Rapport - IPASS

Jimmy Bierenbroodspot

26 juni 2024

1 Probleembeschrijving

In de toekomst willen wij graag een AI-model hebben, dat de beste CV's op de markt kan maken voor onze klanten. Hiervoor hebben we natuurlijk veel data in de vorm van CV's nodig en gelukkig hebben we deze in grote aantallen. Nu is het probleem dat deze CV's in .pdf-formaat opgeslagen zijn en text hieruit halen is niet makkelijk [5].

We hebben het programma genaamd PyPDF[4] waarmee we in staat zijn om text uit .pdf-bestanden te lezen, het probleem is dat dit ongestructureerd is. Als voorbeeld kunnen we het voorbeeld CV in 1 nemen.

Figure 1: Voorbeeld van een Cliq CV

In 2 kunnen we zien wat er gebeurd als we de tekst uit 1 lezen. Dit ziet er uit als een onlogisch rommeltje. We kunnen met PyPDF echter ook de tekstopmaak kopiëren, zoals te zien in 3. Dit ziet er al beter uit maar als we bedenken dat dit allemaal op een regel staat, waarbij de nieuwe regels in dit geval juist worden getoond, maar eigenlijk aangeduid wordt met een \n.

```
Regio [WOONPLAATS]
DRIVE
[TEKST DRIVE - MIDDEN KOLOM
1 GEPOSITIONEERD]
KUNDE
MOGELIJKHEDEN
[TEKST MOGELIJKHEDEN - MIDDEN
KOLOM 1 GEPOSITIONEERD]OPLEIDING
Contactgegevens kandidaatERVARING
Student | startmaand startjaar - eindmaand eindjaarDRIE TREFWOORDEN +
PERSOONSOMSCHRIJVING Functietitel + bij + Bedrijf
[TEKST KUNDE - MIDDEN KOLOM
1 GEPOSITIONEERD]Titel opleiding + opleidinginstituut
[optionele uitleg studie][Meest effectieve zin van het hele CV, kort, krachtig, heel
erg passend bij de vacature.]
Functietiel + bij + Bedrijfstartmaand startjaar - eindmaand eindjaar
[Optionele uitleg van ervaring en ontwikkeling]
Student | startmaand startjaar - eindmaand eindjaarTitel opleiding + opleidinginstituut
[optionele uitleg studie]startmaand startjaar - eindmaand eindjaar
```

Figure 2: Voorbeeld van tekstextractie PyPDF

```
        DRIE TREFWOORDEN +                                  ERVARING
    PERSOONSOMSCHRIJVING                                    F  u  n  ctietitel + b  ij + Bed  rijf
            Re  g  io   [WOONPL  AATS]                      sta  rtma  a  n  d  sta  rtja  a  r - e  in  d  ma  a  n  d
                                                            [Me  e  st e  ffe  ctie  ve  zin  va  n  h  e  t h  e  le
                                                            e  rg  p  a  sse  n  d  b  ij d  e  va  ca  tu  re  .]
                          DRI  VE                           F  u  n  ctietiel + b  ij + Bed  rijf
   [TEKST DRIVE - MIDDEN KOL  OM                            sta  rtma  a  n  d  sta  rtja  a  r - e  in  d  ma  a  n  d
          1  GEPOSITIONEERD]                                [Op  tio  n  e  le  u  itle  g  va  n  e  rva  rin  g  e
                          KUNDE                             OPL  EIDING
  [TEKST KUNDE - MIDDEN KOL  OM                             T  itel o  p  leid  in  g  + o  p  leid  in  g  in  stitu  u
             1  GEPOSITIONEERD]                             Stu  d  e  n  t | sta  rtma  a  n  d  sta  rtja  a  r - e  in
                                                            [o  p  tio  n  e  le  u  itle  g  stu  d  ie  ]
             MOGEL  IJKHEDEN                                T  itel o  p  leid  in  g  + o  p  leid  in  g  in  stitu  u
                                                            Stu  d  e  n  t | sta  rtma  a  n  d  sta  rtja  a  r - e  in
[TEKST MOGEL  IJKHEDEN - MIDDEN                             [o  p  tio  n  e  le  u  itle  g  stu  d  ie  ]
     KOL  OM 1  GEPOSITIONEERD]
                                                            Co  n  ta  ctg  e  g  e  ve  n  s ka  n  d  id  a  a  t
```

