TECHNISCHE UNIVERSITÄT
MÜNCHEN

Universal adversarial triggers - Defense

Legal Data Analysis Lab

Rafid Abyaad
Isaac Olguín
Elif Demirtas
Shanshan Xu

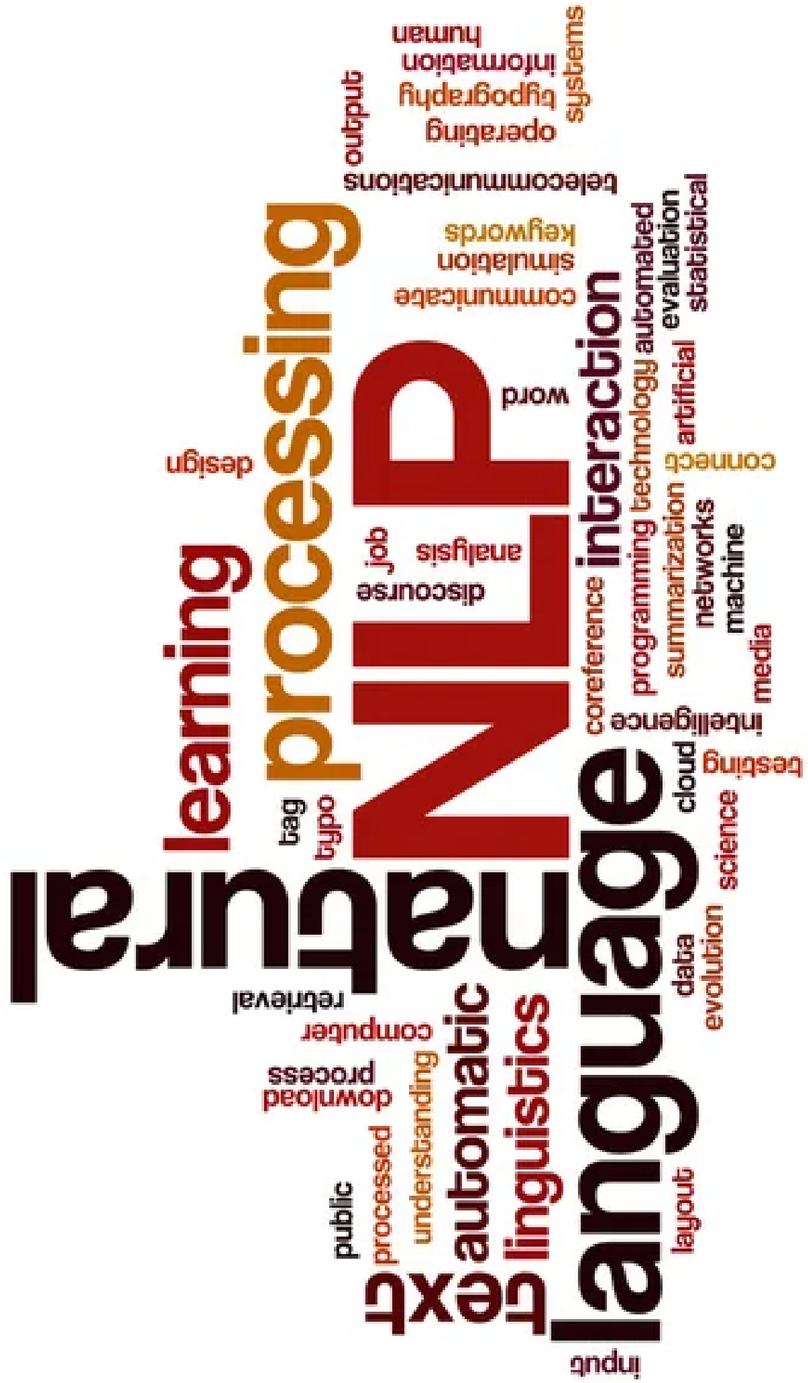
Informatics Department - Legal Tech Chair

Overview

- Introduction
- State of the art
- General implementation
  - Dataset
  - TF-IDF & SVM
  - Baseline model
  - Trigger generation
  - Extension
  - Naturalness
- Defense perspective
  - Perplexity (no natural triggers)
  - Perplexity of naturalness implem.
  - Model enhancement
- Learnings and conclusion
- Future job
- References

Back to Overview

Introduction



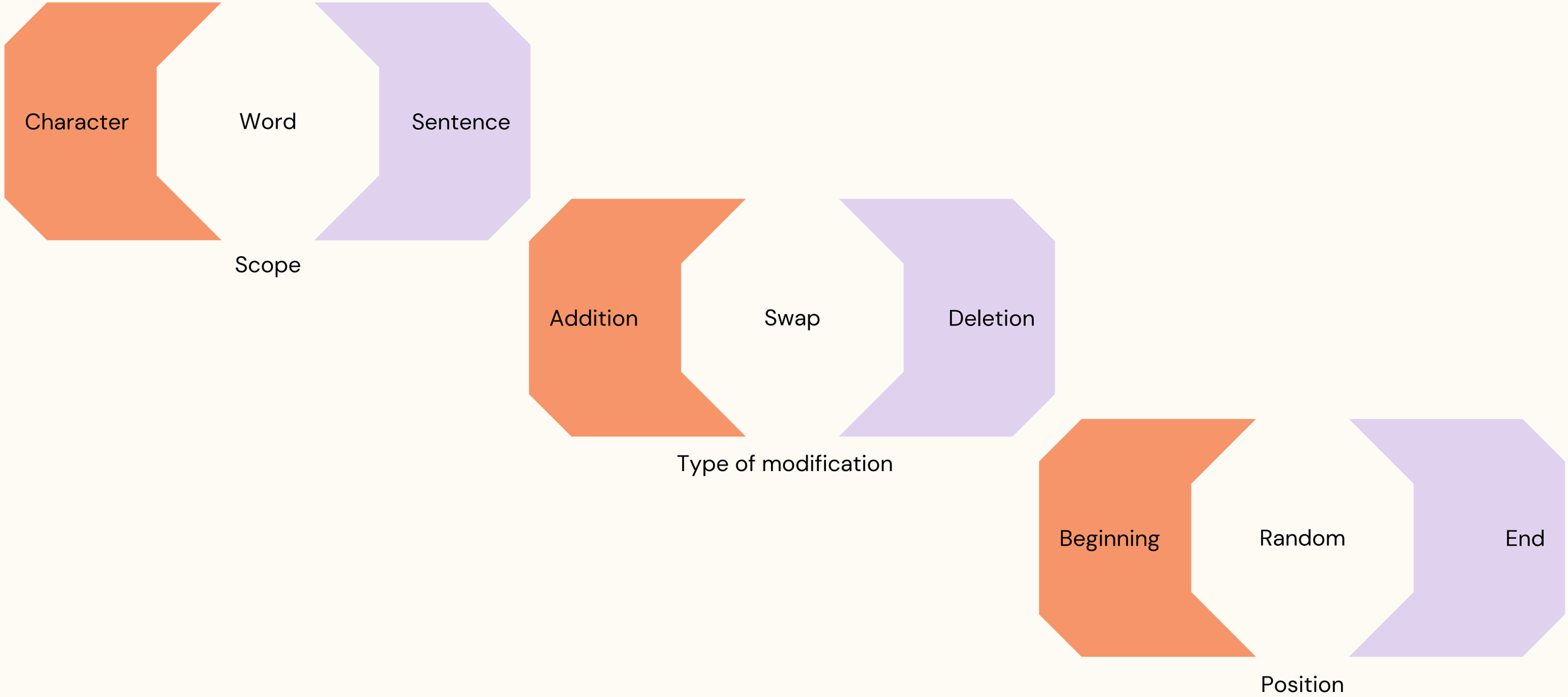
Back to Overview

Introduction

- Classifiers
- Universarial adversarial attack
- Adversarial attacks: perturbations

Trigger operations

Back to Overview



Methodology

Back to Overview

Reproduction	• Implement the approach compatible with the LegalBERT model and CLAUDETTE dataset • Generate triggers
Extension	• Change trigger location
Naturalness	• Make triggers more natural by implementing Song et al 2001
Defense	• Implement defense (Perplexity and enhancement of the model)
Experiments	• Conduct experiments
Results	• Evaluate metric results

Back to Overview

State of the Art

State of the Art

Back to Overview

Papers

- HotFlip: White-Box Adversarial Examples for Text Classification
- Universal Adversarial Triggers for Attacking and Analyzing NLP
- Universal Adversarial Attacks with Natural Triggers for Text Classification
- CLAUDETTE: an Automated Detector of Potentially Unfair Clauses in Online Terms of Service

Extra papers

- Defense against Adversarial Attacks in NLP via Dirichlet Neighborhood Ensemble
- Adversarial Reprogramming of Text Classification Neural Networks

