Lietuvių kalbos teksto sintaksinės-semantinės analizės informacinė sistema

LKSSAIS vystymas

Modernizuojamos LKSSAIS Interneto lietuviškų socialinių tekstų analizės specialiuosius poreikius (neapykantos kalba) tenkinančio analizatoriaus administratoriaus instrukcija

6 puslapiai

2020

TURINYS

1. Įžanga

1.1. Dokumento paskirtis

Šio dokumento paskirtis – pateikti modernizuojamos LKSSAIS Interneto lietuviškų socialinių tekstų analizės specialiuosius poreikius (neapykantos kalba) tenkinančio analizatoriaus administratoriaus instrukcijas.

1.2. Sutrumpinimai

1.1 lentelė. Sutrumpinimai

Santrumpa	Paaiškinimas
LKSSAIS	Lietuvių kalbos teksto sintaksinės-semantinės analizės informacinė sistema
VDU vykdytojai	Užsakovo (VDU) projekto vykdytojų komanda
Diegėjas	Programavimo ir diegimo paslaugų konkursą laimėjusi įmonė (UAB ATEA)

1.3. Panaudotų dokumentų sąrašas

1.2 lentelė. Panaudotų dokumentų sąrašas

Eil. Nr.	Pavadinimas
1.	Neapykantos kalbos atpažinimo IT sprendimo komponento detalios analizės ir projektavimo specifikacija

1.4. Neapykantos kalbos atpažinimo IT sprendimo komponento paskirtis ir tikslai

Modernizuojamos LKSSAIS neapykantos kalbos atpažinimo IT sprendimo komponentas turi tekste atpažinti ar yra neapykantą skatinančios ir įžeidžios kalbos. Komponentas pateiktą tekstą (interneto socialinės žiniasklaidos tekstas) klasifikuoja į dvi klases: tekstas su neapykantos/įžeidžios kalbos požymiais arba neapykantos/įžeidžios kalbos požiūriu neutralus tekstas. Komponentas gauna JSON duomenis su *body* (tekstas lietuvių kalba) lauku, kurį suklasifikuoja, ir grąžina JSON atsakymą kuriame du laukai: klasifikavimo įvertis (0 arba 1) ir tikimybės indikacija.

Pagal techninės užduoties sąlygas, komponentą rengia VDU vykdytojai. Virtualizuoja (konteinerizuoja) ir į sistemą (paslaugą) integruos Diegėjas.

2. Neapykantos kalbos atpažinimo IT sprendimo komponento naudotojo instrukcija

2.1. Parengimas darbui ir paleidimas

Komponento paleidimui gali būti naudojamas Linux, MacOS arba Windows operacinę sistemą turintis kompiuteris. Kompiuteryje turi būti įdiegtas *Docker* įrankis.

Serveryje arba kompiuteryje išarchyvuojamas failas „**hatespeech.zip**“. Atsidarote direktoriją pavadinimu „**hatespeech**“.

Komponento paleidimui reikia sukurti *docker* konteinerį:

```
docker build -t hs .
```

Konteineris paleidžiamas su komanda:

```
docker container run --detach --name hatespeech -p 5000:5000 hs
```

Komponentas startuoja per kelias minutes.

2.2. Veikimo patikrinimas

Norint įsitikinti ar komponentas sėkmingai startavo, galima nueiti adresu http://0.0.0.0:5000/health jei atsakymas yra `{"status": "UP"}`, tai servisas sėkmingai startavo.

2.3. Komponento užklausa

Užklausa turi būti *UTF-8* koduotės ir lietuvių kalba. Užklausa ir atsakymas yra pateikiamas *JSON* formatu.

Užklausos pavyzdys:

```
curl --request POST \
  --url http://localhost:5000/hs \
  --header 'content-type: application/json' \
  --data '{
  "body": "Tekstas"
}'
```

Užklausos atsakymas:

```
{
  "classification": 0,
  "prediction": 0.49
}
```

2.4. Komponento permokinimas

Permokinimui reikalingi duomenys yra subdirektorijoje **data**, kur **training.txt** faile surašyti komentarai. Komentarai yra įžeidžiandūs arba neutralūs. Neigiamus komentaras „**__label__offensive**(tarpas)(komentaras)“, o neutralius „**__label__neutral**(tarpas)(komentaras)“.

Vienoje eilutėje gali būti tik vienas komentaras.

Komponentą galima permokinti per komponento API arba per komandinę eilutę.

2.5. Komponento permokinimas per API

Komponento API sudaro 3 pagrindiniai veiksmai:

- Įžeidaus komentaro pridėjimas
- Neutralaus komentaro pridėjimas
- Komponento permokymo proceso paleidimas

Norint pridėti įžeidų komentarą reikia siųsti *HTTP POST* užklausą:

```
curl --request POST \
  --url http://localhost:5000/add \
  --header 'content-type: application/json' \
  --data '{
  "text"  : "Įžeidus komentaras",
  "label" : "__label__offensive "
}'
```

Norint pridėti neutralų komentarą reikia siųsti *HTTP POST* užklausą:

```
curl --request POST \
  --url http://localhost:5000/add \
  --header 'content-type: application/json' \
  --data '{
  "text"  : "Neutralus komentaras",
  "label" : "__label__neutral"
}'
```

Komponento permokymo procesas paleidžiamas siunčiant *HTTP POST* užklausą:

```
curl --request POST \
  --url http://localhost:5000/retrain
```

Permokymo metu subdirektorijoje **model** perrašomi visi ten esantys failai.

2.6. Komponento permokinimas per komandinę eilutę

Paredaguokite subdirektorijoje **data** esantį **training.txt** failą. Pridedant įžeidžius ar neutralius komentarus. Neigiamus komentarus pradėti tekstu „**__label__offensive**(tarpas)(komentaras)“, o neutralius „**__label__neutral**(tarpas)(komentaras)“.

Prieš paleidžiant permokinimą būtina įrašyti ***python3*** papildomas bibliotekas su komanda:

```
pip3 install -r requirements.txt
```

Papildomai reikia įrašyti *FastText* biblioteką:

```
git clone https://github.com/facebookresearch/fastText.git

cd fastText

pip3 install .
```

Permokimo procesas paleidžiamas:

```
python3 train.py
```

Permokymo metu subdirektorijoje **model** perrašomi visi ten esantys failai.

Pasibaigus apmokinimo procesui, subdirektorijoje **model** sistema sukuria naują embedintos kalbos modelį ir klasifikatorių.

2.7. Komponento paleidimas po permokinimo

Pakartojama veiksmų seka, aprašyta skyriuje „**2.1. Parengimas darbui ir paleidimas**“

2.8. Komponento konfigūravimas

Komponentą galima konfiguruoti redaguojant „**config.ini**“ failą. Pavyzdžiui norint pakeisti komponento nustatytą prievado numerį užtenka faile paredaguoti eilutę „**Port = 5000**“ ir pakeisti į kitą numerį.