Requirements voor Tools-to-data

Deze requirements maken deel uit van het project tools-to-data uit het CLARIAH+-programma. Meer informatie over dit project is te vinden op de projectpagina van de CLARIAH-website: https://www.clariah.nl/projects/tools-to-data

November 2022, Marian Hellema (zelfstandig, adviseur CLARIAH a.i., KB), Steven Claeyssens (conservator digitale collecties, KB), Joris van Zundert (senior researcher and developer in humanities computing, Huygens ING), Melvin Wevers (Assistant Professor in Digital History, Universiteit van Amsterdam), David de Boer (zelfstandig, solutions architect CLARIAH), Martin Brandt (adviseur, SURF), Freek Dijkstra (Project Lead Data Exchange, SURF).

Inhoudsopgave

Visie

Voor onderzoekers zijn de digitale collecties van erfgoedinstellingen heel interessant. Een deel van hen gebruikt hiervoor text- of data-mining-technieken. Zij werken vaak met een kopie van de data, die ze op hun eigen systemen analyseren met algoritmes. Maar de dataleveranciers kunnen lang niet alle data hiervoor beschikbaar stellen vanwege auteursrechten, privacy en andere juridische of contractuele restricties. In de praktijk gebruiken onderzoekers daarom vaak vooral ouder materiaal, omdat dat vrij toegankelijk is. Hierdoor kunnen blinde vlekken in onderzoeksresultaten ontstaan. Zoals een literatuuronderzoeker verzuchtte: “we zijn eigenlijk vooral de geschiedenis van de 19e eeuwse literatuur aan het schrijven”.

Bij ‘tools-to-data’ wordt de zaak omgedraaid. Er wordt geen kopie van de data aan de onderzoeker gegeven, maar de data blijven staan in hun veilige, afgeschermde omgeving. Onderzoekers sturen hun algoritmes naar die omgeving. Daar worden ze uitgevoerd, zonder dat de onderzoeker zelf toegang tot de data heeft. Het resultaat wordt vervolgens beschikbaar gemaakt voor de onderzoeker. Op die manier kunnen onderzoekers toch gebruik maken van collecties die anders ontoegankelijk zijn, terwijl de dataleverancier de garantie heeft dat er geen onrechtmatig toegang tot de data is.

Het idee van ‘tools-to-data’ is overigens niet alleen van toepassing voor data die anders ontoegankelijk zijn. Ook voor open materiaal kan het een bruikbare opzet zijn. Je hoeft dan immers niet steeds een kopie van een selectie uit de collecties naar onderzoekers te sturen. En die onderzoekers hoeven de data niet zelf op te slaan en te beheren.

Een tools-to-data-oplossing moet aansluiten aan bij de onderzoekscyclus van geesteswetenschappers:

- selectie: de onderzoeker kiest welke data / documenten relevant zijn voor het onderzoek
- verrijking: de onderzoeker voegt extra informatie toe die analyses mogelijk maken, bijvoorbeeld parsen van teksten, coderingen toevoegen, gegevens opschonen.
- analyse: met behulp van algoritmes worden de data geanalyseerd. Dit kunnen al bestaande algoritmes zijn, of algoritmes die specifiek voor een onderzoek worden ontwikkeld of aangepast. Algoritme-ontwikkeling gebeurt meestal iteratief, heen-en-weer-gaand tussen programmeercode en data totdat het algoritme naar wens werkt.
- publicatie: voor het publiceren van de onderzoeksresultaten is het belangrijk naar de onderliggende data te kunnen verwijzen, vindplaatsen te kunnen citeren en het onderzoek repliceerbaar te maken.

Opzet van dit document

In dit document worden de *requirements* beschreven voor een tools-to-data-oplossing. Er zijn drie perspectieven van belang: onderzoekers, dataleveranciers en CLARIAH-breed.

De requirements worden beschreven in de vorm van *epics* en *user stories*, op basis waarvan SURF een proof-of-concept heeft ontwikkeld. In de proof-of-concept zijn niet alle user stories geïmplementeerd. Maar de requirements beschrijven alle functionaliteit die nodig is voor een productierijpe oplossing, ook als die niet in het proof-of-concept is uitgewerkt. Dit wordt gebruikt in het vervolgproject SANE, waarvoor tools-to-data een voorloper is. Steeds wordt de prioritering aangegeven: proof-of-concept of lange termijn.

Hieronder worden eerst het domeinmodel en de verschillende rollen beschreven. Daarna volgen de epics en user stories.

Domeinmodel

Het domeinmodel beschrijft de concepten uit het domein waarvoor de tools-to-data-toepassing wordt ontwikkeld. Het model weerspiegelt de termen waarmee de experts zelf over het domein spreken. Dit zorgt voor een gedeeld taalgebruik en begrip tussen onderzoekers, dataleveranciers, CLARIAH, SURF en de ontwikkelaars.

- *Algoritme* (of *tool*): software-code waarmee *data* worden geanalyseerd of bewerkt t.b.v. het onderzoek. Voorbeeld: Named Entity Recognition in historische kranten. (De begrippen algoritme en tools worden hier door elkaar gebruikt als synoniemen).
- *Collectie* (ook wel *dataset* genoemd): een verzameling *data* die door een dataleverancier voor onderzoek beschikbaar wordt gesteld. Meestal is er een samenhang op basis van inhoudelijke, juridische en/of technische aspecten. Voorbeelden: DBNL-boeken, historische kranten uit Delpher, e-pubs van recente Nederlandse publicaties.
- *Data*: alles wat in een *collectie* zit. Voor de proof-of-concept zijn dit *documenten* uit de digitale collecties van de KB.
- *Document*: tekstuele eenheid en daarmee een specifiek soort *data*. Voorbeeld: een digitaal boek of tijdschriftartikel.
- *Metadata*: beschrijving van een *collectie* of item uit een collectie. Voorbeelden: bibliografische metadata van DBNL-boeken; structuurmetadata over de layout van historische kranten.
- *Verrijking*: toevoegen van extra informatie aan de *data* t.b.v. de *analyse*. Ook het opschonen van de data valt hieronder. Voorbeelden: parsen van teksten, toevoegen van coderingen, standaardiseren van gegevens.
- *Analyse*: het zoeken naar informatie of patronen in de *data* om een antwoord te vinden op een onderzoeksvraag.
- *Replicatie*: het herhalen van een onderzoek / *analyse* om de conclusies te verifiëren. Hiervoor moet onder meer duidelijk zijn welke *data* en welke *algoritmes* er in het onderzoek zijn gebruikt.
- *Testset*: een subset van een *collectie* met voorbeeldbestanden die door de onderzoeker worden gebruikt om een algoritme te ontwikkelen. (Soms wordt dit een *sample* of *steekproef* genoemd, maar dat suggereert dat het om een willekeurige, steeds wisselende set data gaat en dat is niet het geval).
- *Onsite* beschikbaarheid: bestanden zijn alleen binnen de muren van het gebouw van de dataleverancier in te zien (bv. vanwege auteursrecht). Dit staat in tegenstelling tot *online* beschikbaar, waarbij bestanden via internet beschikbaar zijn.
- *Recipe*: stappenplan van het onderzoek, de methodologische paragraaf van een onderzoekspublicatie. Hierin worden onder meer de gebruikte data, algoritmes en configuraties beschreven (incl. versies). Er is geen vaste vorm voor, het is proza.
- *Selectie:* de keuze van *data* die de onderzoeker wil gebruiken voor het onderzoek. De selectie kan uit één of meer *collecties* afkomstig zijn.
- *Resultaten*: Het resultaat van een algoritme, de output. Voorbeelden: frequentietellingen van woorden in een tekst, een getraind model bij machine learning of een bag-of-words-representatie per pagina van een document.
- *Rapport:* Vastlegging van de combinatie van:

