TEKSTLOCALISATIE OP BASIS VAN ANALYSE VAN VERBONDEN COMPONENTEN

Bachelorproject

Popke Altenburg, s1444239, paltenburg@gmail.com

Samenvatting: Er is veel onderzoek naar zowel tekstherkenning als automatische schrijversidentificatie op basis van geschreven tekst. Op ingescande documenten staat echter vaak veel meer dan alleen tekst, zoals perforatiegaatjes, vlekken, of plaatjes. Deze onderdelen kunnen de automatische herkenningsmethoden erg verstoren, en hun prestaties verlagen. Mijn methode analyseert een gedigitaliseerd geschreven document om zo voor verdere toepassingen aan te geven wat wél tekst is en wat eventueel genegeerd kan worden. Van elk zogenaamd verbonden component (aaneengesloten vorm, zoals een letter, een woord, of een vlek) wordt een lijst van de volgende eigenschappen bepaald: hoogte, breedte, omtrek, lengte van het skelet en gemiddelde pendikte. Op basis van deze eigenschappen worden de verbonden componenten met 'k-nearest neighbor' classificatie geclassificeerd als tekst of non-tekst, en wordt zo het document gefilterd. Ook wordt er gekeken hoe belangrijk de afzonderlijke kenmerken zijn voor de classificatie.

1. Inleiding

Analyse van geschreven tekst in een veel onderzocht onderzoeksonderwerp. Er wordt veel gedaan aan de automatisering van tekstherkenning, maar ook aan het identificeren van de schrijver van een document, welke bijvoorbeeld gebruikt kan worden voor forensisch onderzoek, of voor het vinden van de schrijver van een historisch document.

Methoden die worden gebruikt, worden meestal toegepast op gedigitaliseerde documenten. Op de grafische bestanden die dit oplevert staat echter vaak veel meer dan alleen tekst, zoals bijvoorbeeld nietjes, perforatie-gaatjes, vlekken, of simpelweg de randen van het document. Methoden voor analyse van handschrift kunnen deze onderdelen niet altijd van echt schrift onderscheiden en ze worden dus vaak meegenomen in de analyse. Dit kan het proces verstoren, en hun prestaties naar beneden halen.

De methode die wordt besproken geeft een oplossing voor dit probleem. Het is een voorbewerkingstap die in een grafisch bestand het gebied wat tekst bevat aangeeft. De manier waarop dit gebeurt is door stukken van het grafische bestand scores mee te geven voor de waarschijnlijkheid dat het ofwel tekst, ofwel niet-tekst is.

Op deze manier kan aan andere methoden voor handschrift analyse worden doorgegeven waar zich de tekst bevindt, zodat storende onderdelen kunnen worden genegeerd.

In de literatuur wordt dit probleem niet zo specifiek behandeld. Er is wel onderzoek naar zogenaamde 'text information extraction', dit richt zich ook op het extraheren van tekst uit documenten en plaatjes, maar dan gaat het vaak over ingewikkelde plaatjes, zoals bijvoorbeeld covers van tijdschriften of videobeelden. Hiervan bestaat een goed overzicht (Jung et al., 2004). Ik richt me specifiek op gedigitaliseerde geschreven documenten.

Verder baseer ik me op een gerelateerd bachelorproject (Renkema, 2006). Hierin wordt hetzelfde probleem behandeld, maar op een algemenere manier. Zijn methode streeft naar zo weinig mogelijk input van de gebruiker, waarbij de methode zoals voorgesteld in deze paper eerst moet leren van enkele voorbeelden die soortgelijk zijn aan te filteren documenten, zodat daarna kan worden gegeneraliseerd over de hele groep van te filteren documenten. Dit heeft als voordeel dat bij het filteren gebruikt wordt gemaakt van de specifieke eigenschappen van de betreffende groep.

2. Methode

Om de gebieden in een document aan te geven die tekst zijn, baseer ik me op de zogenaamde verbonden componenten in dit document, waarover meer uitleg in sectie 2.1.

Er wordt van elk van deze verbonden componenten zo goed mogelijk vastgesteld of

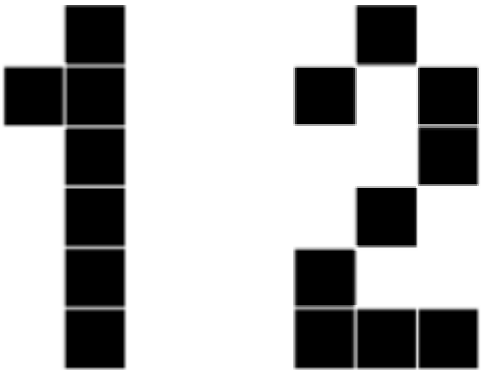
deze tot de tekst behoord (dus of het bijvoorbeeld een letter, of een woord is), of of deze non-tekst is (dus bijvoorbeeld een vlek, of een rand van het papier).

Eerst wordt van elk verbonden component een aantal numerieke eigenschappen vastgesteld. Vervolgens worden deze op basis hiervan geclassificeerd als zijnde tekst, of niet-tekst.

2.1. Verbonden componenten

Verbonden componenten zijn groepen pixels (of: beeldpunten) die met elkaar verbonden zijn.