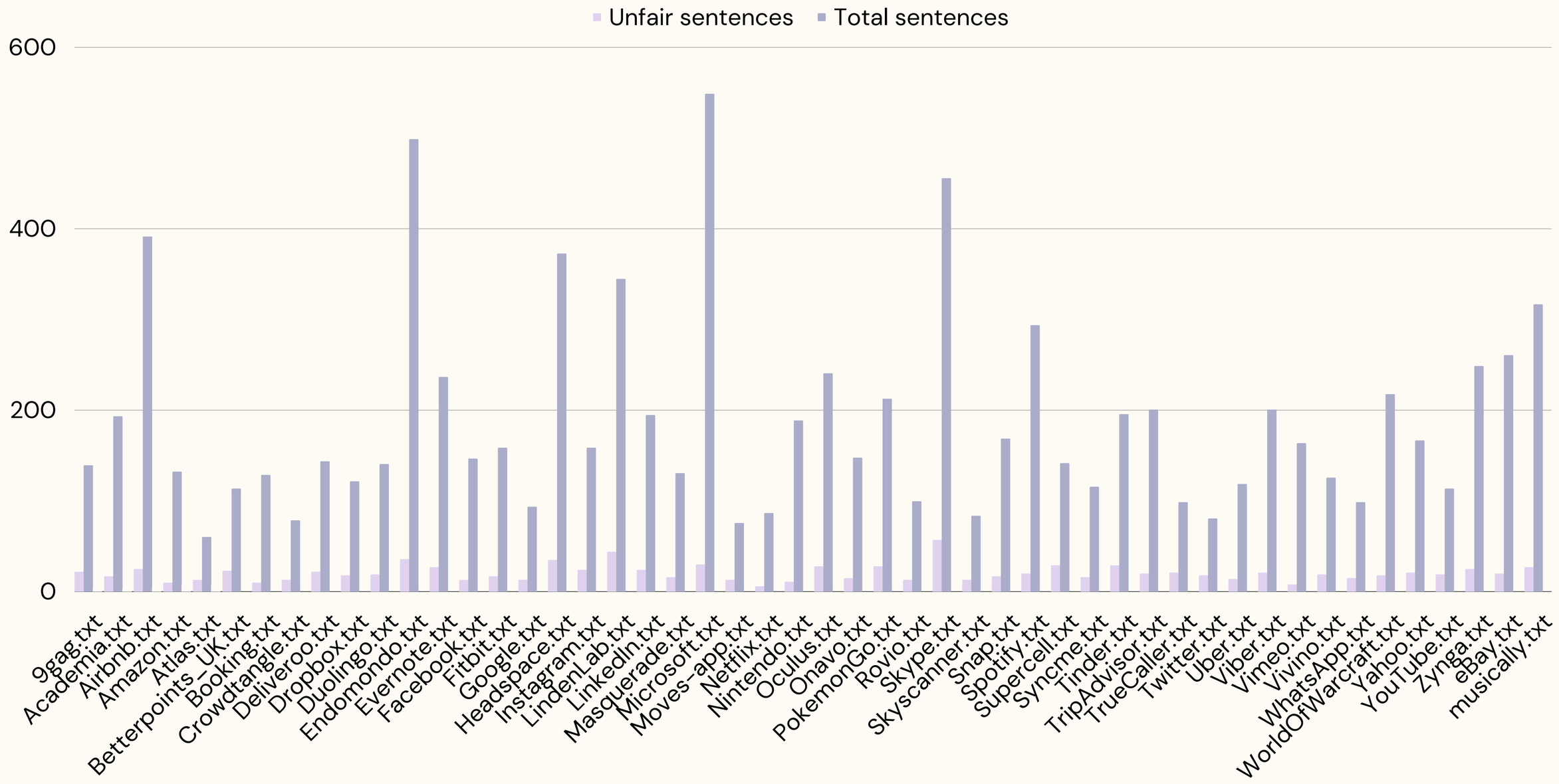
Back to Overview

General implementation

Back to Overview

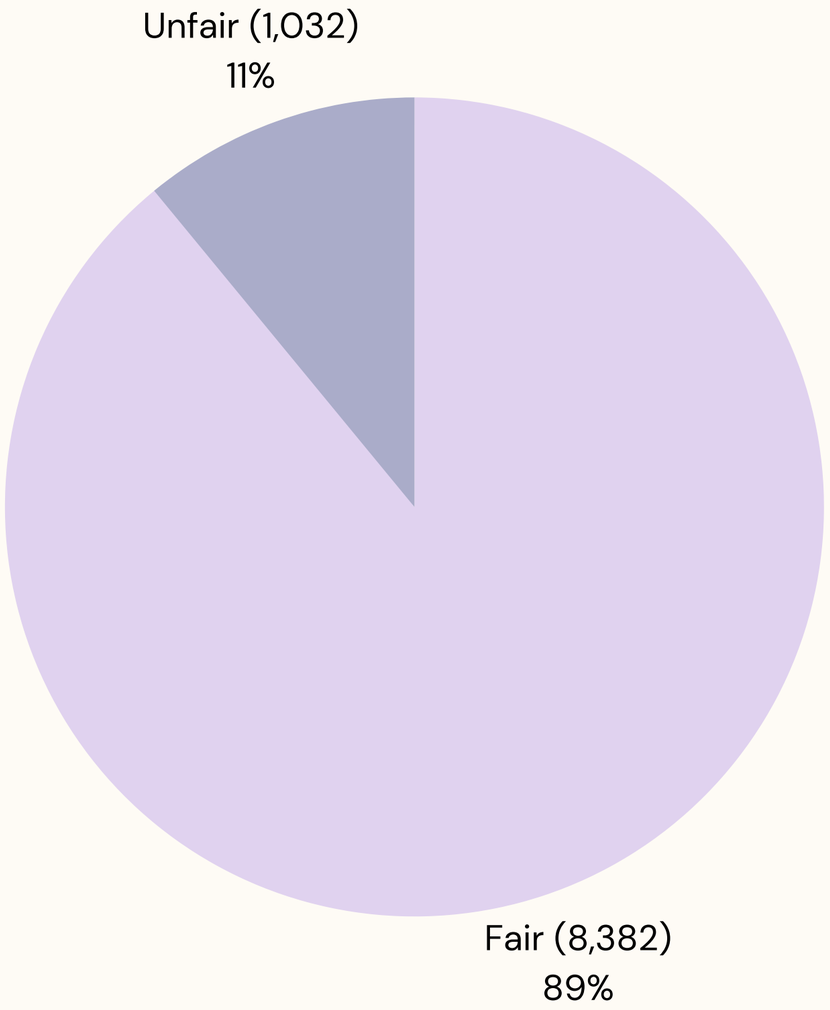
Dataset

Distribution of sentences among files



Back to Overview

Imbalance classes



Strategies against it

- Resampling technique: undersampling or oversampling
- Data augmentation
- Back translation
- Text generation (Generative Adversarial Network)

Back to Overview

Back to Overview

TF-IDF & SVM

TF-IDF

- Statistical measure that evaluates how relevant a word is to a document in a collection of documents.

- Similar words will have similar vectors, i.e. search engine

Text sample:
"parties retain the right to seek relief in small claims court for certain claims, at their option

- TF-IDF features are extracted from Claudette dataset to use on SVM classifier

feature	tf-idf
claims	0.448079
option	0.344648
retain	0.309263
seek	0.306915
small	0.304674
relief	0.291321

Back to Overview

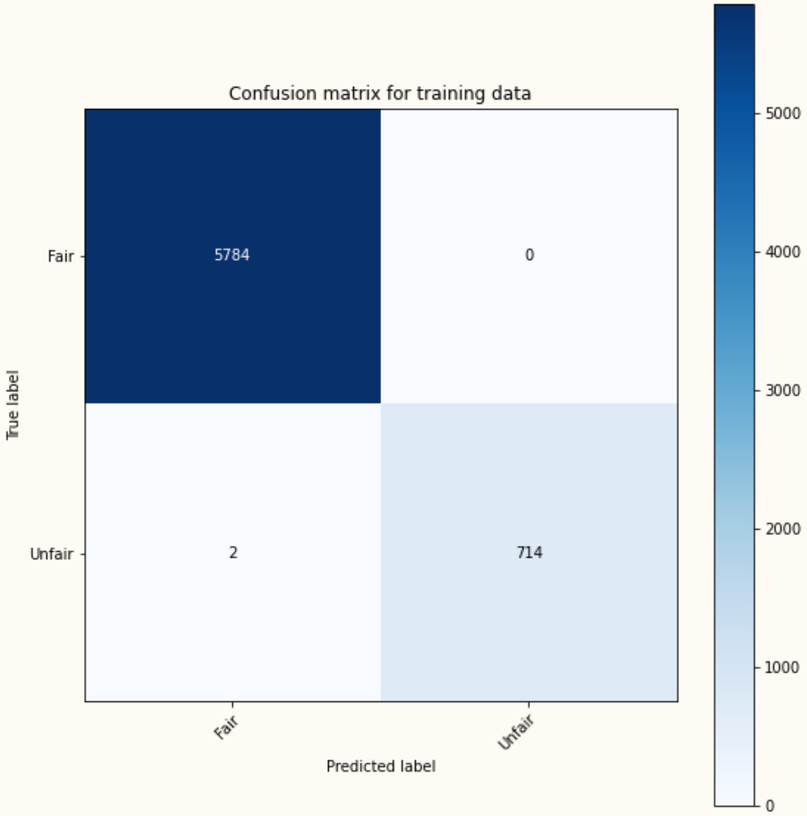
SVM

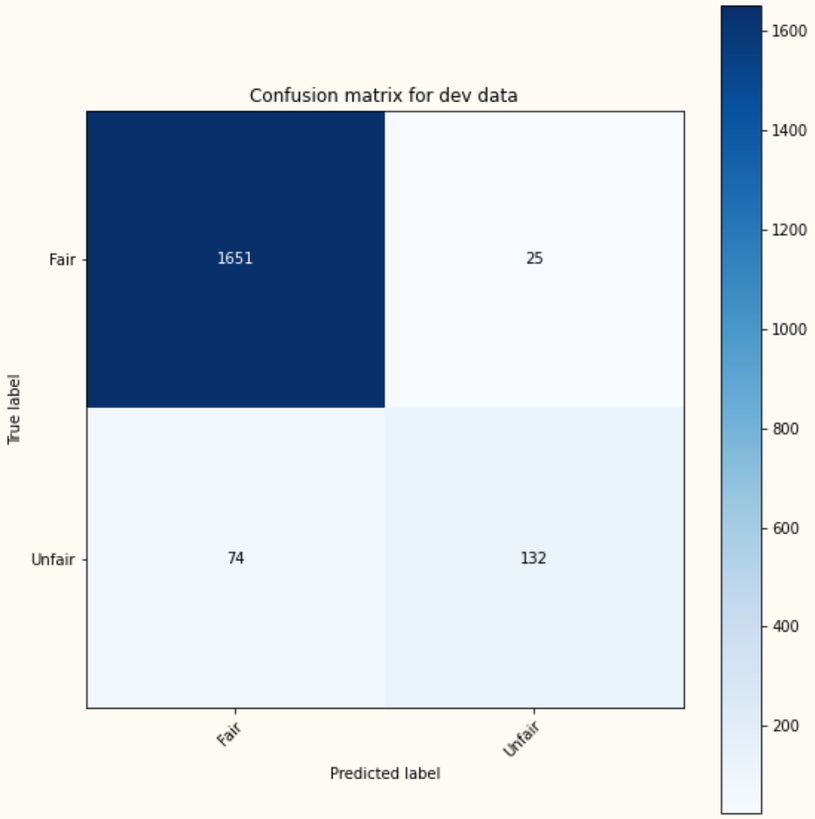
TRAIN:

	precision	recall	f1-score	support
Fair	1.00	1.00	1.00	5784
Unfair	1.00	1.00	1.00	716
accuracy			1.00	6500
macro avg	1.00	1.00	1.00	6500
weighted avg	1.00	1.00	1.00	6500

DEV:

	precision	recall	f1-score	support
Fair	0.96	0.99	0.97	1676
Unfair	0.84	0.64	0.73	206
accuracy			0.95	1882
macro avg	0.90	0.81	0.85	1882
weighted avg	0.94	0.95	0.94	1882





Back to Overview

Baseline model

Baseline model

- Creating a baseline model as a target for attacks

- Binary classification model

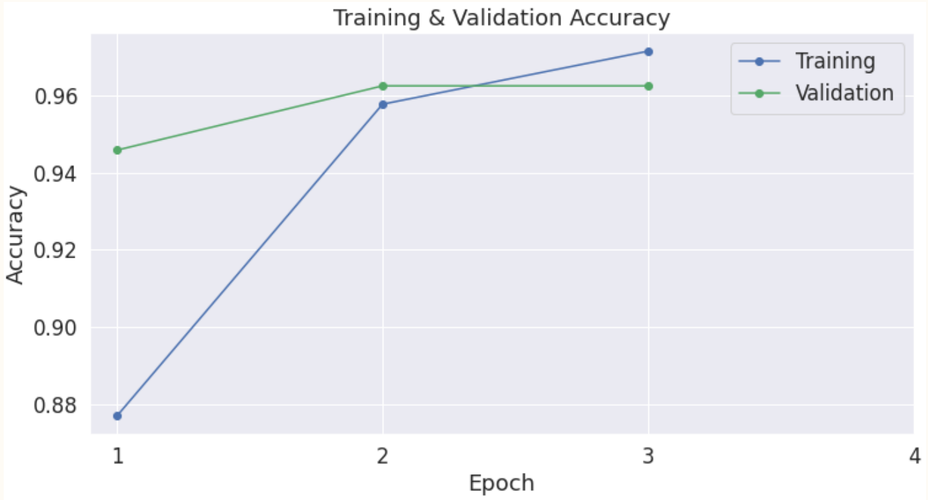
- Implementation:
  - Pre-trained LegalBert backbone
  - Fine-tuned on Claudette
  - CosineAnnealingLR scheduler and Adam optimizer
  - BertForSequenceClassification

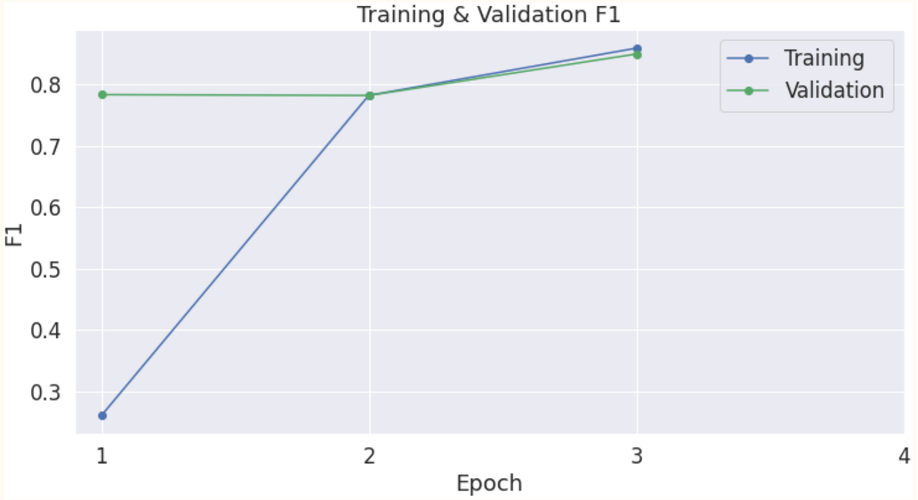
- BERT
  - Embedding size: 512
  - Vocabulary size: 30,522

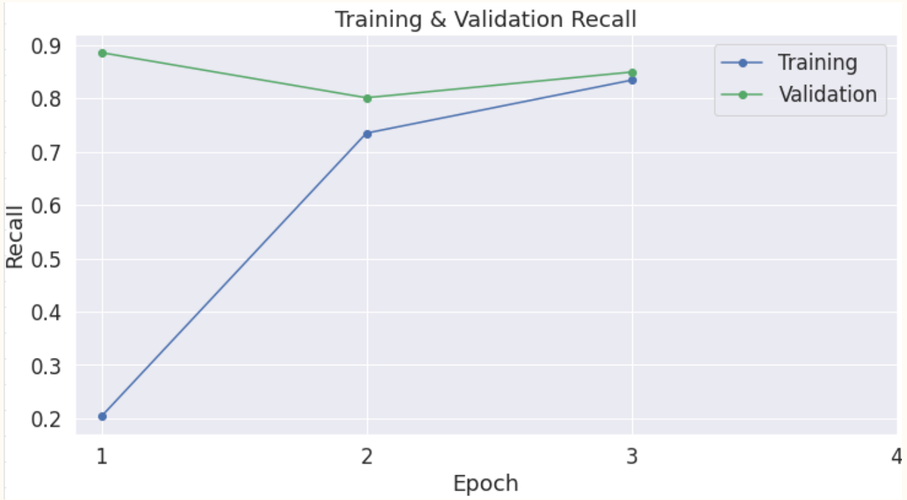
Back to Overview

epoch	Training Loss	Training Accur.	Training F1	Training Precision	Training Recall	Valid. Loss	Valid. Accur.	Valid. F1	Valid. Precision	Valid. Recall	Training Time	Validation Time
1	0.23	0.88	0.26	0.41	0.20	0.16	0.95	0.78	0.71	0.89	0:01:58	0:00:02
2	0.12	0.96	0.78	0.84	0.73	0.12	0.96	0.78	0.77	0.80	0:01:56	0:00:02
3	0.10	0.97	0.86	0.89	0.83	0.11	0.96	0.85	0.85	0.85	0:01:56	0:00:02

```
Running Testing...
[redacted]
=>Accuracy:  0.93
=>Precision: 0.73
=>Recall:    0.75
=>F1:        0.74
Test Loss: 0.09
Test took: 0:00:02
```

