NLP, ML and DL for recognition of consumer-abusive clauses as a real-life application of computational law

Michalina Skibicka, ICM UW

Consumer-abusive clauses

What’s that?

- Defined by Uokik: https://decyzje.uokik.gov.pl/
- All clauses abusing consumer laws or unfair
- Divided into 6 categories
- No. clauses: 7091, ca. 50% labeled
- V. long - avg. length = 1023 tokens

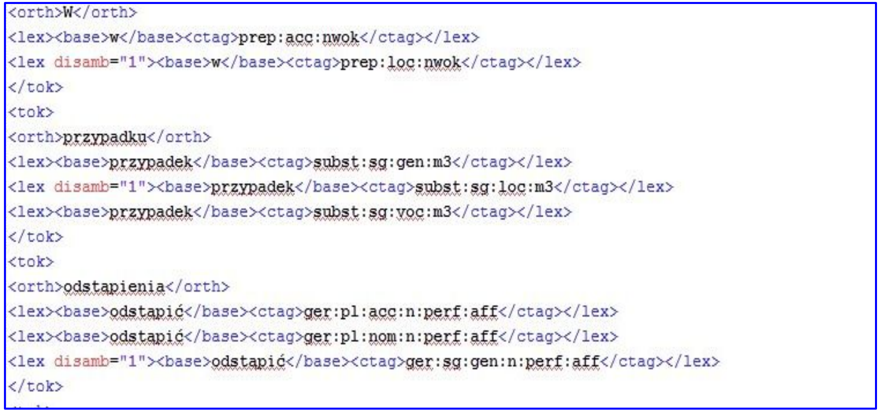
Human labeling / annotation

- Ca. 3300 clauses labeled
- 6 categories: SAD, KARA, OPLATA, OGRPRAW, DYSPROP, WARPRZYM

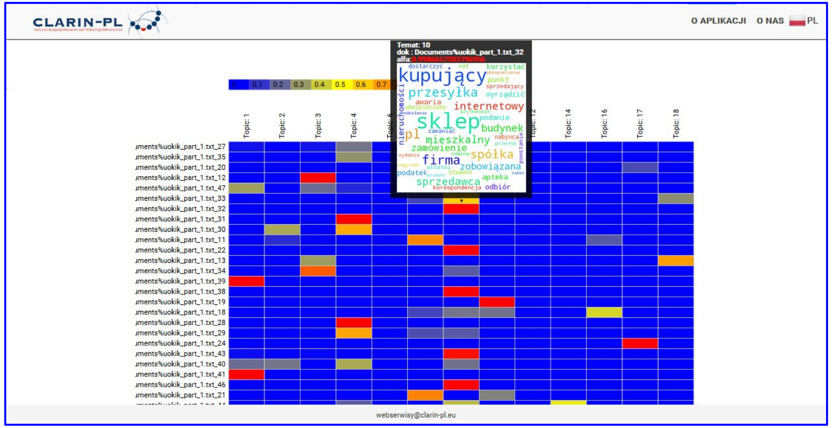
Label	No. clauses	Example
SAD	505	“(...) spór rozstrzygać będzie sąd właściwy rzeczowo dla siedziby Sprzedawcy.”
KARA	232	“(...) odsetki karne za niedotrzymanie terminu (...)”
OPLATA	787	“ (...) otrzyma należność z potrąceniem 2% (...)”
OGRPRAW	742	“(...) zastrzega sobie prawo do nieprzyjęcia zwrotu(...)”
DYSPROP	978	“ (...) dokonuje zakupu na własną odpowiedzialność (...)” “Wszelkie koszty (...) ponosi kupujący”
WARPRZYM	26	“Warunkiem przyjęcia (...) jest sporządzenie protokołu szkód (...)”

Traditional NLP approach

- CLARIN tools: POS-tagger, Korpusomat, TermoPL, Topic



```
<orth>W</orth>
<lex><base>w</base><ctag>prep:acc:nwok</ctag></lex>
<lex disamb="1"><base>w</base><ctag>prep:loc:nwok</ctag></lex>
</tok>
<tok>
<orth>przypadku</orth>
<lex><base>przypadek</base><ctag>subst:sg:gen:m3</ctag></lex>
<lex disamb="1"><base>przypadek</base><ctag>subst:sg:loc:m3</ctag></lex>
<lex><base>przypadek</base><ctag>subst:sg:voc:m3</ctag></lex>
</tok>
<tok>
<orth>odstapienia</orth>
<lex><base>odstapić</base><ctag>ger:pl:acc:n:perf:aff</ctag></lex>
<lex><base>odstapić</base><ctag>ger:pl:nom:n:perf:aff</ctag></lex>
<lex disamb="1"><base>odstapić</base><ctag>ger:sg:gen:n:perf:aff</ctag></lex>
</tok>
```

