Paper bachelorproef: applying NLP

Chun Dongho
Universiteit Hasselt
Hasselt, Belgium
dongho.chun@student.uhasselt.be

***Abstract*—Hergebruik van data wordt in deze gedigitaliseerde wereld steeds belangrijker. Een van de belangrijkste soorten data is gezondheidsdata en we zien pas recent dat er een opkomst is van initiatieven die deze soorten data meer beschikbaar maken en meer structuur hierin brengen. Dit verbetert de toegankelijkheid, maar het hergebruik van deze data blijft nog steeds een uitdaging door het feit dat de relaties tussen deze initiatieven slecht gedocumenteerd zijn. In dit onderzoek probeerden we vooral te achterhalen welke van deze initiatieven een sterk verband hebben met elkaar op basis van hun beschrijvingen.**
Dit werd in 4 stappen gedaan door bepaalde Natural Language Processing (NLP) technieken toe te passen. De eerste stap, *Data Extractie*, bestaat uit *webscraping* en *pre-processing*. Vervolgens probeerden we de sleutelwoorden te identificeren in de data door *Term Frequency-Inverse Document Frequency (TF-IDF)* toe te passen. In de derde stap vergeleken we deze sleutelwoorden met behulp van een *clustering* algoritme. Ten slotte visualiseerden we de resultaten in een tabel.
Deze tabel toont aan welke initiatieven een sterke relatie hebben met elkaar of gelijkaardig zijn aan elkaar door deze initiatieven in dezelfde kolom te groeperen. Daarnaast geven we ook de sleutelwoorden weer die deze groepen of kolommen beschrijven. We verkregen bruikbare en interessante resultaten die zeker het manuele proces om deze initiatieven te identificeren kunnen versnellen. Echter, in deze paper benadrukken we ook dat dit slechts een hulpmiddel is en dat er geen "correcte" oplossing bestaat.

I. Introductie

Het structureren en harmoniseren van gezondheidsdata is noodzakelijk voor het hergebruik van deze data. Data harmonisatie is een zeer actief onderwerp en verwijst naar pogingen die worden gedaan om data die oorspronkelijk uit verschillende bronnen komen en hierdoor ook in verschillende formaten of dimensies staan, op een gestructureerde manier te combineren, zodat deze data meer leesbaar en bruikbaar is voor de gebruikers van deze data [13]. Dit is echter een moeilijke taak door verschillende problemen en uitdagingen die ontstaan tijdens dit proces. Deze zijn vooral socio-technische uitdagingen waarbij beroepsbeoefenaars in de gezondheidssector of gewone burgers vaak niet gemotiveerd zijn of de nood niet zien in het harmoniseren van data. Verder is het ook mogelijk dat verschillende aandeelhouders van een bepaald project andere noden hebben. In een voorbeeld waar binnen een ziekenhuis een werknemer in de ICT sector snel gemotiveerd zou zijn om data te delen met externe partners, zou een arts deze data niet snel willen delen om wille van de bescherming en de aard van deze gevoelige data. Deze coördinatie problemen veroorzaken niet enkel vertraging, maar ook conflicten tussen de aandeelhouders. In het medische domein zijn er verschillende initiatieven die opgericht zijn om zulke problemen op te lossen. Hierdoor ontwikkelden Peeters en Geys [8] een *strategic oversight* die een overzicht toont van deze volgende aspecten van een initiatief:

- Data type, kwaliteit en categorie waarmee ze werken.
- Hoe krijgt men toegang tot de data?
- Welke aandeelhouders zijn erbij betrokken?
- Hoeveel en welke type partners?
- Welke initiatieven proberen de eerder aangehaalde problemen op te lossen?
- Welk specifiek probleem lost een bepaalde initiatief op?
- Geografische scope
- Kosten
- Relaties met andere initiatieven.
- Publicaties

Een van de grote problemen is het gebrek aan goede documentatie van deze informatie op de initiatieven. Dit leidt tot het probleem dat men moeilijk kan achterhalen of twee initiatieven met verschillende namen al dan niet gerelateerd zijn aan elkaar. Een initiatief kan bijvoorbeeld over tijd van naam veranderen, maar door het gebrek aan documentatie kan men niet achterhalen of twee initiatieven met verschillende namen eigenlijk hetzelfde initiatief zijn [8].
Dit probleem is de aanleiding tot dit project. In dit project zal er geprobeerd worden om de volgende onderzoeksvragen te beantwoorden:

1) Hoe kunnen de initiatieven met elkaar vergeleken worden?
2) Hoe kunnen de initiatieven die sterk verbonden zijn met elkaar vervolgens geïdentificeerd worden op basis van hun individuele beschrijving?

Om deze onderzoeksvragen op te lossen wordt er in dit project gebruik gemaakt van Natural Language Processing (NLP). Het *natural language* deel staat voor menselijke taal in tekst of spraak vorm en het *processing* deel staat voor het verwerken van deze taal zodat de computer of de machine deze taal begrijpt [11]. In de context van dit project verwijst de data uit de dataset naar het natural language gedeelte en het verwerken en omzetten van deze data naar numerieke waarden, die de computer begrijpt, naar het processing gedeelte. Vervolgens worden deze numerieke waarden verder verwerkt door specifieke technieken toe te passen om initiatieven te vinden die sterk verbonden zijn met elkaar. Deze technieken zullen verder in deze paper besproken worden.

II. Dataset

De gebruikte dataset voor dit project is afkomstig uit de eerder aangehaalde paper [8]. Deze dataset bestaat uit een Excel sheet die in totaal 68 initiatieven bevat samen met andere belangrijke informatie op deze initiatieven zoals de regio en landen waarin ze opereren, de link naar hun website, een korte beschrijving van de initiatieven, het startjaar, federatieve of centrale aanpak, sociaal of technisch, enkel gezondheidsdomein of meerdere domeinen, type data en contact informatie [7]. In Tabel I wordt een klein deel van deze dataset getoond voor 2 initiatieven. In dit project zijn we vooral geïnteresseerd in de data in de "link" en "short description" kolom. De eerste kolom bestaat uit een URL naar de website van het bijhorende initiatief. Deze websites bevatten informatie over hun bijhorende initiatief die niet in de originele dataset teruggevonden kunnen worden. En de tweede kolom bestaat uit een korte beschrijving van het bijhorende initiatief.
De originele dataset en een groot deel van alle websites zijn in het Engels geschreven. Dit is een belangrijk aspect, omdat deze taal verder in dit project gebruikt wordt. En later in deze paper zal er ook besproken worden waarom het gebruiken van een andere taal slechte resultaten oplevert.

III. Methode

In deze paper representeert de term *term* een woord binnen een *document*. En een document representeert een multiset (bag) van deze termen die het initiatief beschrijft. Verder is een *corpus* de verzameling van deze documenten. Vervolgens zijn er bepaalde variabelen die vaak zullen voorkomen zoals de variabele t die een term voorstelt, d die een document voorstelt en D die het corpus voorstelt.

