Soft Kompjuting E2 – 2020/21

Nedeljni izazov #4 – Ekstrakcija informacija iz polustrukturiranih dokumenata

Motivacija

Ekstrakcija informacija (engl. Information Extraction - IE) je vrsta pretraživanja podataka čiji cilj je automatski dobiti strukturirane informacije, odnosno kategorisane i kontekstualno i semantički dobro definisane podatke iz određene oblasti, iz nestrukturiranih mašinski čitljivih dokumenata, poput skeniranih/fotografisanih stranica.

Funkcionisanje savremenih poslovnih okruženja teško može funkcionisati bez ovakvih sistema. Poslovne procedure generišu veliku količinu štampanih dokumenata koje kasnije treba skladištiti i pretraživati. Zbog toga se ovakvi dokumenti jako često skeniraju, a nakon toga se nad njima vrši OCR (engl. Optical Character Recognition), koji služi da se skenirani dokumenti (najčešće u formatu slike, ili pdf sa skeniranim stranicama) prebace u tekstualnu formu, a nakon toga da se pretražuju i da se takve informacije čuvaju u relacionim ili nerelacionim (noSQL) bazama podataka.

Međutim, ovakve sisteme srećete i u svakodnevnom životu na različitim mestima. Gde god nam je potrebno čitanje informacija sa fotografija dokumenata, koriste se tehnike koje ćemo obraditi kroz ovaj izazov. Da bi problem pojednostavili, koristićemo **polustrukturirane dokumente** čija struktura prati neki šablon, a u nastavku se nalaze primeri takvih dokumenata.

Primer 1: Opcija “Slikaj i plati” u m-banking sistemima

Tipičan primer ovakvog sistema je opcija “Slika i plati” u m-banking sistemima. Ovi sistemi su “obučeni” tako da izvlače tačno određene informacije iz različitih formata uplatnica/platnih naloga. Uplatnica se fotografiše, a nakon toga se koriste tehnike računarske vizije da se iz nje izvuku podaci.

Primer 2: Čitanje ličnih dokumenata

Kada govorimo o graničnim prelazima, nezaobilazna stavka je kontrola ličnih dokumenata. Ovakva kontrola se obično vrši kroz sisteme koji prvo skeniraju dati dokument, a nakon toga proveravaju tačnost podataka u informacionom sistemu i beleže informacije o kretanju ljudi i roba.

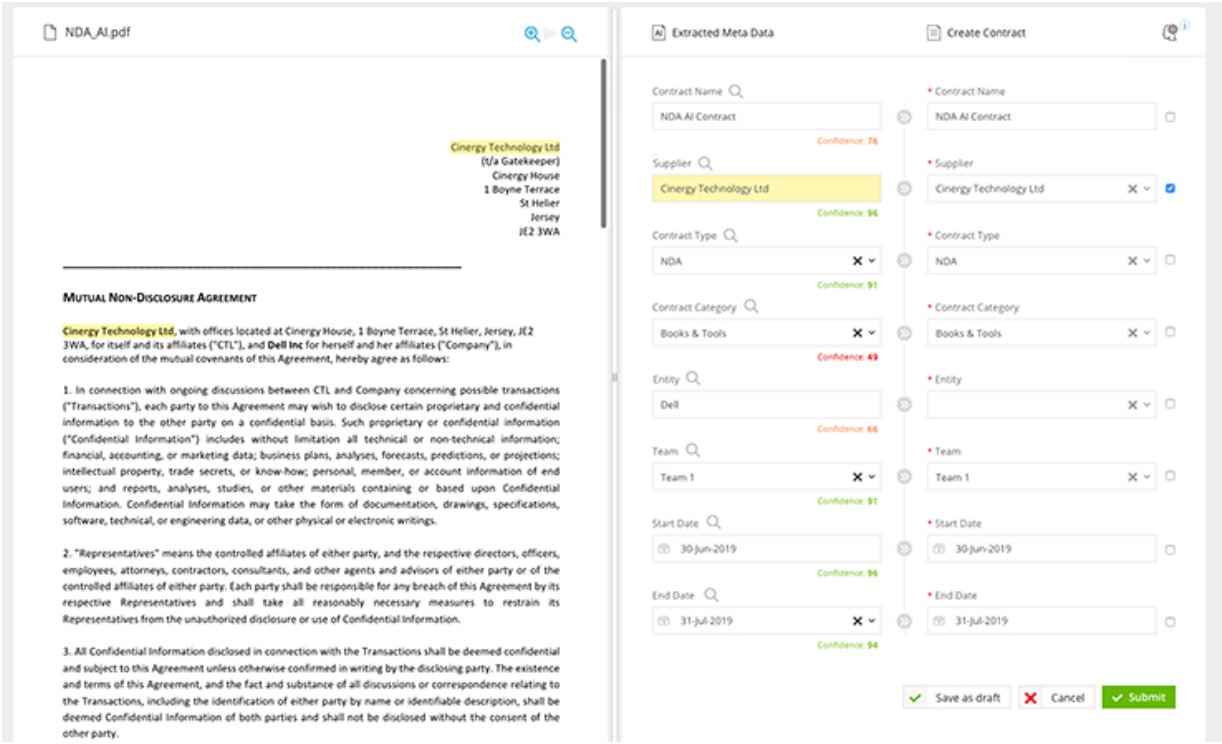
Skeniranje ovakvih dokumenata se može obavljati kroz specijalizovan hardver, koji je opremljen odgovarajućim softverom, “obučenim” za čitanje ličnih dokumenata. Primer izlaza takvog sistema je prikazan na sledećem screenshot-u.