- de broncode van het gebruikte *algoritme* (dan wel een referentie naar het algoritme in een code repository)
- de gebruikte *selectie*
- de *resultaten* van het algoritme
- de door het algoritme geproduceerde logging (stderr en stdout)
- de onderzoeker
- de uitvoerdatum van het algoritme.

Rollen

De rollen beschrijven wat voor soort betrokkenen er zijn in het werken met tools-to-data.

- *Onderzoeker*: iemand die met behulp van algoritmes data onderzoekt om een onderzoeksvraag te beantwoorden. Bij tools-to-data gaat het meer specifiek om text- en datamining. Het kan in principe ook om onderzoekers gaan die niet aan een onderzoeksinstituut verbonden zijn, maar om juridische en praktische redenen richt Clariah zich vooral tot de ‘geïnstitutionaliseerde’ groep.
- *Dataleverancier*: organisatie die collecties digitaal beschikbaar stelt voor onderzoek, inclusief bijbehorende metadata.
- *Ontwikkelaar* van onderzoeksalgoritme: iemand die een algoritme programmeert voor het onderzoek. (Vaak vervult een onderzoeker deze rol zelf, maar het kunnen ook afzonderlijke personen zijn.)

1. Epic: Selectie van data

Onderzoekers kunnen binnen de beschikbare collecties een selectie maken van data die voor hun onderzoek relevant zijn.

1.1. Story: inzicht in inhoud collectie

Als onderzoeker
Wil ik inzicht in de inhoud van een collectie
Zodat ik kan beoordelen of de collectie interessant is voor mijn onderzoek

Prioriteit: lange termijn

Opmerking: het gaat hier om een globaal inzicht in de collectie: wat is de inhoud (bv. Nederlandse literatuur), wat is de spreiding (bv. welke auteurs, welke periode), welke bestandstypen (bv xml), herkomst (bv. gedigitaliseerde boeken van de KB) etc. Dit vraagt om een goede beschrijving (metadata over de collectie als geheel) door de dataleverancier. Het gaat hier niet om metadata van afzonderlijke collectie-items, dat komt in user story Story: kiezen data aan bod.

1.2. Story: kiezen data

Als onderzoeker
Wil ik iteratief kunnen selecteren welke data ik in mijn algoritmes gebruik
Zodat ik alleen relevante data in mijn analyses gebruik.

Prioriteit: lange termijn

Toelichting: de selectie kan op basis van metadata en/of full-text. Het kan met een API of via een GUI. Kan resulteren in een lijst met ID's van bestanden, bv. document-ID's.
De selectie moet iteratief kunnen worden bepaald, bijvoorbeeld omdat de onderzoeker 'outliers' buiten beschouwing wil laten. Inmiddels heeft de KB plannen voor een 'corpusselectie-tool' waarmee gebruikers data kunnen selecteren. Wellicht dat die kan worden aangesloten op tools-to-data (waarbij moet worden aangetekend dat de corpusselectie-tool alleen voor KB-collecties bedoeld is, terwijl uiteindelijk onderzoekers ook data van verschillende collectie-eigenaren moet kunnen selecteren).

1.3. Story: collecties combineren

Als onderzoeker
Wil ik data uit meerdere collecties kunnen combineren bij mijn selectie
Zodat ik gegevens kan combineren en interessante patronen kan ontdekken.

Prioriteit: lange termijn

Toelichting: het kan gaan om het combineren van collecties van één of meerdere dataleveranciers (als voorbeeld: AV-materiaal van Beeld en Geluid combineren met

historische kranten van de KB om te onderzoeken hoe er over bepaalde tv-programma's werd geschreven). (Zie opmerking bij Story: kiezen data over de KB-corpusselectie-tool).

Nog ter discussie: wat is er voor het combineren van collecties nodig? Combinatie, intersectie en verschil van collecties?

1.4. Story: combineren met andere data

Als onderzoeker
Wil ik mijn selectie kunnen combineren met andere data
Zodat ik in mijn algoritmes ook andere relevante data kan gebruiken.

Prioriteit: lange termijn

Toelichting: het gaat hier om data die niet als CLARIAH-collectie beschikbaar is, maar op een andere manier. Dit kan gaan om allerlei soorten data, bijvoorbeeld eigen datasets van de onderzoeker of datasets die buiten het Clariah-domein zijn gemaakt, bijvoorbeeld door het Kadaster.

2. Epic: toepassen en ontwikkelen algoritmes

Onderzoekers kunnen algoritmes toepassen op hun selecties. Dit kunnen bestaande algoritmes zijn, of aanpassingen van bestaande algoritmes, of geheel nieuw ontwikkelde algoritmes. De onderzoeker heeft de vrijheid in het gebruik van algoritmes, mits de resultaten binnen de restricties van de dataleverancier vallen. Met andere woorden: er is geen vaste lijst van algoritmes waar de onderzoeker uit moet kiezen, want daarvoor zijn de behoeften te divers.

Ontwikkelaars van algoritmes kunnen dit iteratief ontwikkelen.

Nog ter discussie: naast eigen algoritmes toch de mogelijkheid van een lijst met bestaande algoritmes aanbieden? Dat heeft als voordeel dat de werking ervan bekend is en dat de dataleverancier niet iedere keer goedkeuring hoeft te geven.

2.1. Story: inzicht in de structuur van een collectie

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik inzicht in de structuur van een collectie (bv. opbouw xml, directorystructuur)
Zodat ik de data in mijn algoritme goed kan verwerken.

Prioriteit: lange termijn

*Toelichting: het gaat hier **niet** om inhoudelijk inzicht in de collectie (dat wordt in user story Story: inzicht in inhoud collectie beschreven). Het gaat hier om meer technische informatie over hoe de data is geordend.*

- *de ordening van bestanden, bv. in een directorystructuur. Dit kan simpelweg een 'ls' / 'dir' commando op de collectie/selectie.*
- *de ordening binnen een bestand, bv. de opbouw van xml-bestanden (xml-schema). (Voor epubs is dit lastiger omdat die niet zijn gestandaardiseerd. De KB overweegt of de inhoud van epubs als ruwe tekst moet worden aangeboden. Nadeel is dat je daarmee eventuele embedded xml kwijtraakt).*

2.2. Story: inzicht in de afzonderlijke collectie-items

Als onderzoeker
Wil ik beschikken over informatie over de afzonderlijke collectie-items (metadata en fulltext)
Zodat ik de informatie kan gebruiken in mijn analyses.