Met 'verbonden' kunnen twee dingen worden bedoeld. Een verbinding tussen twee pixels is '4-connected' als ze direct naast elkaar, of boven elkaar liggen. Echter als ze boven, naast, of ook diagonaal van elkaar liggen, is de verbinding '8-connected', zoals te zien is in figuur 2.1. Alle 4-connected verbindingen zijn dus ook 8-connected, maar niet andersom.



Figuur 2.1: In de linker figuur zijn alle pixels zowel 4- als 8-connected met elkaar verbonden, alle verbindingen tussen de pixels zijn namelijk ofwel horizontaal, of verticaal. In de rechter figuur zijn niet alle pixels 4-connected verbonden, maar wel 8-connected. Alle verbindingen zijn hier namelijk ofwel horizontaal, verticaal of diagonaal.

Ik gebruik 4-connected verbonden componenten. Verbindingen die exclusief 8-connected zijn komen vaak voor in vormen van 1 pixel dikte, zoals dunne lijnen. Als deze echter op een voldoende hoge resolutie worden ingescand, zijn de lijnen dikker dan 1 pixel, zodat deze 4-connected zijn. Ik ga uit van ingescande documenten met een voldoende hoge resolutie zodat er geen belangrijke vormen in voor komen die slechts 1 pixel dik zijn.

2.2. Voorbewerking

Voordat de verbonden componenten vastgesteld kunnen worden, moet het document waarmee wordt gewerkt eerst worden voorbewerkt.

Het document wordt ingescand, wat een pixelbestand oplevert. Dit bestand wordt eerst omgezet van kleur- naar grijswaarden, zodat elke pixel alleen nog informatie bevat over licht en donker. In dit bestand staat dan zwarte, of donkergrijze tekst op een witte, of heel lichtgrijze achtergrond.

Vervolgens wordt dit omgezet naar een bestand met nog slechts twee kleuren. Dit op zo'n manier dat de voorgrond, dus tekst en andere vormen, de kleur zwart krijgt toegewezen, en de achtergrond, dus de lichte kleur van het papier, de kleur wit krijgt. Een veelgebruikte methode hiervoor, die ik ook gebruik, is het algoritme volgens Otsu (Otsu 1979).

In dit bestand worden nu de verbonden componenten vastgesteld, op de manier zoals reeds vermeld, namelijk 4-connected.

2.3. Eigenschappen

Van elke verbonden component wordt nu zes eigenschappen vastgesteld, waarbij elke eigenschap de vorm heeft van een scalaire waarde. Zo krijgt elke component een lijstje van zes waarden, de zogenaamde *featureset*.

De eerste vier eigenschappen zijn:

- hoogte: afstand van de bovenste tot de onderste pixel van het verbonden component, in aantal pixels.
- breedte: afstand van meest linkse, tot de meest rechtse pixel van het verbonden component, in aantal pixels.
- oppervlak: het aantal pixels waar het verbonden component uit bestaat.
- omtrek: het aantal pixels van het verbonden component wat aan de rand ligt, dat wil zeggen wat verbinding maakt met het wit van het document. Ik heb gekozen voor 4-connected verbindigen.

Zie figuur 2.2 voor illustraties van deze eigenschappen.

De volgende twee eigenschappen zijn wat ingewikkelder, en hebben te maken met de middellijn van de vorm van de verbonden component. Dit is een curve van één pixel dik, die loopt door het midden van de vorm (zie

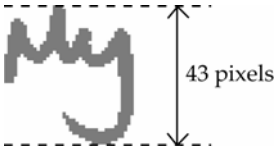


Fig 2.2a: hoogte

Fig 2.2b: breedte

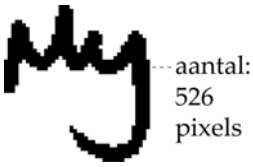


Fig 2.2c: oppervlak

Fig 2.2d: omtrek

figuur 2.3). Ik gebruik deze, omdat deze vrij goed overeenkomt met de curve die de pen of potlood (of welk ander schrijfinstrument dan ook) heeft gemaakt bij het schrijven van de betreffende vorm.

Figuur 2.3a: Middellijn van een verbonden component.

Figuur 2.3b: Het verbonden component waarvan de middellijn in figuur 2.3a getoond wordt.

De middellijn wordt vastgesteld met een zogenaamd thinning-algoritme (Huang et al., 2003). Dit algoritme stelt echter alleen de vorm van de middellijn vast. Ik gebruik een door mijzelf uitgebreide versie die op elk punt van de middellijn ook aangeeft hoe dik de vorm op dat punt is.

Het algoritme volgens Huang et. al. werkt grofweg zo dat een bepaald verbonden component zodanig wordt bewerkt dat de figuur die de middellijn voorstelt overblijft. Deze bewerking bestaat uit herhaalde stappen, waar in elke stap de pixels die aan de rand liggen (dus: verbonden zijn met pixels die behoren tot de achtergrond, in dit geval wit) worden verwijderd, behalve als deze pixels deel uit maken van de middellijn. Zo wordt de dikte van een verbonden component dus bij elke stap verminderd. Deze stap wordt herhaald totdat er alleen nog pixels over zijn die tot de middellijn behoren. Zie het artikel voor de details.