A. Webscraper

Eerst moet er bruikbare data geëxtraheerd worden uit de gegeven dataset. Een website van een bepaald initiatief, die in de "link" kolom wordt teruggevonden zoals in Tabel I, bevat veel informatie over het initiatief. Dus er moet een methode bedacht worden om data (tekst) uit de websites te *scrapen*. Hiervoor kan er een *webscraper* ontwikkeld worden. Een webscraper is een programma dat zich kan verbinden met een website via de URL en de HTML code van de website kan extraheren. Deze HTML code wordt vervolgens gefilterd door elementen die onnodig zijn te verwijderen en elementen die nodig zijn te behouden [14]. De output van de webscraper is een apart .txt bestand voor elk initiatief, waarin de gescrapete data van de bijhorende website uitgeschreven is. In Tabel II wordt een voorbeeld getoond van een zeer klein deel van een willekeurige output van de webscraper. In dit voorbeeld representeert elke zin één document (in de praktijk bestaat het uit meerdere zinnen). Het corpus bestaat dus uit 4 documenten en in de volgende secties zullen deze documenten opnieuw als voorbeeld gebruikt worden.

B. Pre-processing

Vervolgens wordt er een proces gebruikt om deze bestanden op te schonen door onnodige woorden of karakters te verwijderen of te vervangen. Het voornaamste doel van deze stap is dus om de kwaliteit van de ruwe datasets te verhogen door enkel de interessante en relevante woorden en karakters bij te houden. Een bijkomend voordeel is dat door deze onnodige woorden en karakters te verwijderen, we ook tijd en geheugen kunnen besparen tijdens het verwerken van deze data. Dit proces, dat *pre-processing* wordt genoemd, is een cruciale stap voor veel NLP applicaties [18]. Maar deze ruwe datasets kunnen veel onnodige opeenvolgende spaties of *EOL* karakters bevatten tussen 2 verschillende woorden (bv. BDVA is an initiative\n\n\n), waardoor deze eerst uitgefilterd worden voor het volgende proces.

1) Lemmatization en stemming: In deze bestanden komen er veel woorden in voor die in een vervoegde of verbogen vorm staan. In essentie hebben deze woorden dezelfde betekenis als hun stamvorm. Deze stamvorm wordt ook wel *lemma* genoemd. Zo kan een bestand bijvoorbeeld de woorden "studied", "studying" en "studies" bevatten. Deze woorden zijn vervoegingen of verbuigingen van hun lemma "study" en behouden nog altijd dezelfde betekenis als hun lemma. Daarom is het belangrijk om deze onnodige vormen van het stamwoord te vervangen door hun lemma. Om deze stap te implementeren, werden er 2 methodes overwogen: *lemmatization* en *stemming*. Beide methodes vervangen alle verbuigingen en vervoegingen van woorden binnen deze ruwe data naar hun lemma, maar het verschil tussen deze 2 methoden is dat stemming een bepaald woord omzet naar zijn lemma door prefixen en suffixen te verwijderen. Het nadeel hiervan is dat het woord niet altijd grammaticaal correct is. Maar omdat deze techniek enkel voorgedefinieerde regels volgt om de lemma te bepalen heeft het als voordeel dat deze techniek relatief snel is. Bij lemmatization wordt er geprobeerd om de grammaticaal correcte lemma te vinden. Echter is het nadeel dat deze techniek in vergelijking met stemming trager is. Er bestaan verschillende variaties van lemmatization, maar een veel voorkomende variatie gebruikt een dictionary waar een enorme hoeveelheid aan lemma's opgeslagen zijn. En als de gebruikte library van deze techniek een bepaald woord niet herkent in de dictionary, kan dit woord niet vervangen worden door zijn lemma [18].
In dit project werd er gebruik gemaakt van lemmatization.

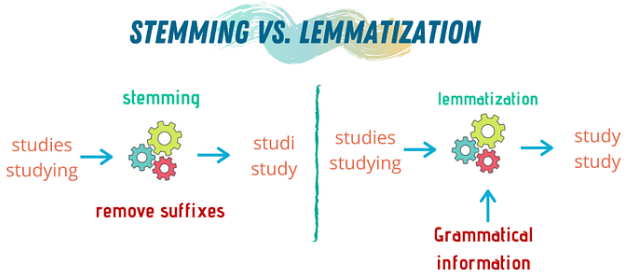

Fig. 1. Voorbeeld stemming vs lemmatization [5]

De belangrijkste reden is dat stemming vooral wordt gebruikt wanneer snelheid een grote factor is. Voor dit project ligt de focus niet op snelheid maar op kwaliteit van de woorden, waardoor lemmatization verkozen werd over stemming. Hiervoor werd `NLTK WordNetLemmatizer` gebruikt om per

TABLE I
EEN DEEL VAN DE ORIGINELE DATASET (ARWEN EN DARWIN) [7]

Name initiative	Region	Countries	Link	short description	Year of Initiation	central/ federated	social/ technical	Health specific or multiple domains	data	contact information
Actionable RWE Network (ARWEN)	Europe	Europe, but also projects specific for: UK The Netherlands	https://arwen.eu/how-it-works/	Platform for RWE projects (RWD!) for hospitals (data owners) prepare, validate, anonymise and harmonise data project outcomes are shared with all members involved ARWEN is committed to build a European data network ...	?	Federated	mainly technical	Health	Claims and care activity data Prescription data CROs PROs Pathology& lab data Other diganostics data Genomics data Hospital data (the data belongs to the hospitals)	contact form on website (”Let’s talk”)
DARWIN (Data Analysis and Real-World Interrogation Network in the European Union)	Europe	Countries of current data partners: Finland France UK Estonia Spain The Netherlands Germany Belgium	https://www.darwin-eu.org/	First priority of the HMA-EMA joint Big data steering group work plan; = a network of data, expertise and services; EMA will have responsibility for managing the network and act as a data holder within it aims to become a node in the EHDS ...	2021	federated	technical	Health	Check out the current list of data sources here Data partners are partners who have access to raw data in-house or remotely, through ownership, public contract, third-party agreement or commercial license. Data partners enable DARWIN EU® to use their data in its scientific studies by ...	https://www.darwin-eu.org/index.php/contact

TABLE II
VOORBEELD: 4 DOCUMENTEN

Document nr.	Document
1	A car is driven on the road.
2	A truck is driven on the highway.
3	A train is driven on the tracks.
4	A plane is flown in the sky.

initiatief alle woorden uit de ruwe dataset om te zetten naar hun lemma. De `NLTK WordNetLemmatizer` gebruikt de `WordNet` database waar zeer grote hoeveelheden aan Engelse woorden opgeslagen zijn. Deze woorden worden gegroepeerd in verzamelingen van synoniemen, deze worden ook *synsets* genoemd. Dit betekent dus dat elke synset woorden bevatten die een gelijke betekenis hebben of hetzelfde concept beschrijven. Het is ook belangrijk om te vermelden dat alle woorden in de synsets in hun stamvorm staan. Wanneer de WordNetLemmatizer een woord als input krijgt, gebruikt het de ingebouwde `morphy` functie van `WordNet`.
Deze functie kan 2 processen uitvoeren:

- Door een lijst te gebruiken met *inflectional endings*, kan de lemmatizer deze endings verwijderen uit het meegegeven woord. Een inflectional ending is een suffix die aan een lemma toegevoegd kan worden om de verbuiging van dat woord te veranderen, zonder dat deze lemma syntactisch verandert (bv. cat, cats). `Morphy` kan vervolgens een look-up uitvoeren in de `WordNet` database om het nieuwe woord te vinden. Als deze gevonden is, is dat woord de lemma van het meegegeven woord.
- Het tweede proces is gelijkaardig aan het eerste proces, maar dit keer gebruikt `morphy` een *exception* lijst voor uitzonderingen die de standaard morfologische regels niet volgen (bv. went, go). Daarna wordt er opnieuw een look-up gedaan in de `WordNet` database.

