Autoriteiten vinden op Twitter

Bachelorscriptie

Naam: Bram Vonk
Studentnummer: 0118281
Begeleider: Prof.dr.ir. Th.P. van der Weide
Datum: 11 mei 2011

1 Inhoudsopgave

1 Inhoudsopgave 3
2 Definities 4
3 Onderzoeksvraag 5
4 Literatuur 6
5 Data verzamelen 8
6 Voorstel systeem 12
7 Voorstel systeem (2) 17
8 Meetmethode 19
9 Resultaten 22
10 Reflectie 25
11 Bibliografie 27
12 Bijlage 1: technische implementatie systeem 29
13 Bijlage 2: een tweet 32

2 Definities

Tweet	Een tekstbericht, waarvan de inhoud uit maximaal 140 karakters bestaat. Wordt door een tweeter geplaatst op zijn *timeline*, en wordt ook zichtbaar bij andere tweeters die hem volgen.
Tweeter	Geregistreerde gebruiker van Twitter, met een twitternaam. In het Nederlands vaak ook wel twitteraar genoemd.
Timeline	Een verzameling van tweets in chronologische volgorde. Elke tweeter heeft een 'home timeline', waarin hij de tweets ziet van de tweeters die hij zelf volgt.
Hashtag	In een tweet kunnen hashtags geplaatst worden, meestal om aan te geven over welk onderwerp de tweet gaat, om zo tweets met gelijksoortig onderwerp te kunnen groeperen, maar vaak ook om simpelweg nadruk te leggen op een woord. Het is simpelweg een #-teken direct gevolgd door het woord. Soms ontstaat een hashtag vanzelf, bijvoorbeeld #egypt voor tweets die over de onlusten in Egypte gaan, maar vaak is het ook afgesproken: zo vroeg het programma *De nationale iq-test* haar kijkers de tag #iqtest te gebruiken.
Usermention	In een tweet kan je een tweeter benoemen door een @ teken te plaatsen, direct gevolgd door zijn twitternaam. Hiermee kan je dus naar een andere tweeter verwijzen, aangeven dat je op hem reageert/antwoordt, of door er nog *RT* voor te plaatsen aangeven dat je een tweet van hem overneemt (zie Retweet).
Following	Een tweeter die jij volgt, en waarvan je de tweets in jouw home timeline ziet.
Followers	Tweeters die jou volgen, en dus jouw tweets in hun home timeline zien.
Retweet	Het kopiëren en zelf tweeten van een tweet van iemand anders, met vermelding van de naam van die tweeter. Doel is om een interessante tweet ook met de eigen volgers te delen. Bijvoorbeeld als @stephenfry heeft gezegd: *Shocking so much of the web can't be used by disabled people. We can all help #fixtheweb - here's how: http://bit.ly/fxweb* dan kan je dat retweeten door zelf te tweeten: *RT @stephenfry: Shocking so much of the web can't be used by disabled people. We can all help #fixtheweb - here's how: http://bit.ly/fxweb*

3 Onderzoeksvraag

Twitter is de afgelopen paar jaar erg populair geworden, en is vooral voor recente gebeurtenissen een snelle bron van informatie .

In dit onderzoek zullen via bestaande information retrieval technieken ‘tweets’ (twitterberichten) beschikbaar gemaakt voor doorzoeking, op zo’n manier dat bepaald kan worden wie over een ingegeven onderwerp op Twitter autoriteit is. De onderzoeksvraag is dan ook:

> *Hoe kunnen we structurele patronen halen uit de onderwerpen waar tweeters over schrijven, op zo’n manier dat bepaald kan worden wie autoriteiten zijn op een bepaald gebied?*

Naast een theoretisch voorstel zal een *proof of concept* geïmplementeerd worden, om de levensvatbaarheid van het antwoord te laten zien.

4 Literatuur

Naast specifieke literatuur, waar op verschillende plekken in dit onderzoek naar zal worden verwezen, is er een aantal artikelen interessant gebleken om een goede denkrichting te verkrijgen. Deze worden hier geïntroduceerd.

4.1 Twitter-specifiek

(Cheong & Lee, 2009) hebben onderzocht wat de relatie is tussen 'trending' topics op Twitter (een lijstje met onderwerpen, genoemd op de hoofdpagina van twitter.com, die volgens Twitter op dit moment 'hot' zijn) en tweets die op dat onderwerp matchten. Omdat ze een groot aantal technieken gebruiken die hier ook gebruikt worden, zoals zoeken op Twitter, het verwerken van een groot aantal tweets en het doorzoeken daarvan, kan hun artikel een aantal handige tips bevatten die bruikbaar kunnen zijn in dit onderzoek.

(Krishnamurthy, Gill, & Arlitt, 2008) geven een overzicht van drie verschillende manieren om datasets van tweets te verzamelen, en welke voor- en nadelen deze methoden hebben.

4.2 Autoriteit

In (Borodiny, Robertsz, Rosenthalx, & Tsaparas, 2002) wordt een viertal algoritmen, en een aantal varianten daarop, vergeleken, om autoriteit op het web te bepalen. Er wordt ook een aantal formele methoden bepaald om deze algoritmen te evalueren en vergelijken. Er worden daarvoor verschillende maten gedefinieerd, zoals:

- monotocity: als voor verschillende nodes j en k in een graaf G geldt dat elke node die naar j verwijst ook naar k verwijst, dat dan de j een gelijk of groter gewicht heeft dan k
- similarity: in hoeverre 2 algoritmen overeenkomstige resultaten opleveren
- stability: in hoeverre worden de gewichten die aan de nodes worden gehangen beïnvloed door kleine veranderingen in de graph
- locality: in hoeverre worden de gewichten van nodes beïnvloed door veranderingen in de graph die niet 'in de buurt' zitten

Verder introduceert het de termen 'focused' en 'balanced': een 'focused' algoritme blijkt beïnvloedbaar door kleine groepjes nodes die naar elkaar verwijzen: deze ranken relatief hoog. Dit terwijl een 'balanced' algoritme hier minder door beïnvloed wordt en ook (geschikte) resultaten buiten deze groepjes nodes hoog laat scoren.
Helaas worden er weinig echte conclusies getrokken in het artikel: het laat nog veel werk open om de algoritmen uit te drukken in hun eigen framework.