Het uitgebreide deel van de versie die gebruikt wordt werkt als volgt: Omdat de dikte van de v.c. elke stap verminderd wordt met een rand van pixels van één pixel dikte, wordt de dikte van de v.c. op een bepaald punt (dus op een bepaalde pixel van de middellijn) uitgedrukt in aantal stappen voordat deze middellijn pixel aan de rand komt te liggen. De dikte van een vorm op een bepaald punt van de middellijn wordt op deze manier gegeven als de afstand in aantal pixels van dat punt tot de dichtstbijzijnde rand van de vorm.

De twee eigenschappen die vervolgens vastgesteld worden aan de hand van de middellijn zijn:

- lengte van middellijn: aantal pixels waar de middellijn uit bestaat. Omdat de middellijn één pixel dik is, is dit een goede maat voor lengte.
- gemiddelde dikte van middellijn: Het gemiddelde van de diktes van alle punten in de middellijn van een verbonden component vastgesteld zoals boven genoemd. Deze dikte is een goede maat voor de dikte van de punt van het instrument waarmee geschreven is.

Nu de zes eigenschappen van de verbonden componenten zijn vastgesteld hebben we voor het document een lijst met data, namelijk voor elk verbonden component de zes waarden van zijn eigenschappen, ofwel de featureset. Zie figuur 2.4 voor een voorbeeld van een featureset.

2.4. Classificatie

Het doel is nu aan de hand van deze waarden zo goed mogelijk onderscheid te maken tussen de verbonden componenten die tekst zijn, en die, die dat niet zijn.

height	width	edgel	surface	skel_l	penthck
35	1	35	35	35	10
17	1	17	17	17	10
16	1	16	16	16	10
10	1	10	10	10	10
2	1	2	2	2	10
1	1	1	1	1	10
6	4	12	14	4	15
106	47	492	1375	268	31
72	33	332	743	180	24
101	17	239	632	113	29
74	36	221	698	109	35
6	6	15	26	1	30
60	38	271	616	137	25
4	4	10	12	3	17
1	1	1	1	1	10
62	22	261	475	128	21
63	83	310	891	162	31
1	1	1	1	1	10
1	1	1	1	1	10
12	23	45	59	24	15
54	85	310	822	167	28
1	1	1	1	1	10
58	20	233	563	112	28

Figuur 2.4: Voorbeeld van een featureset

Dit wordt gedaan met statistische classificatie. De methode die wordt toegepast is de k-nearest neighbour classificatie, afgekort als k.n.n. Het k.n.n. algoritme dat wordt toegepast geeft aan elk verbonden component een klasse score. De twee klasses zijn "tekst" en "non-tekst", en de score is een continue waarde tussen 0 en 1, waar 1 staat voor tekst en 0 voor non-tekst. Hoe dichter de score ligt bij een van beide waarden, hoe waarschijnlijker de bijbehorende klasse is.

2.4.1 referentiesets

Om deze klasse score vast te stellen is informatie nodig over op welke manier de featureset zou moeten worden verdeeld in tekst en non-tekst. Het probleem is dat het moeilijk is te zeggen welke waarden van de eigenschappen in het algemeen, dus voor alle denkbare gevallen van geschreven documenten, behoren tot beide klassen. Dit in de eerste plaats omdat het ondenkbaar is om van alle mogelijke soorten geschreven documenten voorbeelden te krijgen om eigenschappen van vast te stellen. Bovendien is het mogelijk dat verbonden componenten met een bepaalde combinatie van waarden van eigenschappen in de ene soort documenten van een andere klasse zijn dan in de andere soort. Dus dat de verdeling van klassen over de featuredata verschilt per soort document.

Hoe de waarden van eigenschappen verdeeld zijn over een document, en waar de scheiding ligt tussen tekst en non-tekst verschilt dus per soort document. Zo zullen de letters en woorden van een groep historische documenten een andere verdeling van grootte, lijndikte, oppervlakte etc. hebben als die van een groep geschreven getuigenverklaringen. Er kan zelfs verschil zitten tussen bepaalde groepen historische documenten onderling. Voor deze methode wordt de informatie die nodig is om zo juist mogelijke klasse scores vast te stellen gehaald uit andere documenten van dezelfde groep. Voor een goede classificatie is het dus belangrijk dat de verdeling van de eigenschappenwaarden binnen de groep documenten waarmee wordt gewerkt niet teveel van elkaar verschilt.

Uit een paar documenten uit een groep van gelijksoortige documenten wordt handmatig de informatie gehaald hoe de data verdeeld is over de beide klassen, zodat dit automatisch kan worden toegepast op de gehele groep. Dit wordt gedaan aan de hand van een referentie featureset (ofwel: referentieset). Dit is een featureset die vastgesteld is zoals genoemd, maar waar ook de informatie aan is toegevoegd tot welke klasse een verbonden component behoort.