Prioriteit: proof-of-concept.

Toelichting: het gaat hier om metadata op afzonderlijk documentniveau, terwijl het in user story Story: inzicht in inhoud collectie gaat om metadata over de collectie als geheel.
*We leggen hier **niet** vast welke metadata beschikbaar moeten zijn, maar in het algemeen gaat het om beschrijvende metadata en om de identifiers waarmee naar de documenten of delen daarvan kan worden verwezen.*

Deze user story is verwant aan Story: kiezen data omdat het in beide gevallen vereist dat er metadata beschikbaar zijn. Maar voor het kiezen van de data is een zoek- en selectie-interface nodig, terwijl het er hier om gaat dat de metadata ook voor het algoritme en de analyse beschikbaar moet zijn. Een voorbeeld: het jaar van uitgave wordt gebruikt voor een tijdsserieanalyse; of de auteur wordt gebruikt als groepering van de analyses.

Nog ter discussie: hoe moeten de metadata beschikbaar zijn? In ieder geval machineleesbaar met een koppeling naar de data. Een mogelijkheid is om een metadatabestand bij de geselecteerde dataset neerzetten in een vaste directory. Dit kan een csv-, json- of xml-bestand zijn, met daarin in ieder geval ook bij welke data (=bestanden) de metadata horen. De opbouw van het metadatabestand kan met een schema worden gepubliceerd.

2.3. Story: testset t.b.v algoritmeontwikkeling (zandbak)

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik toegang tot een aantal bestanden uit een dataset (testset)
Zodat ik bij het ontwikkelen van mijn algoritme goede testbestanden heb om te zorgen dat het algoritme correct is.

Prioriteit: lange termijn

Toelichting: er is meestal sprake van een iteratieve manier van werken met een wisselwerking tussen data en algoritme: inspecteer data, pas algoritme aan, bekijk resultaat, inspecteer, pas aan, enz.. Hiervoor is het noodzakelijk dat onderzoekers met een testset uit een collectie kunnen werken waar ze wel inzage in hebben (i.t.t. de rest van de collectie). Het is de vraag wat er juridisch gezien mag met zo'n testset. Wellicht kunnen er bestanden worden gebruikt waar geen auteursrecht op rust, of waar het gecleard is. Ook kan dit worden gerealiseerd door een 'zandbak', d.w.z. dat de onderzoeker in een beveiligde omgeving het algoritme kan loslaten op de testset en de testset-bestanden kan bekijken. Als dat niet mogelijk is, kan de onderzoeker naar de KB komen want daar mogen alle bestanden wel ter inzage worden gesteld. Dat betekent dat je een online en een onsite-versie van Tools-to-Data moet hebben, zie Story: onsite-variant van tools-to-data.

Nog ter discussie: wat nu als er op de hele collectie auteursrechten berusten? dan nog kun je waarschijnlijk wel een testset aanbieden in een beveiligde 'zandbak' want de data kunnen er nog steeds niet uitlekken.

3. Epic: algoritme aanleveren

Onderzoekers of algoritmeontwikkelaars zullen hun algoritme meestal op hun eigen systeem ontwikkelen, of ze gebruiken een al bestaande too. Ze kunnen hun algoritme naar keuze op verschillende manieren aanleveren voor de tools-to-data-omgeving: als programmacode, vanuit een repository of als Docker-container. Bij voorkeur wordt het als Docker aangeleverd en aangeboden vanuit een repository met versiebeheer.

Toelichting: een algoritme kan naar keuze op de volgende manieren aan de Tools-to-data-omgeving worden aangeboden:

- *als code (user story Story: algoritme aanleveren als code)*
- *met afhankelijkheden (user story Story: algoritme aanleveren met afhankelijkheden programmeercode)*
- *als container (user story Story: algoritme aanleveren als container)*

Deze drie mogelijkheden worden als afzonderlijke user stories hieronder uitgewerkt, maar de ontwikkelaar zal er in de praktijk dus één kiezen.

3.1. Story: algoritme aanleveren als code

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik mijn algoritme als programmacode aanleveren
Zodat ik een binnen de onderzoekswereld veelgebruikte programmeertaal kan hanteren.

Prioriteit: voor de proof-of-concept is het mogelijk een Pythonscript te uploaden. Voor de lange termijn zullen ook andere programmeertalen moeten worden ondersteund.

Acceptatiecriteria voor Python-script

- ontwikkelaars kunnen één .py-bestand uploaden
- ontwikkelaars kunnen meerdere .py-bestanden uploaden; in dat geval wordt main.py uitgevoerd.

Opmerking: ook R is veelgebruikt en zou goed zijn om te ondersteunen (lange termijn).
Nog ter discussie: moeten onderzoekers ook de mogelijkheid hebben om bestaande tools te gebruiken, zoals OpenRefine, xml-editor e.d.?

3.2. Story: algoritme aanleveren met afhankelijkheden programmeercode

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik opgeven van welke packages (en taalmodellen) mijn algoritme afhankelijk is
Zodat mijn algoritme deterministische output geeft.

Prioriteit: proof-of-concept

Acceptatiecriteria

- ontwikkelaars kunnen een Pipfile en Pipfile.lock aanleveren bij hun code met daarin de afhankelijkheden
- wanneer een Pipfile en Pipfile.lock aanwezig zijn, worden de dependencies daarvan geïnstalleerd tijdens de build pipeline.
- wanneer een requirements.txt aanwezig is, worden de dependencies daarin geïnstalleerd tijdens de build pipeline
- waneer een setup.py aanwezig is, wordt die uitgevoerd tijdens het opbouwen van de omgeving (bijvoorbeeld om een taalmodel te downloaden)

3.3. Story: algoritme aanleveren als container

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik mijn algoritme aanleveren als container
Zodat ik volledige controle heb over afhankelijkheden en zodat mijn algoritme past in de CLaaS-infrastructuur.

Prioriteit: proof-of-concept

Acceptatiecriteria

- ontwikkelaars kunnen een Dockerfile leveren bij hun code, inclusief de vereiste Python-versie
- wanneer een Dockerfile aanwezig is, gebruikt de build pipeline die om een Docker image te bouwen en uit te voeren.

3.4. Story: algoritme aanleveren op basis van repository

Als onderzoeker en als dataleverancier
Wil ik dat het algoritme in een repository met versiebeheer staat
Zodat het onderzoek repliceerbaar is en ik (onderzoeker) mijn algoritme goed kan beheren en zodat ik (dataleverancier) achteraf kan nagaan welk algoritme er is gebruikt.

