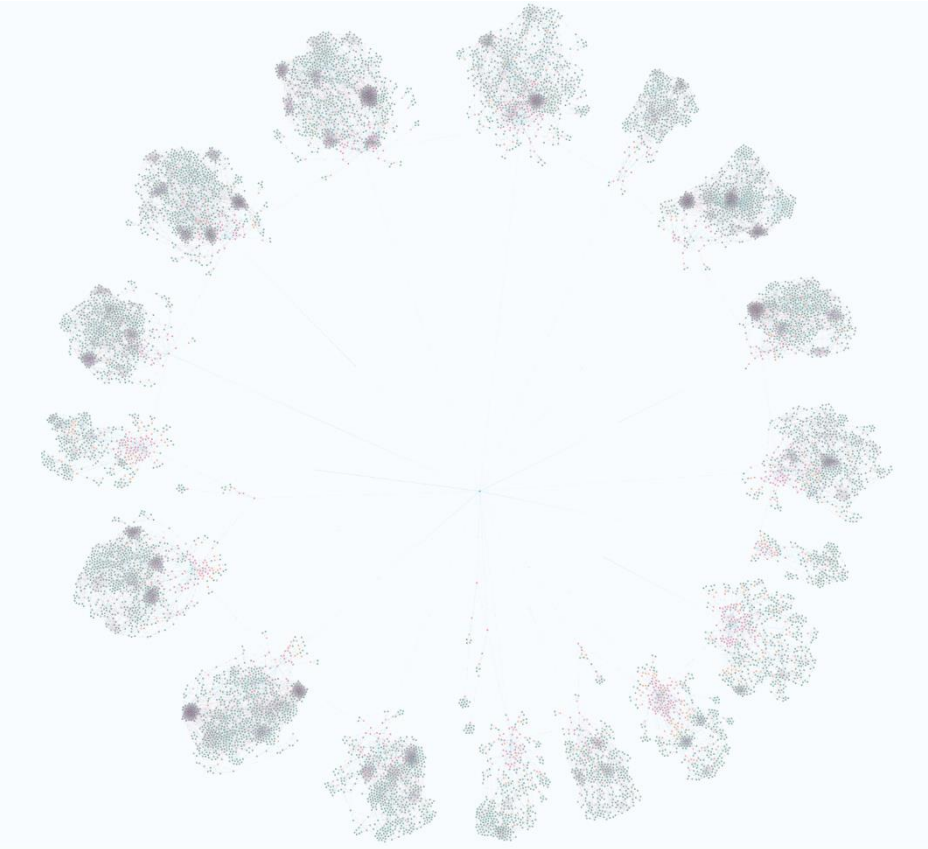
Onderzoeksrapport CirculairNLP



by

Axel Frederiks (2202630)
Jonah Siemers (2202907)
Menno Rompelberg (2204333)
Oscar Theelen (2204154)

19/04/2024

Supervisors

Pascal Slaats
Jeroen Baijens
Shannen Dolls

Requirements

Methode

Elke vrijdag zijn er gesprekken en interviews met de product owner waaruit we nieuwe requirements opnemen in het onderstaande SRS. Verder worden na deze gesprekken ook aanpassingen aan de individuele requirements gedaan wanneer deze niet volledig correct blijken te zijn en worden er in enkele situaties ook requirements verwijderd uit de lijst.

ID	Titel	Prioriteit	Eigenaar
NF1	Functies bevatten maximaal 50 regels code.	Must	Casusgroep
NF2	De bestanden geschreven in de volgende taal kunnen minimaal herkend worden: ENG (extra: NL, FR, DU).	Must	Pascal Slaats
NF3	Het labelen vanuit het systeem heeft een zelflerende functie.	Should	Pascal Slaats
NF4	Minimaal de bestanden opgeslagen in het volgende documenttype kunnen verwerkt worden: PDF (extra: PDFA).	Must	Pascal Slaats
NF5	Apache Hop bewaakt de workflow.	Could	Pascal Slaats
NF6	Apache Hop stuurt de Scripts aan.	Could	Pascal Slaats