(Kleinberg, 1999) geeft een goed algemeen inzicht in hoe autoriteit, en andere structuren die ontstaan door het bestaan van links, 'werkt' op het web.

(Manning, Raghavan, & Schütze, 2008) geeft een handig inzicht in verschillende Information Retrieval technieken, waaronder een hoofdstuk over Link Analysis, waar onder andere PageRank en Hubs and Authorities worden uitgelegd.

5 Data verzamelen

5.1 Mogelijkheden

Voor het onderzoek zou het het mooist zijn om over een langere periode alle tweets te hebben en kunnen doorzoeken. Het moet een periode zijn waarin alle interessante mensen de tijd moeten hebben gehad om iets te zeggen, zodat het op te zetten algoritme voldoende data heeft om met goede resultaten te komen.

Je kunt niet je zomaar een dataset met alle tweets over een bepaalde periode krijgen. Twitter biedt een aantal mogelijkheden om aan (publieke) tweets te komen. Hun api's worden beschreven op (Twitter, Inc.):

- sampling:
    - "spritzer" je krijgt dan een live feed met ongeveer 1% van alle publieke tweets. Deze feed is beschikbaar voor iedereen.
    - "garden hose" je krijgt dan ongeveer 10% van alle publieke tweets. Hiervoor moet je toestemming met een goede reden vragen.
    - "firehose" je krijgt dan alle publieke tweets. Ook hiervoor moet je toestemming vragen. Voor zover dit duidelijk werd is dit alleen beschikbaar voor groot academisch onderzoek.

Zowel "spritzer" als "garden hose" geven – voor dit onderzoek – relatief weinig tweets: de kans dat juist de belangrijke tweets van de persoon waar je naar op zoek bent mist is erg groot: om een beeld van iemand te krijgen is het waarschijnlijk erg onhandig als je maar 1 of 10% van zijn tweets hebt. Uiteraard heb je, als het echt om een belangrijk en bekend persoon gaat, wel genoeg verwijzingen naar hem, en retweets van tweets van hem. Echter: omdat o.a. juist uit dit onderzoek moet komen of het interessant is om te zien hoever je komt met alleen de tweets zelf, of dat je ook op creatieve manieren verwijzingen van andere mensen moet gebruiken, is dit niet voldoende.

Firehose klinkt daarom als een oplossing: je krijgt alle (publieke) tweets. Dit geeft echter meteen wel een praktisch probleem: in september 2010 zei Twitter zelf dat er meer dan 90 miljoen tweets per dag worden gemaakt, waarvan 90% publiek beschikbaar (Siegler, 2010). Het gaat dus in de genoemde meetperiode van een maand om ruim twee miljard tweets, en alleen al de tekst van de tweet (140 tekens) betekent dan ongeveer 300 gigabyte aan data. Dit is veel meer dan in het werkgeheugen van een gemiddelde pc past, en is dus niet eenvoudig te verwerken en doorzoeken. Verder komt er bij een tweet nog allerlei metadata, waardoor de hoeveel data van een tweet gemiddeld 2,8 kilobyte blijkt te zijn, dus in totaal gaat het dan om ruim 6 terabyte aan data. Een oplossing zou zijn om de data goed te indexeren, en alleen de benodigde informatie op te slaan, maar dan nog zou het niet op een gemiddelde harde schijf passen.

De conclusie is dus simpelweg dat geen van de 3 feeds geschikt is voor het onderzoek. Een andere optie zou zijn om niet al van tevoren tweets te verzamelen en indexeren, maar op het moment dat het nodig is te zoeken naar tweets met het ingegeven onderwerp zoeken. Hiervoor heeft Twitter ook

een api, de Search API. Deze heeft echter 2 belangrijke beperkingen: er worden maximaal 1500 tweets geretourneerd, maar vooral dat alleen de laatste 7 dagen aan tweets worden gebruikt en geretourneerd lijkt een erg grote beperking: als de persoon die je zoekt toevallig een weekje (Twitter)vakantie heeft dan vind je deze tweeter niet.

De laatste optie die Twitter biedt is de “filter”-api: hiermee kan je een live feed van een nader te specificeren subset van tweets krijgen. De subset wordt gedefinieerd door één of meer woorden, filterwoorden genoemd, en bepaalt van elke nieuwe publieke tweet of deze ‘matcht’ op het ingegeven filter. Als één of meer van de filterwoorden voorkomt in de tweet, of de twitternaam van de tweeter komt overeen met één van de woorden in je filter, dan wordt die tweet doorgegeven.

Op deze manier is het dus mogelijk een subset van alle publieke tweets krijgen, en als er dan op een slimme manier een aantal onderwerpen ingegeven wordt, kunnen daar een tijd lang tweets over verzameld worden en kunnen die tweets als testset gebruikt worden. De subset is niet ideaal: zo kan het zijn dat je van tweeters die je uiteindelijk vindt eigenlijk alle tweets nodig hebt en misschien zelfs alle tweets die naar deze tweeter verwijzen. Het filter levert echter alleen die tweets die op het filter ‘matchten’. Later zal blijken dat hier grotendeels omheen te werken valt.