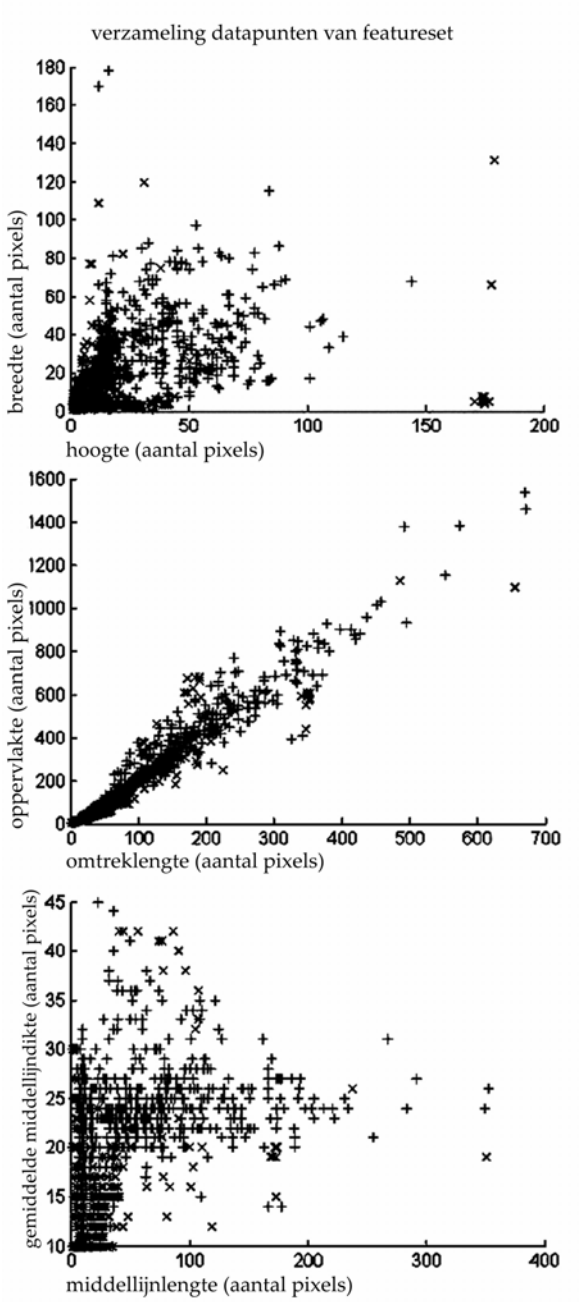
De referentieset wordt handmatig gevormd. Van een (of enkele) gedigitaliseerd(e) document(en) wordt een tweede versie gemaakt waarin handmatig wordt aangegeven welke gebieden tekst zijn en welke niet, door deze gebieden digitaal in te kleuren met een tweewaardige kleur. Deze geeft aan van welke klasse een gebied is, dus tekst of non-tekst. Deze handmatig aangegeven waarden worden ook wel groundtruth waarden genoemd.

De referentieset wordt nu samengesteld door de eigenschappen, van de verbonden componenten uit het (of de) document(en) waarvan een versie met groundtruth waarden is gemaakt, vast te stellen zoals genoemd, met de toevoeging dat ook de bijbehorende klasse waartoe de v.c. behoord (dus tekst of non-tekst) wordt uitgelezen uit de zogenaamde groundtruth versie van het document.

2.4.2 k-nearest neighbor classificatie

De featureset kan worden gezien als een verzameling punten in een zesdimensionale ruimte, waar elk punt staat voor een verbonden component, en waar de zes coördinaten van dat

punt staan voor de zes eigenschappen. Zie figuur 2.5 voor een voorbeeld van een verzameling punten. Vervolgens kan de euclidische afstand tussen twee punten berekend worden.



Figuur 2.5: Voorbeeld van de verdeling van een verzameling datapunten van de featureset van één document, weergegeven in drie figuren met elk twee dimensies. Welke eigenschappen tegenover elkaar staan is vrij arbitrair, maar geeft tenminste een idee van de verdeling van punten.

K.n.n. werkt nu zo dat voor elk te classificeren verbonden component zijn overeenkomstige punt in de zesdimensionale ruimte wordt genomen. Vervolgens worden de euclidische afstanden berekend van dit punt tot alle punten van de referentieset. Van een bepaald aantal punten met de kortste afstand wordt de klasse informatie genomen. Dit aantal is van te voren vastgesteld (het wordt vaak aangeduid met de letter "k", vandaar de naam). Een aantal van 12 blijkt goed te werken. De klasse informatie heeft de waarde 0 als het van de klasse "non-tekst" is, en 1 als het van de klasse "tekst" is. Deze waarden van de 12 genomen punten worden gemiddeld, zodat er een waarde ontstaat tussen de 0 en 1. Dit is de genoemde klasse score.

Voor elk verbonden component wordt dit gedaan zodat elk verbonden component een score krijgt die aangeeft tot welke klasse het het meest waarschijnlijk is dat het component behoort.

2.4.3 z-scores

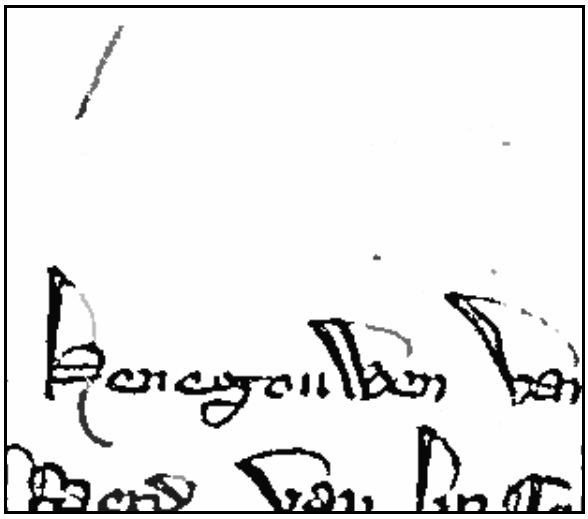
Het bereik van de waarden van de verschillende eigenschappen onderling kan vrij uiteen lopen. Dit betekent dat de eigenschappen die vaak grotere waarden hebben zwaarder worden meegewogen in de k.n.n. classificatie.

