Lietuvių kalbos teksto sintaksinės-semantinės analizės informacinė sistema

LKSSAIS vystymas

Automatinės sentimentų analizės lietuvių kalbos tekste komponento administratoriaus instrukcija

8 puslapiai

2020

TURINYS

1. Įžanga

1.1. Dokumento paskirtis

Šio dokumento paskirtis – pateikti modernizuojamos LKSSAIS aspektais grįstos automatinės sentimentų analizės lietuviškuose socialinės medijos tekstuose, kuriuose išreiškiama nuomonė apie studijas, komponento administravimo 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)
Atpažintuvas	LKSSAIS automatinės aspektais grįstos sentimentų analizės komponentas

1.3. Panaudotų dokumentų sąrašas

1.2 lentelė. Panaudotų dokumentų sąrašas

Eil. Nr.	Pavadinimas
1.	Automatinės sentimentų analizės lietuvių kalbos tekste komponento detalios analizės ir projektavimo specifikacija

1.4. Automatinės sentimentų analizės lietuvių kalbos tekste komponento paskirtis ir tikslai

Čia aprašomas automatinės aspektais grįstos sentimentų analizės komonento (toliau - Atpažintuvas) yra skirtas vertinti lietuviškų interneto socialinės medijos tekstų, kuriuose išreiškiama nuomonė apie studijas, sentimentus (nuomones) tekste išreikštų, su studijomis susijusių aspektų atžvilgiu. Atpažintuvas parengtas dirbti su lietuviškais, normalizuotais interneto socialinės medijos tekstais (komentarai, blogai ir t.t.).

Atpažintuvo paskirtis ir tikslai – tenkinti LKSSAIS teikiamų paslaugų poreikį įvertinti žmogaus socialinės medijos tekste išreikštą sentimentą (nuomonę) tam tikro studijų aspekto atžvilgiu.

Pagal paslaugos koncepciją Atpažintuvas sentimentus (nuomones) klasifikuoja į penkias klases: „Labai neigiami“, „Neigiami“, „Teigiami“, „Labai teigiami“ ir rezervinę klasę „Neutralūs“. Reiantis tyrimo rezultatais, nuomonės labai retai priskirtinos klasei „Neutralus“. Todėl, vertinant visą tekstą, ši klasė yra abejotino poreikio. Tačiau LKSSAIS Atpažintuvas vertina ne visą tekstą bendrai, bet atsiliepimo tekstą skaido į loginius vienetus („Nuomonių vientus“, angl. „opinion unit“), nes atsiliepime dažnai sutinkami atsiliepimai kelių aspektų atžvilgiu, ir vertina kiekvieną atskirai. Todėl klasė „Neutralus“ šiuo atveju reikalinga. Atpažintuvą galima perpanaudoti ir kitoms sentimentų klasėms atpažnti pagal vystytojo poreikius, jį atitinkamai permokinant pagal šiame dokumente aprašytą permokinimo procedūrą.

Pagal paslaugos koncepciją Atpažintuvas aspektus klasifikuoja į penkias klases: Destymas, Studijos, Infrastruktura, Karjera. Komponentas turi rezervinę aspekto klasę „Neutralus“.

Atpažintuvą galima perpanaudoti ir kitoms aspektų klasėms atpažinti pagal vystytojo poreikius, jį atitinkamai permokinant pagal šiame dokumente aprašytą permokinimo procedūrą.

Siekiant gerinti Atpažintuvo veikimo kokybę, jį atitinkamai permokinant pagal šiame dokumente aprašytą permokinimo procedūrą.

Atpažintuvas pats atlieka teksto segmentavimą į saknius ir žodžius, pats atlieka morfologijos ir sintaksės analizę, todėl papildomi kalbos technologijų komponentai jo veikimui nebūtinis.

Atpažintuvas parengtas dirbti su lietuvių kalbos tekstais, kuriuose nėra nukrypimų nuo rašybos normų (šveplas tekstas ir pan.). Todėl LKSSAIS sistemoje jam pateikiami socialinės medijos tekstai, prieš tai apdoroti automatiniu lietuvių kalbos teksto normalizavimo įrankiu. Bet Atpažintuvas neturi privalomos priklausomybės nuo jo, jei jis permokinamas dirbti su kitokiais tekstais. Normalizuoklis reikalingas tik geresnei klasifikavimo kokybei užtikrinti

2. Automatinės sentimentų analizės lietuvių kalbos tekste komponento administravimo instrukcija

2.1. Parengimas darbui

Atpažintuvas suprogramuotas Python kalba ir yra patalpintas Docker konteineryje. Docker konteineryje Apažintuvas veikia Linux aplinkoje. Naudojat komponentą ne Docker konteineryje, jis veikia bet kokioje operacinėje aplinkoje, kurioje sudiegtos reikalingos proklausomybės. Pateikiamas programinis kodas orientuotas į Linux aplinkos specifiką, todėl Atpažintuvą leidžian ne Linux aplikoje, reikia atlikti pakeitimus kodėl pagal kitos operacinės sistemos specifiką.