5.2 Tweet

Een tweet zoals die uit het systeem van Twitter komt bevat meer informatie dan alleen de tekst van de tweet. Het bevat onder andere:

- de inhoud van de tweet, dus de daadwerkelijke tekst
- wanneer de tweet gemaakt is
- welke tweeter de tweet toegevoegd heeft:
    - zijn twitternaam
    - echte naam (zoals zelf ingevuld)
    - locatie (zoals zelf ingevuld)
    - beschrijving (hoe hij zichzelf omschrijft)
    - hoeveel volgers hij heeft
    - hoeveel tweeters hij zelf volgt
    - of de identiteit van de gebruiker is vastgesteld (vaak bij beroemdheden, bedrijven en instanties)
    - wanneer de account is aangemaakt
    - time zone van de tweeter (zoals zelf ingevuld)
    - taal van de tweeter (zoals zelf ingevuld)

    - hoeveel tweets hij al geschreven heeft

- als de tweet een retweet is, dan is dezelfde informatie over de tweeter die geretweet is beschikbaar als hierboven gedefinieerd over de tweeter
- indien beschikbaar: de locatie van de tweeter op het moment van plaatsen
- een lijst van hashtags die in de tekst voorkomt
- een lijst van usermentions die in de tekst voorkomt
- een lijst van URLs die in de tekst voorkomen
- het globale id (oplopend nummer) van de tweet

Voor een compleet voorbeeld van een tweet, zie *13 Bijlage 2: een tweet*.

5.3 Filterwoorden

De data die met de filter-api van Twitter is verzameld, is in eerste instantie verzameld door de onderstaande filterwoorden in te geven. Deze woorden zijn gekozen omdat er op dat moment erg veel getwitterd werd naar aanleiding de politieke ontwikkelingen in onder andere Tunesië en Egypte.

egypt
tunesia
mubarak
jan25
tahrir
qatar
cairo
palastine
freeegypt
tunisia
tunesia

Hier zijn de onderstaande filterwoorden aan toegevoegd. Het is een subset van de zoektermen die gebruikt zijn in (Haveliwala, 2003), dit artikel heeft deze set verzameld uit weer andere artikelen. Alleen de zoektermen bestaande uit één woord zijn gebruikt, omdat de filter-api van Twitter niet overweg kan met 2 termen: het werkt alleen als OF en niet als EN. Uiteraard is hieromheen te werken door op allebei de termen te matchen en dan alleen tweets door te laten die beide termen bevatten, maar dit zou relatief veel programmeerwerk kosten: er zijn dan veel randgevallen, bijvoorbeeld als een van de andere filterwoorden ook voorkomt in de tweet.

alcoholism
architecture
bicycling
blues
buddhism
cheese

cruises
gardening
hiv
laserey
lipari
lyme
shakespeare
sushi
telecommuting
volcano
zener

Tijdens de campagnetijd voor de provinciale verkiezingen 2011 zijn de volgende filterwoorden toegevoegd:

d66
pvda
cda
groenlinks
sgp
sp
vvd

Dit alles leverde tussen 9 februari 2011 en 9 maart 2011 ongeveer 22,3 GB aan tweets op. Deze zijn gebruikt als testset.

6 Voorstel systeem

6.1 Hypothesen

Een snelle brainstorm leverde al een heel rijtje mogelijke gedachten op waarmee een indicatie van autoriteit zou kunnen worden bepaald:

- maakt 'meer twitteren over een bepaald onderwerp' je een autoriteit?
    - en zijn normale tweets dan een sterkere aanwijzing dan retweets?
- maakt meer geretweet worden je autoriteit?
    - of is gementiond worden ook al goed genoeg?
        - moet het gezochte onderwerp dan in deze tweets genoemd worden?
    - en moet je retweets van mensen met meer volgers extra sterk laten meetellen?
- helpt het als je meer volgers hebt?
    - en moet dan ook gekeken worden naar het aantal mensen dat de tweeter zelf volgt? Dit omdat in de praktijk blijkt dat mensen volgen vaak ervoor zorgt dat die mensen jou 'terugvolgen'.
- het veel verwijzen naar en/of retweeten van andere interessante tweeters zou een tweeter ook interessant kunnen maken: via een dergelijke tweeter krijg je ook veel informatie. In dat geval is het algoritme van (Kleinberg, 1999) misschien bruikbaar: een dergelijke tweeter is zelf geen autoriteit, maar wel een hub. Dit valt echter waarschijnlijk buiten de scope van dit onderzoek.
- omdat volgens (Cheong & Lee, 2009) ongeveer 96% van alle usermentions slechts is om persoonlijke berichten naar elkaar te schrijven, is PageRank misschien helemaal ongeschikt. Onduidelijk is of hierbij retweets wel of niet zijn meegeteld.

In principe is het mogelijk om al deze hypothesen te testen, maar in de praktijk kost dit alles veel tijd. Ook is het goed mogelijk dat blijkt dat er een combinatie van verschillende factoren nodig is voor 'de perfecte oplossing', waardoor er veel geëxperimenteerd moet worden: hoe sterk moet elke factor meewegen?

Er is daarom gekozen om slechts een klein aantal factoren mee te nemen. Het gaat hierbij om de hypothese dat als je meer over een onderwerp twittert, je met grotere kans een autoriteit op dat gebied bent, en dat als je meer gementiond wordt dat je dan belangrijker bent. In de praktijk komt dit neer op het uitvoeren van vrij standaard IR-technieken, namelijk een standaard tekst-index waarin gerankt wordt met behulp van het Vector Space Model voor de eerste factor, en PageRank voor de tweede factor.

6.2 Tweet of timeline als document?

Een keuze die gemaakt moet worden is of je een enkele tweet als 'document' ziet, of alle tweets van een tweeter bij elkaar. Elke tweet apart zien geeft mogelijk een aantal problemen. Ten eerste levert het een zeer groot aantal documenten op, mogelijk veel meer dan werkbaar met standaard technieken. Ten tweede is het maar de vraag of standaard IR-technieken, zoals TF/IDF, goed werken met erg korte documenten zoals een tweet: een analyse van de verzamelde data levert op dat de gemiddelde tweet uit zo'n 16 woorden bestaat. Zoals (Metzler, Dumais, & Meek, 2007) al schreven: de meeste similarity measures zijn ontwikkeld om als input een korte query te krijgen, en te werken op relatief grote documenten, en werken duidelijk minder goed op korte teksten. Ten derde moet het resultaat van het op te zetten systeem een lijst van tweeters worden, niet een lijst van tweets: het zou dus lastig zijn om een tweet als atomair item te nemen.