Daarom wordt, alvorens de k.n.n. classificatie toe wordt gepast, elke waarde omgezet naar zijn z-score. Het gemiddelde en de standaard deviatie die hierbij wordt gebruikt worden genomen over alle waarden van de eigenschap waartoe de desbetreffende waarde behoort. Op deze manier worden grote verschillen in waardebereik tussen de eigenschappen onderling vermeden.

2.4.4 filteren naar dichtheid

Elk verbonden component in het document heeft nu een score tussen 0 en 1, welke staat voor de mate waarin het tekst is of niet. Deze score is alleen gebaseerd op zijn individuele eigenschappen.

Een probleem wat hierbij voor kan komen is dat er een verbonden component is wiens eigenschappen erg lijken op die van een letter of een woord, ondanks dat het er niet een is, zoals te zien op figuur 2.6, Deze is dus niet te scheiden van tekst alleen op basis van zijn eigenschappen. Er is echter nóg een manier waarop we het onderscheid kunnen maken, namelijk met behulp van de aanname dat alle tekst in een document gegroepeerd staat.



Figuur 2.6: De streep linksboven is niet onderdeel van de tekst, maar wordt toch als zodanig gezien.

Als er nu een verbonden component is dat qua zijn eigenschappen lijkt op een letter of een woord, maar buiten de tekst ligt, en omringt door allerlei non-tekst componenten, is het duidelijk dat het niet als tekst geclassificeerd zou moeten worden. Ditzelfde geld ook voor het omgekeerde geval, dus dat een v.c. dat qua eigenschappen niet op tekst lijkt, maar er wel middenin ligt, wel behoort tot het tekstgebied.

Dit is waar de volgende verwerkingsstap op is gebaseerd. Deze werkt als volgt:

Voor elk verbonden component wordt een nieuwe score vastgesteld. Deze nieuwe score neemt de scores zoals gegeven door het k.n.n. algoritme van de omliggende verbonden componenten mee.

Om voor een verbonden component een nieuwe score op te stellen wordt gekeken naar alle andere omliggende componenten. Van elk van deze omliggende componenten worden de volgende drie eigenschappen genomen:

- de score zoals vastgesteld door het k.n.n.-algoritme.
- het oppervlak. Dit om zijn belang ervan mee te wegen, zodat bijvoorbeeld een v.c. met een bepaalde oppervlakte twee keer zo zwaar meeweegt als twee andere v.c.'s die samen datzelfde oppervlak in beslag nemen.
- de afstand tot het verbonden component waarvan de nieuwe score wordt vastgesteld.

De afstand wordt nu getransformeerd op zo'n manier dat het een maat voor 'dichtbijheid' wordt. Dat wil zeggen hoe kleiner de afstand, hoe groter de getransformeerde waarde. De transformatie gebeurt met de volgende functie:

$$T = \left(1 - \frac{1}{C}\right)^{A}$$

Waar A staat voor de afstand van de omliggende component tot die waarvoor de nieuwe score moet worden vastgesteld, T voor de getransformeerde waarde, en C voor de constante die bepaald hoe snel de getransformeerde waarde afneemt naarmate de afstand toeneemt. Een grotere waarde voor C leidt tot een kleinere waarde van $\left(1 - \frac{1}{C}\right)$, wat dus leidt tot een snellere afname. Een waarde van 1/50 van de breedte van het document, gemeten in aantal pixels, blijkt goed te werken.

Voor elke omliggende component worden de k.n.n.-score, het oppervlak en de transformeerde afstand met elkaar vermenigvuldigd. Dit geeft een waarde die staat voor de bijdrage van deze component aan de nieuwe score. De bijdrages van alle omliggende componenten worden bij elkaar opgeteld, wat resulteert in de nieuwe score.

Dit wordt gedaan voor iedere verbonden component, zodat de meeste valse classificaties worden gecorrigeerd.

3. Resultaten

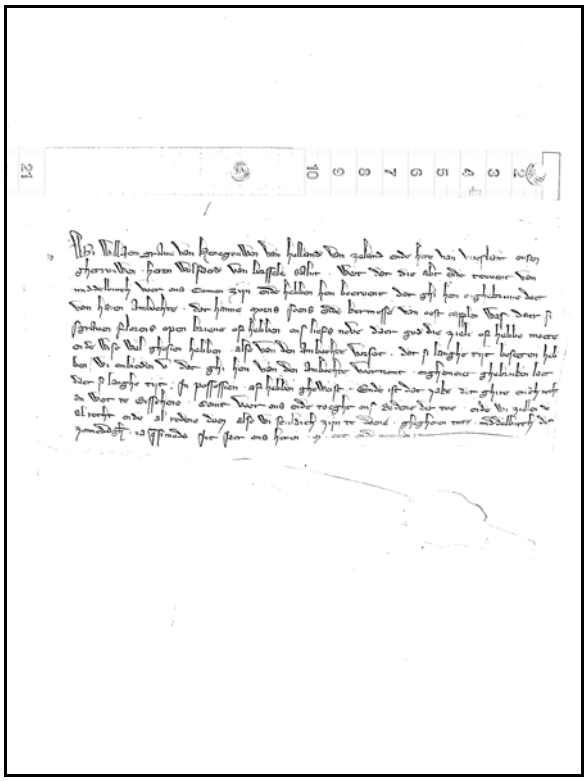
In deze sectie zal ik de resultaten bespreken van de methode.