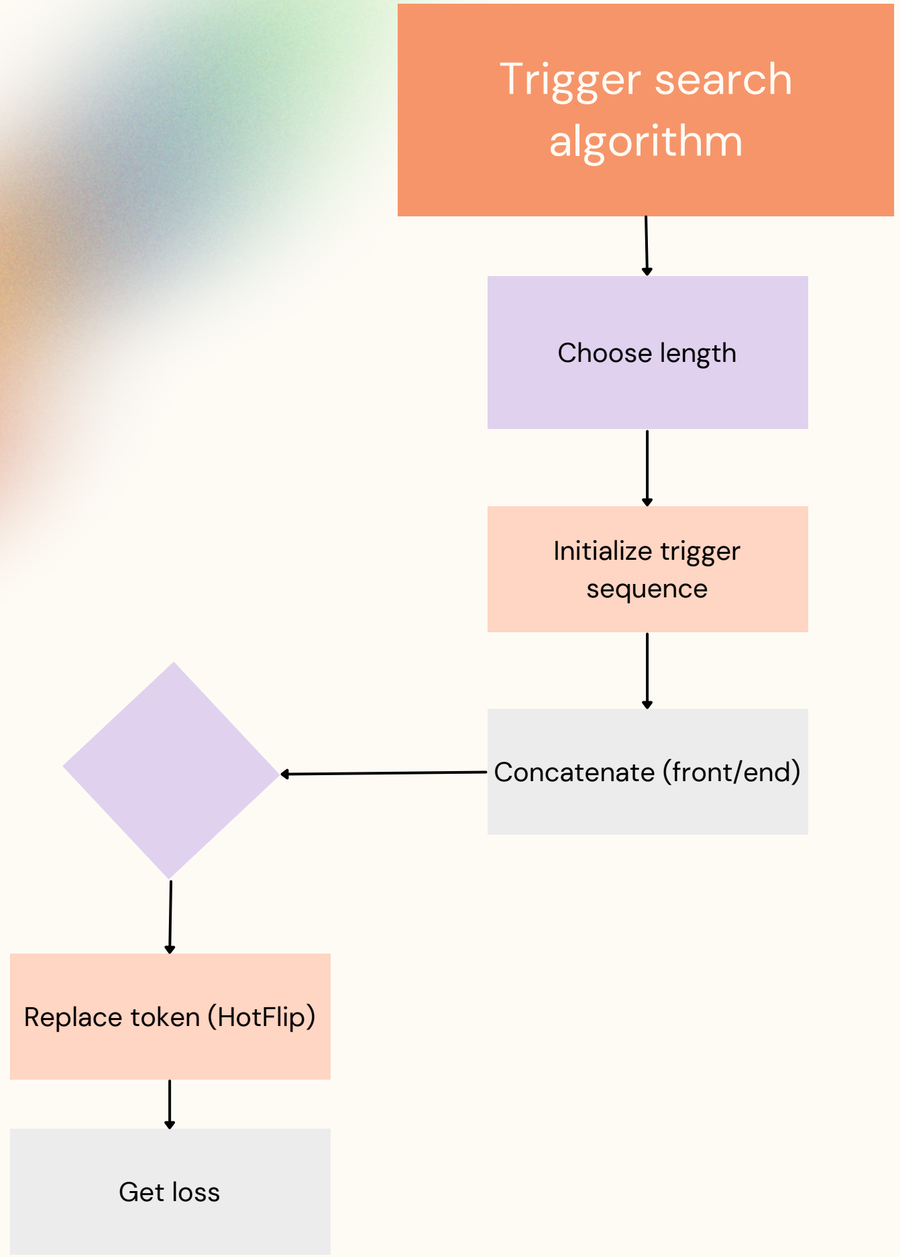


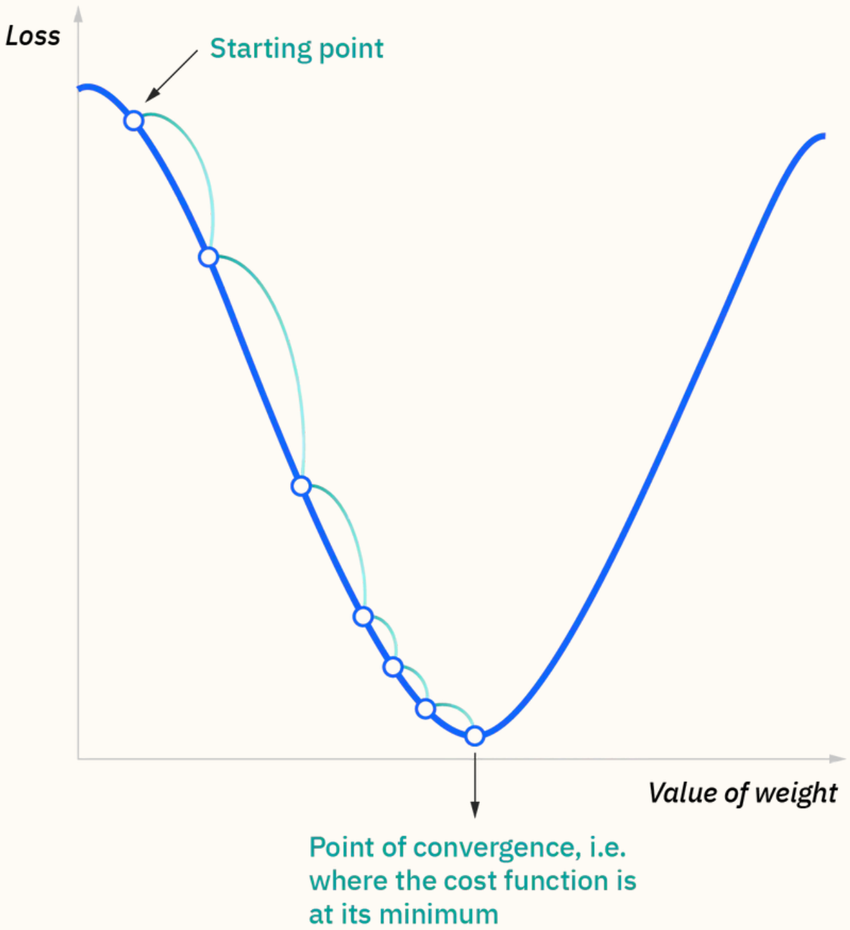




Back to Overview

Adversarial Attacks





Back to Overview

Original with special tokens

```
tensor([[  101, 12017,   179,  ...,    -1,    -1,    -1],
        [  101,   233,   223,  ...,    -1,    -1,    -1],
        [  101, 12017,   179,  ...,    -1,    -1,    -1],
                                ...,
        [  101,   206,  4313,  ...,    -1,    -1,    -1],
        [  101,   206,  4313,  ...,    -1,    -1,    -1],
        [  101,   218,  1260,  ...,    -1,    -1,    -1]])
```

Vector representation with "the the the" concatenated at the beginning

```
tensor([[  101, 207,  207,  ...,    -1,    -1,    -1],
        [  101, 207,  207,  ...,    -1,    -1,    -1],
        [  101, 207,  207,  ...,    -1,    -1,    -1],
                                ...,
        [  101, 207,  207,  ...,    -1,    -1,    -1],
        [  101, 207,  207,  ...,    -1,    -1,    -1],
        [  101, 217,  207,  ...,    -1,    -1,    -1]])
```

Iteration of the candidates in the first position

```
tensor([[  101, 7484,  207,  ...,    -1,    -1,    -1],
        [  101, 7484,  207,  ...,    -1,    -1,    -1],
        [  101, 7484,  207,  ...,    -1,    -1,    -1],
                                ...,
        [  101, 7484,  207,  ...,    -1,    -1,    -1],
        [  101, 7484,  207,  ...,    -1,    -1,    -1],
        [  101, 7484,  207,  ...,    -1,    -1,    -1]])
```

Iteration of the candidates in the second position

```
tensor([[  101, 621,  13890,  ...,    -1,    -1,    -1],
        [  101, 621,  13890,  ...,    -1,    -1,    -1],
        [  101, 621,  13890,  ...,    -1,    -1,    -1],
                                ...,
        [  101, 621,  13890,  ...,    -1,    -1,    -1],
        [  101, 621,  13890,  ...,    -1,    -1,    -1],
        [  101, 621,  13890,  ...,    -1,    -1,    -1]])
```