Een logische keuze lijkt daarom om alle tweets van één tweeter samen te voegen tot een document. Dit geeft echter wel een probleem qua data verzamelen: het is niet mogelijk om een verzameling van alle tweets te krijgen, alleen data van tweets die 'matchten' op ingegeven filterwoorden kon worden verzameld. Helaas zijn dus niet alle tweets van de gevonden tweeters voor het onderzoek beschikbaar, maar alleen de tweets waarin hij woorden gebruikte die in het filter zaten. Het bepalen van de Term Frequency is daarom lastig: het is niet bekend wat de tweeter nog meer getweet heeft.

Gelukkig kan er om deze beperking heen gewerkt worden: er is wel beschikbaar hoeveel tweets de tweeter geschreven heeft in de periode dat we data verzameld hebben: bij elke verzamelde tweet kregen we ook het aantal tweets dat de tweeter in totaal geschreven heeft. Dit aantal kan gebruikt worden om te bepalen hoeveel tweets de tweeter tussen de eerstgevonden en de laatstgevonden tweet getweet heeft, en zo kan bepaald worden hoeveel tweets het filter 'gemist' heeft. Door dit aantal te vermenigvuldigen met het gemiddeld aantal woorden per tweet, wordt een goede maat voor de totale documentlengte bepaald.

6.3 Voorstel systeem

Het plan is om de alle tweets per tweeter te indexeren, en tegelijkertijd de usermentions te gebruiken om via het PageRank-algoritme per tweeter te bepalen wat zijn algemene autoriteit is.

6.3.1 Indexering

De verzamelde data bestaat uit een enorme lijst van tweets, op volgorde van binnenkomst. Elke tweet bevat de informatie zoals beschreven in *5.2 Tweet*. Een klein voorbeeld van een aantal tweets, met de gegevens die gebruikt worden om te indexeren:

twitternaam	aantal geschreven tweets	datum/tijd	tekst	usermentions
markreid	2265	Fri Dec 17 13:58:29 +0000 2010	Having issues with Google Apps mail and reverse DNS. Anyone know anything about that stuff?	[]
nevali	64361	Fri Dec 17 13:59:13 +0000 2010	@markreid I use Google Apps and know a shedload about DNS	[markreid]
NiersLucia	13603	Sat Dec 18 08:41:31 +0000 2010	Amazon stellt Cloud DNS Dienst Amazon Route 53 vor	[]
nevali	64366	Fri Dec 17 21:33:45	RT @MuscleNerd: @saurik one time after explaining to TSA why I had so many iPhones ("to hack them"), they asked if I	[MuscleNerd, saurik]

		+0000 2010	was with wikileaks!	
mbjunior	461	Mon Dec 20 19:33:48 +0000 2010	RT @demuxer: #Cablegate 06SAOPAULO675 shows Microsoft trying to Take Over Brazilian Schools http://rod.gs/v6e via @schestowitz #wikileaks	[demuxer, schestowitz]
mbjunior	465	Mon Dec 20 20:07:35 +0000 2010	Studying the IETF's DNSSEC specifications. #security #infosec #networking #dns #dnssec #authentication	[]

Omdat het praktischer bleek om niet per tweet te indexeren, maar alle tweets van één tweeter samen te voegen, worden de tweets eerst gesorteerd op twitternaam, en vervolgens samengevoegd:

twitternaam	aantal gevonden tweets	aantal geschreven tweets (min, max)	datum/tijd oudste	datum/tijd nieuwste	teksten	usermentions
markreid	1	2265, 2265	Fri Dec 17 13:58:29 +0000 2010	Fri Dec 17 13:58:29 +0000 2010	Having issues with Google Apps mail and reverse DNS. Anyone know anything about that stuff?	[]
mbjunior	2	461, 465	Mon Dec 20 19:33:48 +0000 2010	Mon Dec 20 20:07:35 +0000 2010	RT @demuxer: #Cablegate 06SAOPAULO675 shows Microsoft trying to Take Over Brazilian Schools http://rod.gs/v6e via @schestowitz #wikileaks Studying the IETF's DNSSEC specifications. #security #infosec #networking #dns #dnssec #authentication	[demuxer(1), schestowitz(1)]
nevali	2	64361, 64366	Fri Dec 17 13:59:13 +0000 2010	Fri Dec 17 21:33:45 +0000 2010	@markreid I use Google Apps and know a shedload about DNS RT @MuscleNerd: @saurik one time after explaining to TSA why I had so many iPhones ("to hack them"), they asked if I was with wikileaks!	[markreid(1), MuscleNerd(1), saurik(1)]
NiersLucia	1	13603, 13603	Sat Dec 18 08:41:31 +0000 2010	Sat Dec 18 08:41:31 +0000 2010	Amazon stellt Cloud DNS Dienst Amazon Route 53 vor	[]

Vervolgens zijn deze samengevoegde tweets geïndexeerd in Apache Lucene. Deze software maakt een doorzoekbare index aan de hand van standaardtechnieken als TF/IDF.