Ik zal ook enkele tussenresultaten laten zien om te illustreren wat het effect is van de verschillende onderdelen van de methode.

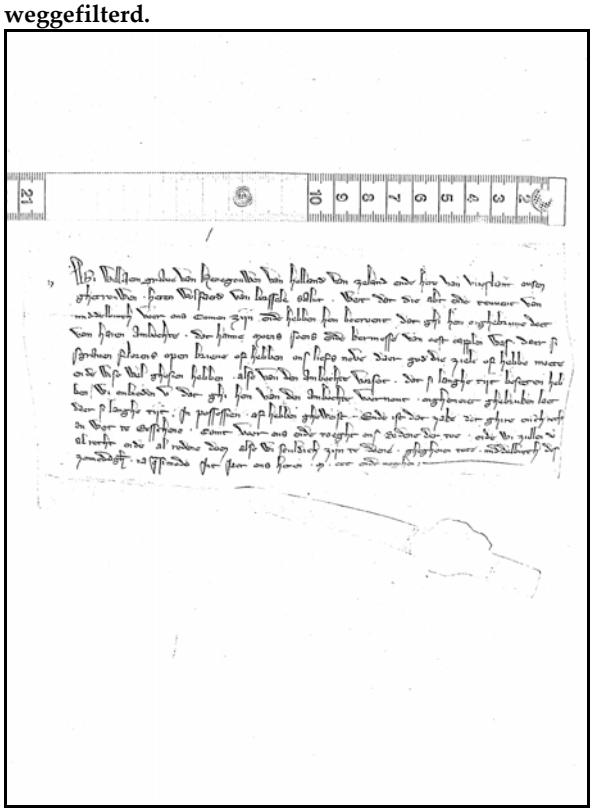
3.1. K.n.n. resultaten

Hier bespreek ik de resultaten van een document dat gefilterd is door een deel van voorgestelde methode. Het document is verwerkt tot zover als de k.n.n. classificatie, en is nog niet naar dichtheid gefilterd zoals beschreven wordt in sectie 2.4.4. Voor het resultaat: zie figuur 3.1.

Het is een historische oorkonde (met dank aan Jinna Smit, UvA). Zoals te zien zijn er verbonden componenten niet goed geclassificeerd. Er staan in de tekst enkele v.c.'s die geclassificeerd zijn als non-tekst, terwijl er buiten de tekst enkele zijn die onterecht als tekst zijn geclassificeerd. Op figuur 2.6 is een



Figuur 3.1a: Document verwerkt tot en met de k.n.n. classificatie. Niet alle non-tekst componenten zijn weggefilterd.



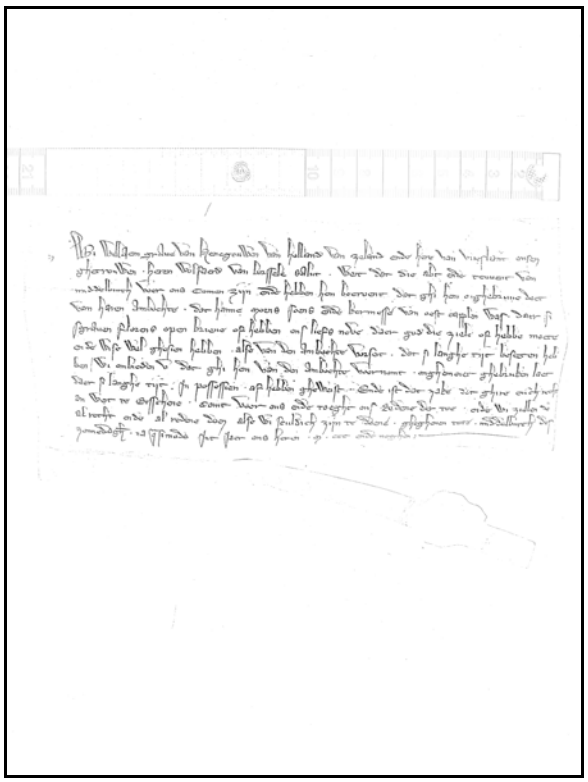
Figuur 3.1b: Het origineel van figuur 3.1a, voordat de methode werd toegepast.

voorbeeld te zien van een incorrect geclassificeerde v.c.

Het maakt het probleem goed duidelijk van het classificeren alleen op basis van de eigenschappen alleen. Zoals namelijk te zien is, heeft deze veel eigenschappen gemeen met een vorm die wel tekst is.