`Morphy` combineert deze 2 processen op een efficiënte manier om de lemma van het meegegeven woord te vinden [16]. De `WordNetLemmatizer` toegepast op de documenten uit Table II, geeft Table III als resultaat.

TABLE III
VOORBEELD: RESULTATEN NA LEMMATIZATION

Document nr.	Document
1	A car is drive on the road.
2	A truck is drive on the highway.
3	A train is drive on the track.
4	A plane is fly in the sky.

2) Stopwoorden: In de volgende pre-processing stap worden de *stopwoorden* verwijderd uit de dataset verkregen uit de lemmatization stap. Stopwoorden zijn woorden die zeer vaak voorkomen in zinnen, maar weinig betekenis geven aan deze zinnen (bv. the, a, an, so, ...). Door deze woorden te verwijderen, worden de datasets kleiner en als gevolg heeft dit invloed op de uitvoertijd van het programma dat deze datasets moet verwerken. Maar het grootste voordeel is dat termen die onbelangrijk zijn en niet veel betekenis geven aan een document uitgefilterd kunnen worden [9]. Voor deze implementatie werd opnieuw de `NLTK` library gebruikt om een verzameling van alle Engelse stopwoorden in te laden. Door voor alle woorden uit de datasets na te gaan of ze in deze verzameling voorkomen, kunnen deze woorden verwijderd worden uit de dataset als ze in de verzameling voorkomen. Het resultaat van deze stap is een .txt bestand voor elk initiatief, dat verkregen werd uit de vorige stap (lemmatization), zonder de stopwoorden.
In Tabel IV wordt het resultaat getoond van de documenten uit Tabel III na het verwijderen van de stopwoorden.

3) Leestekens en nummers: Pre-processing kan nog verder geoptimaliseerd worden door leestekens en nummers uit de datasets te verwijderen. Leestekens en nummers zijn opnieuw karakters die geen of weinig betekenis geven aan een zin en door deze uit te filteren, kunnen de dataset groottes opnieuw gereduceerd worden [18]. In Tabel V wordt er opnieuw getoond hoe de documenten uit Tabel IV eruit zien na deze pre-processing stap. Dit is de finale pre-processing stap die

TABLE IV
VOORBEELD: RESULTATEN NA VERWIJDERING STOPWOORDEN

Document nr.	Document
1	car drive road.
2	truck drive highway.
3	train drive track.
4	plane fly sky.

geïmplementeerd werd in dit project en aan de resultaten is het te zien dat elk document enkel nog de belangrijke en betekenisvolle termen bevat.

TABLE V
VOORBEELD: RESULTATEN NA VERWIJDERING LEESTEKENS EN NUMMERS

Document nr.	Document
1	car drive road
2	truck drive highway
3	train drive track
4	plane fly sky

C. TF-IDF

TF-IDF staat voor Term Frequency - Inverse Document Frequency en is een techniek die in dit project gebruikt wordt om alle termen in de documenten, die verkregen zijn uit de pre-processing stap, om te zetten naar numerieke scores tussen 0 en 1. Een TF-IDF score van een term is een meting die de "importance" van een term representeert binnen een bepaald document relatief tot het corpus. Dit concept komt uit het idee dat een document beschreven of gerepresenteerd kan worden door woorden die vaak voorkomen in dat document, maar weinig of zelden voorkomen in andere documenten (corpus). De TF score is een meting voor hoe frequent een bepaalde term voorkomt binnen een specifiek document. Er wordt een hoge TF score toegewezen aan een bepaalde term als deze term frequent voorkomt relatief tot andere termen in het document waar deze term zich bevindt. Er bestaan verschillende methodes om de TF score te berekenen, maar de meest gebruikte en bekende methode is de volgende formule [15, 17]:

$$\mathrm{tf}(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$f_{t,d}$ = het aantal keer dat term t voorkomt in document d
$\sum_{t' \in d} f_{t',d}$ = het totaal aantal termen in document d

De IDF score is opnieuw een score die toegewezen wordt aan een bepaalde term, maar deze score bepaalt hoe vaak of zelden deze term voorkomt in het corpus. Een bepaalde term in een document krijgt een hoge IDF score als dat woord weinig of zelden voorkomt in alle documenten. De meeste gebruikte formule is de volgende [15, 17]:

$$\mathrm{idf}(t,D) = \log \frac{1+N}{1+|\{d : d \in D \text{ and } t \in d\}|} + 1$$

N = totaal aantal documenten in het corpus D
$|\{d : d \in D \text{ and } t \in d\}|$ = het aantal documenten waar term t voorkomt

Intuïtief zouden we kunnen zeggen dat enkel de deling in de formule voldoende is om IDF te berekenen. Maar in praktijk wordt de logaritme genomen van deze deling, zodat zeer kleine of grote waarden van de noemer niet te veel invloed hebben op de score. In een voorbeeld waar het corpus 1000 documenten bevat zou een woord zoals "serendipity", dat slechts in 1 document voorkomt, een IDF score krijgen van 1000 ($= \frac{1000}{1}$). Maar een veelvoorkomend woord zoals "the" dat in alle 1000 documenten voorkomt, zou een IDF score van 1 ($= \frac{1000}{1000}$) krijgen. Het is duidelijk dat de IDF score van 1000 te veel invloed zal hebben op de totale TF-IDF score, 1000 keer de invloed van de IDF score van "the" en dit zou onevenredig blijven stijgen als het totaal aantal documenten wordt verhoogd. Door de logaritme te nemen van deze scores, normaliseren we deze invloed ($\log 1 = 0$, $\log 1000 = 3$).
Ten slotte wordt de TF-IDF score van een bepaald woord berekend door de TF score van dat woord te vermenigvuldigen met de IDF score van dat woord:

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D)$$

Het is vanzelfsprekend dat een woord dat vaak voorkomt in een bepaald document (hoge TF score), maar in weinig andere documenten voorkomt (hoge IDF score), een hoge TF-IDF score krijgt. Meer in de context van dit project betekent het dat een woord met een hoge TF-IDF score een "belangrijk" woord is binnen het document waar het zich bevindt relatief tot het volledige corpus. Deze belangrijke woorden in een document, zijn dus sleutelwoorden die betekenis geven aan dat document en het document ook "beschrijven". Zoals eerder vermeld zijn er verschillende variaties van formules voor TF en IDF. Elke variatie kan gebruikt worden voor andere doeleinden. Dit betekent ook dat er verschillende implementaties bestaan die verschillende variaties van deze formules gebruiken. Voor dit project werd de `TfidfVectorizer` van de `sklearn` library gebuikt [1] om alle documenten om te vormen naar een lijst van TF-IDF scores. Er is een grote hoeveelheid aan parameters die geïnitialiseerd kunnen worden, maar hier zullen enkel de meest relevante parameters besproken worden. De eerste parameter die hier besproken zal worden is de `smooth_idf` parameter. Deze parameter kan op True of False gezet worden afhankelijk van de keuze voor de IDF formule die in de `TfidfVectorizer` gebruikt wordt [2]:
`smooth_idf` = False:

$$\mathrm{idf}(t,D) = \log \frac{N}{|\{d : d \in D \text{ and } t \in d\}|} + 1$$