Prioriteit: proof-of-concept

Acceptatiecriteria

- wanneer ontwikkelaars hun algoritme plaatsen in een *code repository* (bijvoorbeeld GitHub of GitLab), kunnen zij de URL van die repository opgeven voor de uitvoer van het algoritme
- ontwikkelaars kunnen tools-to-data toegang geven tot hun code repository wanneer dat niet publiek is
- wanneer een URL wordt opgegeven, leest de build-pipeline de repository uit en controleert op de aanwezigheid van Pipfile, requirements.txt of Dockerfile.

Nog ter discussie: wat voor metadata van het algoritme zijn er voor Tools-to-data nodig?

4. Epic: resultaten van algoritmes

Onderzoekers en algoritme-ontwikkelaars kunnen hun algoritmes draaien en krijgen (na goedkeuring van de dataleverancier) beschikking over het resultaat en kunnen het downloaden. Daar horen ook eventuele loggingsgegevens van het algoritme, foutmeldingen en andere debugging bij. Ook kan het algoritme gebruik maken van tijdelijke werkbestanden.

4.1. Story: resultaten inzien en downloaden

Als onderzoeker en als ontwikkelaar van een onderzoeksalgoritme
Wil ik de resultaten van een algoritme kunnen inzien en downloaden
Zodat ik conclusies kan trekken voor mijn onderzoek of mijn algoritmes verder kan ontwikkelen

Prioriteit: proof-of-concept

Acceptatiecriteria

- ontwikkelaars kunnen alle output downloaden die het algoritme naar een vooraf bepaalde locatie op het bestandssysteem heeft geschreven (bijvoorbeeld directory `output/`)
- de (gedeeltelijke) output is ook inzichtelijk als het algoritme tijdens de uitvoer vastloopt
- voordat de uitvoer wordt vrijgegeven aan ontwikkelaar/onderzoeker controleert de dataleverancier of het mag worden vrijgegeven (zie user story Story: controle over resultaten en Story: bredere toestemming).

Opmerking: onderzoekers willen soms snippets van de teksten kunnen zien, bv. de context van een vindplaats van hun analyse. Die kunnen ze zelf in hun algoritme als uitvoer opnemen. Dit maakt ook onderdeel uit van deze user story. Hier hoeft dus geen afzonderlijke functionaliteit voor te worden ontwikkeld.

4.2. Story: logs van het algoritme

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik de logs van mijn algoritme kunnen inzien
Zodat ik kan debuggen.

Prioriteit: proof-of-concept

Acceptatiecriteria

- het algoritme moet log-output schrijven naar `stdout/stderr` (overeenkomstig CLARIAH-requirements)
- ontwikkelaars kunnen de logs inzien en downloaden.

Opmerking: er kunnen in die logs ook teksten weglekken. Die moeten dus ook van tevoren door de dataleverancier kunnen worden gecontroleerd voordat ze worden vrijgegeven aan de onderzoeker. Anders geformuleerd: deze logs zijn onderdeel van de resultaten van een algoritme.

4.3. Story: opslagruimte voor werkbestanden

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik tussenresultaten van mijn algoritmes kunnen opslaan tussen verschillende runs
Zodat ik die weer kan gebruiken als invoer voor een volgende stap van mijn algoritmes.

Prioriteit: proof-of-concept

Toelichting: vaak zijn conversies, data-cleaning en verrijkingen de eerste stap in onderzoeksalgoritmes (preprocessing). Bijvoorbeeld: een xml-file wordt gelemmatiseerd of geparst. De resultaten van die eerste stap zijn de invoer van de volgende stappen in het algoritme. In deze user story gaat het om tijdelijke bestanden die alleen voor dit specifieke algoritme van belang zijn. Als het gaat om bewerkte data die op een later moment kunnen worden hergebruikt door andere personen of andere algoritmes, dan is de user story Story: collecties combineren of Story: combineren met andere data van toepassing.

4.4. Story: inzien data voor debugging

Als ontwikkelaar van een onderzoeksalgoritme
Wil ik waar nodig inzicht in de data
Zodat ik onverwachte fouten kan oplossen.

Prioriteit: lange termijn, maar wel juridische vraagstuk onderzoeken.

Toelichting: een algoritme wordt ontwikkeld op een testset van de data, maar kan daarna alsnog misgaan als het op de complete dataset wordt gedraaid. Bv. omdat er in een bestand ineens vreemde tekens staan waar de code op vastloopt. Nog ter discussie: hoe lossen we dit op als het bestanden zijn die de onderzoeker niet mag zien? Moet die dan naar het gebouw van de dataleverancier komen? In SANE-Tinker is dit geen probleem.

Deze story is heel handig, maar niet essentieel. Je kunt altijd eromheen werken door in het algoritme heel precieze uitvoer voor debugging te genereren.

5. Epic: controle over toegang

Dataleveranciers hebben controle over welke onderzoekers met welke data welke algoritmes mogen draaien en welke resultaten mogen ontvangen.

5.1. Story: onderzoekers toegang geven (ingangscontrole)

Als dataleverancier
Wil ik kunnen bepalen wie toegang krijgt tot tools-to-data
Zodat ik onrechtmatige toegang kan voorkomen.

Prioriteit: lange termijn, voorlopig via SURF-account (Research Drive).

Nog ter discussie: wat voor procedure is hiervoor nodig? Voorstel in onderstaand scenario.

Opmerking: het is aan te bevelen dat onderzoekers van tevoren opgeven wat voor onderzoek ze gaan doen en wat voor soort resultaten dit oplevert.

5.1.1. Scenario: onderzoekers toegang geven

1. Onderzoeker verzoekt de dataleverancier om met het tools-to-data-systeem te mogen werken (bv. per e-mail). Het verzoek geeft een korte beschrijving van welke collecties de onderzoeker wil gebruiken en welk soort algoritmen. Ook geeft het een tijdsperiode waarbinnen toegang tot Tools-to-Data nodig is. .
2. Dataleverancier beoordeelt het verzoek.
3. Indien de dataleverancier akkoord is, stemt de onderzoeker in met de voorwaarden over hoe de data wel en niet mogen worden gebruikt. Ook wordt zo nodig opgenomen dat de dataleverancier tijdelijk gebruiksgegevens mag bewaren voor monitoring en voor onderzoek bij incidenten. (Deze instemming kan per mail worden geregeld, of er kan een contract worden afgesloten).
4. Na instemming door de onderzoeker, registreert de dataleverancier de onderzoeker zodat die toegang heeft tot het tools-to-data-systeem. In ieder geval machineleesbaar en zo mogelijk via Clariah-brede autorisatie- en authenticatie-functionaliteit.
5. De toegang kan door de dataleverancier aan een periode worden gebonden, waarna de toegang voor de onderzoeker vanzelf vervalt.
6. Dataleverancier kan eventueel de toestemming tussentijds weer intrekken.