ID	Titel	Prioriteit	Eigenaar
F1	Er is een functie die de tekst uit een bestand omzet naar een lijst van alle zinnen.	Must	Pascal Slaats
F2	Er is een functie die tabellen uit bestanden herkent en omzet naar een lijst van cel-inhouden.	Must	Pascal Slaats
F3	Er is een functie die tekst uit afbeeldingen herkent.	Must	Pascal Slaats
F4	Er is een functie die afbeeldingen uit het document herkent en opslaat.	Must	Pascal Slaats
F5	Er is een functie die verwijzingen tussen woorden/zinnen/alinea’s herkent.	Must	Pascal Slaats
F6	Er is een optie om tussen 2 bestanden alle gelijkenissen te vinden.	Should	Pascal Slaats
F7	Er is een functie die voor elke zin in de lijst met zinnen deze zin ontleed.	Must	Pascal Slaats

Grounding

Onderzoeksonderdelen:

1. PDF-tekst omzetten naar enkele zinnen.
2. Afbeeldingen uit PDF-bestanden halen.
3. Tekst uit afbeeldingen omzetten naar enkele zinnen of woorden.
4. Werken met Apache Hop.
5. Werken met Neo4J.
6. PDF Tabellen omzetten naar enkele zinnen of woorden.
7. Verwijzingen tussen zinnen/woorden vinden.
8. Libraries voor het herkennen en verwerken van Europese talen.
9. Tools voor zinnen ontleden.
10. Tekst opdelen in woorden, zinnen, paragrafen, hoofdstukken.
11. Bestanden met elkaar vergelijken.
12. Mogelijkheden voor PDF en PDFA.

Doelstelling

Met dit onderzoek willen we antwoorden krijgen op de eerdergenoemde onzekerheden zodat we een zo goed als mogelijke applicatie kunnen neerzetten voor de product owner.

De hierboven genoemde onzekerheden zullen een voor een beantwoord worden, zodat de focus gelegd kan worden op het vinden van literatuur en antwoorden per vraag. Dit voorkomt dat bronnen door elkaar gehaald gaan worden. De zoekstrategie die gebruikt gaat worden bestaat uit zoektermen en selectiecriteria die gebruikt gaan worden om bronnen te verzamelen en om hiermee aan antwoorden te komen. Ten slotte is er nog de ranglijstcriteria die bepaalt welke bronnen belangrijker zijn binnen deze context.

Zoekstrategie (zoektermen)

- PDF-tekst uit bestand halen
- PDF-afbeeldingen uit bestand halen
- Tekst uit een afbeelding halen
- Apache hop
- Apache hop werking
- PDF-tabellen uit bestand halen
- Verwijzingen tussen zinnen met code
- Library voor taalherkenning
- Zinnen ontleden met code
- Inhoud van bestanden vergelijken in code

Selectiecriteria

Inclusie:
- Informatie over PDF, PDFA.

Exclusie:
- Informatie over bestandtypes buiten PDF en PDFA.

Ranglijst

Des te hoger in deze ranglijst een factor staat des te zwaarder deze telt voor de ranglijst van werkelijk gevonden bronnen.

1. Relevantie met betrekking tot de onzekerheden.
2. Des te meer unieke inhoud in een bron des te hoger te rang binnen de ranglijst.
3. Bronnen door instanties met aantoonbare professionaliteit.

Resultaten:

PDF-tekst omzetten naar enkele zinnen

Enkele software tools die voor het lezen van tekst uit PDF s' gebruikt kunnen worden zijn:

GROBID

- GROBID is een machine learning gebaseerde tool dat gestructureerde informatie uit PDF-documenten kan lezen, over het algemeen wordt deze tool met name gebruikt op technische en wetenschappelijke bestanden.
- GROBID biedt functionaliteiten voor het analyseren en formaten van ruwe PDF-documenten naar XML/TEI-gecodeerde documenten dit maakt tekst mining en semantische analyses makkelijker.
- GROBID biedt functionaliteiten zoals het scheiden van bijvoorbeeld headers, referenties, citatiecontexten, en volledige tekstinhoud waardoor het veelzijdig is voor verschillende doeleinden binnen een onderzoek.