3.2. Resultaten na dichtheidsfilter

Hier zal ik de resultaten bespreken van een toepassing van de methode op het hetzelfde document dat wordt gebruikt in sectie 3.1, met toevoeging van de laatste stap welke de methode compleet maakt.



Figuur 3.2: Het resultaat na de filtering naar dichtheid.

Zoals op figuur 3.2 te zien wordt de klasse scores van een componenten beïnvloed zijn door hun buren. Het voordeel hiervan is dat veel foute classificaties worden gecorrigeerd door de omgeving waar ze in liggen.

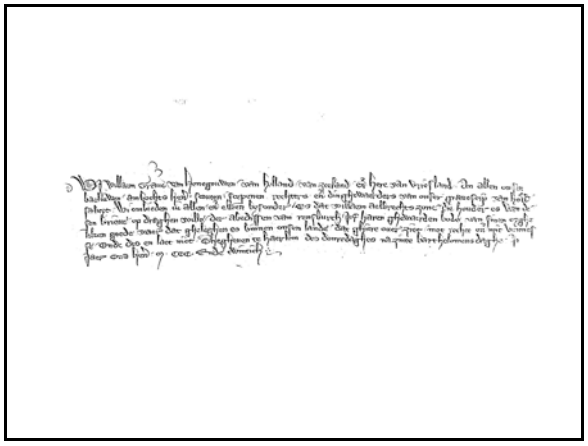
Het nadeel is echter dat componenten die aan de rand van een tekstgebied liggen invloed kunnen hebben van non-tekst componenten. In veel gevallen valt dit mee, omdat de dichtheid van componenten binnen het tekstgebied vaak groter is dan die van componenten búiten het tekstgebied. Zo worden componenten die aan de rand liggen sterker door de omliggende tekst-componenten beïnvloed, dan door non-tekst

componenten. Echter als er veel non-tekst componenten dicht bij een tekstgebied liggen, kan hun invloed wél de overhand hebben, wat resulteert in foute classificaties.

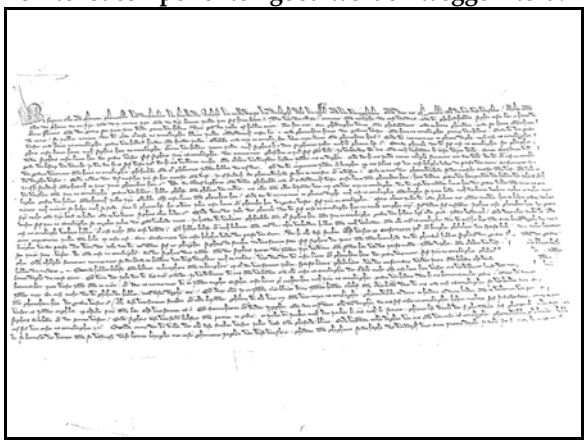
3.3. Toepassing op grote groep

Hier zal ik de resultaten bespreken van een toepassing van de methode op een groep van documenten. De methode wordt toegepast op een groep historische oorkondes, waar de oorkonde gebruikt in sectie 3.1 er een exemplaar van is. Deze groep is interessant, omdat de eigenschappen van de verbonden componenten in de oorkondes onderling overeenkomen, en op andere punten van elkaar verschillen. Dit geeft een goed beeld van de diversiteit die de methode aankan.

Wat in figuur 3.3 te zien is, is dat het effect dat tekstcomponenten aan de rand lagere classificatiescores krijgen (dus minder waarschijnlijk als tekst worden gezien, dan v.c.'s die midden in een tekstgebied liggen) verschilt per oorkonde. Dit effect is sterker bij oorkondes die qua eigenschappen meer afwijken van het document waar de refentie featureset op is gebaseerd.



Figuur 3.3a: Voorbeeld van een oorkonde uit de groep waarbij vrij goed te tekst bewaard blijft, en de non-tekst componenten goed worden weggefilterd.



Figuur 3.3b: Voorbeeld van een oorkonde waarbij non-tekst componenten ook goed worden weggefilterd, maar waar rechtsonder duidelijk te zien is dat er te veel van de tekst wordt weggefilterd.

4. Discussie

De voorgestelde methode filtert de meeste non-tekst componenten vrij goed. Er zijn echter punten waarop het kan worden verbeterd.

Ik zal eerst enkele punten bespreken waarop voorgestelde methode mogelijk verbeterd kan worden. Daarna zal ik enkele andere benaderingen van het probleem bespreken, welke mogelijk ook goed zouden kunnen werken.

4.1. Mogelijke verbeteringen

De voorgestelde methode heeft continue klasse scores als output. Deze waardes tussen 0 en 1 bevatten echter een bepaalde onzekerheid, omdat ze niet tweewaardig zijn, zoals "tekst" en "non-tekst". Het voordeel hiervan is dat deze wel informatie bevatten hoe zeker het is dat ze tekst of non-tekst zijn, afhankelijk van hoe dicht de waarde zit bij respectievelijk 1 of 0.

Het zou echter een verbetering zijn om tussen deze twee waarden automatisch een grens te kunnen stellen waaronder met een bepaalde zekerheid te zeggen dat het non-tekst is, en waarboven met een bepaalde zekerheid is te zeggen dat het tekst is. Zo'n grens zou mogelijk gesteld kunnen worden met een methode soortgelijk aan die van Otsu (Otsu, 1979), welke automatisch een grenswaarde bepaald voor grijswaarden in bijvoorbeeld een digitaal grafisch bestand.

