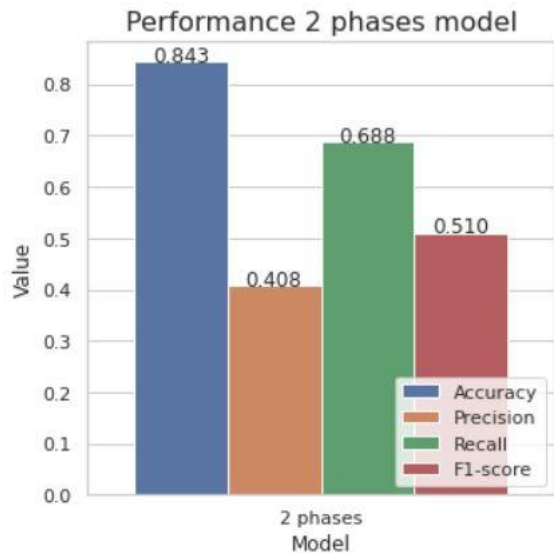


## Best models:

- precision: GRU
- recall: LSTM
- F1-score: LSTM

# Evaluation - Combined classifiers

LSTM binary + LSTM multilabel

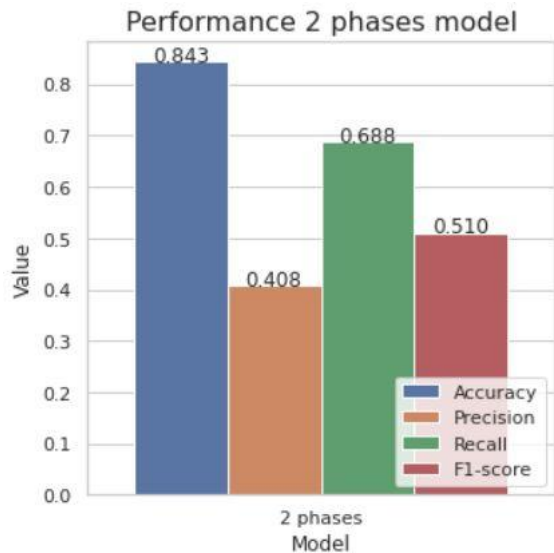


Toxic		Severe toxic		Obscene	
51536	5892	62559	591	56648	3179
604	5485	168	199	619	3071

Threat		Insult		Identity hate	
63082	224	57296	2795	62220	585
101	110	896	2530	292	420

# Evaluation - Combined classifiers

LSTM binary + LSTM multilabel



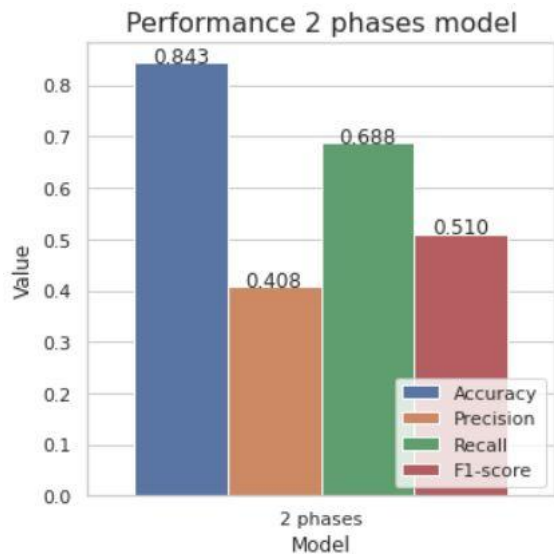
Toxic		Severe toxic		Obscene	
51536	5892	62559	591	56648	3179
604	5485	168	199	619	3071

Threat		Insult		Identity hate	
63082	224	57296	2795	62220	585
101	110	896	2530	292	420



# Evaluation - Combined classifiers

LSTM binary + LSTM multilabel



Toxic		Severe toxic		Obscene	
51536	5892	62559	591	56648	3179
604	5485	168	199	619	3071

Threat		Insult		Identity hate	
63082	224	57296	2795	62220	585
101	110	896	2530	292	420

# Considerations

- **two-phases** approach allows to:
  - set a different classification threshold for each phase
  - combine optimal configurations of different models
- **binary model**: lots of false positives
- **multilabel model**: high labels imbalance → minority labels misclassified
- **best configuration**: LSTM binary + LSTM multilabel

# Improvements and future works

- improve first model to better filter toxic comments
- augment exclusively minority labels examples
- put more effort on the BERT based models
- ensemble techniques (i.g., models based on TF-IDF, text features, etc.)



Thanks for the attention!