PyPDF2

- PDF2 is een softwaretool die PDF-bestanden kan omzetten naar andere formaten, zoals tekstbestanden of afbeeldingen.
- De PDF2 library biedt functionaliteiten voor het lezen van tekst, afbeeldingen en metadata uit PDF-documenten, waardoor het nuttig is voor gegevensanalyse, tekstverwerking en archivering.
- PDF2 ondersteunt verschillende programmeertalen en platforms, hierdoor is het zeer veelzijdig. In dit geval is de ondersteuning voor Python het belangrijkst
- PDF2 kan worden gebruikt voor batchverwerking van meerdere PDF-bestanden, hierdoor past het goed binnen deze opdracht aangezien er meerdere PDF-documenten verwerkt dienen te worden.
- PDF2 wordt regelmatig geupdate.

PyMuPDF

- PyMuPDF is een library gebaseerd op MuPDF waarmee PDF-bestanden kunnen worden geanalyseerd op vrijwel elk gebied.
- PyMuPDF kan afbeeldingen inlezen.
- PyMuPDF heeft een ingebouwde OCR voor het herkennen van tabellen.
- PyMuPDF is erg actief en wordt regelmatig bijgewerkt als of april 2024.

Dit haalt de onzekerheid weg over het inlezen van aparte zinnen en afbeeldingen uit een PDF-bestand, aangezien dit al te doen is via de PyDPF2 en PyMuPDF tools.

Afbeeldingen uit pdf halen

PyPDF2

Voordelen
- Eenvoudig te gebruiken voor het lezen en manipuleren van PDF-bestanden.
- Biedt basisfunctionaliteit voor het extraheren van afbeeldingen, tekst, pagina's en metadata uit PDF's.
- Vereist geen externe afhankelijkheden buiten de Python-standaardbibliotheek.

Nadelen
- Minder uitgebreide ondersteuning voor het extraheren van afbeeldingen in vergelijking met gespecialiseerde bibliotheken zoals PyMuPDF of pdfplumber.
- Mogelijk niet zo krachtig of efficiënt voor complexe taken zoals het manipuleren van PDF's met complexe structuren.

Een basisfunctie is gemaakt om afbeeldingen te extraheren maar hier kwamen veel corrupte afbeeldingen en errors bij kijken. Daarom is het waardevol andere libraries te proberen.

PyMuPDF (fitz)

Voordelen
- Biedt krachtige mogelijkheden voor het parsen van PDF's.
- Maakt extractie van zowel afbeeldingen als tekst mogelijk.
- Biedt goede prestaties.

Nadelen
- Vereist installatie van de MuPDF-bibliotheek, wat een extra stap kan zijn voor gebruikers.

pdfplumber

Voordelen
- Eenvoudig te gebruiken voor het extraheren van zowel tekst als afbeeldingen uit PDF's.
- Biedt goede compatibiliteit met verschillende PDF-formaten.

Nadelen
- Biedt mogelijk niet zoveel controle- of afstemmingsmogelijkheden als meer fundamentele bibliotheken zoals PyMuPDF.
- Beperkte documentatie vergeleken met meer populaire bibliotheken.

PyMuPDF heeft de meeste functionaliteiten van de drie en heeft een zeer uitgebreide documentatie met onder andere een User Guide en heeft uitleg voor extractie van afbeeldingen en tekst uit meerdere bestands types en afbeelding types.

Image captioning
Astica.ai heeft heel veel relevante en goed werkende functionaliteiten, maar is wel extern en maakt dus gebruik van API-calls die niet onder onze invloed vallen, maar dit kan een waardevolle back-up zijn voor tijdelijke functionaliteit die aan de rand van onze scope valt.

Tekst uit afbeeldingen lezen

Om ervoor te zorgen dat afbeeldingen ook inhoudelijk geanalyseerd kunnen worden is het nodig dat er elementen zoals tekst en tabellen hieruit gelezen kunnen worden. Hiervoor zijn al een aantal tools beschikbaar, voornamelijk OCR-software. Deze kunnen samen met PyPDF2 worden gebruikt. Met OCR zal de onzekerheid van tekst uit afbeeldingen halen worden opgelost.