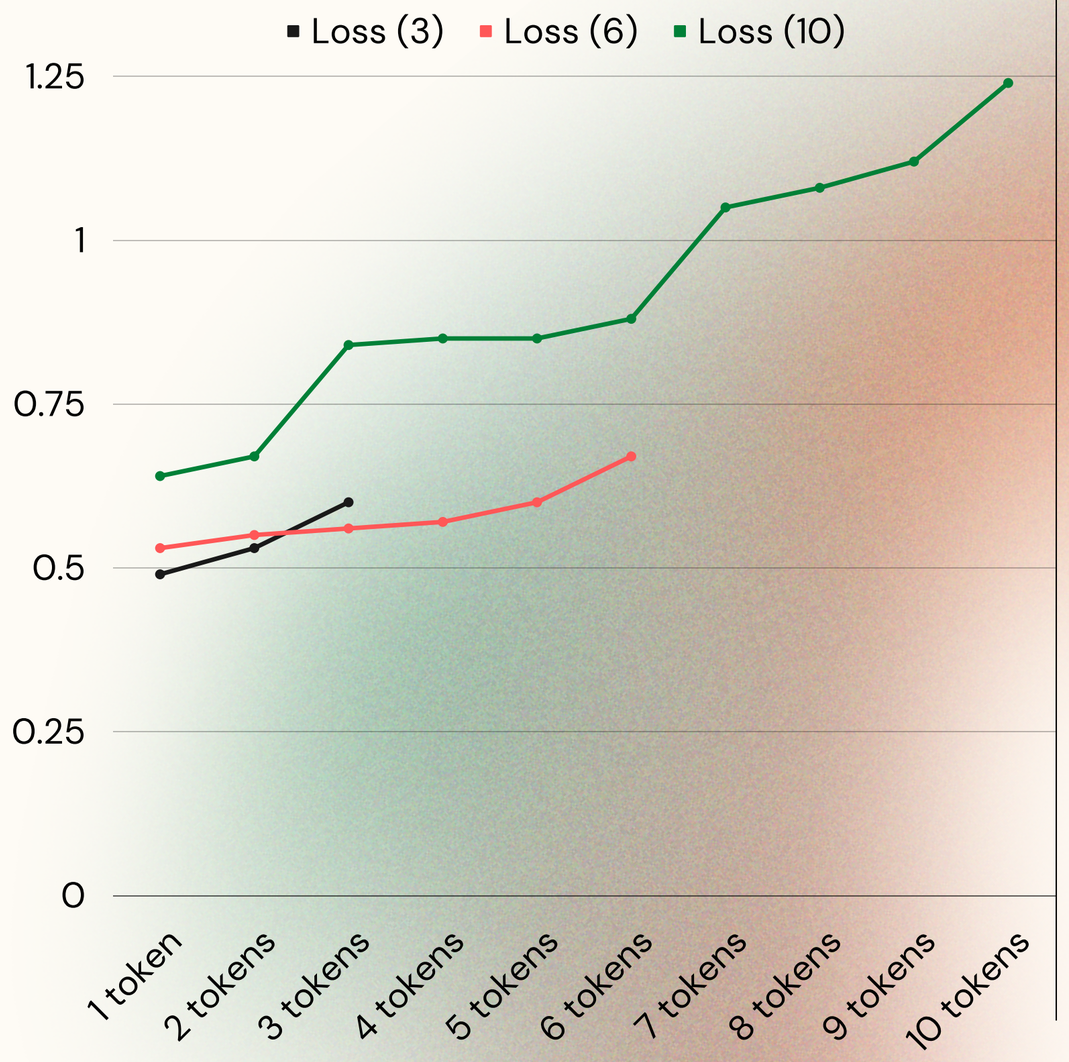
Back to Overview

Loss metric

[11858, 23347, 700] = telegram bnym him

[11858,9843,207,18086,1459] = telegram reprint acquitted the directorship how

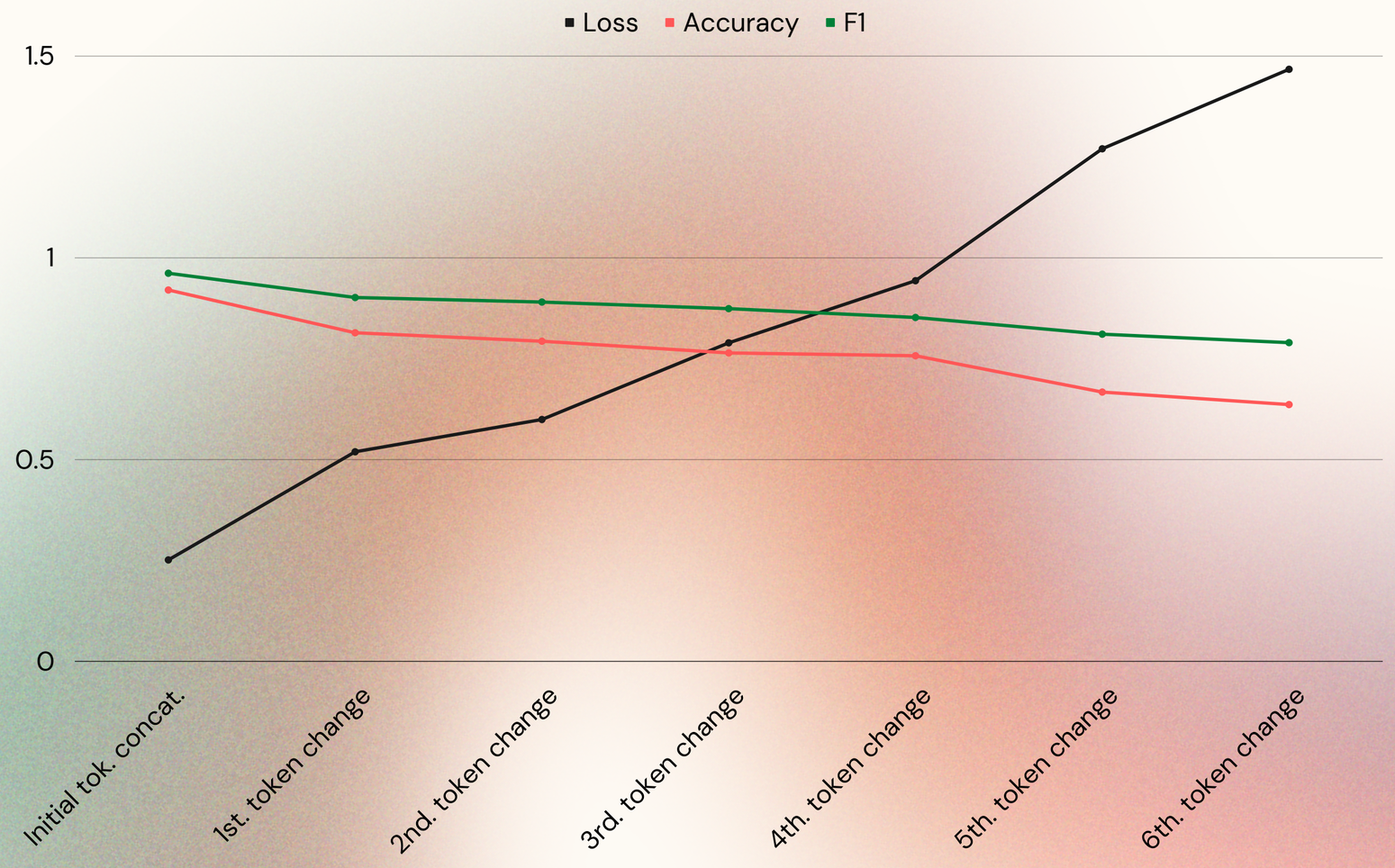
[11858, 6900, 5312, 4162, 207, 30162, 1227, 2659, 3756, 6339] = telegram naphth testify injunctive theenskap notify waive severability inducement



	Loss	
1 token	0.49	telegram the the
2 tokens	0.53	telegram bnym the
3 tokens	0.60	telegram bnym him

Back to Overview

Metrics with the whole dataset



Loss: unless communist normativ encroachments as anything
Accuracy: unless communist tolerate political dismissed disjunctive
F1: unless communist tolerate political dismissed symmetrical

Back to Overview

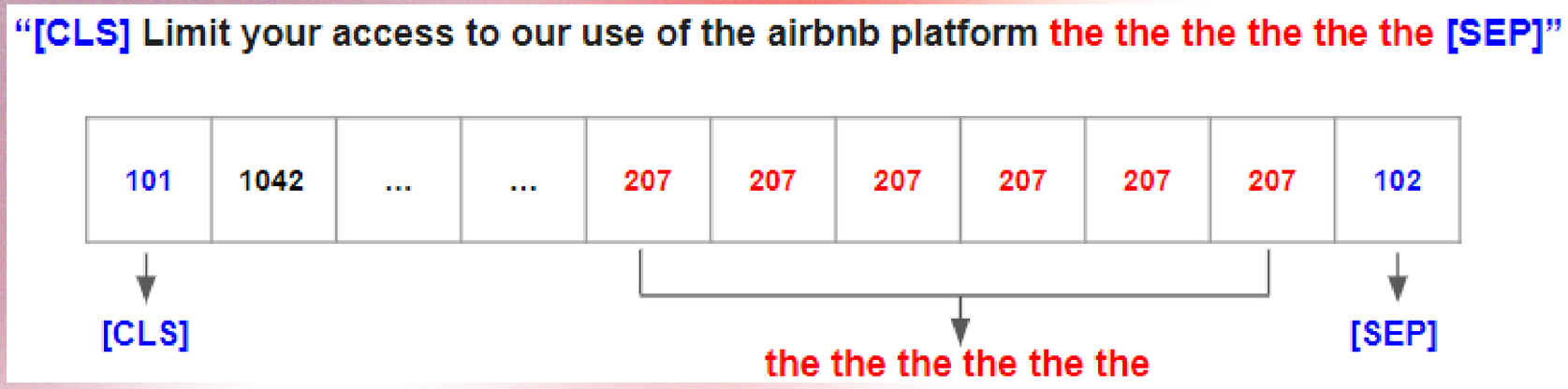
Back to Overview

Extension

Extension – Triggers at the end of the sentece

- Create triggers from the end of sentences

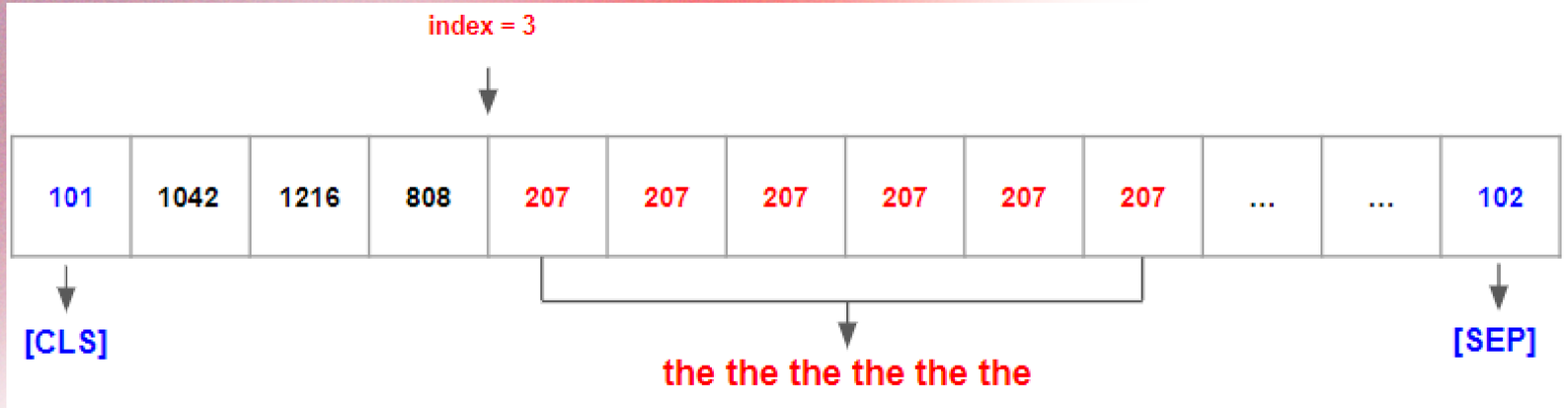
- Implementation
  - Tokenize the sentence
  - Tokenize the triggers
  - Add triggers at the end of tokenized sentence
  - Save triggers position



Extension – Triggers given a specific index

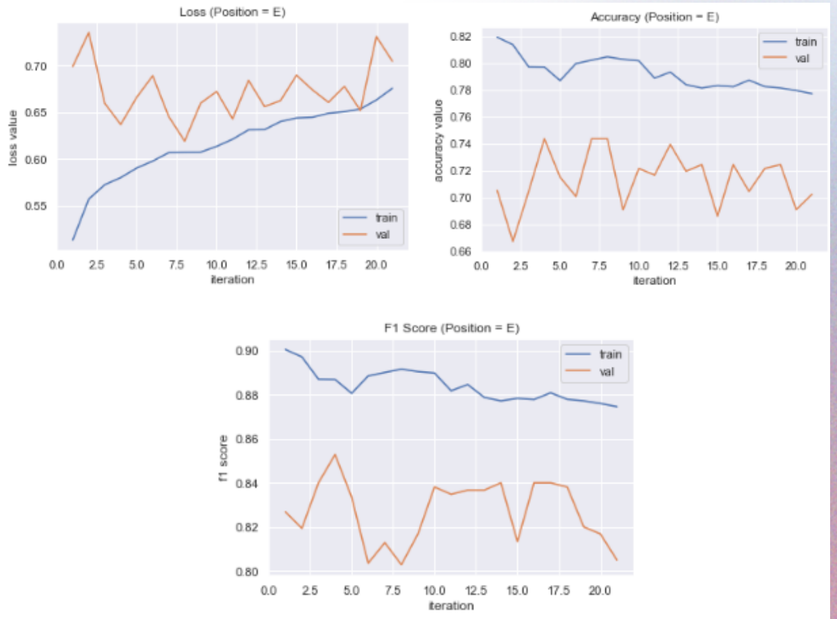
- Create triggers with a given index

- Implementation
  - Tokenize the sentence
  - Tokenize the triggers
  - Add triggers at given position
  - Check if index is out of range (i.e. short sentences)
  - Check if token overlap with [SEP] token
  - Save triggers position



Analysis of extension model

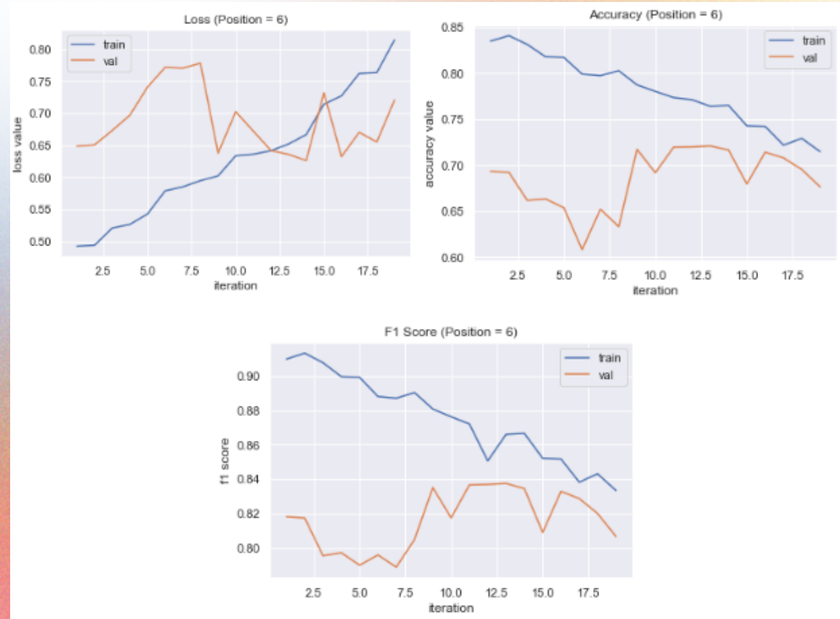
End of the sentence



	Precision	Recall	F1 Score
Fair	0.95	0.97	0.96
Unfair	0.71	0.55	0.62
Acc.			0.93

"assume defense differentiati failure compating advocate"

Given an index



	Precision	Recall	F1 Score
Fair	0.94	0.97	0.96
Unfair	0.69	0.50	0.58
Acc.			0.94

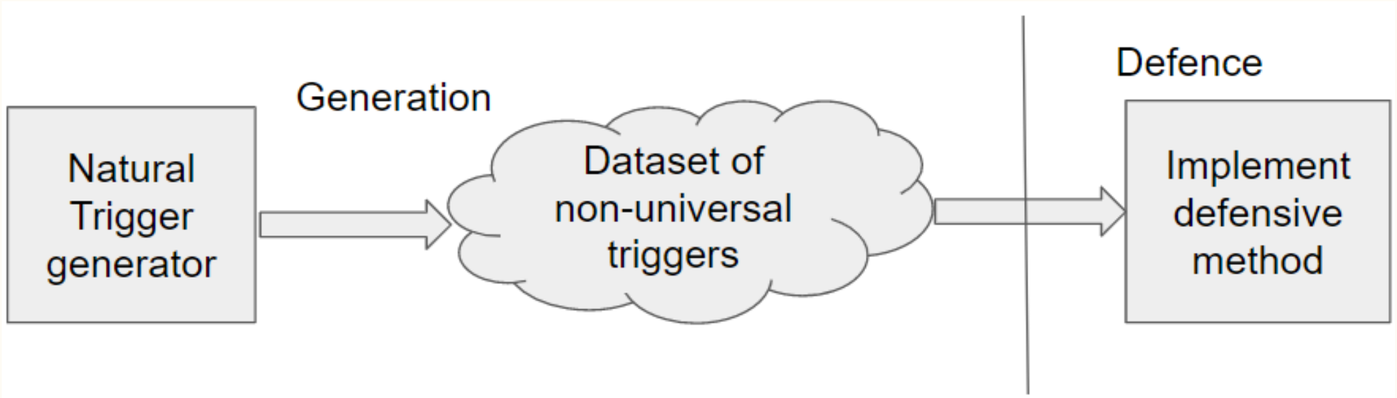
"capitation kirov testified scarcely closing comparator"