`smooth_idf` = True:

$$\mathrm{idf}(t,D) = \log \frac{1+N}{1+|\{d : d \in D \text{ and } t \in d\}|} + 1$$

[1] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
[2] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

Deze formules zijn zeer gelijkaardig aan de vorige standaard formule voor IDF. Het opvallendste verschil is dat er + 1 wordt gedaan vanachter in beide formules. Dit zorgt ervoor dat een term die in elk document voorkomt niet 0 als IDF waarde krijgt, zodat deze term niet volledig genegeerd wordt. In de tweede formule wordt er nog + 1 gedaan in de teller en in de noemer, zodat deling door 0 niet mogelijk is. Er is ook een optie om IDF volledig uit te zetten door de `use_idf` parameter of False te zetten waardoor de IDF score standaard op 1 gezet wordt voor alle termen. Vervolgens zijn er een aantal parameters die geïnitialiseerd kunnen worden voor bepaalde pre-processing stappen zoals `lowercase` en `stop_words` en verder zijn er optimalisatie parameters zoals `min_df` en `max_df` die ervoor zorgen dat de termen genegeerd worden die respectievelijk onder een minimum *df* of boven een maximum *df* zitten (*df* = in hoeveel documenten een bepaalde term voorkomt).
Het uiteindelijke resultaat van de `TfidfVectorizer` is een matrix waar de TF-IDF scores van alle termen binnen elk document erin staan. Afbeelding 2 toont een voorbeeld van de TF-IDF matrix dat het resultaat is van de `TfidfVectorizer` toegepast op de documenten in Table II. Zoals eerder vermeld, worden deze documenten eerst verwerkt in de pre-processing stap. Alle unieke termen die na de pre-processing stap overblijven, staan in de kolomtitels. En verder komt elke rij ook overeen met één document.

car	drive	fly	highway	plane	road	sky	track	train	truck
0.645	0.412	0	0	0	0.645	0	0	0	0
0	0.412	0	0.645	0	0	0	0	0	0.645
0	0.412	0	0	0	0	0	0.645	0.645	0
0	0	0.578	0	0.578	0	0.578	0	0	0

Fig. 2. Voorbeeld TF-IDF matrix na pre-processing

D. Clustering

Clustering is de techniek die in dit project gebruikt wordt om gelijkaardige initiatieven te groeperen met elkaar. Zulke groepen worden clusters genoemd en er bestaan verschillende clustering algoritmes zoals k-means of agglomerative hierarchical clustering algoritmes. Voor dit project werd het k-means clustering algoritme verkozen, omdat het één van de populairste clustering algoritmes is.
Het finale doel van het k-means algoritme is om k aantal clusters te genereren waar elke cluster datapunten bevat die "gelijkaardig" zijn met elkaar. In dit algoritme wordt elke cluster gerepresenteerd door zijn *centroid*. Een centroid is het middelpunt van een cluster en doorheen het algoritme worden de coördinaten van deze centroids telkens geüpdatet totdat een bepaalde conditie bereikt wordt, dit wordt "convergence" genoemd. Het "k" gedeelte van k-means staat voor de parameter k die eerst geïnitialiseerd moet worden voordat het algoritme gebruikt wordt. Deze k representeert het aantal clusters of centroids dat men uiteindelijk wilt verkrijgen. Het algoritme werkt als volgt [12]:

1) Kies een k
2) Initialiseer willekeurig k centroids
3) Voor elk datapunt (= document):
   a) Zoek de dichtstbijzijnde centroid
   b) De cluster van die centroid is nu waar dit datapunt bijhoort
4) Update de coördinaten van de centroids door voor elke cluster het middelpunt/gemiddelde te berekenen van alle datapunten binnen de bijhorende cluster
5) Herhaal stap 3 en 4 totdat de eindconditie bereikt wordt (convergence)

Omdat elk document gerepresenteerd wordt als een vector of lijst van TF-IDF scores, kan elke vector of lijst in deze context gerepresenteerd worden als een datapunt. Hierdoor kan dit algoritme de Euclidische afstand tussen elk datapunt en de centroids berekenen om de dichtstbijzijnde centroid te identificeren. De formule om de euclidische afstand tussen 2 punten (x, y) te meten is als volgt [1]:

$$d_E(x, y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + ... + (x_n - y_n)^2}$$

n = dimensie van de datapunten

De eindconditie of convergence kan door 2 manieren bereikt worden:

1) Updates op de coördinaten van de centroids zijn niet significant genoeg.
2) Een bepaald aantal iteraties zijn bereikt.

Er wordt in dit project gebruik gemaakt van de `KMeans` klasse uit de `sklearn` library om k-means clustering toe te passen op de datapunten uit de vorige stap. Deze klasse vereist opnieuw een aantal parameters die meegegeven moeten worden. De meest vanzelfsprekende parameter is de `n_clusters` parameter, deze bepaalt het aantal clusters of centroids die gemaakt moeten worden. De `init` parameter is een zeer belangrijke parameter die de initialisatie van de centroids bepaalt in de beginfase van het algoritme. Deze kan een array ontvangen waar men handmatig de centroid posities initialiseert, maar een meer geautomatiseerde methode is het gebruiken van de strings `k-means++` of `random`. Bij random worden de centroids op willekeurige posities geïnitialiseerd en bij `k-means++` wordt er een algoritme gebruikt die op een empirische manier de centroids zo goed mogelijk probeert te initialiseren [2]. De initialisatie van de centroids is een zeer belangrijke stap, omdat het afhankelijk van deze initialisatie een slecht of een goed resultaat oplevert. Door het randomness aspect van `k-means++`, is het ook belangrijk om het algoritme meerdere keren uit te voeren en het beste resultaat eruit te halen.
De `max_iter` parameter zorgt ervoor dat de tweede eindconditie kan bereikt worden. Deze parameter verwacht een integer die het maximum aantal iteraties dat uitgevoerd mag worden binnen het k-means algoritme specificeert.
Een aantal termen die vermeld moeten worden voor de eerste eindconditie zijn "inertia" (e_{in}) en "distortion" (e_{dis}). Inertia is de som van de kwadratische afstanden tussen de datapunten en hun dichtstbijzijnde centroid of in het Engels "Within-Cluster

Sum of Squares (WCSS)". Bij distortion wordt het gemiddelde van de inertia genomen door deze te delen door het totaal aantal datapunten N [6].
Inertia [4]:

$$e_{in} = \sum_{i=1}^{N} d_E(x_i, C_k)^2$$

Distortion:

$$e_{dis} = \frac{1}{N} \sum_{i=1}^{N} (x_i - C_k)^2$$

Deze termen werden hier aangehaald, omdat deze gebruikt worden in de `tol` parameter om de eerste eindconditie te bereiken. Deze parameter neemt als waarde een floating point getal aan dat aangeeft hoe groot het verschil in inertia mag zijn tussen 2 opeenvolgende iteraties, concreet betekent dit dat het algoritme stopt wanneer het verschil in inertia tussen 2 opeenvolgende iteraties kleiner is dan de waarde die aan `tol` meegegeven werd.
De laatste parameter die besproken zal worden is de `algorithm` parameter. Deze kan 2 strings aannemen: `lloyd` en `elkan`. Het Elkan's algoritme is een variatie van het Lloyd's algoritme die de snelheid van het algoritme optimaliseert door veel afstandsberekeningen te vermijden. Het algoritme vermijdt deze berekeningen door de driehoeksongelijkheid te gebruiken, maar het nadeel is dat dit algoritme intensief geheugen gebruikt en als gevolg geen goede keuze is voor grote hoeveelheden aan clusters [10, 3].