Tesseract OCR (pytesseract):

Installatie

Tesseract is een veelgebruikte tool om tekst uit afbeeldingen te lezen. Om pytesseract te gebruiken moeten er een aantal instructies worden gevolgd aangezien deze niet alleen via pip te installeren is (downloadpagina).

Taalondersteuning

Tesserect bevat een aantal mogelijkheden voor talen. Het is wel belangrijk om op te merken dat alleen Engels standaard geïnstalleerd is. Andere taalpakketten zijn te downloaden vanaf internet. Deze zijn te downloaden op https://muthu.co/all-tesseract-ocr-options/

Configuratie

Door pytesseract te gebruiken kan tesseract vanuit python gebruikt worden, via pytesseract is de configuratie in te stellen waarmee Tesseract zal werken, opties luiden als volgt.

--psm NUM Specify page segmentation mode.
--oem NUM Specify OCR Engine mode.

Page segmentation modes:
0 Orientation and script detection (OSD) only.
1 Automatic page segmentation with OSD.
2 Automatic page segmentation, but no OSD, or OCR.
3 Fully automatic page segmentation, but no OSD. (Default)
4 Assume a single column of text of variable sizes.
5 Assume a single uniform block of vertically aligned text.
6 Assume a single uniform block of text.
7 Treat the image as a single text line.
8 Treat the image as a single word.
9 Treat the image as a single word in a circle.
10 Treat the image as a single character.
11 Sparse text. Find as much text as possible in no particular order.
12 Sparse text with OSD.
13 Raw line. Treat the image as a single text line,
bypassing hacks that are Tesseract-specific.

OCR Engine modes: (see https://github.com/tesseract-ocr/tesseract/wiki#linux)
0 Legacy engine only.
1 Neural nets LSTM engine only.
2 Legacy + LSTM engines.
3 Default, based on what is available.

Afkomstig uit: https://muthu.co/all-tesseract-ocr-options/

Output van Tesseract

Tesseract probeert de tekst uit de ingevoerde afbeelding te herkennen en geeft deze in de vorm van een string terug, afhankelijk van de opties die zijn geselecteerd in de config.

Werken met Apache Hop

Apache Hop is een open-source data-integratie- en ETL (Extract, Transform, Load) tool. Apache Hop biedt een grafische interface om data-integratieworkflows te ontwerpen en uit te voeren, waardoor gebruikers eenvoudig verbinding kunnen maken met verschillende databronnen, transformaties op de data kunnen uitvoeren en deze in doelsystemen kunnen laden. Daarnaast biedt Apache Hop mogelijkheden voor het plannen en monitoren van workflows, waardoor betrouwbare en efficiënte dataverwerkingspijplijnen worden gegarandeerd.

Over het algemeen is Apache Hop een krachtige en veelzijdige tool voor het beheren van de complexiteit van data-integratietaken in zowel enterprise- als open-sourceomgevingen. Hier volgt een overzicht van hoe het proces doorgaans verloopt:

- **Ontwerp van workflows**: Gebruikers ontwerpen data-integratieworkflows met behulp van de grafische interface van Apache Hop. Dit omvat het slepen en neerzetten van verschillende stappen, zoals het verbinden met databronnen, uitvoeren van transformaties en laden van gegevens in doelsystemen.
- **Verbinding met databronnen**: Apache Hop ondersteunt verschillende databronnen, waaronder databases, bestanden, API's en meer. Gebruikers kunnen verbinding maken met deze bronnen om toegang te krijgen tot de gewenste gegevens.
- **Data Transformaties**: Gebruikers kunnen transformatiestappen toevoegen aan de workflow om de gegevens te manipuleren en te transformeren volgens de vereisten van het project. Dit omvat taken zoals het filteren van rijen, het wijzigen van datatypes en het samenvoegen van gegevens.
- **Data Laden**: Nadat de gegevens zijn getransformeerd, kunnen ze worden geladen in doelsystemen, zoals databases, datawarehouses, bestanden, of andere eindpunten.
- **Plannen en Uitvoeren**: Gebruikers kunnen workflows plannen om op specifieke tijden of gebeurtenissen uit te voeren. Apache Hop biedt mogelijkheden voor het plannen en monitoren van workflows, zodat gebruikers de uitvoering kunnen beheren en eventuele fouten kunnen oplossen.
- **Monitoring en Rapportage**: Apache Hop biedt ook mogelijkheden voor het monitoren van de uitvoering van workflows en het genereren van rapporten over de gegevensintegriteit en prestaties van de processen.