Zoals al beschreven, zijn alleen tweets verzameld zijn die daadwerkelijk matchten op één van de ingegeven termen, en misten de overige tweets die een tweeter heeft geschreven. Mensen die maar weinig tweets hebben geschreven kwamen daardoor erg hoog in de rankings: hun TF/IDF is logischerwijs erg hoog, omdat ze maar een kort 'document' hebben. Dit kon worden opgelost door te bekijken hoeveel tweets de tweeter tijdens de meetperiode heeft geschreven. Door dit af te trekken van het aantal tweets dat wel gevonden was kon worden achterhaald hoeveel de tweeter over andere onderwerpen geschreven heeft. Uit eigen onderzoek bleek dat een tweet gemiddeld uit zo'n 16 woorden bestaat, zo kon IDF dus worden 'gerepareerd' door voor het indexeren een aantal loze woorden toe te voegen dat overeenkomt met dat aantal gemiste tweets.

Een uitzondering was er voor tweeters die tijdens de meetperiode maar 1 tweet over een van de ingegeven onderwerpen heeft geschreven, of meerdere maar dan slechts verdeeld over 1 of 2 dagen. Het bepalen van hoeveel tweets er in de tussentijd geschreven zijn kon dan niet bepaald worden, of niet op een betrouwbare manier. Besloten is om deze tweeters buiten de zoekindex te

houden: met zo weinig tweets over de onderwerpen is de kans dat het om een autoriteit op een van de ingegeven gebieden is vrij klein. Overigens zijn de usermentions van deze tweeters wel meegenomen in het PageRank-algoritme.

Een andere optimalisatie is het niet in de tekst-index opnemen van tweeters die zelf nooit (door anderen) gementiond worden. Deze mensen zullen waarschijnlijk geen autoriteit zijn op Twitter, bovendien zullen ze een zodanig lage PageRank-score krijgen dat ze uiteindelijk toch niet hoog gaan scoren. Overigens moet worden opgemerkt dat het best kan zijn dat deze tweeters mogelijk wel door mensen gementiond worden, maar dat dit niet in de dataset is gekomen, omdat er in die betreffende tweets geen van de ingegeven zoektermen zijn gebruikt. Dit zou opgemerkt kunnen worden als een beperking van de dataset, aan de andere kant is het misschien ook mogelijk een zegen: er zijn alleen verwijzingen beschikbaar die over een van de ingegeven onderwerpen gaan. Dit zijn dus sterke indicaties van het feit dat de gementionde tweeters daadwerkelijk autoriteit zijn op het genoemde gebied. Wat dat betreft gaat het hier al bijna vanzelf om een Kleinberg/Hubs and Authorities-algoritme, en is een zuiver PageRank-algoritme niet mogelijk: PageRank geeft een waarde aan een document op een onderwerponafhankelijke wijze.

6.3.2 PageRank

Per tweeter is bepaald welke andere tweeters hij heeft gementiond. Hierop is met Java Universal Network/Graph Framework (The JUNG Framework Development Team) het standaard PageRank-algoritme losgelaten, zoals beschreven in (Page, Brin, Motwani, & Winograd, 1998), met een alpha van 0.2. Er is niet geëxperimenteerd met de alpha-waarde: deze experimententen zouden een tijdsinspanning kosten waarvan te verwachten was dat deze niet op zou wegen tegen de opbrengsten.

De uiteindelijke ‘ruwe’ PageRank-waarde wordt gecorrigeerd tot een normaliseerbare waarde met behulp van een logfunctie. (Page, Brin, Motwani, & Winograd, 1998) zelf hebben het over ‘een log’, het is onduidelijk welk grondtal gebruikt wordt. Een zoekactie op internet levert een aantal discussies op, maar geen duidelijke conclusie. “Ergens tussen 5 en 10” wordt het meest genoemd, maar echt een betrouwbare bron wordt niet aangegeven. Mogelijk houdt Google dit soort dingen ‘geheim van de kok’, of variëren ze dit afhankelijk van bijvoorbeeld de grootte van het web. In dit onderzoek is de keuze voor het grondtal arbitrair op 8 vastgesteld, ook hiermee is uit tijdsinspannsingsoverwegingen niet geëxperimenteerd.

6.3.3 Queryen

Aan de hand van de ingegeven query worden eerst potentieel interessante tweeters verzameld die zelf relatief veel over dat onderwerp schreven door deze query te voeren aan Apache Lucene. Hieruit kwam een ranking van een aantal tweeters. Van elk van deze tweeters werd vervolgens de PageRank-waarde opgehaald.

Het resultaat hiervan zijn twee rankings van dezelfde tweeters: één op volgorde van score van Lucene, en een op volgorde van PageRank.

Het samenvoegen van deze rankings is niet triviaal. Zoals (Page, Brin, Motwani, & Winograd, 1998) in hun beschrijving van PageRank al schreven: “*Rank merging is known to be a very difficult problem,*

and we need to spend considerable additional effort before we will be able to do a reasonable evaluation of these types of queries."

De gekozen oplossing is er een zoals besproken in (Craswell, Hawking, & Thistlewaite, 1999): normaliseer alle waarden naar een getal tussen 0 en 1 en tel vervolgens de PageRank-waarde en de documentscore uit Apache Lucene bij elkaar op, met een weging voor elk van de twee. Deze weging kan worden ingesteld. Praktisch gezien gaat het berekenen van een score bestaande uit een aantal componenten dus als volgt:

$$score = \sum_{scorecomponent} \frac{waarde - minwaarde}{maxwaarde - minwaarde} weging$$

In dit geval, met een documentscore uit Apache Lucene en een PageRank-waarde gaat het dus om:

$$score = \frac{documentscore - mindocumentscore}{maxdocumentscore - mindocumentscore} weging_{documentscore}$$
$$+ \frac{pagerankwaarde - minpagerankwaarde}{maxpagerankwaarde - minpagerankwaarde} weging_{pagerank}$$

Hierbij worden de wegingsfactoren zodanig gekozen dat geldt: $weging_{documentscore} + weging_{pagerank} = 1$. Hierdoor wordt de uiteindelijke score ook een getal tussen 0 en 1.

6.4 Technische implementatie systeem

Voor een beschrijving van de technische implementatie van deze methode, zie *1 Bijlage 1: technische implementatie systeem*.