IV. De pipeline

In deze sectie wordt er beschreven welke stappen er gevolgd werden om het doel te realiseren door gebruik te maken van een pipeline met 4 stappen.

Fig. 3. De pipeline met 4 stappen

A. Data extractie

De eerste stap in de pipeline is om bruikbare data te extraheren uit de dataset. Zoals eerder vermeld, ligt de focus op 2 bronnen voor data. De eerste bron is de "short description" kolom uit de Excel sheet voor elk initiatief. De Excel sheet wordt dus eerst ingeladen, zodat er toegang is tot deze kolommen. Voor elk initiatief wordt er een nieuw betand aangemaakt waar de tekst uit de corresponderende "short description" wordt toegevoegd. Daarna haalt de webscraper nodige data uit de bijhorende website door tot niveau 2 te scrapen. Dit houdt in dat de webscraper eerst naar de home-pagina navigeert en deze pagine scrapet. Daarna worden alle links geïdentificeerd die op de home-pagina staan door alle `<a>` tags te selecteren en deze links te scrapen. Door de `langid` library te gebruiken wordt de gebruikte taal op een webpagina achterhaald, hierdoor worden de webpagina's die niet in het Engels staan automatisch uitgefilterd. In deze stap worden ook de eerder aangehaalde pre-processing stappen toegepast. Elk bestand wordt eerst ingelezen en vervolgens gelemmatized. Deze data wordt opnieuw in een apart bestand opgeslagen voor elk initiatief. Verder wordt opnieuw elk bestand uit de lemmatization stap ingelezen waar de stopwoorden verwijderd worden en opnieuw opgeslagen worden in een nieuw bestand voor elk initiatief. En ten slotte kunnen de leestekens en nummers uitgefilterd worden uit deze bestanden. Dit werd geïmplementeerd door voor elk karakter uit de datasets na te gaan of ze binnen dit domein zitten: a-z of A-Z. Als een karakter niet binnen dit domein zit, wordt deze vervangen door een leeg karakter. Als resultaat krijgen we voor elk initiatief een apart bestand met enkel sleutelwoorden die het initiatief "beschrijven". Deze bestanden worden vervolgens in de volgende stappen gebruikt.

B. Sleutelwoorden vinden

In deze stap moeten de belangrijkste woorden geïdentificeerd worden binnen de gefilterde datasets, die uit de laatste pre-processing stap verkregen zijn. Hiervoor wordt de `TfidfVectorizer` gebruikt om TF-IDF toe te passen op de woorden uit deze datasets. Zoals eerder aangehaald, vormt elke dataset van een initiatief een document en de verzameling van deze documenten een corpus. In de implementatie worden deze documenten in een lijst opgeslagen en verder meegegeven als parameter aan de `fit_transform` functie (van `TfidfVectorizer`).

C. Sleutelwoorden vergelijken

De voorlaatste stap in de pipeline is het vergelijken van de sleutelwoorden. Hiervoor wordt de TF-IDF matrix uit de vorige stap gebruikt, omdat de elementen in deze matrix de "importance" van een specifiek woord binnen elk document representeren. Deze elementen zijn numerieke waarden waardoor ze met elkaar vergeleken kunnen worden.
In de implementatie werd k-means clustering gebruikt om deze waarden in groepen te verdelen. Voordat dit algoritme toegepast wordt, moet de parameter k gekozen worden. Deze parameter representeert het gewenste aantal clusters. Deze werd initieel manueel gekozen om te experimenteren met verschillende waarden en om de resultaten te observeren, maar dit werd later vervangen door de *elbow method* om automatisch de meest optimale waarde voor k te vinden. Dit werd geïmplementeerd door voor elke mogelijke k (1 tot 68), het k-means algoritme 10 keer uit te voeren. Voor elke iteratie van k wordt de gemiddelde inertia en distortion berekend nadat het algoritme 10 keer uitgevoerd is. Deze waarden worden ten slotte geplot waarbij de x-as de waarden voor k representeert en de y-as de inertia of distortion representeert. Door de "elbow" in deze functie te identificeren, wordt de meest optimale waarde voor k gevonden.
Voor het k-means clustering algoritme moeten er verschillende parameters meegegeven worden. In de tabel hieronder worden de gebruikte parameters vermeld. De waarde

van `n_clusters` is k, omdat er geëxperimenteerd werd met verschillende waarden voor k en er is geen correcte oplossing voor deze parameter. Vervolgens krijgt de `init` parameter de waarde `k-means++`, zodat de centroids zo goed mogelijk geïnitialiseerd worden. De `n_init`, `max_iter`, `random_state`, `copy_x`, `algorithm` en `tol` parameters behouden hun default waarden, omdat deze niet veranderd moeten worden. `Verbose` wordt op 1 gezet, zodat we feedback krijgen op de uitvoering van het algoritme.

TABLE VI
GEBRUIKTE PARAMETERS VOOR KMEANS

Paramater	Waarde
n_clusters	k
init	'k-means++'
n_init	'auto'
max_iter	300
tol	'0.0001'
verbose	1
random_state	None
copy_x	True
algorithm	'lloyd'

D. Resultaten visualiseren

Voor deze laatste stap wordt de `plotly` library gebruikt om de resultaten uit de vorige stap te visualiseren. Voor deze implementatie zijn er 2 belangrijke parameters: een lijst waarin de namen van initiatieven gegroepeerd staan volgens hun bijhorende cluster en een lijst van de top 3 woorden per cluster. De eerste lijst kan gemaakt worden door de `labels_` variabele te gebruiken waarin de cluster nummers opgeslagen zijn voor elk document. Voor de tweede lijst worden alle centroids eerst geïdentificeerd. Deze centroids worden gerepresenteerd als een lijst van numerieke waarden (TF-IDF scores). Voor elke centroid worden de indices van de top 3 grootste getallen bijgehouden, vervolgens worden de top 3 woorden geïdentificeerd door de woorden terug te vinden op de plaats van deze indices in de lijst van alle unieke woorden. Deze worden tenslotte opgeslagen samen met de bijhorende TF-IDF scores in een lijst. Deze twee lijsten worden verder verwerkt via de `plotly` library om automatisch een tabel, die zich in afbeelding 5 bevindt, uit te tekenen waarin de clustering ordelijk gevisualiseerd worden.