Over het algemeen biedt Apache Hop een flexibele en krachtige oplossing voor het beheren van complexe data-integratietaken, met een intuïtieve interface en uitgebreide functies voor het ontwerpen, uitvoeren en beheren van workflows.

Hop heeft java runtime nodig om te werken of je kunt op een docker image donwloaden. De officiële java runtime versie voor Apache hop os versie 11.

Gebruikte bronnen:
https://hop.apache.org/manual/latest/getting-started/hop-download-install.html#_download
https://hop.apache.org/manual/latest/getting-started/index.html

Werken met Neo4J

Neo4J is voor de groep een onbekend onderwerp. Het is daarentegen wel een voorwaarde vanuit de opdrachtgever om dit als database te gebruiken en dit is dan ook de reden waarom we onderzoek hebben gedaan naar dit onderwerp, dit onderzoek is vooral bedoeld om te achterhalen hoe Neo4J werkt en om te kijken waarom deze methode in databases normaal wordt gebruikt.

Een Neo4j grafiekdatabase slaat knooppunten en relaties op in plaats van tabellen of documenten. Data wordt opgeslagen zoals je misschien ideeën op een whiteboard zou schetsen. Je data wordt opgeslagen zonder het te beperken tot een vooraf gedefinieerd model, wat een zeer flexibele manier van denken over en gebruiken ervan mogelijk maakt.

Wat maakt Neo4j de gemakkelijkste grafiek om mee te werken?

- Cypher, een declaratieve querytaal vergelijkbaar met SQL, maar geoptimaliseerd voor grafieken.
- Constante tijdtraversals in grote grafieken voor zowel diepte als breedte vanwege efficiënte representatie van knooppunten en relaties. Maakt opschaling mogelijk naar miljarden knooppunten op gematigde hardware.
- Flexibel eigenschappengrafiekschema dat zich in de loop van de tijd kan aanpassen, waardoor het mogelijk is om later nieuwe relaties te materialiseren en toe te voegen om de domeingegevens te versnellen wanneer de zakelijke behoeften veranderen.
- Drivers voor populaire programmeertalen, waaronder Java, JavaScript, .NET, Python en nog veel meer.

Terwijl bestaande relationele databases deze relaties kunnen opslaan, navigeren ze erdoorheen met dure JOIN-operaties of cross-lookups, vaak gekoppeld aan een rigide schema. Het blijkt dat "relationele" databases relaties slecht afhandelen. In een grafiekdatabase zijn er geen JOINs of lookups. Relaties worden native opgeslagen naast de gegevenselementen (de knooppunten) in een veel flexibelere indeling. Alles aan het systeem is geoptimaliseerd voor het snel traverseren van gegevens; miljoenen verbindingen per seconde, per kern.

Gebruikte bronnen:
https://neo4j.com/docs/getting-started/get-started-with-neo4j/graph-database/#:~:text=A%20Neo4j%20graph%20database%20stores,thinking%20about%20and%20using%20it.

PDF Tabellen inlezen.

Op een blog van gradient worden onder andere de volgende uitdagingen en problemen genoemd. “Inconsistent Layout, Lack of Structural Metadata, Complex Layouts and Formats, Quality Issues, Multimodal Content” (Gradient Blog: Untangling the Complexities of PDF Extraction, n.d.) Ook wordt in de documentatie van de Apache PDF tool tika gezegd: “Note that as of 2023, we still see papers at research conferences on extracting structural elements (including tables) from PDFs – THIS IS NOT A SOLVED PROBLEM. And, humorously, this kind of task is sometimes called "PDF remediation".” (Allison & Apache, 2023).

Hieruit is te concluderen dat het extraheren van tabellen uit PDF bestanden een modern probleem is waar nog steeds geen voleldige oplossing voor is. Dit is dus zeker waard om meer onderzoek naar te doen.

