Generate BA thesis with GPT

Search

# RuATD

Anya Aksenova
Katya Taktasheva

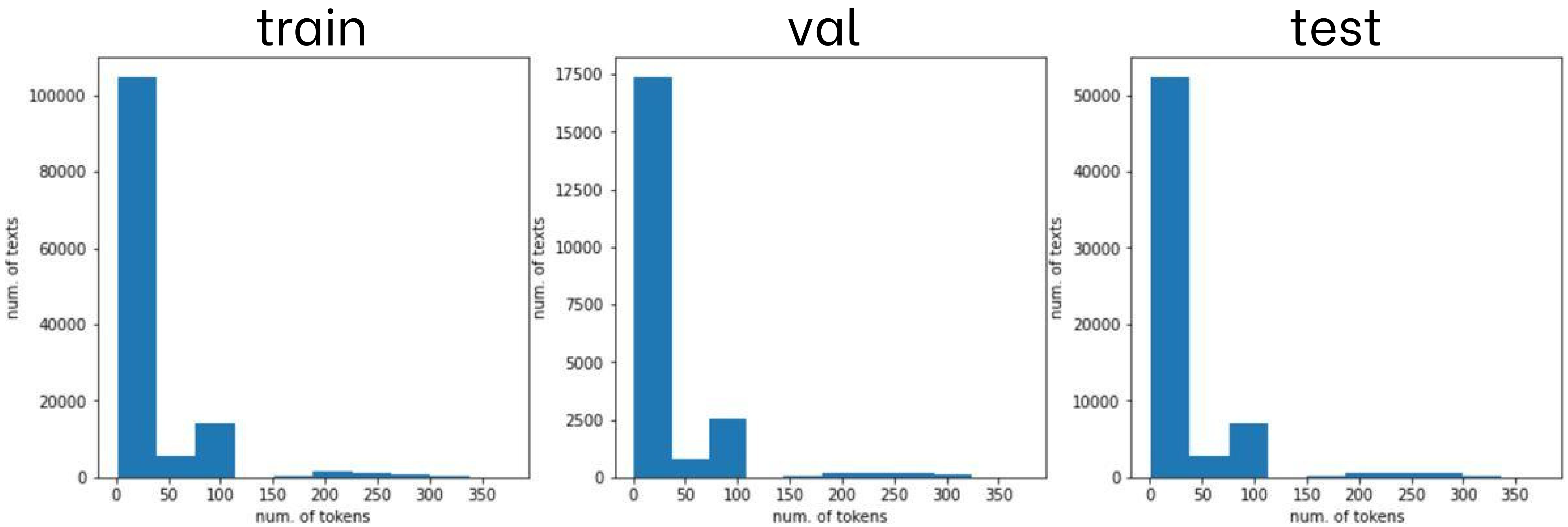
# RuATD challenge

How this game works

- Generated text VS Human written text
- Balanced classes
- Fine-Tuned RuBERT and logreg over tf-idf baseline
- Accuracy

# Exploratory Data Analysis

# Distribution of sentence lengths




| train | val | test |
|---|---|---|
| min = 1 | min = 1 | min = 1 |
| mean = 30.96 | mean = 30.97 | mean = 30.99 |
| max =376 | max =360 | max =374 |

| | | | |
|---|---|---|---|
| **6484** | 10876 | супруга______________________________________... | H |
| **48920** | 81476 | zakupki.zapis.org/doc/148462-zakopis-dnya-chel... | M |
| **64899** | 108146 | дети:__________________________________________... | H |
| **72562** | 120881 | http://www.entlebucher-anzeiger.ch/2014/06/ski... | H |
| **79372** | 132271 | ===Census/index | M |
| **79649** | 132746 | П.Д.ФРИЗЕН | H |
| **89753** | 149491 | http://www.transportaward.com/index.php/histor... | H |
| **90038** | 149955 | дети:__________________________________________... | H |
| **91810** | 152888 | president.go.kr/ | H |
| **99406** | 165658 | ttp://twitter.com/TrumpTramp/status/1025219074... | M |
| **99450** | 165752 | тыс.кв.м. | H |

**Length distribution (in tokens) by class:**

Train:
H mean = 30.07
M mean = 31.85

Val:
H mean = 30.08
M mean = 31.86

## Readability Metrics

| | H | M |
|---|---|---|
| dale_chall_readability_score | 20.13 | 20.43 |
| flesch_reading_ease | 90.49 | 88.83 |
| gunning_fog | 11.78 | 12.35 |
| text_standard | 0 | 0 |