V. RESULTATEN

Voor het k-means algoritme werden er interessantere resultaten verkregen wanneer de waarde voor k manueel gekozen werd. Volgens de resultaten van de elbow method in afbeelding 4 is 58 de beste waarde voor k. Als deze waarde voor k wordt gebruikt voor het k-means clustering algoritme, betekent het dat er 58 clusters gecreëerd worden. Met $k = 58$ worden er 58 clusters gecreëerd. Hieruit kunnen we afleiden dat er slechts 10 (68-58) initiatieven of datapunten zijn die gedistribueerd kunnen worden onder 58 bestaande clusters. Anders verwoord, betekent het dat er maximaal 10 clusters zijn (van 58 clusters) die 2 of meer initiatieven bevatten en als gevolg zijn er 48 (58-10) initiatieven of datapunten die hun eigen individuele cluster

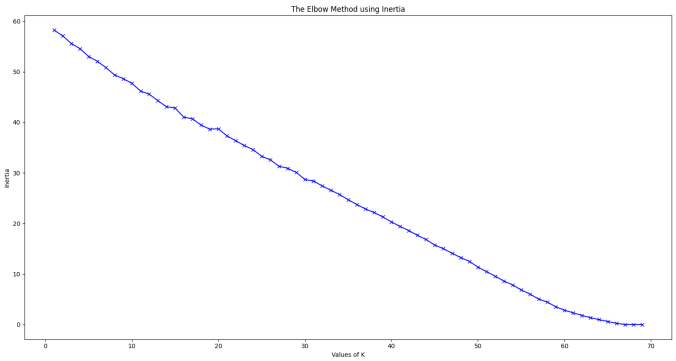

Fig. 4. Grafiek elbow method op inertia

krijgen (*pigeonhole principle*). Hieruit volgt dat het clustering algoritme geen goede resultaat oplevert voor $k = 58$, bepaald door de elbow method. Wanneer deze methode niet wordt gebruikt, maar er manueel geëxperimenteerd wordt met verschillende waarden voor k zien we interessantere resultaten op afbeelding 5.
Wanneer we manueel een kleinere waarde kiezen voor k, zien we dat er een vaak voorkomend patroon is. Op afbeelding 5 zien we dat:

- Cluster (1), (2), (9) en (10) elk 3 initiatieven bevatten.
- Cluster (4) en (5) elk 2 initiatieven bevatten.
- Cluster (3) en (6) elk 4 initiatieven bevatten.
- Cluster (7) 5 initiatieven bevat.
- Cluster (8) 39 $(68 - 29)$ initiatieven bevat.

Verder zien we dat 9 van de 10 clusters relatief goed geclusterd zijn, maar enkel in cluster (8) zien we dat de clustering niet goed gelukt is. Anders verwoord, betekent het dat de initiatieven binnen deze cluster niet samen gegroepeerd zijn omdat ze gelijkaardig zijn met elkaar. In de overige $k - 1$ clusters zien we dus dat er kleinere hoeveelheden aan initiatieven zijn binnen elke cluster, maar door handmatig deze clusters te analyseren, zagen we dat een groot deel van de initiatieven binnen deze clusters effectief gelijkaardig zijn met elkaar of een sterk verband hebben met elkaar.
Over het algemeen zien we in elke cluster interessante betekenisvolle sleutelwoorden, maar het is ook opvallend dat een woord zoals "cookie" erin voorkomt. In eerdere bevindingen, kwamen ook woorden zoals "javascript" of "http" erin voor doordat de dynamische website die uit JavaScript code bestond ook gescrapet werd. Deze zijn natuurlijk geen woorden die in deze context de initiatieven beschrijven, maar meer technische woorden die gebruikt worden bij het maken van de website. Dit probleem wordt initieel veroorzaakt door het feit dat het moeilijk is om op een intelligente manier woorden zoals http, javascript of cookie uit te filteren. In de pre-processing stap worden deze woorden ook niet verwijderd, omdat ze zeker geen stopwoorden zijn.
In de beginfase van dit project werd er in de implementatie van de webscraper geen controle gedaan op de gebruikte taal op de webpagina's, dit leverde als gevolg interessante resultaten op. Op afbeelding 6 kunnen we zien dat BelRAI een aparte cluster

toegewezen krijgt. Het valt hier op dat de top 3 woorden van deze cluster Nederlandse woorden bevatten. Hieruit kunnen we 2 conclusies trekken:

1) De gebruikte dataset of document voor BelRAI is in het Nederlands geschreven.
2) Deze Nederlandse woorden krijgen een hoge TF-IDF score, omdat Nederlandse woorden "uniek" zijn binnen een corpus dat volledig uit Engelse woorden bestaat.

Het tweede punt wordt veroorzaakt door een hoge IDF score voor de Nederlandse woorden, omdat deze woorden zelden voorkomen binnen het corpus.

VI. Discussie

Vooral in de implementatiefase zijn we een aantal uitdagingen tegengekomen. In deze sectie zullen deze uitdagingen en de oplossingen besproken worden.

A. Schaalbaarheid van de webscraper

Intuïtief zouden we enkel de interessante en bruikbare elementen uit een website willen scrapen. Echter, is elke website anders gestructureerd waardoor dit een probleem oplevert in de schaalbaarheid. Bijvoorbeeld, een initiatief, dataeuropa, bevat bruikbare data binnen de `<article>` tag, terwijl een ander initiatief, ARWEN, deze vooral binnen de `<main>` tag heeft. Echter, is deze methode onpraktisch, omdat we voor elke website in de dataset handmatig zou moeten nagaan welke specifieke elementen interessant zijn. Daarom is de volgende stap het ontwikkelen van een methode die dit proces veralgemeent. Dit betekent dat we een patroon moeten zoeken dat consistent voorkomt op alle websites. Zodra dit patroon geïdentificeerd is, kunnen we data extraheren op basis van deze patronen.

- Eerste optie - whitelisting: Een veel voorkomend patroon zijn 2 types webpagina's, de home-pagina en de about-pagina. Deze specifieke pagina's zijn pagina's die in alle professionele websites voorkomen en verder ook veel bruikbare informatie bevatten die hun initiatief beschrijven. In de implementatie is het dan mogelijk om de URL's met de structuur, https://domeinnaam/index en https://domeinnaam/about, te scrapen.
- Tweede optie - alles scrapen: Een andere methode is om niet voor een specifiek patroon te zoeken, maar alle data die beschikbaar is te scrapen. Dit werd geprobeerd door alle links die op de website staan te scrapen en vervolgens binnen deze gescrapete links opnieuw alle links te scrapen. Dit wordt herhaald totdat alle mogelijke links gescrapet zijn geweest.
- Derde optie - tot niveau 2 scrapen: Deze methode is vergelijkbaar met de tweede optie, maar hier wordt een limiet gesteld aan het aantal links dat per website gescrapet kan worden. Dit kon geïmplementeerd worden door voor elke website enkel tot niveau 2 te scrapen. Dit betekent dat de home-pagina eerst gescrapet wordt en vervolgens alle links die op deze pagina staan gescrapet worden, maar in tegenstelling tot de tweede optie wordt er niet verder genavigeerd.

Om het schaalbaarheid probleem op te lossen, werd er gekozen voor de derde optie. De eerste optie brengt een probleem met zich mee dat hoewel de beschreven patronen in elke website voorkomen, dat er veel uitzonderingen in de structuur van de URL waardoor deze randgevallen telkens handmatig afgehandeld moeten worden. De tweede optie zorgt ervoor dat er geen rekening gehouden moet worden met deze randgevallen, maar voor complexe websites met een diepe nesting van links, is deze optie niet haalbaar door de uitvoertijd. Als gevolg werd de derde optie geïmplementeerd, omdat dit het schaalbaarheid probleem oplost door voldoende data te scrapen zonder dat de uitvoertijd een probleem is.