Back to Overview

Naturalness

Natural trigger generation

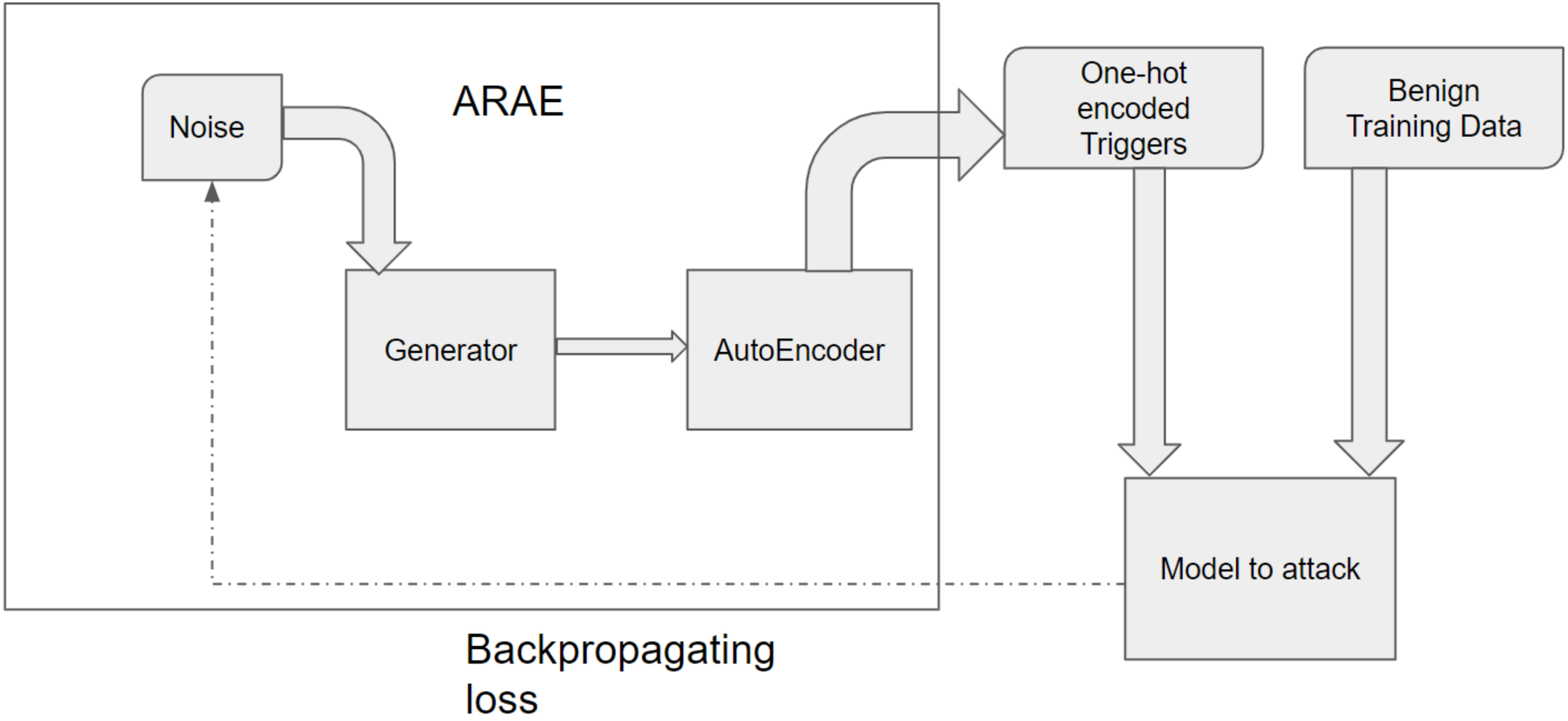
- Generating natural triggers with defence perspective in mind
- Generate multiple trigger sequences than one universal trigger



- We need a text generator that can generate semantically sound sentences.
- Adversarially Regularized Autoencoder
- Consists of an autoencoder and a generative adversarial network.
- The ARAE model is pre-trained on one billion word benchmark.

Back to Overview

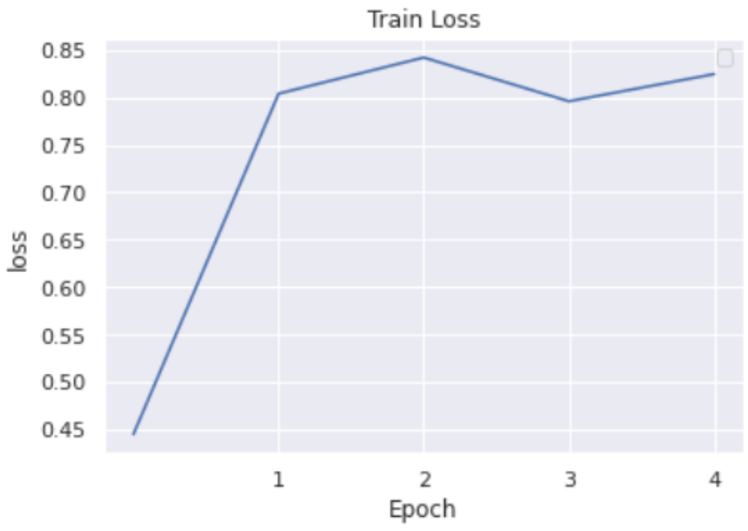
Generating natural triggers

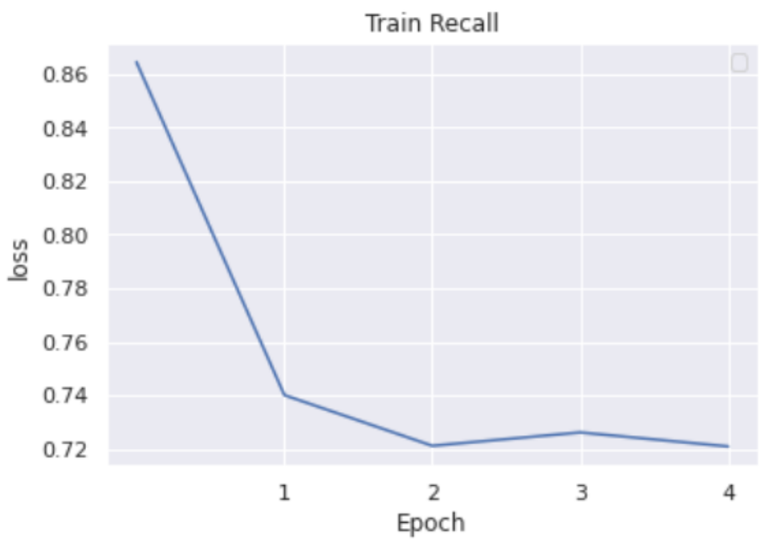


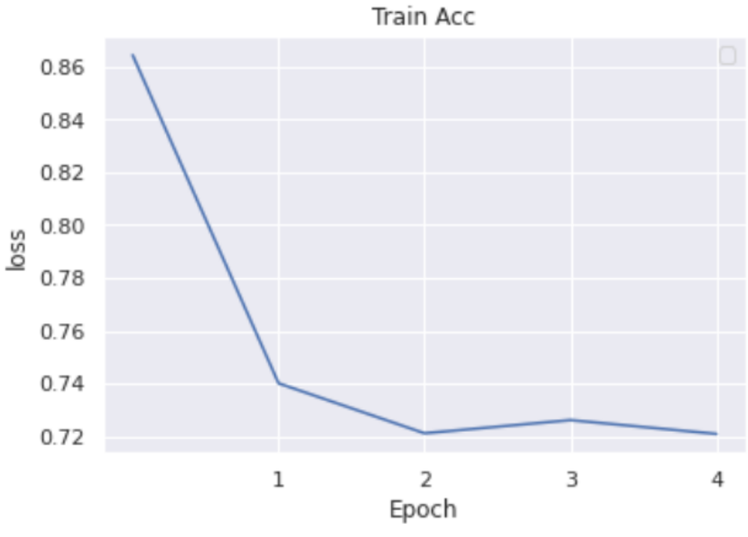
Generating natural triggers

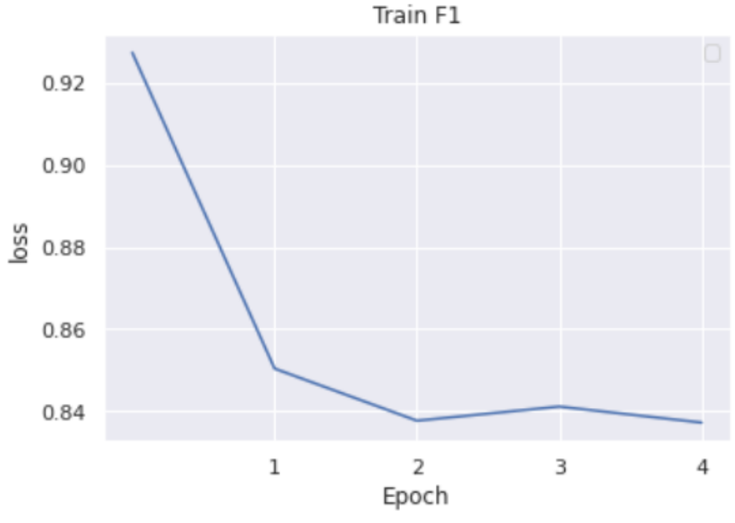
- Split dataset 60% train, 20% valid, 20% test
- Use train set to get model loss
- If current_loss >= last_loss, generate a new trigger
- Check performance of model in valid set
  - If under a threshold, discard
  - If over a threshold, accept
- Finally, Performance is tested over unseen test set

Back to Overview









Trigger sequence	Test Accuracy
this is a proposed	0.6976
the government will surely pass	0.6343
in future talks, the	0.6574
this award included in its	0.7248
other sites of information	0.7273

Back to Overview

Defense perspective

Perplexity

- Is a metric used for language models
- Describes how well an input is predicted by a probability distribution
- Also can be seen as how confused a model is, given the input

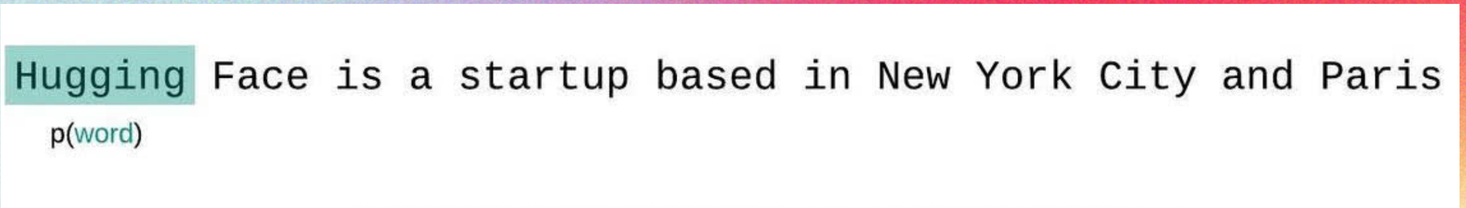
$$\text{Perplexity}(M) = M(s)^{-1/n}$$

$$= \sqrt[n]{\prod_{k=1}^{n} \frac{1}{M(w_k|w_0 w_1 \cdots w_{k-1})}}$$

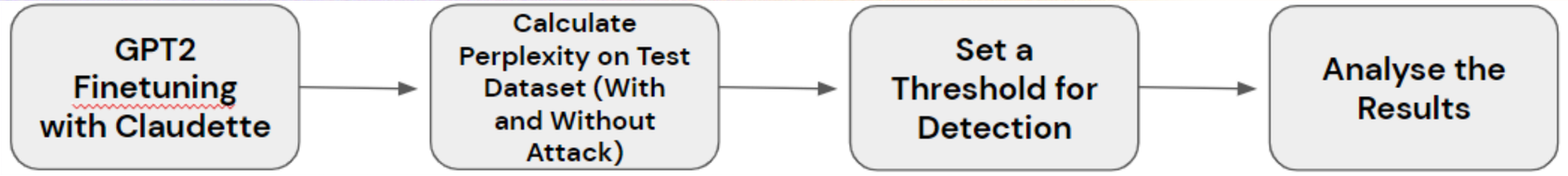
Back to Overview

Perplexity on non-natural triggers

Perplexity Calculation on non-natural triggers



- Applicable to classical language models, and not well defined for a masked model like BERT