Voor het inlezen van de tabellen is de eerder genoemde PyMuPDF te gebruiken.

Verwijzingen tussen zinnen/woorden vinden

StanfordNLP biedt ondersteuning voor een verscheidenheid aan NLP-taken, waaronder coreferentie-resolutie, en is ontwikkeld door Stanford University.

pip install stanfordnlp

AllenNLP, ontwikkeld door het Allen Institute for AI, is een krachtige bibliotheek voor natuurlijke taalverwerking, inclusief coreferentie-resolutie.

pip install allennlp

NeuralCoref, gebouwd boven op de SpaCy NLP-bibliotheek, bood een gebruiksvriendelijke aanpak voor coreferentie-resolutie en was ontwikkeld door Hugging Face. Echter, de ontwikkeling van NeuralCoref is stopgezet, en het wordt nu als verouderd beschouwd.

Fast-Coref is een nieuwe bibliotheek gericht op efficiëntie en snelheid, gebaseerd op transformer-architecturen voor diep leren. Het behaalt uitstekende prestaties op verschillende benchmarks en wordt actief onderhouden.

In ons onderzoek hebben we uitgebreide evaluaties uitgevoerd van zowel **AllenNLP** als **NeuralCoref**. Helaas voldeden beide bibliotheken niet aan onze verwachtingen. Zelfs SpaCy's experimentele coreferentiefunctionaliteit was niet voldoende en werd als verouderd beschouwd, wat ons dwong elders te zoeken naar een geschikte oplossing.

Uiteindelijk bleken zowel AllenNLP als NeuralCoref, evenals de experimentele coreferentie van SpaCy, als verouderd te worden beschouwd. Dit betekent dat ze niet langer actief worden onderhouden of bijgewerkt, waardoor hun bruikbaarheid en betrouwbaarheid worden ondermijnd. Als gevolg daarvan richten we ons op meer actief onderhouden en bijgewerkte bibliotheken die voldoen aan moderne standaarden en vereisten voor NLP.

Tijdens ons onderzoek hebben we ook **StanfordNLP** grondig onderzocht als mogelijke coreferentie-oplossing. Hoewel StanfordNLP een robuuste set tools biedt voor verschillende NLP-taken, ontdekten we dat het op het gebied van coreferentie-resolutie niet voldeed aan onze verwachtingen omdat deze ook deprecated was. Na zorgvuldige overweging en evaluatie van verschillende opties hebben we besloten om te kiezen voor Fast-Coref als onze primaire coreferentie-oplossing. We kozen voor Fast-Coref vanwege zijn indrukwekkende prestaties op diverse benchmarks en zijn actieve onderhoud, wat in lijn is met onze behoefte aan een moderne en betrouwbare coreferentie-tool.

https://nlp.stanford.edu/projects/coref.shtml
https://docs.allennlp.org/v0.9.0/api/allennlp.models.coreference_resolution.html
https://spacy.io/universe/project/neuralcoref

https://github.com/shtoshni/fast-coref

Library voor het herkennen en verwerken van Europese talen

Als onderdeel van het onderzoek is gezocht naar een geschikte oplossing voor het herkennen en verwerken van Europese talen. Het vermogen om nauwkeurig talen te identificeren en te analyseren is van belang voor diverse toepassingen, zoals gebruikersinput begrijpen en tekstverwerkingsprocessen uitvoeren. Verschillende bibliotheken en tools zijn onderzocht om taalherkenning en -verwerking te vergemakkelijken, met als doel een oplossing te vinden die zowel nauwkeurigheid als efficiëntie biedt voor het onderzoeksproject.