Alternatief scenario (toekomstig): alle onderzoekers die toegang hebben tot CLARIAH-infrastructuur via CLARIAH-brede authenticatie en autorisatie, hebben daarmee ook toegang tot tools-to-data.

5.2. Story: controle over gebruik collecties (ingangscontrole)

Als dataleverancier

Wil ik kunnen bepalen welke collecties door een bepaalde onderzoeker te gebruiken zijn
Zodat ik kan voorkomen dat er onrechtmatige toegang wordt gegeven.

Prioriteit: lange termijn.

Opmerking: dit kan door onderzoekers toestemming te geven om met bepaalde collecties te werken (bv. DBNL, Delpher-kranten e.d.). Daarbinnen kunnen ze dan hun eigen selectie maken.

Nog ter discussie: wil de KB de toestemming per collectie kunnen geven? M.a.w. wat zijn redenen om een onderzoeker wel met de ene collectie en niet met de andere te laten werken? Misschien wel relevant voor andere dataleveranciers. En misschien ter geruststelling van rechthebbenden dat er differentiatie mogelijk is.

Opmerking: als alternatief voor deze user story is er ook de mogelijkheid een onderzoeker meer algemene toestemming te geven, zie user story Story: bredere toestemming

5.3. Story: controle over output (uitgangscontrole)

Als dataleverancier
Wil ik alle output en het rapport kunnen inzien en kunnen besluiten of de resultaten daarin aan de onderzoeker of ontwikkelaar worden vrijgegeven
Zodat ik kan voorkomen dat een algoritme onrechtmatig te veel data vrijgeeft.

Opmerking: Het gaat hier om alle output: resultaat van het algoritme, loggegevens van het algoritme zelf, eventuele rapportages, en andere relevante gegevens die het algoritme oplevert.

Prioriteit: proof-of-concept.

Toelichting: het is denkbaar dat de resultaten het mogelijk maken om de oorspronkelijke tekst te reconstrueren. Bijvoorbeeld als het algoritme een kopie van een (deel van de) tekst maakt, of als het alle woorden en hun positie in de tekst zou produceren. Om dit te voorkomen, moet de dataleverancier eerst de resultaten kunnen bekijken. Ook de log-gegevens van het algoritme moeten door de dataleverancier kunnen worden beoordeeld, omdat daar immers ook data kunnen weglekken. De beoordeling zal door een persoon worden gedaan (hoewel het misschien in de toekomst deels kan worden geautomatiseerd).

Nog ter discussie: volgt hieruit dat de output van algoritmes alleen in een vorm mag zijn die door mensen leesbaar is (geen binary's, bijvoorbeeld afbeeldingen)?

Opmerking: als alternatief voor deze user story is er ook de mogelijkheid een onderzoeker meer algemene toestemming te geven, zie user story Story: bredere toestemming

5.4. Story: controle over algoritmes (ingangscontrole)

Als dataleverancier

Wil ik een door onderzoeker aangeleverd algoritme kunnen toestaan of afwijzen
Zodat ik kan voorkomen dat algoritmes onrechtmatige informatie vrijgeven.

Prioriteit: lange termijn.

Opmerking: het bekijken van het algoritme is een optie voor de dataleveranciers. Ze kunnen het overslaan (en vooral de resultaten beoordelen, zie user story Story: controle over resultaten), maar als ze het willen, kunnen ze het algoritme zien. Dit zou ook nuttig kunnen zijn als bestaande algoritmes door andere onderzoekers worden hergebruikt; dan wil je bijvoorbeeld kunnen aangeven dat een onderzoeker een bepaald NER-algoritme mag gebruiken.

Opmerking: als alternatief voor deze user story is er ook de mogelijkheid een onderzoeker meer algemene toestemming te geven, zie user story Story: bredere toestemming

5.5. Story: bredere toestemming (ingangscontrole)

Als dataleverancier
Wil ik een onderzoeker toestemming kunnen geven om alle algoritmes op een collectie te mogen draaien
Zodat ik niet voor alle stappen die een onderzoeker doet iedere keer afzonderlijk toestemming hoef te geven.

Prioriteit: proof-of-concept.

Toelichting: dit is een mogelijkheid die naast de eerder beschreven fijnmazige toestemmingen moet bestaan. Het idee is dat je als dataleverancier aan het begin van een onderzoek misschien per stap wilt volgen wat er gebeurt. Maar na een tijdje vertrouw je de onderzoeker en wil je algehele toestemming geven om met een collectie alle algoritmes te mogen draaien en de uitvoer direct te ontvangen. De onderzoeker heeft immers ingestemd met de gebruiksvoorwaarden die vastleggen wat onrechtmatig gebruik van de data is.

Nog ter discussie: geldt deze bredere toestemming per collectie? Of krijgt een onderzoeker ook toestemming om alle collecties van de dataleverancier te gebruiken?

5.5.1. Scenario: fijnmazige controle over data, resultaten

Opmerking: in dit scenario wordt de toestemming fijnmazig geregeld, per afzonderlijke stap van de onderzoeker. Als alternatief is er ook een bredere toestemming mogelijk, die wordt beschreven in Alternatief scenario: bredere toestemming.

1. Onderzoeker logt in op tools-to-data
2. Systeem controleert of onderzoeker geautoriseerd is voor toegang tot het systeem
3. Zo ja, dan kiest de onderzoeker één of meer collecties voor het onderzoek. Ook geeft de onderzoeker een einddatum op voor de toegang tot de collectie.
4. Systeem stuurt bericht aan dataleverancier dat er een verzoek klaarstaat.

5. Dataleverancier (na inloggen) beoordeelt het verzoek. Indien akkoord wordt de toestemming gegeven en in het systeem vastgelegd. De dataleverancier heeft hierbij de mogelijkheid een einddatum in te stellen.
6. Vanaf dan mag deze onderzoeker met de goedgekeurde collecties werken, totdat de einddatum is verstreken of de dataleverancier de toestemming tussentijds intrekt.
7. De onderzoeker selecteert de data om mee te werken (d.w.z. eigen subsets van de toegestane collecties). (dit is een afzonderlijk scenario).
8. De ontwikkelaar ontwikkelt algoritme of de onderzoeker kiest een bestaand algoritme.
9. De ontwikkelaar / onderzoeker draait het algoritme op de geselecteerde data.
10. Het systeem bewaart de resultaten maar toont ze nog niet aan de onderzoeker.
11. Het systeem stuurt bericht aan de dataleverancier dat er een verzoek klaarstaat om de resultaten te mogen zien (per e-mail).
12. De dataleverancier beoordeelt het verzoek en kan daarvoor de resultaten van het algoritme inzien. Zo kan o.a. bekeken worden of er in de resultaten niet te veel data weglekken (bv. een algoritme dat de dataset of delen daarvan kopieert).
13. De dataleverancier geeft toestemming.
14. Het systeem stuurt de ontwikkelaar / onderzoeker per e-mail een bericht dat de resultaten beschikbaar zijn.
15. De ontwikkelaar / onderzoeker kan de resultaten bekijken en downloaden.