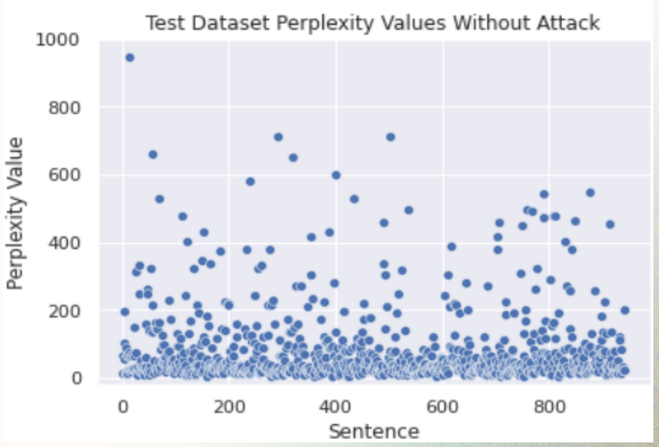


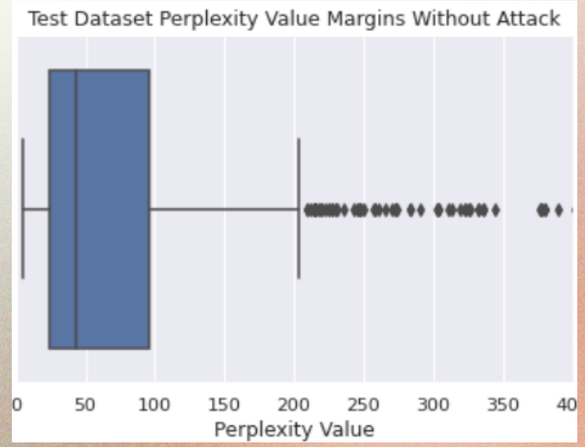
- Test dataset:
  - 942 samples
    - 839 fair
    - 103 unfair

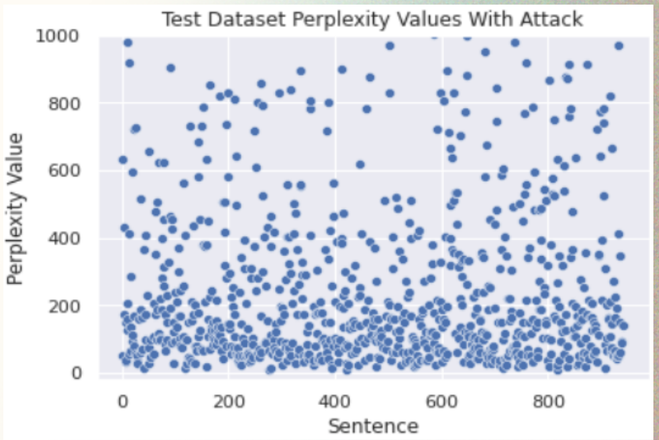
$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

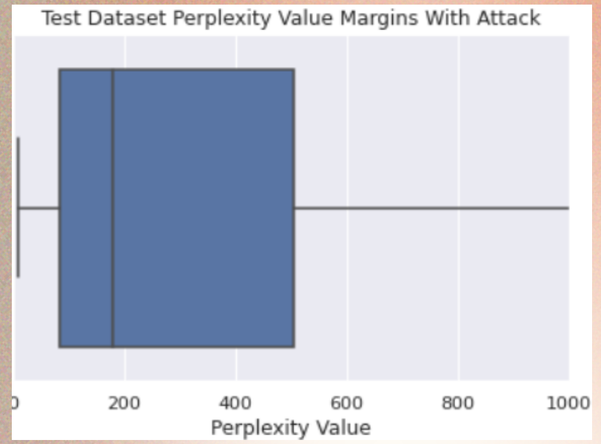
https://huggingface.co/docs/transformers/perplexity

Perplexity Calculation on Test Dataset









For attack, added 10 different triggers randomly to test dataset.

<u>Threshold</u>: 200

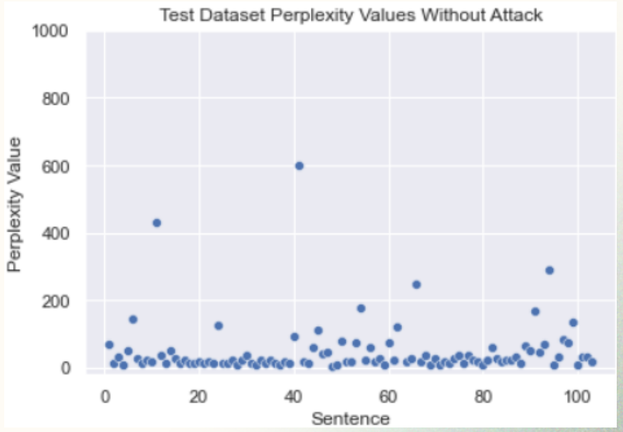
<u>Attack detection</u>:

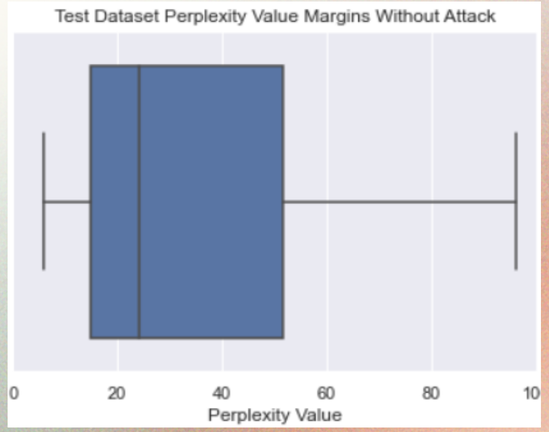
942 Samples -> 48.1% Detection

839 Fair -> 50.3% Detection
103 Unfair -> 30.1% Detection

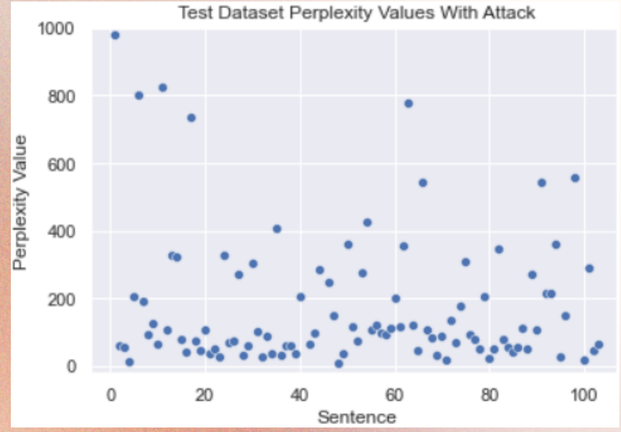
Perplexity Calculation on Unfair and Fair Samples

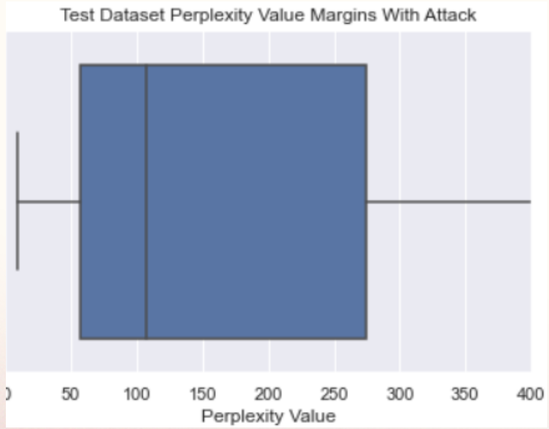
Unfair



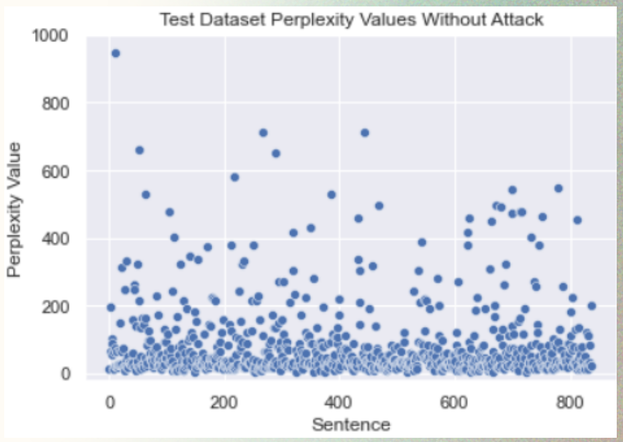


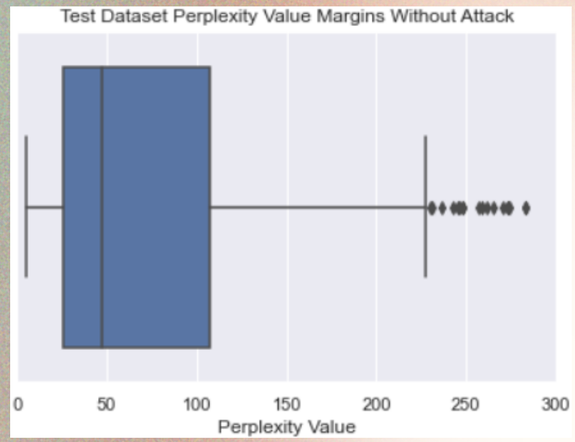
Unfair Attack



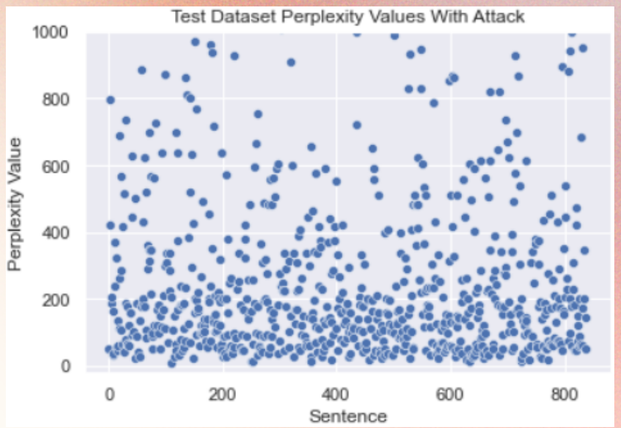


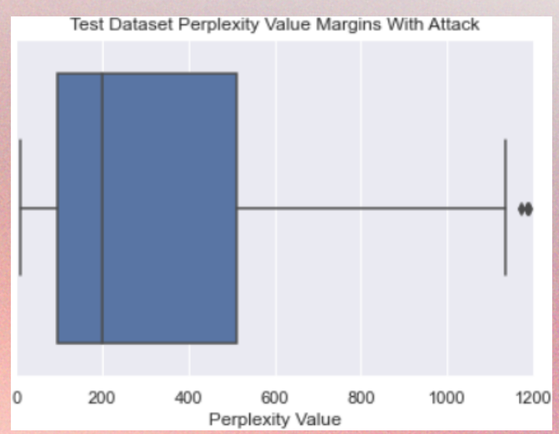
Fair





Fair Attack



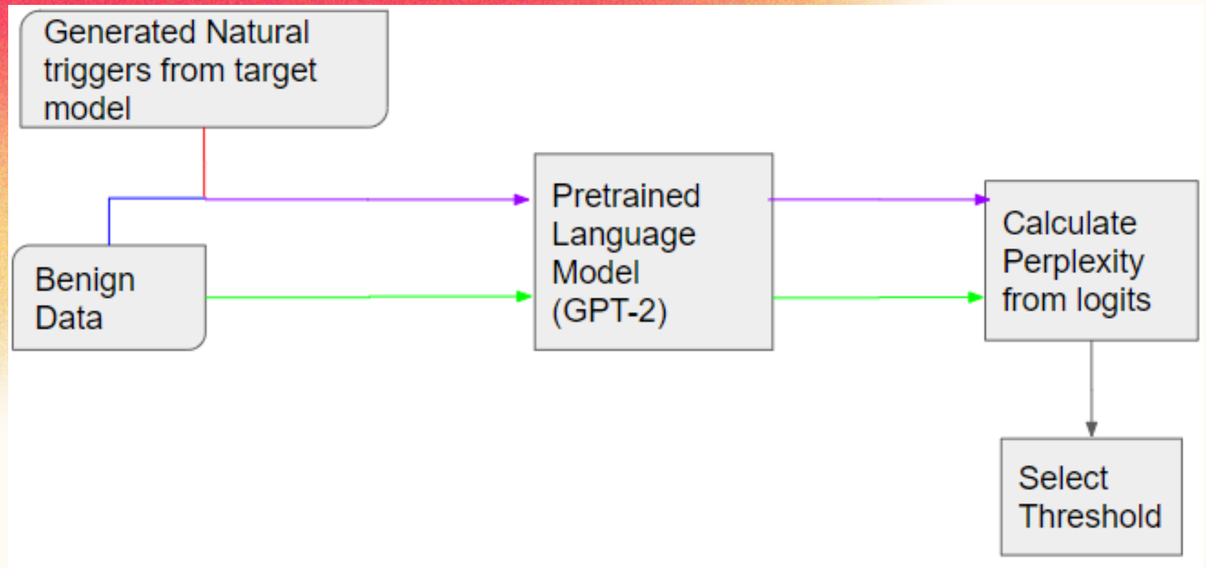


Back to Overview

Perplexity on natural triggers

Perplexity for Defence

- Pre-trained language models have :
  - Low perplexity when sample text is feasible
  - High perplexity if not
- Why GPT-2:
  - Perplexity well defined
  - Pretrained model weights openly available
- We use this idea to our advantage.
- Pre-trained GPT-2 to estimate perplexity of
  - Normal samples
  - Samples with added triggers



Back to Overview

Result on test set (20%)

Token Sequence Length	Perplexity
0 (no triggers)	133.98
3	329.01
4	310.21
5	271.76
6	212.19

Result on complete dataset (100%)

Token Sequence Length	Perplexity
0 (no triggers)	122.18
3	301.76
4	288.33
5	228.14
6	202.12

Back to Overview

Results

- Perplexity higher on unseen test set
  - If vocab is available, defence is more susceptible
- Decreases with sequence length
  - Trigger sequence themselves are natural
  - Perplexity is averaged down with the sequence length
- Longer triggers are inherently hard to use
  - Not stealthy
- Difference wide enough for a threshold : 150