- **Langdetect** is een eenvoudige Python-bibliotheek voor taalherkenning. Het ondersteunt meer dan 55 talen en is gebaseerd op de Compact Language Detector 2-bibliotheek van Google.

  pip install langdetect

- **Polyglot** is een natuurlijke taalpijplijn die massale meertalige toepassingen ondersteunt. Het biedt functionaliteiten zoals tokenisatie, herkenning van vernoemde entiteiten, taalherkenning en vertalingen.

  pip install polyglot

- **NLTK** is een populaire Python-bibliotheek die een breed scala aan taalverwerkingstaken ondersteunt, waaronder tokenisatie, lemmatisering, taggen van spraakdelen, en het uitvoeren van sentimentanalyse. Het biedt ook toegang tot verschillende taalcorpora en woordenboeken, waardoor het een veelzijdige tool is voor het verwerken van tekst in Europese talen.

pip install nltk

Er is gekozen voor **NLTK** vanwege de diverse taalcorpora die het biedt en de snelheid waarmee het verschillende taalverwerkingstaken kan uitvoeren. Deze keuze is gerechtvaardigd door de uitgebreide functionaliteit van **NLTK**, dat het in staat stelt om een breed scala aan taalkundige analyses uit te voeren. Bovendien maakt de snelheid van **NLTK** het een efficiënte keuze voor het verwerken van grote hoeveelheden tekstgegevens in Europese talen.

https://pypi.org/project/langdetect/
https://polyglot.readthedocs.io/en/latest/

Tools voor zinnen ontleden

Als onderdeel van het onderzoek naar natuurlijke taalverwerking zijn verschillende tools geëvalueerd die ondersteuning bieden bij het analyseren en verwerken van tekstuele gegevens.

- **Polyglot** is een natuurlijke taalpijplijn die massale meertalige toepassingen ondersteunt. Het biedt functionaliteiten zoals tokenisatie, herkenning van vernoemde entiteiten, taalherkenning en vertalingen.

pip install polyglot

- **SpaCy** is een krachtige bibliotheek voor NLP die een breed scala aan taalkundige verwerkingstaken ondersteunt, waaronder tokenisatie, lemmatisering, benoemde entiteitherkenning (NER), zinsanalyse en meer.

pip install spacy

De keuze voor SpaCy is gebaseerd op zijn uitgebreide functionaliteiten en efficiëntie, waardoor het een geschikte optie is voor het verwerken van grote hoeveelheden tekstuele data. Deze bibliotheek maakt diepgaande analyses mogelijk op tekstuele gegevens, wat cruciaal is voor het onderzoek.

https://pypi.org/project/langdetect/
https://polyglot.readthedocs.io/en/latest/

PDF reading tools

	Tekst	Afbeeldingen	Tabellen	Metadata	Positie	OCR	Fonts	API usage	Dependencies
PyPDF2	Ja	Ja	Nee[18]	Ja	Nee[19]	Nee[20]	Ja	Nee	Nee
Textract	Ja	Nee	Nee	Nee	Nee	Ja	Nee	Nee	Ja[21]
PyMuPDF	Ja	Ja	Ja[22]	Ja	Ja	Ja	Ja	Nee	Ja
PDFtotext	Ja	Nee	Nee[23]	Nee	Ja	Nee	Nee	Nee	Nee
PDFMiner	Ja	Ja	Nee	-	Ja	Nee	Ja	Ja[24]	Ja
Tabula	Nee	Nee	Ja	Nee	Nee	Nee[25]	Nee	Nee	Ja
Tika	Ja	Ja	Nee[26]	Ja	Ja	Ja[27]	Ja	Ja[28]	Ja

Hierboven is een tabel te zien waar de mogelijkheden voor het inlezen van PDF's met elkaar worden vergeleken. Uiteindelijk is gekozen voor PyMuPDF aangezien deze de meeste mogelijkheden biedt. Na verder praktisch onderzoek is deze keuze gerechtvaardigd door de goede resultaten.

Mogelijkheden voor PDF en PDFA.

PDFA is een type PDF bestand dat bestaat uit afbeeldingen. Om hieruit tekst te kunnen lezen is er OCR nodig(Optical Character Recognition). De gekozen PyMuPDF library ondersteund dit

Design

PDF-reader

Onze PDF-reader gebruikt functies van de PyMuPDF library om alle onderdelen uit de geleverde PDF's te halen. Hoe bepaalde onderdelen uit de PDF worden gehaald en wat er precies verder wordt gegeven aan andere onderdelen van de applicatie wordt binnen dit stukje beredeneerd.