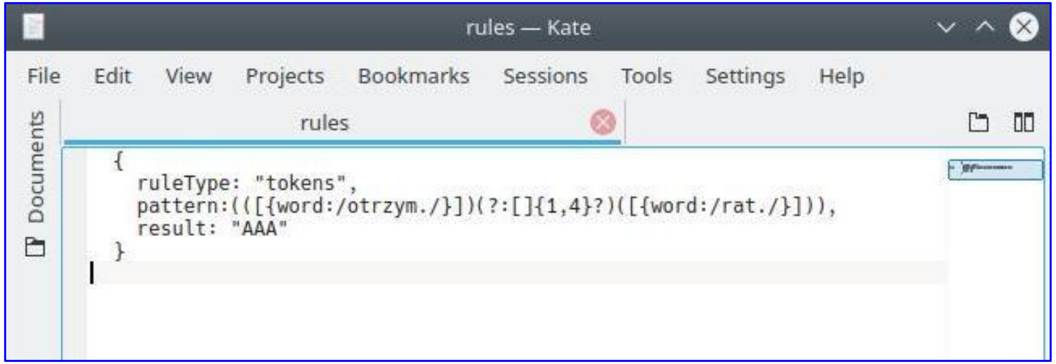


Temat: 10
dok : Documents%uokik_part_1.txt_32
alfa:0.9986657303196966

CLARIN-PL
O APLIKACJI
O NAS
PL
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
Topic: 1
Topic: 2
Topic: 3
Topic: 4
Topic: 6
Topic: 12
Topic: 14
Topic: 16
Topic: 17
Topic: 18
dostarczyć
vat
korzystać
kupujący
ubezpieczenie
punkt
sprzedający
przesyłka
wyrządzić
nieruchomości
awaria
ubezpieczony
internetowy
użytkownik
odwołanie
podanie
sklep
budynek
pl
zamawiać
nabywca
mieszkalny
przerwa
zamówienie
odmowa
spółka
wydanie
powstanie
firma
nagroda
zobowiązana
ostatni
podatek
produkt
student
żaden
apteka
sprzedawca
korespondencja
odbiór
ıments%uokik_part_1.txt_27
ıments%uokik_part_1.txt_35
ıments%uokik_part_1.txt_20
ıments%uokik_part_1.txt_12
ıments%uokik_part_1.txt_47
ıments%uokik_part_1.txt_33
ıments%uokik_part_1.txt_32
ıments%uokik_part_1.txt_31
ıments%uokik_part_1.txt_30
ıments%uokik_part_1.txt_11
ıments%uokik_part_1.txt_22
ıments%uokik_part_1.txt_13
ıments%uokik_part_1.txt_34
ıments%uokik_part_1.txt_39
ıments%uokik_part_1.txt_38
ıments%uokik_part_1.txt_19
ıments%uokik_part_1.txt_18
ıments%uokik_part_1.txt_28
ıments%uokik_part_1.txt_29
ıments%uokik_part_1.txt_24
ıments%uokik_part_1.txt_43
ıments%uokik_part_1.txt_40
ıments%uokik_part_1.txt_41
ıments%uokik_part_1.txt_46
ıments%uokik_part_1.txt_21
webserwisy@clarin-pl.eu

Traditional NLP approach

- Plan - to be used in SemGrex rule writing



```
{
  ruleType: "tokens",
  pattern:(([{word:/otrzym./}])(?:[]{1,4}?)([{word:/rat./}])),
  result: "AAA"
}
```

- Failed miserably - Java Regex limitations, package structure, knowledge of programming language

Traditional NLP approach

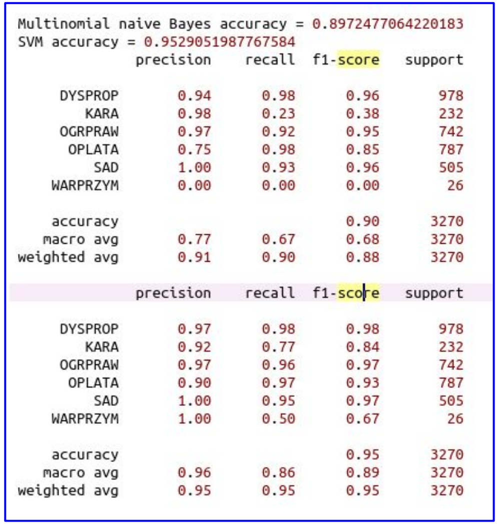
- Failed miserably - Java Regex limitations, package structure, knowledge of programming language

?