5.5.2. Alternatief scenario: bredere toestemming

NB in plaats van dit scenario kan ook voor Scenario: fijnmazige controle over data, resultaten worden gekozen.

1. De onderzoeker logt in op tools-to-data
2. Het systeem controleert of onderzoeker geautoriseerd is voor toegang tot het systeem
3. Zo ja, dan kiest de onderzoeker één of meer collecties voor het onderzoek. Ook geeft de onderzoeker een einddatum op voor de toegang.
4. Het systeem stuurt een bericht aan de dataleverancier dat er een verzoek klaarstaat.
5. De dataleverancier beoordeelt het verzoek. Indien akkoord wordt de toestemming gegeven en in het systeem vastgelegd. De dataleverancier heeft hierbij de mogelijkheid een einddatum in te stellen.
6. Vanaf dan mag deze onderzoeker met de goedgekeurde collecties werken, totdat de einddatum is verstreken of de dataleverancier de toestemming tussentijds intrekt.
7. De ontwikkelaar ontwikkelt algoritme of de onderzoeker kiest een bestaand algoritme.
8. De ontwikkelaar / onderzoeker draait het algoritme op de geselecteerde data.
9. De ontwikkelaar / onderzoeker kan resultaten bekijken en downloaden.
10. De dataleverancier kan op ieder moment zien wat de onderzoeker gedaan heeft met de data en welke resultaten dat heeft opgeleverd (zie monitoring). Desgewenst kan de dataleverancier de toestemming van de onderzoeker intrekken, bv. als er misbruik is gebleken.

5.6. Story: inzicht in gebruik

Als dataleverancier
Wil ik inzicht in hoe mijn datasets worden gebruikt
Zodat ik de impact van mijn digitale collecties kan zien.

Prioriteit: lange termijn.

Nog ter discussie: wat voor soort informatie heb je hiervoor nodig?

5.7. Story: controle op basis van historische rapporten

Als dataleverancier
Wil ik na verloop van tijd kunnen beschikken over historische rapporten
Zodat ik te allen tijde kan achterhalen wat er met de data is gebeurd, kan controleren of er geen onrechtmatige toegang is ontstaan en kan nagaan wat er precies is gebeurd bij incidenten waarin toch data naar buiten zijn gelekt.

Directe controle is voorzien in Story: controle over rapport (uitgangscontrole); het gaat hier om controle na verloop van tijd.

Acceptatiecriteria

- Rapporten worden bewaard zodat de dataleverancier ze ook later nog inzien.

Prioriteit: lange termijn

Zie 6.1.

5.8. Story: automatische controle op basis van output

Als dataleverancier
Wil ik dat de output automatisch wordt gecontroleerd
zodat ik minder tijd kwijt ben met handmatig controleren.

Prioriteit: lange termijn.

Opmerking: het gaat er in deze user story om dat de outputcontrole voor een deel geautomatiseerd zou kunnen worden, bijvoorbeeld door te zoeken of er 'gewone taal' in de uitvoer staat. Dit zou de benodigde inspanning voor de controles verlichten. De verwachting is echter dat deze controle niet helemaal geautomatiseerd kan worden, maar dat er ook een mens naar zal moeten kijken.

6. Epic: replicatie

Onderzoekers kunnen verwijzen naar de gebruikte data en ze kunnen hun onderzoek repliceerbaar maken. Deze mogelijkheden zijn met name van belang voor wetenschappelijke publicaties.

6.1. Story: rapportage over uitvoeren van algoritmes

Als onderzoeker
Wil ik kunnen beschikken over het *rapport*
Zodat ik dat kan gebruiken voor mijn ‘recipes’ en voldoe aan de wetenschappelijke eis van replicatie.

Prioriteit: proof-of-concept.

Opmerking: Vanuit de onderzoekers komt ook de wens om in de fasen van het ontwikkelen en analyseren je eigen 'geschiedenis' te kunnen zien: op welk moment heb je welk algoritme op welke data gedraaid en wat was de output? Ter discussie: moet de output ook worden bewaard?

Acceptatiecriteria

- In het rapport is opgenomen: de code van het algoritme dan wel een referentie daarnaar in een code repository (Story: beheer algoritmes), de gebruikte data (selectie en versie van de data), de uitvoerende onderzoeker, de uitvoerdatum. Nog ter discussie: moeten de resultaten van het algoritme ook in het rapport worden bewaard?
- De gebruikte data kan bijvoorbeeld een lijst met document-ID’s zijn met een versie-aanduiding en/of tijdstempel.

Vraag: als iemand anders het onderzoek wil repliceren, heeft die dan toegang tot tools-to-data nodig?

7. Epic: veilige data- en analyseomgeving

Onderzoekers en algoritmeontwikkelaars werken in een omgeving waarin geen onrechtmatige toegang tot de data mogelijk is.

7.1. Story: onrechtmatige toegang voorkomen

Als dataleverancier
Wil ik dat de data in een veilige omgeving staan waarin kan worden voorkomen dat er kopieën van de data worden gemaakt en dat onderzoekers buiten onze muren de data kunnen inzien
Zodat ik kan garanderen dat er geen onrechtmatige toegang is.

Prioriteit: proof-of-concept.

Toelichting: het gaat erom dat de data kunnen worden afgeschermd, zodat er geen onrechtmatige toegang of kopiëren mogelijk is. Afhankelijk van de juridische status zijn de restricties die de KB op toegang moet leggen meer of minder stringent. Bijvoorbeeld: voor recente publicaties waar auteursrecht op berust, mag alleen binnen de muren van de KB toegang worden gegeven. Voor andere publicaties mag wellicht wel online inzage worden gegeven, maar er mogen geen kopieën van worden gemaakt.

In de proof-of-concept is een stringente afscherming ontwikkeld, waarin de onderzoekers de data niet mogen zien.

7.2. Story: onsite-variant van tools-to-data

Als dataleverancier en als onderzoeker
Wil ik dat de tools-to-data-oplossing onsite wel toegang biedt tot alle bestanden
Zodat onderzoekers binnen de muren van de dataleverancier wel de data kunnen inzien, bv. voor het ontwikkelen van hun algoritme, het oplossen van bugs of het zien van de context van vindplaatsen.

Prioriteit: lange termijn.

Toelichting: publicaties mogen altijd binnen de muren van de KB ter inzage worden gegeven, mits er kopieerbeveiliging is.

Opmerking: dat zou betekenen dat in de tools-to-data-oplossing onderscheid moet worden gemaakt tussen gebruikers binnen en buiten het gebouw van de dataleverancier. Sommige bestanden zouden onsite wel getoond mogen worden, maar niet online. Of dit nodig is, hangt af van de discussie met de juridische adviseurs.