Back to Overview

Back to Overview

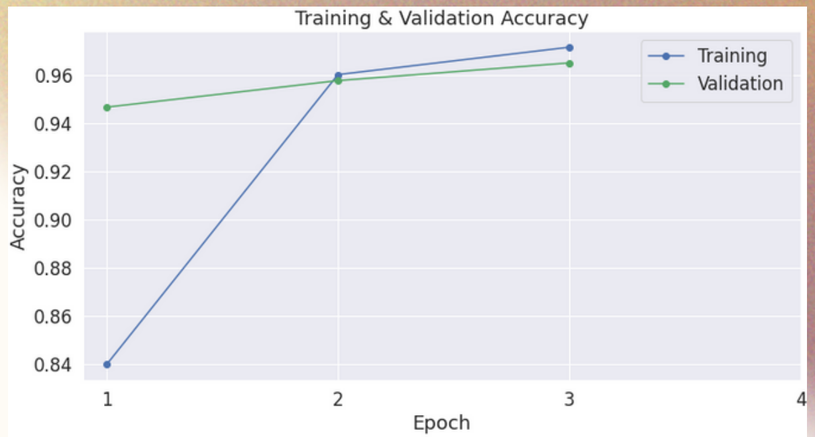
Model enhancement

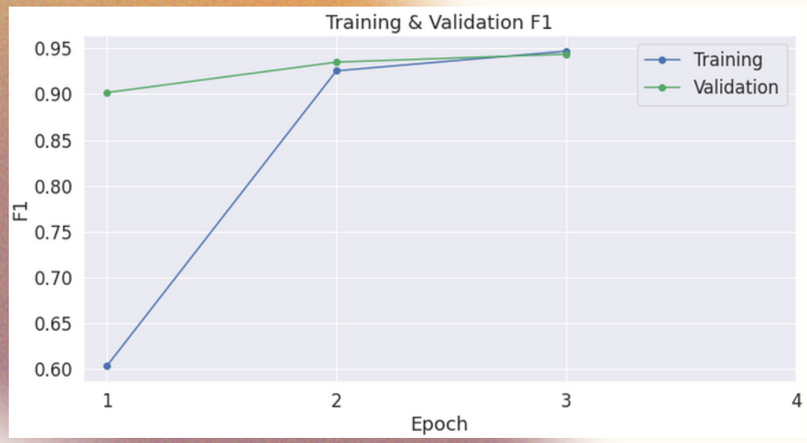
Baseline model

epoch	Training Loss	Training Accur.	Training F1	Training Precision	Training Recall	Valid. Loss	Valid. Accur.	Valid. F1	Valid. Precision	Valid. Recall	Training Time	Validation Time
1	0.23	0.88	0.26	0.41	0.20	0.16	0.95	0.78	0.71	0.89	0:01:58	0:00:02
2	0.12	0.96	0.78	0.84	0.73	0.12	0.96	0.78	0.77	0.80	0:01:56	0:00:02
3	0.10	0.97	0.86	0.89	0.83	0.11	0.96	0.85	0.85	0.85	0:01:56	0:00:02

Model trained with tokens at the beginning

epoch	Training Loss	Training Accur.	Training F1	Training Precision	Training Recall	Valid. Loss	Valid. Accur.	Valid. F1	Valid. Precision	Valid. Recall	Training Time	Validation Time
1	0.24	0.84	0.60	0.81	0.50	0.13	0.95	0.90	0.92	0.89	0:02:18	0:00:02
2	0.11	0.96	0.93	0.95	0.90	0.12	0.96	0.93	0.95	0.92	0:02:18	0:00:02
3	0.08	0.97	0.95	0.96	0.93	0.10	0.97	0.94	0.96	0.92	0:02:18	0:00:02





Back to Overview

Unseen dataset

Baseline model

```
=>Accuracy:  0.96
=>Precision: 0.79
=>Recall:    0.86
=>F1:        0.82
Test Loss: 0.09
Test took: 0:00:02
```

Model trained with three tokens

```
=>Accuracy:  0.98
=>Precision: 0.92
=>Recall:    0.82
=>F1:        0.86
Test Loss: 0.08
Test took: 0:00:02
```

Unseen dataset with three tokens concatenated at the beginning of the unfair sentences

Baseline model

```
=>Accuracy:  0.97
=>Precision: 0.87
=>Recall:    0.83
=>F1:        0.85
Test Loss: 0.11
Test took: 0:00:43
```

Model trained with three tokens

```
=>Accuracy:  0.97
=>Precision: 0.91
=>Recall:    0.79
=>F1:        0.85
Test Loss: 0.10
Test took: 0:00:44
```

Unseen dataset with six tokens concatenated at the beginning of the unfair sentences

Baseline model

```
=>Accuracy:  0.96
=>Precision: 0.86
=>Recall:    0.79
=>F1:        0.82
Test Loss: 0.11
Test took: 0:01:23
```

Model trained with six tokens

```
=>Accuracy:  0.93
=>Precision: 0.86
=>Recall:    0.49
=>F1:        0.63
Test Loss: 0.20
Test took: 0:01:24
```

Back to Overview

Learnings and Conclusions

Rafid

- Pytorch
- HuggingFace
- Autoencoders and GANs
- BERT and transformers

Elif

- Need to analyze the data carefully
- Data imbalance issue
- Token positions are important
- Perplexity is a good approach but not best
- Make the model robust in the training part

Isaac

- Impact of attacks
- Security implications
  - Attack perspective
  - Defense perspective. (*Always one step behind)*
    - Shield
    - Response
    - Countermeasures
- Learning curve
  - Collect blog, data, tutorials
- Re-implementation of algorithms
- Dev. of NLP applications with good results is not enough

Back to Overview

Back to Overview

Future job

Future Job

- Empirical analysis of natural/non-natural triggers and effect on LM perplexity
  - Various positions
  - Various lengths
- Generate adversarial dataset using reinforcement learning
- Re-implementation of Defense against Adversarial Attacks in NLP via Dirichlet Neighborhood Ensemble
- Transferability

Back to Overview

Back to Overview

References

References

Back to Overview

- Ebrahimi, Javid and Rao, Anyi and Lowd, Daniel and Dou, Dejing, `"HotFlip: White-Box Adversarial Examples for Text Classification", in Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 31-36
- Wallace, Eric and Feng, Shi and Kandpal, Nikhil and Gardner, Matt and Singh, Sameer, "Universal Adversarial Triggers for Attacking and Analyzing NLP", in Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2153-2162
- Song, Liwei and Yu, Xinwei and Peng, Hsuan-Tung and Narasimhan, Karthik, "Universal Adversarial Attacks with Natural Triggers for Text Classification", in Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 3724-3733
- Marco Lippi and Przemyslaw Palka and Giuseppe Contissa and Francesca Lagioia and Hans-Wolfgang Micklitz and Giovanni Sartor and Paolo Torroni, "CLAUDETTE: an Automated Detector of Potentially Unfair Clauses in Online Terms of Service", in Artif. Intell. Law 27 (2019) 117-139
- Yi Zhou and Xiaoqing Zheng and Cho-Jui Hsieh and Kai-Wei Chang and Xuanjing Huang, "Defense against Adversarial Attacks in NLP via Dirichlet Neighborhood Ensemble", in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 5482–5492, 2021
- Neekhara, P., Hussain, S., Dubnov, S. and Koushanfar, F "Adversarial Reprogramming of Sequence Classification Neural Networks", 2018

Back to Overview

Q&A Session

Thank you for listening!

Overview Appendices

- Adversarial Reprogramming
- Examples of unfair sentences
- Distribution of just unfair sentences
- Distribution of types of unfair sentences
- Reproduction progress
- Defense Against Adversarial Attacks in NLP via DNE

Back to Overview

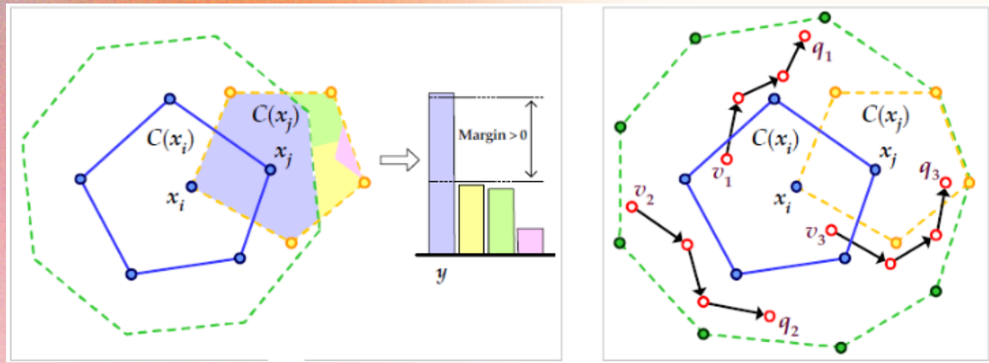
Adversarial Reprogramming

Back to Overview

- Repurpose the target model to predict labels as wanted.
- Transform input space to target models input space.
- Remap labels from the output of the target model into initial input space.
- Black box attack.
- Can be used to generate adversarial triggers
  - Use RL algorithm to learn policy that returns adversarial tokens
  - Insert tokens from the vocab that fools the target model.

Defense Against Adversarial Attacks in NLP via Dirichlet Neighborhood Ensemble (DNE)

- Making the model robust during the training phase to defense substitution-based attacks
- In the training, DNE creates virtual sentences using sampled embedding vectors which represents the synonyms of each word in an input sentence.
- Synonym words are collected via spanned convex hull
- Gradient-guided optimization on convex hull
- This approach outperforms the methods fixing the word vectors during the training or updating individual word vectors independently.



Back to Overview

Example of unfair sentences

Back to Overview

Facebook

by using or accessing the facebook services , you agree to this statement , as updated from time to time in accordance with section 13 below .