Scikit classifier implementations

- **Linear SVM** and **Naive Bayes** + TF-IDF feature
- First tested on two intentions: SAD and KARA
- NB acc = **0,9796**, SVM acc = **0, 9932**

- Implemented to multiple labels:
    - NB acc = **0,8972** SVM acc = **0,9529**
    - **Best** as far



```
Multinomial naive Bayes accuracy = 0.8972477064220183
SVM accuracy = 0.9529051987767584
              precision    recall  f1-score   support

     DYSPROP       0.94      0.98      0.96       978
        KARA       0.98      0.23      0.38       232
     OGRPRAW       0.97      0.92      0.95       742
      OPLATA       0.75      0.98      0.85       787
         SAD       1.00      0.93      0.96       505
    WARPRZYM       0.00      0.00      0.00        26

    accuracy                           0.90      3270
   macro avg       0.77      0.67      0.68      3270
weighted avg       0.91      0.90      0.88      3270

              precision    recall  f1-score   support

     DYSPROP       0.97      0.98      0.98       978
        KARA       0.92      0.77      0.84       232
     OGRPRAW       0.97      0.96      0.97       742
      OPLATA       0.90      0.97      0.93       787
         SAD       1.00      0.95      0.97       505
    WARPRZYM       1.00      0.50      0.67        26

    accuracy                           0.95      3270
   macro avg       0.96      0.86      0.89      3270
weighted avg       0.95      0.95      0.95      3270
```

Neural networks implementations

- Tensorflow / Keras implementations

- ANN + TF-IDF vectors
- LSTM + Fasttext
- LSTM + word2vec
- BERT + ktrain wrapper

ANN + TF-IDF

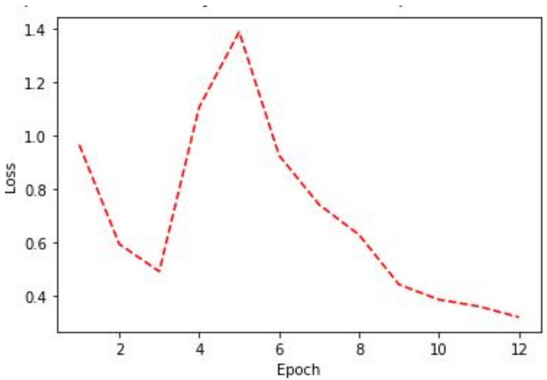
- Simple model: 3 layers, activation=RELU + Sigmoid, loss=categorical crossentropy, optimizer=Adam, d = 0,2
- T_time = 100 epochs

- Score for 2 labels: loss= **0,14151**, acc= **0,9633**
- Score for multiple labels: loss=**0,2930**, acc=**0,9440**

- Diff. parameters tested: limiting features - decrease in score, use of TF-IDF transformer - similar. Best scores: on CountVectorizer.

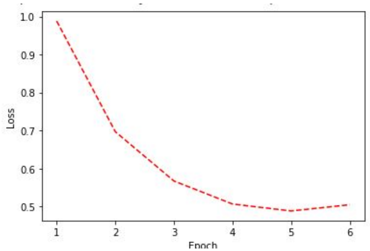
LSTM + fasttext

- Activation: softmax, loss=categorical crossentropy, optimizer=Adam, d = 0,2
- T_time = 12 epochs (ca. 50 min on Colab)
- Fasttext for Polish
- Score for 2 labels: loss= **0,4632**, acc= **0,9189**
- Score for multiple labels: loss=**0,5622**, acc=**0,8685**
- Diff. parameters tested: loss=cosine_proximity basically non-relevant
- Training loss curve:



LSTM + word2vec

- Same model
- Tested for 2 labels with general and IPI PAN Polish word2vecs for comparison:
  - General: loss= **0,5712**, acc=**0,7837**
  - IPI nkjp+wiki-forms-restricted-300-cbow-hs: loss=**0,3784**, acc=**0,9594**
- For multiple labels:
  - CBOW-hs: loss= **1,929** acc=**0,5504**
  - Best scores: loss= **1,08**, acc=**0,74** with nkjp+wiki+lemmas-all-300-skipg-ns (worst for 2 labels)
- V.large loss - why?
- Training loss curve:



BERT + ktrain

- **Ktrain** wrapper for Keras: https://github.com/amaiya/ktrain with BERT
- BERT Uncased Base (?)
- T_time: 1 epoch (ca. 3 hours on Colab)
- Train: 2943 samples, validate: 327 samples
- Scores (2 labels): loss= **0,276**, acc=**0,8727**;
- Scores(multiple): loss= **0,1783**, acc=**0,9297**; val_loss= **0,0645**, val_acc=**0,9837**

- Better scores for multiple labels
- Comparable to other reported results

Conclusions

What’s next?

- High values of loss functions - optimize, research
- Validate on the rest of clauses and real-life contracts

Thank you!

michalina.skibicka@gmail.com