7.3. Story: API-variant van tools-to-data

Als dataleverancier

Wil ik dat tools-to-data de API's van mijn data-infrastructuur kan gebruiken
Zodat ik kan garanderen dat bepaalde data niet buiten de 'virtuele muren' van mijn organisatie komen.

Toelichting: dit speelt bijvoorbeeld bij de KB bij auteursrechtelijk beschermde publicaties of bij het webarchief. Zonder instemming van de rechthebbenden is het wettelijk alleen toegestaan dat de KB deze binnen de eigen muren beschikbaar stelt. Dat betekent dat de data binnen de 'muren' van de KB moeten blijven. Tools-to-data kan deze data dan via data-API's benaderen.

Nog ter discussie: het is juridisch nog niet uitgekristalliseerd wat er wel of niet mag met een tools-to-data-omgeving.

Niet-functionele requirements

This section describes quality attributes of the solution that influence how well it performs, scales and how reliable it is. These requirements are based on the CLARIAH development requirements.

1. Algorithms are executed in containers

If researchers provide algorithms as code (story) this means the software must be able to build the code's dependencies and package the whole in a container.

The data itself must be *mounted* in the container, not built into it. Exchange data using S3 token?

2. Containers run on a cluster

To enable scalability and portability, it must be possible to scale containers horizontally. SURF's Data Exchange currently runs on a VM (SURF HPC VM).

3. Containers are checked for security

Whether the container image is provided by researchers (story) or built by the application (story), its security must be validated before it is run. For example, the following criteria must be met:

- the container runs as a user other than root;
- the container does not include vulnerable packages.

4. Containers are isolated

Task containers will be provided with the data that the algorithm will act on, so no network access is allowed. Therefore, the application runtime must not rely on network access. Each container has access only to the corpus selection made by the researcher. So task containers must have no dependencies on other containers or services and any data other than the corpus selection.

5. Tasks are queued

To prevent overloading the host system, tasks must not directly be executed but published to a message queue first. Queue subscribers monitor load and run new tasks when capacity becomes available. The queue acts as a throttling mechanism.

6. The software can read text metadata

To enable Epic: Selectie van data, the software must be able to consume metadata that is provided with the texts by the data provider. Preferably, that metadata is available in RDF.

7. The software is tested

The software's source is validated by a suite of automated tests that are automatically run when commits are pushed to the source code. Code coverage must be 80% or higher.

8. Software releases are well-managed

Software releases follow Semantic Versioning so third parties using the software can rely on backwards compatibility within major versions.

9. The software separates code and configuration

No configuration values are hard-coded; all configuration is done through environment variables.

10. The software is documented

The source code must be accompanied by at least a README.md file that explains how to run the software locally, how to run the tests and how to contribute improvements.

11. Scalability

Bijlage: requirements voor de dataleverancier

Een tools-to-data-omgeving stelt ook eisen aan de manier waarop dataleveranciers hun datasets beschikbaar stellen. Dit zijn geen requirements voor de tools-to-data-omgeving zelf, maar het zijn voorwaarden om op een goede manier met tools-to-data te kunnen werken. De dataleveranciers zullen ervoor moeten zorgen dat aan deze requirements wordt voldaan.

Ook deze requirements zijn in de vorm van epics en user stories beschreven en worden in deze bijlage opgenomen.

8. Epic: metadata / informatie over de digitale collecties

Dataleveranciers moeten voldoende informatie bieden over de datasets, zodat onderzoekers kunnen beoordelen welke data relevant zijn voor hun onderzoek en hoe hun algoritmes gebruik kunnen maken van de bestanden.

8.1. Story: inzicht geven in de inhoud van een collectie

Als dataprovider
Wil ik inzicht geven in de inhoud van een collectie
Zodat ik kan zorgen dat onderzoekers een goede keuze kunnen maken welke collecties ze selecteren.

Opmerking: het gaat hier om een inhoudelijk inzicht in de collectie: wat voor soort inhoud (bv. Nederlandse literatuur), herkomst (bv. gedigitaliseerde boeken van de KB), eventueel een indruk van de spreiding (bv. welke auteurs, welke periode), etc. De dataprovider zal hiervoor een goede vorm moeten vinden.

Dit zijn metadata op collectie/dataset-niveau. Hieronder vallen ook 'provenance'-metadata vallen, die informatie geven over de herkomst van de data, die de onderzoeker helpen de betrouwbaarheid van de data te beoordelen. En eventuele metadata over de kwaliteit van de data, waarmee de onderzoeker de bruikbaarheid van de dataset kan beoordelen.

8.2. Story: inzicht in de structuur van een collectie

Als dataleverancier
Wil ik inzicht bieden in de structuur van een collectie (bv. opbouw xml, directorystructuur)
Zodat ik de data zodanig kan aanbieden dat de onderzoeker weet welke datastructuren het algoritme kan verwachten.

Toelichting: het gaat hier om technische informatie over hoe de data is geordend. Ten eerste de ordening van bestanden, bv. in een directorystructuur. Ten tweede de ordening binnen een bestand, bv. de opbouw van xml-bestanden (xml-schema). Dit vereist dat de dataleverancier hier goede beschrijvingen van levert. De dataleverancier zal hiervoor een goede vorm moeten vinden.

Ter discussie: in welke vorm moet deze informatie beschikbaar worden gemaakt? Een readme-file?

8.3. Story: inzicht in de afzonderlijke collectie-items

Als dataleverancier
Wil ik informatie bieden over de afzonderlijke collectie-items (metadata en fulltext)
Zodat ik de data kan aanbieden op een manier waarop de onderzoeker kan selecteren welke data relevant zijn en deze informatie kan gebruiken in de analyses.

Toelichting: het gaat hier om metadata op afzonderlijk document/item-niveau.

9. Epic: verwijzen naar data

Dataleveranciers moeten goede referentiemogelijkheden voor hun data bieden, zodat onderzoekers goed kunnen verwijzen naar vindplaatsen in de collectie, om te kunnen tonen waar ze hun onderzoeksresultaten op baseren.

9.1. Story: duurzame identifiers

Als dataleverancier
Wil ik duurzame identifiers voor mijn data
Zodat onderzoekers naar mijn data kunnen verwijzen om hun resultaten te verantwoorden en zodat replicatie mogelijk is, ook op de langere termijn.

Toelichting: deze identifiers zullen worden opgenomen in de rapporten van tools-to-data.

9.2. Story: fijnmazige identifiers

Als dataleverancier
Wil ik fijnmazige identifiers voor mijn data,
Zodat onderzoekers kunnen verwijzen naar de precieze vindplaatsen voor hun bevindingen en zo hun resultaten kunnen verantwoorden.