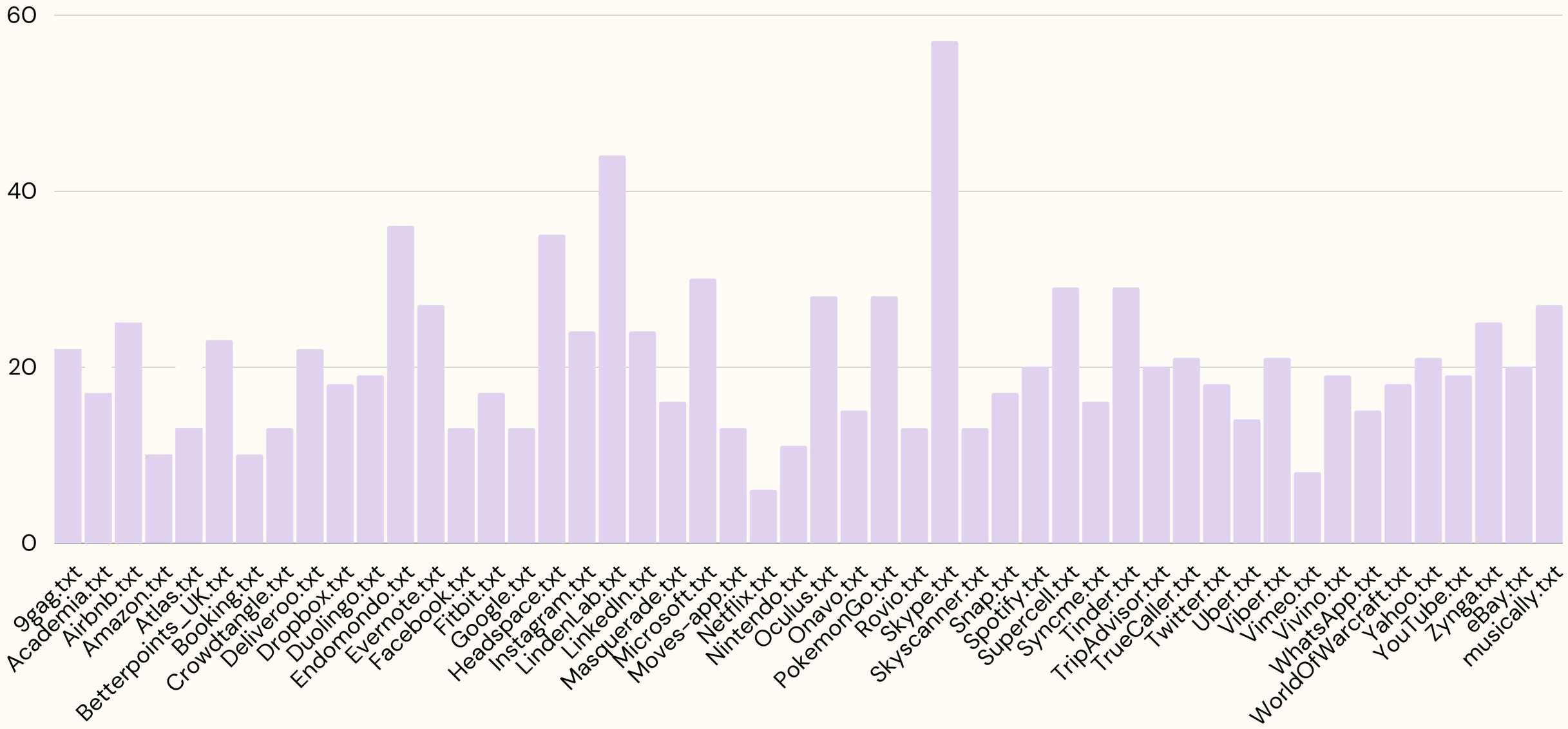
Netflix

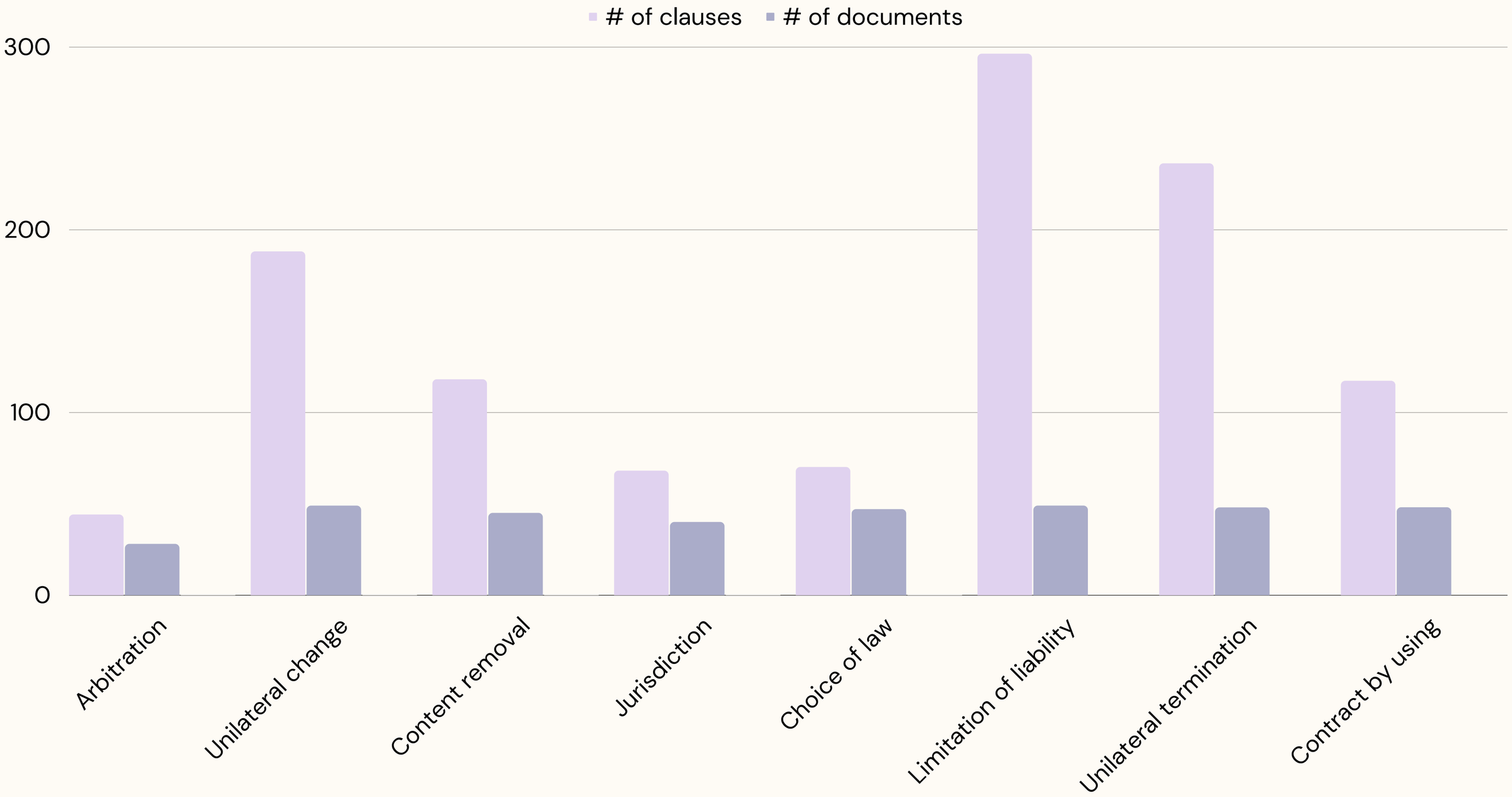
these terms of use shall be governed by and construed in accordance with the laws of the netherlands .

Skype

skype reserves the right to change the fair usage policy at any time .

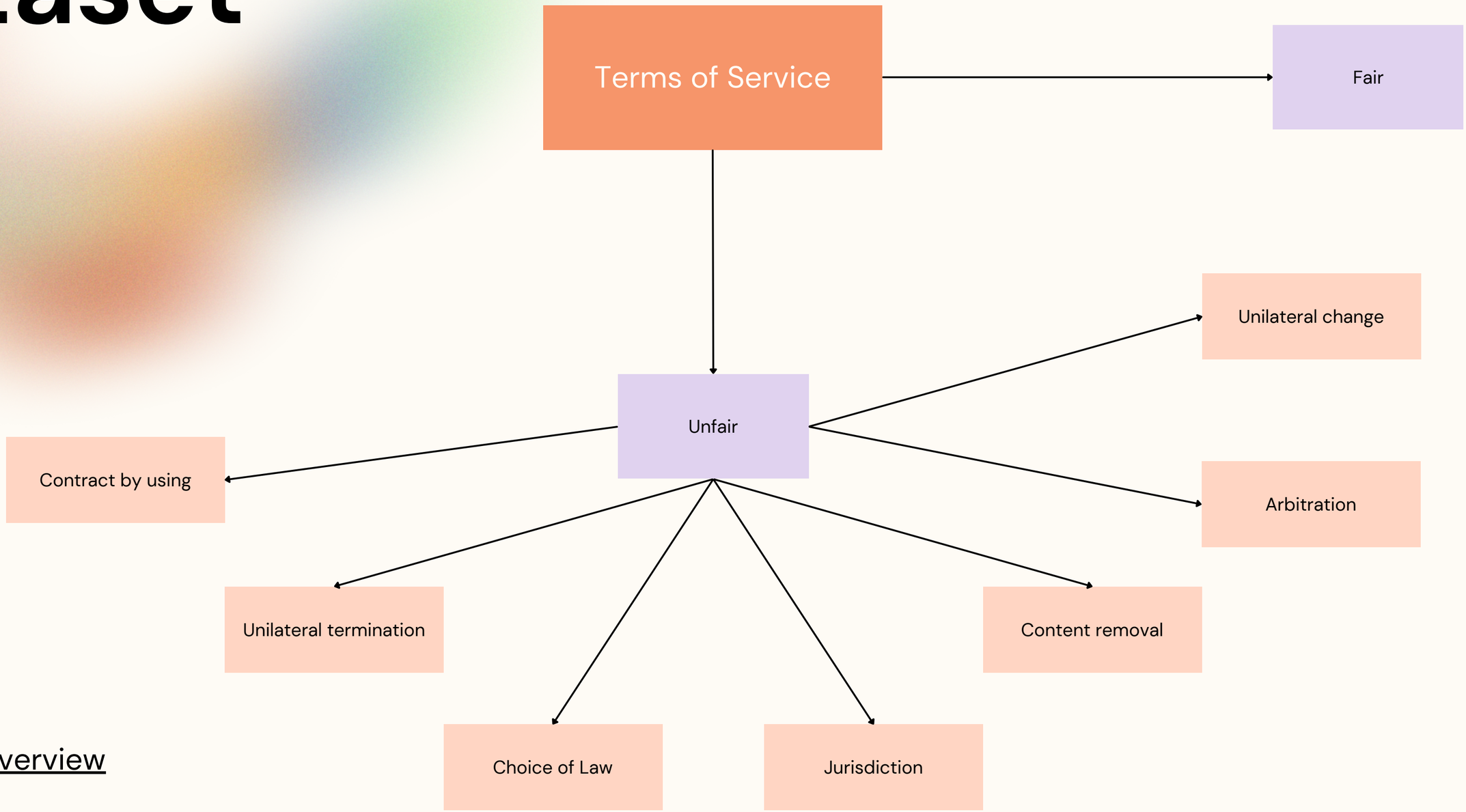
Distribution of unfair sentences among files





Back to Overview

Dataset



Back to Overview

Categories of Adversarial Attacks

01 BY MODEL ACCESS

- White-box attack
- Black-box attack

02 BY SEMANTIC APPLICATION

- Single modal
- Cross modal

- Classification, machine translation, machine comprehension, text summarization, text entailment, POS tagging, relation extraction, dialogue system.
- Optical character recognition (OCR), scene text recognition, image captioning, visual question answering, visual-semantic embedding, and speech recognition.

03 BY TARGET TYPE

- Targeted
- Untargeted

04 BY SEMANTIC GRANULARITY

- Character-level
- Word-level
- Sentence-level

05 BY ATTACKED DNN

- Feed forward
- Reinforcement learning models
- Generative adversarial networks
- Hybrids models

- CNNs
- RNNs
- Auto-Encoder

Back to Overview

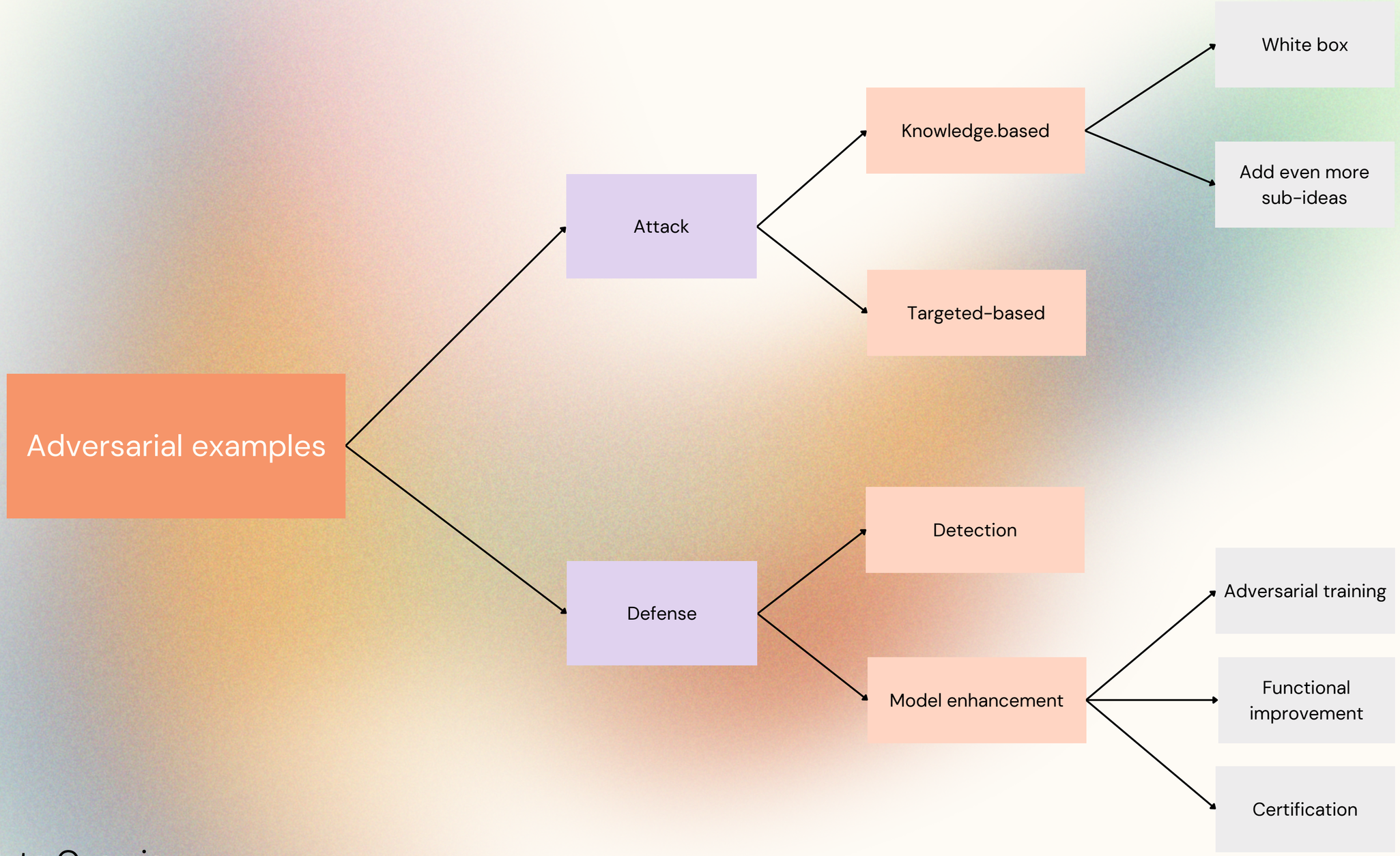
Defense

How do universal adversarial triggers help?

Useful to

- Expose vulnerabilities and thus security implications.
- Evaluate and interpret.
- Facilitate interpretation.

Back to Overview



Back to Overview

"Towards a Robust Deep Neural Network in Texts: A survey", Wang, W. et al.

Defense methods

Detection of misspelling

- Word
- Character

Functional improvement

Naturalness

- Perplexity

Challenges:

- Length of triggers
- Position (B|E|R)
- Imbalance of classes

Back to Overview