Međutim, procesiranje dokumenata može biti jako kompleksno ukoliko su u pitanju p**otpuno nestrukturirani dokumenti** (ne postoji jasan šablon po kom su dokumenti pravljeni), pa mašina mora da “pročita” tekst na sličan način kao i čovek, da ga razume i da sama odluči koje informacije su relevantne za izvlačenje a koje nisu. Takav tip problema spada u oblast proceiranja prirodnog jezika (engl. Natural Language Processing – NLP), trenutno jednu od najperspektivnijih oblasti veštačke inteligencije. Međutim, ove tehnike nećemo obrađivati na ovom predmetu i o njima možete čuti više u predmetima na master studijama inteligentnih sistema.

…This Contract is made this 4th day of January in the year 2017 by and between the BIG BANK CORPORATION (hereinafter referred to as the "Purchaser"), having its principal office at 10 A Street, N.W., Washington D.C. 20433 and TechnologyVendor, LLC hereinafter referred to as the "Contractor"), a corporation incorporated under the laws of the state of New York having a principal place of business at 30 Broad Rd, Union NJ 07083…

```
<eligible_contract_participants>
<purchaser>
    <name>BIG BANK CORPORATION</name>
    <location><address>10 A Street, N.W.</address><city>Washington D.C.</city><zip>20433</zip></location>
</purchaser>
<contractor>
    <name>TechnologyVendor, LLC</name>
    <location><address>30 Broad Rd</address><city>Union</city><state>New Jersey</state><zip>07083</zip></location>
</contractor>
</eligible_contract_participants>
```

Primer izvlačenja informacija iz nestrukturiranog dokumenta (različiti ugovori, klauzule i slično):



Zadatak

Naš ovonedeljni zadatak će biti ekstrakcija informacija iz polustrukturiranih dokumenata, na primeru ličnih dokumenata. Ideja jeste da korišćenjem tehnika sa vežbi pročitate sve tražene podatke iz dokumenta.

Result

Firstname : PATRICIA
Lastname : DELAIRE
NID/CIN : 0123456789
Birthdate : 01-01-1987
Gender : F
Place of birth : Département Quest, Commune Port-au-Prince

Izvršiti čitanje podataka iz fotografija dokumenta korišćenjem tehnika računarske vizije kroz OpenCV biblioteku, Tesseract biblioteku i modela mašinskog učenja kroz Keras, Scikit-Learn i/ili DLIB, kao i programski jezik Python.

Dozvoljeno je korišćenje **svih** biblioteka sa prva tri izazova, plus nove biblioteke koje smo uveli u vežbi 4.