7 Voorstel systeem (2)

7.1 Aanleiding

De eerste resultaten bleken niet veelbelovend. Precieze getallen staan in het hoofdstuk *9 Resultaten*, maar het komt hierop neer: zonder gebruik van PageRank, dus alleen met de resultaten van de tekst-index, was de score matig. Het toevoegen van PageRank maakte het – zodra de weging daarvan meer werd dan 10% - alleen maar slechter. Met een weging van 10% was de score marginaal hoger dan met alleen de tekst-index, maar dit mag geen naam hebben.

Bij nadere inspectie bleek een aantal dingen op te vallen, maar de belangrijkste is dat de aanname dat het PageRank-algoritme door de manier van data verzamelen bijna vanzelf als een Hubs and Authorities-algoritme zou werken niet altijd waar blijkt te zijn.

De aanname was dat omdat er alleen tweets zijn verzameld op bepaalde keywords, een 'mention' naar iemand betekent dat deze tweeter een autoriteit is op het gebied van een van de keywords. De manier van queryen, namelijk eerst een standaard tf/idf-zoek-algoritme, en dan pas kijken welke PageRank-waarden daarbij horen, zou ervoor zorgen dat alleen mensen die iets over dat keyword te zeggen hebben worden meegenomen, dus ook alleen van die mensen de bijbehorende PageRank-waarde wordt gebruikt. In de praktijk blijkt echter dat de enorme hoeveelheid tweets die gematcht zijn door de keywords rond de recente Midden-Oostenproblematiek voor een aantal tweeters heeft gezorgd met vrij hoge PageRank. Als die vervolgens ook maar 1 of 2 keer iets zeggen over bijvoorbeeld 'bicycling' dan komen ze ineens erg vrij hoog in de ranking. Tussen de individuele resultaten zitten hierdoor een aantal tweeters die een hoge PageRank-waarde en een erg lage Lucene-waarde hebben.

Een voorbeeld: bij de query bicycling, met een weging van PageRank van 40%, staan hieronder de eerste 20 resultaten. Aangestreept zijn tweeters die waarschijnlijk weinig met bicycling te maken hebben, en veel meer met de Midden-Oostenproblematiek. Opvallend is dan ook de hoge PageRank-waarde, en de lage Lucene-score.

PageRank-waarde	Lucene-score	Totale score	Twitternaam	Waarom geen goed resultaat voor query 'bicycling'
0,0000	1,0000	0,6000	Undriving	
0,0001	0,9899	0,5940	bikeclimbr	
0,0001	0,8000	0,4801	BikeMN	
0,0031	0,7289	0,4385	qbike	
1,0000	0,0625	0,4375	beleidy	'Bio': "(…) Egypt."
0,3344	0,4961	0,4314	gnudarwin	
0,0126	0,7071	0,4293	Streetfilms	
0,0001	0,7071	0,4243	20BY2020	
0,0000	0,6708	0,4025	BikeFairfax	
0,8848	0,0625	0,3914	washingtonpost	
0,2112	0,5000	0,3845	miabirk	
0,0001	0,6062	0,3638	readride	
0,7177	0,0625	0,3246	pascaluccelli	Bio: "Muslim."
0,7046	0,0625	0,3193	shoshido	Veel tweets over Midden-Oostenproblematiek.

0,0006	0,5196	0,3120	SavBikeCampaign	
0,0004	0,5196	0,3119	lipeolipe	
0,6744	0,0625	0,3072	shunradan	Veel tweets over Midden-Oostenproblematiek.
0,2045	0,3674	0,3023	ellyblue	
0,0003	0,5000	0,3001	SustainableComm	
0,6509	0,0625	0,2979	advorec_II	Bio: "(…) Furiously active in spreading news about anything green, right and human. (…)".

Waar PageRank op het web vrij goed werkt, werkt het blijkbaar niet goed op Twitter. Een verklaring zou kunnen zijn dat daar waar op het web de meeste webpagina's maar over 1 onderwerp gaan, tweeters vaak over meerdere onderwerpen iets schrijven (of retweeten). De keuze om alle tweets van 1 tweeter samen te voegen zou dus wel eens niet goed kunnen samenwerken met deze manier van queryen.

Los daarvan moet worden opgemerkt dat de manier waarop Listorious de zoekresultaten oplevert ook in de verste verten niet vergelijkbaar is met de manier waarop een geautomatiseerd systeem dit doet, zie hiervoor het hoofdstuk *8 Meetmethode*.

7.2 Nieuw voorstel systeem

Na bestudering van de verschillende literatuur leek de methode Hubs and Authorities, hierna te noemen: H&A, uit (Manning, Raghavan, & Schütze, 2008) een mogelijk geschikte keuze te zijn om bovenstaande problemen op te lossen. Hiermee wordt, aan de hand van een query, eerst een subset van 'het web' genomen door middel van een standaard tekst-index. Dit wordt de root set genoemd.

Vervolgens wordt deze root set uitgebreid tot een base set door daaraan alle pagina's toe te voegen die linken naar pagina's uit de root set, en pagina's die worden gelinkt door pagina's uit de root set. Vervolgens wordt over de base set het HITS-algoritme (zie (Kleinberg, 1999)) uitgevoerd. Hieruit komt o.a. een ranking op autoriteit, binnen deze base set. Deze ranking wordt samengevoegd met de ranking die uit Lucene komt, op dezelfde manier als bij de methode met PageRank.

7.3 Technische implementatie systeem

Voor een beschrijving van de technische implementatie van deze methode, zie *12 Bijlage 1: technische implementatie systeem*.

8 Meetmethode

De bedoeling van het te bouwen zoeksysteem is, gegeven een query, een lijst van tweeters terug te geven die autoriteit op dat gebied zijn, gesorteerd op mate van autoriteit. Zo'n lijst wordt ook wel een ranking genoemd.