Docker atvaizdas atsisiunčiamas adresu: semantikadocker.vdu.lt/sentiments:latest

Docker konteneris paleidžiamas standartine Docker konteinerių paleidimo procedūra.

Komponento programinės įrangos struktūra:

1) šakninė direktorija:

app.py – pagrindinis failas, kuris paleidžia komponentą, jo serverį ir mikroservisą. Jei reikia pakeisti mikroserviso IP ir porto numerį, tai nustatoma šiame faile esančio programinio kodo 175 eilutėje.

requirements.txt - priklausomybių sąrašas

Docker - Docker failas.

2) subdirektrija „noun“:

tokenizer_exceptions.py – Spacy tokenizavimo išimčių rinkmena, kuria Atpažintuvo gamintojai papildę lietuvių kalbos trumpiniais, gerinančiais teksto segmentavimo kokybę. Naudojant Atpažintuvą ne Docker konteineryje, šį failą reikia nukopijuoti į Spacy instaliacijos direktoriją **./lang/lt**.

syntax_itarators.py - standartinę Spacy istaliaciją šio failo neturi lietuvių kalbai, todėl veveikia sintaksinis skaidymas į daitvavardines frazes, be kurio Apažintuvas neveikia.

Serveryje arba kompiuteryje sukuriama direktorija „**Sentiment**“. Joje išskleidžiamas failo „**sentiment.zip**“ turinys. Direktorijoje „**Sentiment**“ atsiranda: a) du failai **sentiment.py** ir **train.py** ir b) dvi subdirektorijos **training** ir **models**. Naudojant Atpažintuvą ne Docker konteineryje, šį failą reikia nukopijuoti į Spacy instaliacijos direktoriją **./lang/lt**.

init.py - modifikuotas Spacy failas, būtinas, kad veiktų aukųčiau aprašytų failų funkcijos. Naudojant Atpažintuvą ne Docker konteineryje, šį failą reikia nukopijuoti į Spacy instaliacijos direktoriją **./lang/lt**.

3) Subdirektorija „resouces“:

sentimen_model.json – Tensorflow formatu išsaugotas sentimentų klasifikatoriaus neuroninio tinklo verčių modelis

sentimen_model.h5 – Tensorflow formatu išsaugota sentimentų klasifikatoriaus neuroninio tinklo architektūra

aspect_model.h5 – Tensorflow formatu išsaugota aspektų klasifikatoriaus neuroninio tinklo architektūra

aspect_model.json – Tensorflow formatu išsaugotas aspektų klasifikatoriaus neuroninio tinklo verčių modelis

absa.csv – failas, kuriame yra atsiliepimų tekstynas, vektorių modelio sudarymui, atraminiai senitmentų ir aspektų terminai, aspektų ir sentimentų klasių etiketės

4) Subdirektorija „services“:

modeliai.py - nuo app.py prikalusantis failas, kurio funkcijos reikalingos į Atpažintuvo serverį užkrauti viso veikimo metu RAM‘e reziduojančius neuroninių tinklų modelius

text_pre.py - nuo app.py priklausantis failas, kuriame patalpintos teksto valymo ir frazavimo į nuomonių vienetus taisyklės

2.2. Komponento paleidimas

Komponento veikimui iš Docker konteinerio, koteineryje yra visos reikalingos priklausomybės.

Komponento paleidimui be Docker konteinerio, sistemoje turi būti įdiegta requirements.txt nurodyta programinė priklausomybių įrangą.

Paleidimas Linux aplinkoje: komandinėje eilutėje rašoma **python3 app.py** ir spaudžiama **Enter** klavišas.

Paleidimas Windows aplinkoje: komandinėje eilutėje rašoma **python app.py** ir spaudžiamas **Enter** klavišas

Linux aplinkoje: komandinėje eilutėje rašoma **python3 sentiment.py** ir spaudžiama **Enter** klavišas.

Windows aplinkoje: komandinėje eilutėje rašoma **python sentimen.py** ir spaudžiamas **Enter** klavišas.

Abiem atvejais paleidžiamas komponento veikimas, paleidžiamas komponento serveris, aktyvuojasi komponento mikro servisas.

PASTABA: paleidus komponento veikimą, pirma užklausa apdorojama kiek ilgiau, nes tik pirmos užklausos apdorojimo metu į serverio RAM įkeliami ir suformuojami sentimentų ir aspektų klasifikatorių neuroniniai modeliai. Po pirmos užklausos apdorojimo, jie RAM‘e lieka reziduoti iki komponento veikimo sustabdymo. Todėl visos tolesnės užklausos apdorojamos ženkliai greičiau.

Komponento veikimo nutraukimas atliekamas klavišų kombinacija „Ctrl“+C.