# Solutions

1

# Feature-based

1. **Features**
    a. LEX: readability + diversity + lexical richness
    b. LEX + TF-IDF N-grams (2-3)
2. **Models**
    a. LogisticRegression
    b. KNN
    c. RandomForest

| Model | Acc | LEX + N-grams |
|---|---|---|
| LogReg | 0.55 | 0.62 |
| KNN | 0.59 | - |
| RandomForest | 0.63 | 0.64 |
| BERT-baseline | | 0.79622 |
| TF-IDF baseline | | 0.63562 |

1

## NN-based

1. Transfer learning
    a. RuRoberta
    b. Cointegrated RuBERT-tiny
2. Fine-tuning
    a. RuGPT3small
    b. Cointegrated RuBERT-tiny
3. Custom arhitechure
    a. CNN + LSTM
    b. CNN + LSTM + attention

| Model | Acc |
|---|---|
| ruRoBERTa TL | 0.56 |
| RuBERT TL | 0.62 |
| RuBERT FT | 0.50 |
| RuGPT3 FT | 0.79 |
| CNN-LSTM1 | 0.67 |
| CNN-LSTM2 | 0.67 |
| CNN-LSTM + Attention | 0.68 |
| BERT-baseline | 0.79622 |
| TF-IDF baseline | 0.63562 |

# Solutions 2.0

## Going Deeper...

## ruTS Readability

- Тест Флеша-Кинкайда
- Индекс удобочитаемости Флеша
- Индекс Колман-Лиау
- Индекс SMOG
- Автоматический индекс удобочитаемости
- Индекс удобочитаемости LIX

## ruTS Lexical Diversity

- Root Type-Token Ratio
- Corrected Type-Token Ratio
- Herdan Type-Token Ratio
- Summer Type-Token Ratio
- Mass Type-Token Ratio
- Dugast Type-Token Ratio
- Moving Average Type-Token Ratio
- ...

## Most important featues

’flesch_kincaid_grade’,
’flesch_reading_easy’,
’coleman_liau_index’,
’automated_readability_index’,
’lix’, ’dttr’, ’mtld’, ’mamtld’,
’simpson_index’, ’hapax_index’

**Selected based on the difference between H and M**

# Data processing tricks

What else??

# Back-translation

Train dataset [H] (ru)

Helsinki_NLP ru-en

Train dataset [H] (en)