Om te bepalen of de resultaten van dit systeem 'goed' zijn, moet er een maat zijn om de kwaliteit van de resultaten te geven. Een veelgebruikte – en logische – methode is om zo'n ranking te vergelijken met een ranking waarvan bekend is dat deze goed is.

8.1 Vergelijkingsmateriaal

Na de start van dit onderzoek heeft Twitter zelf een 'who to follow'-functionaliteit toegevoegd aan zijn website, die qua idee eigenlijk op dezelfde manier lijkt te werken: als je in Twitter zoekt naar tweets krijg je naast de resultaten ook een lijstje met 'People results'. Een kleine steekproef met een aantal termen geeft aan dat deze functionaliteit niet echt goed werkt: zo is de eerste hit op de query *buddhism* de tweeter daily_buddhism, die weliswaar relatief veel volgers heeft (6279), en het woord buddhism zowel in z'n naam als in z'n 'description' heeft staan, maar hij heeft al sinds 2007 geen tweets meer heeft geschreven. Verder is het ook niet duidelijk hoe Twitter de lijst samenstelt: het gaat hier om een eigen algoritme. Overigens omschrijft (Twitter, Inc.) het ook niet als functionaliteit die autoriteit weergeeft, maar als een zoekfunctie op naam: "*To Find People by Name - The easiest way to find people is by typing their name into the search box at the top of your Twitter homepage.*".

Andere geschikte zoekmogelijkheden geeft Twitter niet. Een andere optie zou Google zijn, maar die lijkt niet echt de inhoud van tweets mee te nemen, maar vooral de namen van gebruikers, en de namen van *lists*. Lists zijn lijsten die tweeters kunnen samenstellen, om degenen die ze volgen in groepen op te delen. Er blijkt ook een meta-website te zijn, listorious.com, die op handmatige wijze goede lists van tweeters verzamelt en categoriseert. Tweeters die lijsten hebben gemaakt kunnen deze zelf aanmelden bij Listorious. De zoekfunctie, die ze zelf "*Find experts on Twitter*" noemen, lijkt uitstekend te werken: de zoekresultaten hiervan lijken veelal actieve tweeteraccounts te zijn die daadwerkelijk veel met de onderwerpen te maken hebben. Probleem is dat tweeters zelf hun eigen lijsten moeten toevoegen, waardoor veel interessante lijsten en tweeters mogelijk worden gemist. Ook hoeft het niet zo te zijn dat tweeters überhaupt autoriteiten in lijsten stoppen. Door dit alles is het goed mogelijk dat goede autoriteiten op een bepaald gebied volledig door Listorious worden gemist.

In dit onderzoek is er – ondanks de mogelijke beperkingen – voor gekozen om te vergelijken met de resultaten van listorious.com. Hieruit zijn de tweeters verwijderd die tijdens de meetperiode niet getweet hebben verwijderd omdat het geen eerlijk vergelijkingsmateriaal zou zijn voor het gebouwde systeem. Het is de bedoeling een zoeksysteem te maken die werkt op alle tweets; als ware het dat het systeem de beschikking over alle tweets die ooit zijn gemaakt, of in ieder geval die over een langere periode. Dat het systeem geen tweets heeft van voor de meetperiode, is het resultaat van het onderzoek, niet een beperking van het beoogde systeem.

8.2 Similarity measures

In (Haveliwala, 2003) wordt gebruik gemaakt van een tweetal similarity measures , daar OSim en KSim genoemd. OSim is een maat voor overlap: voor twee rankings A en B met beide lengte k wordt bepaald welk deel van de resultaten uit ranking A ook voorkomen in ranking B. Het wordt als volgt berekend:

$$\mathrm{OSim}(A, B) = \frac{|A \cap B|}{k}$$

Hierbij is *k* het aantal resultaten dat wordt vergeleken (voor beide rankings evenveel), en A en B zijn de sets van resultaten zelf. OSim geeft dus alleen een maat voor overlap, maar neemt niet de volgorde van de rankings mee.

KSim daarentegen is een maat voor overeenkomst in volgorde tussen 2 rankings, en is gebaseerd op Kendalls τ afstandsmaat. Echter, omdat OSim een overeenkomst geeft, heeft Haveliwala KSim ook als een overeenkomst gedefinieerd, in plaats van een afstandsmaat, zodat een getal dichter bij 1 een hogere overeenkomst betekent.

Het wordt als volgt berekend: neem twee partieel geordende lijsten van URLs (in het geval van dit onderzoek: tweeters), τ_1 en τ_2, elk van lengte *k*. Laat U de vereniging zijn tussen deze lijsten. Construeer nu τ'_1 door uit τ_1 de URLs te verwijderen die niet in U zitten, en construeer τ'_2 op analoge wijze. KSim is nu gedefinieerd als:

$$\mathrm{KSim}(\tau_1, \tau_2) = \frac{|(u, v) : \tau'_1, \tau'_2 \text{ zijn het eens over de volgorde van } (u, v), u \neq v|}{|U| \cdot (|U| - 1)}$$

In andere woorden: KSim geeft de kans dat een τ_1 en τ_2 het eens zijn over de relatieve ordening van een willekeurig geselecteerd paar van verschillende URLs uit U×U.

Als er geen paren zijn, dan is KSim niet gedefinieerd.

Bij volledig willekeurige rankings neigt KSim naar 0,5, omdat 'het eens zijn over een volgorde van een willekeurig paar' bij willekeurige rankings als ware een muntje opgooien is: het is wel zo, of niet zo, en beide mogelijkheden hebben evenveel kans. Alles wat (structureel) boven de 0,5 zit is dus een indicatie van dat de ingevoerde rankings qua volgorde overeenkomen.