Figure 3: Voorbeeld van tekstextractie PyPDF met tekstopmaak

1.1 Waarom willen we dit?

In datawetenschap kennen we het concept GIGO, oftewel garbage in garbage out [3]. Als we aannemen dat ditzelfde zal gelden voor het model dat we gaan gebruiken dan zouden we graag willen dat onze invoerdata gestructureerd is.

Als we aannemen dat de tekst op een regel zou moeten komen te staan om ingelezen te worden door het model, dan zijn beide manieren van text extraheren met PyPDF niet gestructureerd. We willen dus dat de tekst op een logische volgorde komt te staan. Als een CV geen kolommen zou bevatten dan kunnen we met PyPDF de tekstopmaak kopiëren, nu weten we alleen van tevoren niet of een CV kolommen heeft of niet en als dat wel zo is weten we niet hoeveel kolommen er zijn. Om hierachter te komen moeten we naar het CV kijken en dat gaat niet met een geautomatiseerd proces.

2 Eisen

De opdrachtgever heeft in ieder geval de volgende eisen gegeven:

1. Er moet een CV bestand ingelezen worden.
2. Het aantal kolommen moet teruggegeven worden.

Verder zijn er geen functionele of prestatie-eisen.

3 Algoritme

In 3 kunnen we zien dat er een best duidelijke scheiding zit tussen de twee kolommen. We hebben besloten om hier Lloyds algoritme (zie[2]) voor te gebruiken omdat het een makkelijke en snelle oplossing leek te zijn voor dit probleem.

Om erachter te komen hoeveel kolommen het optimale aantal zou zijn gebruiken we de elleboog methode, deze werkt als volgt[1]:

1. Voor elk gewilde aantal clusters wordt het model getraind.
2. Elke keer dat het model getraind is wordt de error opgeslagen.
3. Er wordt een rechte lijn van de eerste observatie naar de laatste observatie getrokken.
4. Bereken de afstand tussen deze lijn en elke score.
5. Neem het aantal clusters van de score die het verst van deze lijn verwijderd is.

4 Resultaat

Na het evalueren van het model op een dataset van 116 bestaande CV's zien we een teleurstellend resultaat. We weten maar van ongeveer 15% van de gevallen het juiste aantal kolommen te vinden. Dit is echt veel te weinig om in een productieomgeving te gebruiken.

5 In het vervolg

We zouden in de toekomst willen kijken of de volgende verbeterpunten het presteren van de clusteringsaanpak zal vergroten:

- het model meerdere keren uitvoeren met verschillende willekeurige clustercentra voor elke keer dat het model getraind wordt.
- Kmeans++ gebruiken.
- De invoerdata standaardiseren.

Ook zouden we dit probleem graag als classificatieprobleem willen proberen aanpakken aangezien dit daadwerkelijk ondersteund wordt door bestaande literatuur.

References

[1] *Find the "elbow" of a curve.* URL: https://docs.ropensci.org/pathviewr/reference/find_curve_elbow.html (visited on 06/26/2024).

[2] S. Lloyd. "Least squares quantization in PCM". In: *IEEE transactions on information theory* 28.2 (Mar. 1, 1982), pp. 129–137. DOI: 10.1109/tit.1982.1056489. URL: https://doi.org/10.1109/tit.1982.1056489.

[3] Merriam-Webster. *GIGO*. In: URL: https://www.merriam-webster.com/dictionary/GIGO (visited on 05/30/2024).

[4] *pypdf*. Apr. 7, 2024. URL: https://pypi.org/project/pypdf/ (visited on 05/30/2024).

[5] Amit Timalsina. *How to extract data from PDF?* May 2, 2024. URL: https://www.docsumo.com/blog/extract-data-from-pdf (visited on 05/30/2024).

List of Figures

List of Tables