Na vežbama smo prošli sasvim dovoljno teorijskih i praktičnih osnova za rešavanje ovog izazova. ***Budite kreativni i primenite ih na svoj način, tako da dobijete što bolje rezultate.***

Dozvoljene biblioteke i podešavanje okruženja

U sklopu ovog izazova je dozvoljeno koristiti sledeće biblioteke uz Python 3.6:

- numpy
- openCV verzija **3.x.y** (bilo koja verzija koja počinje sa 3)
- matplotlib
- scikit-learn (verzija **0.21.3**)
- keras verzija 2.1.5
  - za FeedForward potpuno povezane NM
  - **nije dozvoljeno koristiti konvolutivne mreže, odnosno *Conv* slojeve**
- joblib (za serijalizaciju scikit-learn modela)
- Dlib
- Tesseract verzija 4

Instaliranje:

Za kreiranje okruženja i instalaciju biblioteka je potrebno preuzeti najnoviju Anaconda distribuciju sa njihovog zvaničnog sajta i instalirati je. Anaconda postoji za sve moderne operativne sisteme. Nakon instaliranja možete preći na kreiranje virtuelnog okruženja i instaliranje biblioteka u njega.

Detaljniji opis šta virtuelna okruženja predstavljaju možete naći u sklopu **v0** na github repozitorijumu predmeta (https://github.com/ftn-ai-lab/sc-2019-e2/blob/master/v0-priprema/podesavanje-okruzenja.ipynb)

1. Kreirati virtuelno okruženje (iz terminala na Linux i MacOS, ili Anaconda prompt na Win)

   ```
   conda create -n soft-env python=3.6
   ```

2. Aktivirati okruženje (**obavezno pre naredbi iz koraka 3**)

   ```
   source activate soft-env
   ili
   conda activate soft-env
   ```

3. Instalirati biblioteke

   ```
   pip install opencv-python==3.4.1.15
   pip install scikit-learn
   pip install imutils
   pip install matplotlib
   pip install keras==2.1.5
   pip install theano
   pip install h5py
   pip install pillow
   pip install pyocr

   conda install -n soft-env -c conda-forge dlib
   ili
   ```

```
conda install -n soft-env -c menpo dlib
```

Instalirajte tesseract biblioteku (samo za Linux OS):

```
sudo apt install tesseract-ocr
sudo apt install libtesseract-dev
```

Za instalaciju tesseract biblioteke (verzija 4) na drugim operativnim sistemima prođite kroz njihovo zvanično uputstvo. **Mada je naša preporuka da koristite VM koji prilažemo uz izazov**, jer instalacije na ne-linux sistemima mogu napraviti dosta problema pa ne morate gubiti vreme na to.

VM možete pronaći na sledećem linku:
https://drive.google.com/drive/folders/1C2C1FmPI2fb1B_8gqGIfsazbN1Bag0_P?usp=sharing

Ukoliko budete imali problema sa numpy bibliotekom (**dobijate DLL greške vezane za numpy** prilikom pokretanja rešenja), znači da imate više numpy verzija i da ih treba obrisati. Izvršavajte (više puta) „pip uninstall numpy“ dok ne dobijete obaveštenje da numpy ne postoji, a onda ga instalirajte ponovo sa „pip install numpy“.

4. Preuzeti i instalirati pyCharm Community razvojno okruženje, koje je preporuka za razvoj python projekata. Otvoriti projekat koji je deo ovog izazova.

5. Podesiti interpreter za projekat tako što ćete se povezati na python instancu iz prethodno kreiranog virtuelnog okruženja. Uputstvo je nalazi na kraju sledećeg fajla, koji je na github repozitorijumu predmeta (https://github.com/ftn-ai-lab/sc-2019-e2/blob/master/v0-priprema/podesavanje-okruzenja.ipynb)

6. Sve je spremno. Desni klik na odgovarajući fajl i onda Run ili Debug. Ukoliko importovanje cv2 biblioteke puca u skripti, proverite da li ste dobro instalirali openCV i da li ste dobro povezali projekat sa virtuelnim okruženjem.

Format koda za ocenjivanje (isto za sve izazove)

Kod koji upload-ujete u GoogleDrive folder treba da zadovolji neke kriterijume da bi ga platforma za ocenjivanje analizirala na pravi način. Glavna ograničenja su sledeća:

1. **Fajl main.py se mora nalaziti u korenu vašeg foldera.** Ukoliko to nije slučaj, platforma neće biti u mogućnosti da pokrene vaše rešenje i nećete biti ocenjeni.