B. Technische uitdagingen van de webscraper

1) Verbindingsfouten: Een vijftal websites genereren verbindingsfouten. Deze fouten worden veroorzaakt door websites die niet meer bestaan of door bepaalde websites die geen verbinding met de webscraper lijken toe te laten. Dit probleem kon niet opgelost worden, maar omdat we ook de data uit de "short description" kolom nemen uit de originele dataset (bv. "short description" uit tabel I), was er toch nog data die verwerkt kon worden voor de initiatieven met deze problemen.

2) Dynamische websites: Dynamische websites zijn websites die eerst bestaan uit JavaScript code en verder gerenderd moeten worden door een webbrowser om de HTML code te genereren. Wanneer de webscraper een verbinding maakt met deze website, ziet het alleen de JavaScript code. Door extra implementatie toe te voegen kunnen we dit probleem oplossen. Maar aangezien er enkel 1 website (https://dssc.eu/) is die niet gescrapet kan worden door dit probleem, werd er besloten dat dit probleem verwaarloosbaar is.

3) URL-formaat: Verder waren er een aantal randgevallen in verband met het formaat van de URL's. Verschillende checks waren nodig om na te kijken of de URL's die geëxtraheerd werden uit een webpagina effectief door de webscraper gebruikt kon worden. Doordat verschillende websites die links toevoegen in de HTML code geen "https://" of "http://" van voor in de links zetten, maar op sommige andere websites wel moest hier op gelet worden. Verder bestaan er ook links die de gebruiker doorverwijzen naar bijvoorbeeld een PDF bestand waardoor de link een ".pdf" extensie krijgt vanachter in de link. Deze randgevallen werden één voor één opgelost, maar dit betekent natuurlijk niet dat alle randgevallen opgelost zijn voor nieuwe bijkomende websites die niet in de originele dataset staan waardoor dit opnieuw een probleem zou kunnen zijn voor de schaalbaarheid.

C. Praktische nadelen van de lemmatizer

In sectie III-B1 werd er aangehaald dat de `WordNetLemmatizer` werd gebruikt voor de lemmatization, maar er werd ook aangehaald dat een vervoegd woord zo gelaten wordt als de lemma van dat woord niet teruggevonden kan worden in de `WordNet` database. Doordat we veel data hebben, zien we dit probleem

regelmatig terugkomen. In het resultaat, zoals weergegeven in Figuur 5, valt op dat twee van de top 3 woorden die cluster (9) representeren, namelijk "space" en "spaces", niet gelemmatized zijn. Een oplossing zou bijvoorbeeld zijn om een lemmatizer te gebruiken van een andere library, omdat het mogelijks die woorden wel herkent. Maar dit werd niet geïmplementeerd, omdat het buiten de scope van dit project valt.

D. Interne werking `TfidfVectorizer`

In Figuur 5 zien we dat het woord "data" veel voorkomt en ook telkens een hoge "importance" krijgt. Intuïtief zou dit woord geen hoge TF-IDF score mogen krijgen, omdat dit woord een heel populair woord is en dus door de zeer lage IDF score ook een lage TF-IDF score zou moeten krijgen. Hier zijn 2 redeneringen mogelijk:

- TF-score wordt niet genormaliseerd of er wordt een andere formule gebruikt waardoor deze score te groot wordt en vervolgens een zeer grote invloed heeft op de TF-IDF score.
- IDF-score is toch groot en onderschatten we hoe weinig documenten dit woord bevatten.

Door te testen, zagen we dat "data" een IDF score van 1.03 krijgt. Dit is een zeer lage IDF score aangezien de gebruikte IDF formule in sectie III-C de score met 1 verhoogt. Een ander woord dat in weinig document voorkomt is "arwen" en deze kreeg een IDF score van 4.16. Hieruit volgt dat de tweede redenering niet correct is. Voor cluster (4) in Figuur 5 zou het betekenen dat de TF score van "data" 0.34 (0.35/1.03) moet zijn. Volgens de standaard TF formule betekent het dat gemiddeld 1/3 van de termen in de documenten van Arwen en Athena uit het woord "data" bestaat. Dit is een onverwacht resultaat, omdat de dataset van Arwen wel veel termen "data" bevat, maar enkel 174 termen zijn "data" van de 3093, dus de TF score zou 0.056 (174/3093) moeten zijn. Gezien de beperkte tijd voor dit project en dit buiten de scope van het project viel, werd er besloten om geen verdere onderzoeken naar te doen. Dit zou in een opvolgend onderzoek uitgevoerd moeten worden.

CONCLUSIES

In dit project werd er geprobeerd om 2 onderzoeksvragen op te lossen:

1) Hoe kunnen de initiatieven met elkaar vergeleken worden?
2) Hoe kunnen de initiatieven die sterk verbonden zijn met elkaar vervolgens geïdentificeerd worden op basis van hun individuele beschrijving?

Door stapsgewijs de pipeline te volgen werd er een poging gedaan om deze vragen te beantwoorden. Eerst werd er een methode ontwikkeld om bruikbare data te extraheren. Deze bruikbare data is data in tekst formaat die uit de originele dataset verkregen werd en uit de websites van elk initiatief door een webscraper te ontwikkelen. Het volgende doel in deze stap was om de kwaliteit van deze data te verhogen door pre-processing technieken toe te passen zoals lemmatization, stopwords en leestekens/nummers removal. In de tweede stap werd TF-IDF toegepast op deze data om de woorden om te zetten naar numerieke waarden. En deze stap beantwoordt meteen de eerste onderzoeksvraag, omdat de data van elk initiatief door deze stappen vergeleken kunnen worden met elkaar, omdat de data die nu uit numerieke waarden bestaan door de machine/computer geïnterpreteerd kan worden. De tweede onderzoeksvraag werd beantwoord door de derde stap. Er werd het k-means clustering algoritme gebruikt om de resultaten uit de tweede stap in groepen te verdelen waarbij elke groep initiatieven bevat die een sterk verband hebben met elkaar en in de vierde stap werd dit gevisualiseerd voor de gebruiker.
Beide onderzoeksvragen werden dus beantwoord, maar dit betekent niet dat het probleem is opgelost. Ten eerste is het belangrijk om te vermelden dat het uiteindelijke resultaat echter een hulpmiddel is die een deel van het manuele werk kan automatiseren. Zoals eerder vermeld, bevat het k-means algoritme veel randomness waardoor elke uitvoering van het programma een ander resultaat kan opleveren. Verder heeft dit probleem ook niet 1 specifieke oplossing. Of 2 verschillende initiatieven effectief een sterk verband hebben of gelijkaardig zijn aan elkaar is niet volledig objectief waardoor er externe hulp nodig is om dit te verifiëren, wat buiten de scope van dit project valt. Dit zien we ook aan de subjectiviteit voor de keuze van k. Wat precies de correcte waarde is van k, werd uiteindelijk een manueel proces door voor verschillende waarden van k de resultaten te vergelijken met elkaar. De elbow method werd toegepast om dit proces te automatiseren, maar dit gaf onbruikbare resultaten terug. Hieruit werd er geconcludeerd dat er mogelijks een betere aanpak is door de methodes die gebruikt werden in verschillende stappen te vervangen door andere methodes. Voor het clustering algoritme zou bijvoorbeeld het *agglomerative hierarchical clustering* algoritme een beter resultaat kunnen opleveren, maar het zou natuurlijk ook een slechter resultaat kunnen opleveren. Hoewel het grote probleem niet volledig opgelost is, zijn we tevreden met het hulpmiddel dat gecreëerd is dat bruikbare resultaten opleverde om een deel van het manuele proces te automatiseren.