Helsinki_NLP en-ru

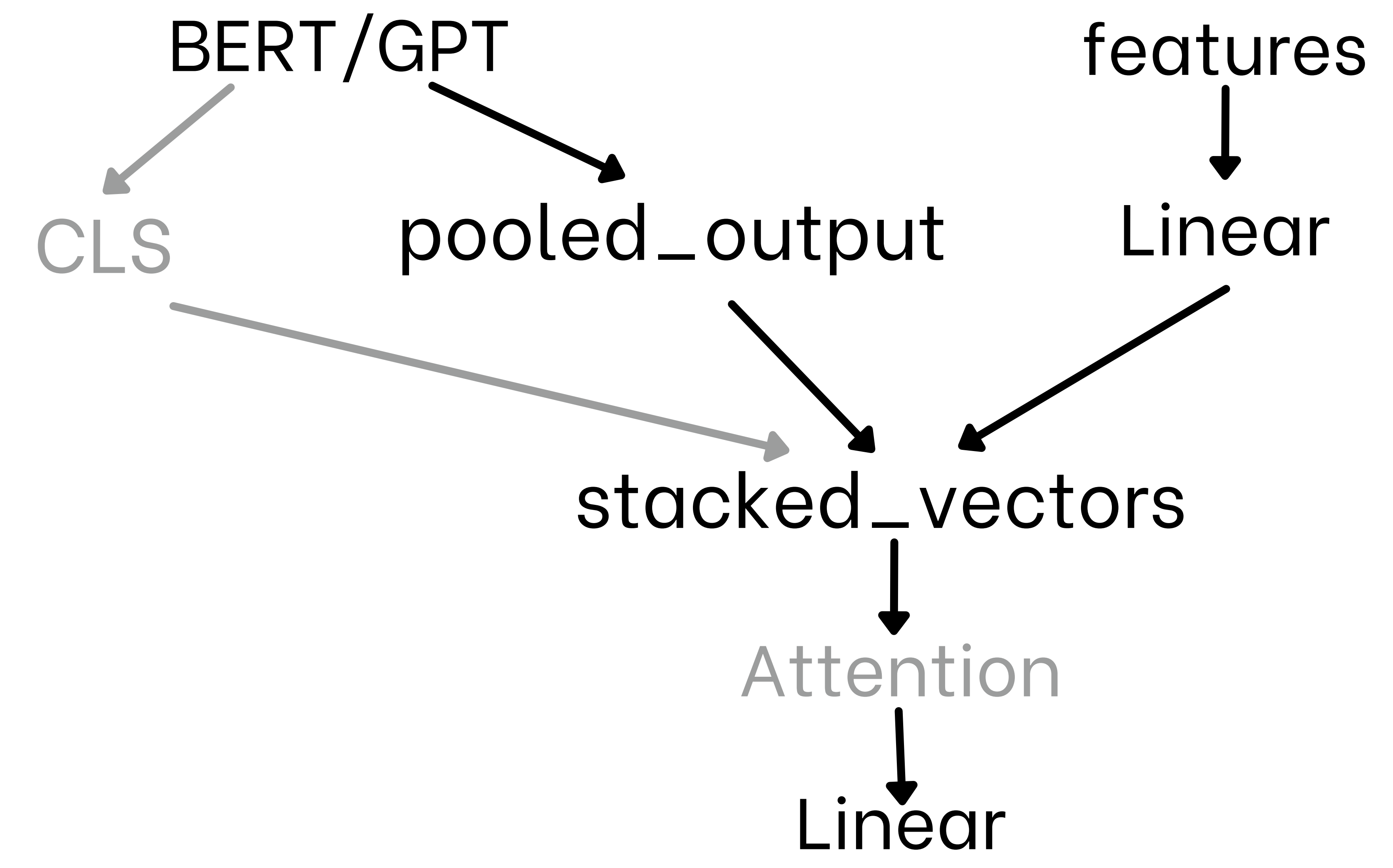
Train dataset [M] (ru)

What else??

# Sentence Clip

```
If len(text) == 1 sent:
    use it
else:
    use text[1] + text[-1]
```

# Architectures



1

## NN-based

1. Transfer learning
    a. RuRoberta
    b. Cointegrated RuBERT-tiny
2. Fine-tuning
    a. RuGPT3small
    b. Cointegrated RuBERT-tiny
3. Custom arhitechure
    a. CNN + LSTM
4. Features/Clips/BackTranslation

| Model | Acc |
|---|---|
| RuRoBERTa TL | 0.56 |
| RuBERT TL | 0.62 |
| RuBERT FT | 0.50 |
| **RuGPT3 FT** | **0.79** |
| CNN+LSTM1 | 0.67 |
| CNN+LSTM+Attention | 0.68 |
| CNN FEATS | 0.66 |
| GPT3 FEATS 10000 | 0.72 |
| GPT3 FEATS | 0.74 |
| GPT3+Attention FEATS | 0.59 |
| GPT3-clip | 0.73 |
| GPT3-clip FEATS | 0.74 |
| MBERT-clip FEATS | 0.77 |
| MBERT-clip-192 FEATS | 0.77 |
| **MBERT CLS FEATS** | **0.78** |
| GPT3 BT | 0.59 |
| BERT-baseline | 0.79622 |
| TF-IDF baseline | 0.63562 |

# Conclusions

# Feature-Engineering is all you need

Who?

# Roles

**Katya:** feature-based models

**Anya:** NN architectures

P.S. *But in the end everything got mixed up...*

Links

https://github.com/aaaksenova/RuATD_katana

# References


1. *Automatic Detection of Machine Generated Text: A Critical Survey* (Jawahar 2020)
    a. An overview of NN-based methods for ATD
    b. RoBERTa (TweepFake) mistakes analysis
2. *Computer-Generated Text Detection Using Machine Learning: A Systematic Review* (Beresneva 2016)
    a. Use of lexicographical and statistical features for ATD
3. *Defending Against Neural Fake News* (Zellers et al. 2019)
    a. GROVER – model for text generation and ATD
    b. GPT2 model as discriminator used for text classification
4. *Giant Language model Test Room* (Gehrmann 2019)
    a. Use GPT to detect text generated by GPT
    b. Frequency analysis is important