2. **Fajlove main.py i evaluate.py nije dozvoljeno menjati.** Ovi fajlovi su direktno korišćeni od strane platforme da bi ocenjivanje bilo moguće.

3. **Vaša implementacija treba da bude u fajlu process.py.** U ovom fajlu se nalazi neimplementirana metoda koja ima jasno naznačen ulaz i izlaz. Metoda je automatski uklopljena u ostatak koda (poziva se iz main.py) i nema potrebe da je ručno pozivate. **Vaš zadatak je da implementirate tražene metode i da obezbedite da vraćaju ono što se od vas traži.**

4. **Dozvoljeno je kreiranje novih python fajlova, koje možete pozivati iz process.py.** Ukoliko želite da deo koda izdvojite u druge fajlove I da onda kroz python import koristite u process.py, to je dozvoljeno. Dok god poštujete sve prethodne korake, ne bi trebalo biti problema.

5. U kodu koji okačite na platformu nemojte koristiti sistemske pauze i slične mehanizme koji zahtevaju reakciju korisnika, pošto u tom slučaju rešenje neće biti pokrenuto.

Pokretanje rešenja i evaluacija

Da biste pokrenuli rešenje na svojoj mašini i proverili kolika je postignuta tačnost, potrebno je uraditi sledeće:

1. Implementirati metodu u **process.py** traženom logikom. Ovaj fajl **ne** pokrećete direktno.
2. Pokrenuti **main.py** (iz pycharm-a na Run, ili iz terminala komandon “python main.py” uz prethodno aktiviranje odgovarajućeg virtuelnog okruženja). Pokretanje main.py fajla će izgenerisati **result.csv** fajl, tako što će pozvati prethodno implementiranu metodu za sve primerke iz skupa podataka.
3. Pokrenuti **evaluate.py** fajl (iz pycharm-a na Run, ili iz terminala komandon “python evaluate.py” uz prethodno aktiviranje odgovarajućeg virtuelnog okruženja). Ovaj fajl će učitati result.csv koji je prethodno generisan i izračunati tačnost. Izlaz ovog fajla je samo broj koji pokazuje procenat tačnosti trenutnog rešenja.

Ocenjivanje (isto za sve izazove)

Ocenjivanje upload-ovanog koda će biti izvršavano iterativno, po sledećim pravilima:

1. Platforma će automatski vršiti download koda, jednom u 24h i vršiti ocenjivanje.

2. U toku jednog dana možete imati neograničen broj upload-a. Ocenjivanje će svakako biti pokrenuto samo jednom na kraju dana i biće ocenjen kod koji u tom trenutku bude u folderu na Google Drive-u.

3. Platforma vrši ocenjivanje za prethodni dan u periodu **od 3:00 iza ponoći do 8:00 ujutru narednog dana**, pa u tom periodu nije dozvoljeno menjanje fajlova.

4. Ukoliko izazov traje 7 dana, studenti tehnički imaju 7 pokušaja da reše izazov. Platforma će ocenjivati kod svaki dan. Na rang listu će se računati **najbolji rezultat** iz svih ciklusa ocenjivanja. Zbog toga je bolje da što ranije rešite izazov, pošto ćete imati više pokušaja da ispravite nešto i postignete još bolji rezultat. Ako bilo koji pokušaj bude detektovan kao plagijat, student dobija godinu dana zabrane polaganja.

5. Svaki dan će studenti dobijati izveštaj u formi txt fajla u svom Google Drive folderu. Ovo se odnosi samo na studente koji su postavili nešto u svoj folder. Izveštaj se generiše svaki dan, bez obzira na to da li ste šta menjali u folderu tog dana. Tako ćete na dnevnom nivou biti ažurirani činjenicom gde se nalazite na rang listi.

6. U izveštaju niko neće imati informaciju gde se tačno nalazi na rang listi. Dobićete informaciju da li se nalazite u TOP 5, TOP 10, TOP 25, TOP 50, TOP 100 ili van TOP 100 studenata. Ako ste dobili informaciju da ste u TOP 25, to znači da se nalazite izmedju 11. i 25. pozicije i da možete poboljšati rešenje da popravite rang. Tačan rang će biti objavljen na kanvasu, tek na kraju izazova.