DANKWOORD

Eerst zou ik graag mijn begeleider, Marcel Parciak, willen bedanken voor zijn waardevolle begeleiding en hulp doorheen deze bachelorthesis. Verder wil ik mijn promotors, prof. dr. Liesbet Peeters en prof. dr. Stijn Vansummeren, bedanken voor de onmisbare adviezen tijdens mijn tussentijdse evaluatie.

1	2	3	4	5	6	7	8	9	10
phiri, 0.216	cookie, 0.322	gaiax, 0.209	data, 0.35	data, 0.275	hl, 0.446	data, 0.326	data, 0.241	data, 0.291	ohdsi, 0.293
covid, 0.209	cooky, 0.313	belrai, 0.188	arwen, 0.212	citizen, 0.269	fhir, 0.205	eu, 0.176	health, 0.121	spaces, 0.19	cooky, 0.229
esfri, 0.195	fiware, 0.21	twinning, 0.144	hospital, 0.166	tehdas, 0.237	ihe, 0.168	open, 0.165	research, 0.073	space, 0.133	pioneer, 0.176
european strategy forum on research infrastructures (esfri)	fiware (foundation)	belrai	arwen	healthy data e-consultation	helios	big data value public-private partnership (bdv)	actionable rwe network (arwen)	big data value association (bdva)	european health data and evidence network (ehden) (incl. ehden academy)
opensafely	health outcomes observatory (h2o)	dhu (digital health uptake) and dhu radar	athena (augmenting therapeutic effectiveness through novel analytics)	tehdas (towards european health data space)	hl7 (health level seven) international	data.europa.eu	amdex	data space support centre (dssc)	ohdsi (observational health data sciences and informatics)
phiri (population health information research infrastructure)	nihr health informatics collaborative	digitalhealtheurope catalogue of digital solutions supporting the digital transformation of health and care			hl7 belgium	dataeuropa	bdva	vlaamse smart data space (vsds)	pioneer
		gaia-x			integrating the healthcare enterprise (ihe)	european open science cloud (eosc)	big data for better outcomes (bd4bo)		
						european platform on rare disease registration	darwin (data analysis and real-world interrogation network in the european union)		
							data saves lives		
							data sharing coalition		
							ebmt (european society for blood and marrow transplantation)		
							ebrains		
							edith		
							elixir belgium		
							elixir		
							european health data space (ehds)		
							european medical		

Fig. 5. Resultaat na clustering op k=10

de, 0.442	data, 0.389	darwin, 0.522	data, 0.39	data, 0.352
belrai, 0.338	sharing, 0.169	eu, 0.432	arwen, 0.157	eu, 0.212
van, 0.295	health, 0.165	ema, 0.284	hospital, 0.142	open, 0.187
belrai	amdex	darwin (data analysis and real-world interrogation network in the european union)	actionable rwe network (arwen)	bdva
	big data for better outcomes (bd4bo)	the data analysis and real-world interrogationnetwork in the european union (darwin)	arwen	big data value association (bdva)
	data saves lives		athena (augmenting therapeutic effectiveness through novel analytics)	data.europa.eu
	data sharing coalition		big data value public-private partnership (bdv)	dataeuropa
	data space support centre (dssc)			

Fig. 6. BelRAI heeft een individuele cluster

Referenties

[1] *Afstand (wiskunde)*. In: *Wikipedia*. Page Version ID: 67494735. May 7, 2024. URL: https://nl.wikipedia.org/w/index.php?title=Afstand_(wiskunde)&oldid=67494735#Gewone_metriek_of_euclidische_afstandsfunctie (visited on 05/28/2024).

[2] David Arthur, Sergei Vassilvitskii, et al. "k-means++: The advantages of careful seeding". In: *Soda*. Vol. 7. 2007, pp. 1027–1035.

[3] Alejandra Ornelas Barajas. "K-Means clustering accelerated algorithms using the triangle inequality". In: (2015).

[4] Patrick Brus. *Clustering: How to Find Hyperparameters using inertia*. 2022.

[5] D212digital. *What is Lemmatization and Stemming in NLP?* Medium. Nov. 7, 2022. URL: https://212digital.medium.com/what-is-lemmatization-and-stemming-in-nlp-e25e142332c4 (visited on 05/27/2024).

[6] *Elbow Method for optimal value of k in KMeans*. GeeksforGeeks. Section: Machine Learning. June 6, 2019. URL: https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/ (visited on 05/28/2024).

[7] Lotte Geys and Liesbet M. Peeters. "Strategic Oversight Across Real-World Health Data Initiatives in a Complex Health Data Space". In: (Jan. 2, 2024). Publisher: Zenodo. DOI: 10.5281/zenodo.10451144. URL: https://zenodo.org/records/10451144 (visited on 05/27/2024).

[8] Lotte Geys and Liesbet M. Peeters. "Strategic Oversight Across Real-World Health Data Initiatives in a Complex Health Data Space: A Call for Collective Responsibility". In: ().

[9] Chetna Khanna. "Text pre-processing: Stop words removal using different libraries". In: *Towards Data Science* 10.02 (2021).

[10] *KMeans*. scikit-learn. URL: https://scikit-learn/stable/modules/generated/sklearn.cluster.KMeans.html (visited on 05/28/2024).

[11] Ekaterina Kochmar. *Getting started with natural language processing*. Simon and Schuster, 2022.

[12] Trupti M Kodinariya, Prashant R Makwana, et al. "Review on determining number of Cluster in K-Means Clustering". In: *International Journal* 1.6 (2013), pp. 90–95.

[13] Ganesh Kumar et al. "Data harmonization for heterogeneous datasets: a systematic literature review". In: *Applied Sciences* 11.17 (2021), p. 8275.

[14] Sanit Kumar et al. "Web scraping using Python". In: *International Journal of Advances in Engineering and Management* 4.9 (2022), pp. 235–237.

[15] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Higher Education from Cambridge University Press. ISBN: 9780511809071 Publisher: Cambridge University Press. July 7, 2008. DOI: 10.1017/CBO9780511809071. URL: https://www.cambridge.org/highereducation/books/introduction-to-information-retrieval/669D108D20F556C5C30957D63B5AB65C (visited on 05/28/2024).

[16] *morphy(7WN)*. WordNet. URL: https://wordnet.princeton.edu/documentation/morphy7wn (visited on 05/27/2024).

[17] Shahzad Qaiser and Ramsha Ali. "Text mining: use of TF-IDF to examine the relevance of words to documents". In: *International Journal of Computer Applications* 181.1 (2018), pp. 25–29.

[18] Ayisha Tabassum and Rajendra R Patil. "A survey on text pre-processing & feature extraction techniques in natural language processing". In: *International Research Journal of Engineering and Technology (IRJET)* 7.06 (2020), pp. 4864–4867.