Norint įsitikint, ar Atpažintuvo miroservisas veikia, pateikima GET metodo užklausa http://0.0.0.0:[port]/health. Jei komponentas veikia, gaunamas atsakymas: 200 OK, JSON ({"status": "UP"}).

JSON užklausa su analizuotinu tekstu pateikiama metodu POST adresu http://0.0.0.0:[port]/absa.

Jei užklausa paduodama ne JSON užklausa arba JSON užklausoje nėra laukelio 'normalised_body', komponentas duoda klaidą: 400, status: bad request.

Užklausos JSON pavyzdys:


```
{
    "normalised_body": "X ir Y informatikos dėstytojai protingi. X dėstytojai fantastiški, bet internetas neveikia. Y dėstytojai fantastiški, bet kompiuteriai neveikia. Z profesoriai nieko neišmano. Šiaip W gali pasigirti geresne technika straipsnyje aiškiai parašyta tai, Y gal kažkiek griežtesne tvarka ir gal kai kurie dėstytojai aukštesnio lygio."
}
```


Atsakymo pavyzdys:


```
[
  {
    "sentence_index": 0,
    "aspect_opinion": [
```

```
    {
      "aspect": "Destymas_asp",
      "opinion": "T"
    }
  ]
},
{
  "sentence_index": 1,
  "aspect_opinion": [
    {
      "aspect": "Studijos_asp",
      "opinion": "T"
    },
    {
      "aspect": "Studijos_asp",
      "opinion": "N"
    }
  ]
},
{
  "sentence_index": 2,
  "aspect_opinion": [
    {
      "aspect": "Destymas_asp",
      "opinion": "T"
    },
    {
      "aspect": "Studijos_asp",
      "opinion": "N"
    }
  ]
},
{
  "sentence_index": 3,
  "aspect_opinion": [
    {
      "aspect": "Studijos_asp",
      "opinion": "N"
    }
  ]
},
{
  "sentence_index": 4,
  "aspect_opinion": [
    {
      "aspect": "Studijos_asp",
      "opinion": "T"
    },
    {
      "aspect": "Studijos_asp",
      "opinion": "T"
    }
  ]
}
```

]

2.3. Komponento konfigūravimas

Atpažintuvas veikia mikro servisų pagrindu.

Užklausoje turi būti pateiktas JSON (užklausos pavyzdys pateiktas šiame dokumente žemiau). Užklausoje turi būti pateikiamas lietuvių kalbos, pageidautina normalizuotas, UTF-8 koduotės tekstas. Teksto ilgis neribojamas. Šių parametrų keitimas-konfigūravimas nenumatytas. Bet tekstas gali būti pateikiamas ir nenormalizuotas, tačiau nuo to kris Atpažintuvo kokybė.

Jei reikia pakeisti Atpažintuvo mikro serviso IP ir porto numerį, tai nustatoma **app.py** faile esančio programinio kodo 175 eilutėje.

Jei vystytojui reikia, užklausoje esančio JSN laukelio pavadinimas gali būti pakeistas pagal poreikį, pakeičiant jį app.py failo 92 eilutėje.

Jei komponentas paleidžiamas ne Docker koteineryje ir ne Linux aplinkoje, reikia rankiniu būdu app.py ir modeliai.py failuose pataisyti failų takelių sintaksę, pagal konkrečios OS specifiką.

Susiejimas su Kafka karkasu aprašytas šiame dokumente žemiau.

Kitų konfigūravimų Atpažintuvo veikimui atlikti nereikia.

2.4. Komponento permokymas

Norint permokyti komponentą, iš https://github.com/Semantika2/ABSA_studies parsisiunčiamas failas **retrain_ABSA.py**

Parengiamas failas absa.csv pagal komponento sudėtyje esantį pavyzdį. Sentimentų ir aspektų klasių skaičius ir etikečių pavadinimai priklauso nuo vystytojo poreikių. Todėl permokymas gali būti atliekmas: 1) siekiant didinti atpažintuvo kokybę; 2) permokinti dirbti su kitos srities nuomonėmis (politika ir pan.)

Faile retrain_ABSA.py patikslinama apmokymo failo lokalizacijos takelis ir takelis, kur planuojama išsaugoti apmokymo rezultus.

Failas paleidžiamas:

Linux - python3 retrain_ABSA.py

Win - python retrain absa.py

Po permokymo pasirinktoje direktorijoje atsiranda failai aspect_model_json, aspect_model.h5, sentiment_model.json, sentiment_model.h5, Emp.pickle. Šiuo faulus reikia nukopijuoti į komponento veikimo subdirektoriją „resources“ (panaikinant ten esančius.

Į tą pačią direktoriją reikia nukopijuoti pasirengtą failą absa.csv su aspektų bei sentimentų klasių etiketėmis, atraminiai terminais ir tekstynu vektorių modelio sudarymui.

2.5. Komponento paleidimas po permokinimo Komponento permokinimas

Pakartojama veiksmų seka, aprašyta skyriuje „**Komponento paleidimas**“