Stap 1: Haal alle tekst op uit het PDF-bestand.
`self.read_text_from_pdf ( file )`

Stap 2: Deel de volledige tekst op in Textblocks.
`self.extract_textblocks_from_pdf ( file )`

Stap 3: Voeg de Metadata toe aan het tekstbestand.
`self.get_metadata ( file )`

Stap 4: Haal alle Afbeeldingen op uit het PDF-bestand.
`self.extract_images_from_pdf ( file, output_folder )`

Stap 5: Vraag de inhoudsopgave op als deze in dit bestand aanwezig is.
`self.get_table_of_contents ( self.file )`

Stap 6: Haal alle Tabellen op uit het PDF-bestand.
`self.get_tables ( file )`

Stap 7: Vraag de structuur op van het PDF-bestand.
`self.extract_full_text_structure ( self.file )`

Stap 8: Creëer een json met een gestructureerde versie van alle verzamelde informatie uit de PDF.

```
self.generate_paragraphs_per_file ( file )
self.generate_paragraphs_per_page ( file )
+
self.structure_paragraphs_per_file(structure_dict_for_file)
```

Dit leidt tot de volgende json-bestanden die worden doorgegeven aan het proces van neo4J;

- extracted_text_blocks.json uit stap 2
- metadata.json uit stap 3
- table_of_contents.json uit stap 5
- file_structured.json uit stap 7
- page_layout_structured_per_page.json uit stap 8
- page_layout_structured_per_file.json uit stap 8
- ierarchical_structure_for_file.json uit stap 8

Neo4j

De volgende onderdelen worden opgeslagen in de Neo4J graph database:

- Woorden
- Zinnen
- Textblocks
- Paragrafen
- Subtitels
- Titels
- Subteksten
- Pagina's
- Tabellen
- Afbeeldingen
- Bestanden
- Entiteiten

De structuur is als volgende opgebouwd:

1. Woorden zijn opgeslagen in zinnen.
2. Entiteiten zijn Woorden.
3. Zinnen zijn opgeslagen in Textblocks.
4. Paragraven/ Subtitels/ Titels/ Subteksten zijn Textblocks.
5. Textrblocks/ Tabellen/ Afbeeldingen zijn opgeslagen Pagina's.
6. Pagina's zijn opgeslagen in Bestanden.

Elke query is zo opgebouwd dat wanneer een onderdeel nog niet bestaat in de database dat deze wordt aangemaakt en wordt gekoppeld aan de andere onderdelen. De werkelijke structuur van een bestand wordt zo opgeslagen: Titel > Subtitel > Paragraaf > subtekst.

Queries die worden uitgevoerd voor het uploaden van data:

- Upload text
- Upload images
- Upload tables
- Upload hierarchial interpreted structure

Queries die worden uitgevoerd voor het strucureren van de database:

- Organize pages
- Organize blocks
- Organize words
- Connect matching words
- Connect blocks (verbind geïnterpreteerde structuur met absolute structuur)
- Make lemmetized nodes
- Connect words to lemmetized nodes
- Resolve coreferences and connect words

Coreference

Om verbanden en verwijzingen te herkennen binnen de tekst wordt er een vorm van Coreference toegepast, het model dat hiervoor wordt gebruikt is te vinden in de repository onder het volgende pad:

CirculairNLP\Utils\Coreference\assets

Het hoofddoel van Coreference binnen onze applicatie is dat deze wordt gebruikt om entiteiten te herkennen en te vergelijken over files om zo te herkennen in hoeverre er over circulariteit en duurzaamheid gesproken wordt.

Zo werkt het:

Stap 1: Het model voor Coreferencing wordt geïnitialiseerd.
Self.__init__ (self, model_path)

Stap 2: Coreferencing wordt uitgevoerd.
Self.resolve_coreferences (self, doc)

Stap 3: Clusters uit de Coreferencing worden opgedeeld.
Self.resolve_coreferences (self, doc)

Stap 4: Woorden uit de tekst worden getokenized en stopwoorden worden verwijderd.
Self.text_similarity (self, text1, text2)

Stap 5: Gelijkenissen worden herkend.
Self.text_similarity (self, text1, text2)