Een andere verbetering kan liggen in de eigenschappen die worden geanalyseerd. In de voorgestelde methode wordt bijvoorbeeld kleur niet meegenomen, terwijl het best denkbaar is dat in een bepaald document kleur een onderscheidende factor is, bijvoorbeeld als de tekst met een blauwe pen is geschreven, kun je verbonden componenten die erg afwijken van die kleur als non-tekst classificeren.

Ook eigenschappen die de moeite van het noemen waard is, zijn die, die door Bulacu (Bulacu, 2003) gebruikt wordt voor het

determineren van de schrijver van een document. Deze neemt analyse van de vorm uitvoeriger mee. De output hiervan bestaat echter uit tweedimensionale histogrammen, welke dus niet direct geschikt zijn voor k.n.n. classificatie. Hiervoor kan óf de classificatiemethode worden aangepast, óf de histogrammen op een bepaalde manier omgezet worden naar scalaire waarden, zoals die van de eigenschappen in de voorgestelde methode.

4.2. Andere benaderingen

In voorgestelde methode wordt een te verwerken document vergeleken met een ander document van dezelfde soort, waarvan met de hand is aangegeven welke onderdelen tekst zijn, en welke niet. Het voordeel hiervan is dat de methode hiervan kan leren welke eigenschappen in welke combinaties horen bij tekst en welke bij non-tekst. Een nadeel hiervan is natuurlijk dat het de gebruiker tijd kost. Een ander nadeel is dat de methode alleen werkt op groepen documenten, en niet op losse. Dit nadeel valt wel enigszins mee, omdat het automatisch filteren vooral op grote groepen documenten aanzienlijk tijd bespaard, terwijl het handmatig filteren van enkele losse documenten niet veel werk is. Niettemin is een benadering voorstelbaar welke niet hoeft te worden vergeleken met andere documenten waarin de tekst is aangegeven.

Zo kan er worden gekeken naar de verdeling van de waarden die de eigenschappen van de verbonden componenten hebben. Vaak kan worden aangenomen dat tekst-v.c.'s vaker voorkomen dan non-tekst-v.c.'s. Als nu wordt gekeken welke combinaties van eigenschappen het meest voorkomen kunnen deze gekoppeld worden aan het vermoeden dat het tekst is, omdat is aangenomen dat tekstcomponenten ook het vaakst voorkomen, en zo componenten die sterk van deze eigenschappen, of combinaties ervan, afwijken classificeren als non-tekst. Welke combinaties van eigenschappen het meest voorkomen is bijvoorbeeld te bepalen door de featureset te beschouwen als punten in een multidimensionale ruimte, zoals gedaan wordt bij de k.n.n. classificatie, en dan te bepalen waar de dichtheid van de punten het grootst is.

Een andere benadering is om niet uit te gaan van verbonden componenten, maar het document met een raster op te delen in vierkante vakjes. Als op ieder vakje vervolgens een bepaalde vorm van patroonanalyse wordt toegepast kan op deze manier van het vakje bepaald worden of het tekst is of niet. Een voorbeeld van zo'n patroonanalyse is: Het vakje wordt opgedeeld in horizontale lijnen van één pixel dikte, binnen deze lijnen worden de lijnstukken van eenzelfde kleur bepaald, en vervolgens wordt een histogram op gesteld van de lengtes van deze stukken. De verdeling van lengtes is binnen een tekstgebied anders dan buiten een tekstgebied. Zo kan door te kijken naar de vorm van de histogram van het vakje worden bepaald of het tot een tekstgebied behoort of niet.

Natuurlijk zijn andere vormen van patroonanalyse, of de manier van indelen van het document mogelijk

5. Conclusie

De voorgestelde methode filtert in een document de meeste non-tekst component er vrij goed uit. Vooral voor grote groepen documenten is het een voordeel dat deze manier van voorbewerken met de voorgestelde methode automatisch kan, in plaats van handmatig.

De prestaties zijn echter wel sterk afhankelijk de mate van diversiteit van de documenten in de groep waar de methode op wordt toegepast, en werkt het best als de documenten zo veel mogelijk gelijksoortig zijn. Dit is helaas iets om rekening mee te houden, en onder andere daardoor is de methode nog vatbaar voor verbeteringen.

Referenties

Bulacu, M. (2003). Writer identification using edge-based directional features. In *Proc. of 7th Int. Conf. on Document Analysis and Recognition (ICDAR 2003)* (pp. 937-941). Edinburgh: IEEE Computer Society.

Huang, L., Genxun, W., Changping, L. (2003) An improved parallel Thinning Algorithm. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition* (pp. ?-?) Essex: IEEE Computer Society

Jung, K., Kwang, I. K., Jain, A. (2003) Text information extraction in images and video: a survey. *Pattern Recognition*, 37, 977-997.

Otsu, N. (1979). A threshold selection method from grey level histograms. *IEEE Trans. Systems, Man and Cybernetics*. 9, 62-66.
Renkema, S. (2006). Text Area Marking.