Toelichting: voor de datasets van de KB geldt dat meestal publicaties als geheel een persistente identifier hebben, maar de onderzoekseenheid is vaak veel kleiner bv. een paragraaf, zin of woord. Idealiter zijn daar ook duurzame identifiers voor. Het is de vraag hoe fijnmazig dataleveranciers dit kunnen en willen maken. Als alternatief kunnen fijnmazige identifiers ook door het algoritme worden toegekend ipv door de dataleverancier.

9.3. Story: versies van data

Als dataleverancier
Wil ik versiebeheer op mijn data

Zodat ik identifiers kan aanbieden die verwijzingen naar de data leiden die onderzoekers hebben gebruikt heb voor hun onderzoek om zo hun resultaten controleerbaar en repliceerbaar te mkaen.

Toelichting: strikt genomen vloeit dit al voort uit de requirement voor persistent identifiers, omdat een persistent identifier altijd naar hetzelfde moet blijven wijzen. Een nieuwe versie van de data vraagt dus om een nieuwe identifier, terwijl de oude blijft bestaan.

10. Epic: beoordeling van verzoeken

De dataleverancier moet procedures ontwikkelen voor de afhandeling van verzoeken van onderzoekers.

10.1. Story: afhandeling toegangsverzoeken

Als dataleverancier
Wil ik heldere criteria en procedures voor het behandelen van toegangsverzoeken
Zodat deze efficiënt kunnen worden afgehandeld.

Toelichting: het gaat hier om de verzoeken van onderzoekers om gebruik te mogen maken van de Tools-to-Data-omgeving. (Specifieke verzoeken voor het gebruik van afzonderlijke collecties of algoritmes worden binnen de Tools-to-Data-omgeving afgehandeld). Denk aan:

- *hoe kunnen onderzoekers een verzoek indienen*
- *wie beoordeelt dat*
- *wat zijn de criteria voor het verlenen van toegang*

10.2. Story: beoordelingscriteria voor vrijgeven resultaten

Als dataleverancier
Wil ik dat er duidelijke criteria zijn voor het vrijgeven van de resultaten van algoritmes
Zodat dit eenduidig en efficiënt kan worden beoordeeld.

Toelichting: als een algoritme wordt gedraaid binnen de Tools-to-Data-omgeving, dan kan de dataleverancier het resultaat eerst controleren voordat het wordt vrijgegeven aan de onderzoeker. Zo kan worden voorkomen dat er data weglekt, bv. doordat het algoritme de oorspronkelijke tekst kopieert. Het moet duidelijk zijn wanneer hier sprake van is. Bijvoorbeeld: als het algoritme woorden in context laat zien (een aantal woorden vóór en ná een zoekwoord). De criteria voor deze beoordeling zullen ook juridisch getoetst moeten worden.

11. Epic: collecties beschikbaar stellen

Dataleveranciers kunnen hun digitale collecties als verschillende datasets beschikbaar stellen, waarin de collectie-items op een logische manier worden gegroepeerd. Dit zal verschillend zijn per soort dataset en per dataleverancier.

11.1. Story: indeling collecties op basis van juridische status

Als dataleverancier
Wil ik een verzameling items gebundeld als collectie aanbieden op basis van hun juridische status
Zodat ik de onderzoekers toestemming kan geven met data te werken die dezelfde juridische status hebben.

Toelichting: bij de KB kunnen bijvoorbeeld recente e-books waar auteursrecht op rust, samen één collectie vormen. Of historische kranten waarover met belangenorganisatie afspraken zijn gemaakt.

Opmerking: deze user story kan gecombineerd worden met Story: indeling collecties op basis van structuur.

11.2. Story: indeling collecties op basis van structuur

Als dataleverancier
Wil ik een verzameling items gebundeld als collectie aanbieden op basis van hun structuur
Zodat ik onderzoekers hierover goede informatie en voorbeeldbestanden kan geven.

Toelichting: het gaat hier om de manier waarop de bestanden in elkaar zitten. Bij de KB zijn dit bv. xml-bestanden volgens een bepaald xml-schema, e-pubs e.d. De DBNL-boeken kunnen bijvoorbeeld één dataset vormen (TEI-formaat), de gedigitaliseerde boeken in Delpher een andere (ander xml-schema).

Opmerking: deze user story kan gecombineerd worden met Story: indeling collecties op basis van juridische status.

11.3. Story: alle collecties beschikbaar via tools-to-data

Als onderzoeker
Wil ik dat alle collecties van de dataleveranciers beschikbaar zijn ongeacht hun juridische status
Zodat ik ze kan combineren voor mijn onderzoeksalgoritmes.

Toelichting: de tools-to-data-oplossing is weliswaar in eerste instantie bedacht om onderzoekers te kunnen laten werken met 'gevoelige' collecties wegens auteursrecht, privacy of contractuele beperkingen. Maar een onderzoeker is eigenlijk niet geïnteresseerd in de juridische status, die wil bijvoorbeeld alle Nederlandse fictie uit de periode 1800-2020

kunnen analyseren. Dit betekent dat ook auteursrechtvrije boeken beschikbaar moeten zijn voor tools-to-data. Dit kan eventueel een andere collecties zijn, maar onderzoekers kunnen deze collecties combineren.

Nog ter discussie: moet de tools-to-data differentiëren in wat er mag met de data? D.w.z. moet binnen de tools-to-data-oplossing sommige bestanden wel kunnen worden ingezien door de onderzoeker en andere niet? Of wordt dat te complex?

11.4. Story: testset t.b.v algoritmeontwikkeling

Als dataleverancier
Wil ik per dataset een aantal testbestanden beschikbaar stellen (testset)
Zodat ik die beschikbaar kan stellen in een 'zandbak' waarmee de ontwikkelaar een algoritme kan ontwikkelen.

Toelichting: onderzoekers/ontwikkelaars gebruiken een iteratieve manier van werken met een wisselwerking tussen data en algoritme: verrijk, inspecteer, pas aan, verrijk opnieuw. Hiervoor is het noodzakelijk dat zij met een testset uit een collectie kunnen werken, die zij (in tegenstelling tot de complete dataset) wel kunnen bekijken. Dit is ook nodig om fouten uit het algoritme te ontdekken. De zandbak biedt hiervoor een oplossing, waarbij de data nog steeds in een veilige omgeving staan, maar wel kunnen worden bekeken.

De dataleverancier moet hiervoor een aantal bestanden ter beschikking stellen. Hiervoor kunnen bijvoorbeeld auteursrechtenvrije delen van de collectie worden gebruikt.

Nog ter discussie:

- *wat nu als er op de hele collectie auteursrechten berusten?*
- *gebeurt dit ontwikkelen op de eigen systemen van de ontwikkelaar?*

Het is de vraag wat er juridisch gezien mag met zo'n testset. Wellicht kunnen er bestanden worden gebruikt waar geen auteursrecht op rust, of waar het gecleard is. Als dat niet mogelijk is, kan de onderzoeker naar de KB komen want daar mogen alle bestanden wel ter inzage worden gesteld. Dat betekent dat je een online en een onsite-versie van Tools-to-Data moet hebben.