8.3 Query's en variabelen

De volgende filterwoorden zijn gebruikt als testquery's:

alcoholism
architecture
bicycling
blues
buddhism
cheese
cruises
gardening
hiv

lipari
lyme
shakespeare
sushi
telecommuting
volcano
zener

Deze filterwoorden hebben een maand lang meegedraaid in het verzamelen van tweets, en hebben daarom een goede dekking in de dataset.

Deze query's zijn een aantal keer op het geïmplementeerde systeem losgelaten, met verschillende variabelen:

- met verschillende wegingsfactoren van de PageRank- danwel H&A-waarden, namelijk 0%, 10%, 30%, 50%, 70% en 90%.
  Hiermee kan goed worden bepaald op welke manier deze rankingalgoritmen de resultaten uit Lucene beïnvloeden.
- met verschillende aantallen documenten/tweeters waarmee gerekend wordt, en wordt geretourneerd. Hiermee kan worden bepaald of het nodig is veel documenten/tweeters mee te nemen in de berekening. De volgende combinaties van variabelen is getest:
  - voor PageRank:
    - 'klein': na het ophalen van 150 documenten wordt hiervan de PageRank-waarde opgehaald, de ranking met behulp van de wegingsfactor bepaald, en de eerste 15 resultaten geretourneerd
    - 'groot': na het ophalen van 300 documenten wordt hiervan de PageRank-waarde opgehaald, de ranking met behulp van de wegingsfactor bepaald, en de eerste 50 resultaten geretourneerd
  - voor H&A:
    - 'klein': de beste 15 documenten worden uit Lucene opgehaald, het H&A-algoritme wordt hierover uitgevoerd, de ranking met behulp van de wegingsfactor bepaald, en de eerste 15 resultaten geretourneerd
    - 'groot': de beste 30 documenten worden uit Lucene opgehaald, het H&A-algoritme wordt hierover uitgevoerd, de ranking met behulp van de wegingsfactor bepaald, en de eerste 50 resultaten geretourneerd

Opmerking: bij H&A is een kleiner aantal documenten opgehaald dan bij PageRank: H&A haalt zelf nog documenten/tweeters op die naar de opgehaalde documenten verwijzen, en waarnaar de opgehaalde documenten verwijzen. Hierdoor wordt de graaf vanzelf ruim voldoende groot. Overigens haalt het algoritme in eerste instantie veel meer documenten uit Lucene, maar gebruikt alleen een klein aantal daarvan – maar wel de beste – als 'seed' voor de root set van het H&A-algoritme. Van de overige documenten wordt alleen de documentscore gebruikt, om samen met de autoriteitswaarde van H&A een ranking op te kunnen bouwen.

9 Resultaten

De testquery's zijn uitgevoerd met zowel de eerste versie van het systeem, die een tekst-index en PageRank combineert, als met de tweede versie van het systeem, die een tekst-index en H&A combineert.

Tekst-index & PageRank, 15 resultaten ('klein'), OSim						
Weging PageRank:	0%	10%	30%	50%	70%	90%
alcoholism	0,00	0,00	0,00	0,00	0,00	0,00
architecture	0,07	0,07	0,07	0,07	0,13	0,13
bicycling	0,00	0,00	0,07	0,00	0,00	0,00
blues	0,00	0,00	0,00	0,00	0,00	0,00
buddhism	0,00	0,00	0,00	0,00	0,00	0,00
cheese	0,13	0,13	0,07	0,07	0,07	0,07
cruises	0,00	0,00	0,00	0,00	0,00	0,00
gardening	0,07	0,07	0,07	0,07	0,00	0,00
hiv	0,00	0,00	0,07	0,13	0,13	0,13
lipari	0,00	0,00	0,00	0,00	0,00	0,00
lyme	0,00	0,00	0,00	0,00	0,00	0,00
shakespeare	0,13	0,13	0,13	0,13	0,13	0,13
sushi	0,07	0,07	0,07	0,07	0,07	0,07
telecommuting	0,00	0,00	0,00	0,00	0,00	0,00
volcano	0,00	0,00	0,00	0,00	0,00	0,00
zener	0,00	0,00	0,00	0,00	0,00	0,00
Gemiddelde:	**0,029**	**0,029**	**0,033**	**0,033**	**0,033**	**0,033**

Omdat de OSim-waarden zo laag zijn, zijn de KSim-waarden bijna nietszeggend: als er een overlap van 1 of minder is, is KSim niet gedefinieerd. Deze gegevens zijn dan ook weggelaten.

De tweede versie van het systeem doet het – met 15 resultaten – niet beter:

Tekst-index & H&A, 15 resultaten ('klein'), OSim						
Weging H&A:	0%	10%	30%	50%	70%	90%
alcoholism	0,00	0,00	0,00	0,00	0,00	0,00
architecture	0,07	0,07	0,07	0,07	0,07	0,13
bicycling	0,00	0,00	0,00	0,00	0,00	0,00
blues	0,00	0,00	0,00	0,00	0,00	0,00
buddhism	0,00	0,00	0,00	0,00	0,00	0,00
cheese	0,13	0,13	0,13	0,13	0,00	0,00
cruises	0,00	0,00	0,00	0,00	0,00	0,13
gardening	0,07	0,07	0,07	0,07	0,07	0,07
hiv	0,00	0,00	0,00	0,07	0,13	0,13
lipari	0,00	0,00	0,00	0,00	0,00	0,00
lyme	0,00	0,00	0,00	0,00	0,00	0,00
shakespeare	0,13	0,13	0,13	0,13	0,13	0,00
sushi	0,07	0,07	0,07	0,07	0,00	0,00
telecommuting	0,00	0,00	0,00	0,00	0,00	0,00
volcano	0,00	0,00	0,00	0,00	0,00	0,00
zener	0,00	0,00	0,00	0,00	0,00	0,00
Gemiddelde:	**0,029**	**0,029**	**0,029**	**0,033**	**0,025**	**0,029**

Daarom zijn ook van deze metingen de KSim-waarden weggelaten.

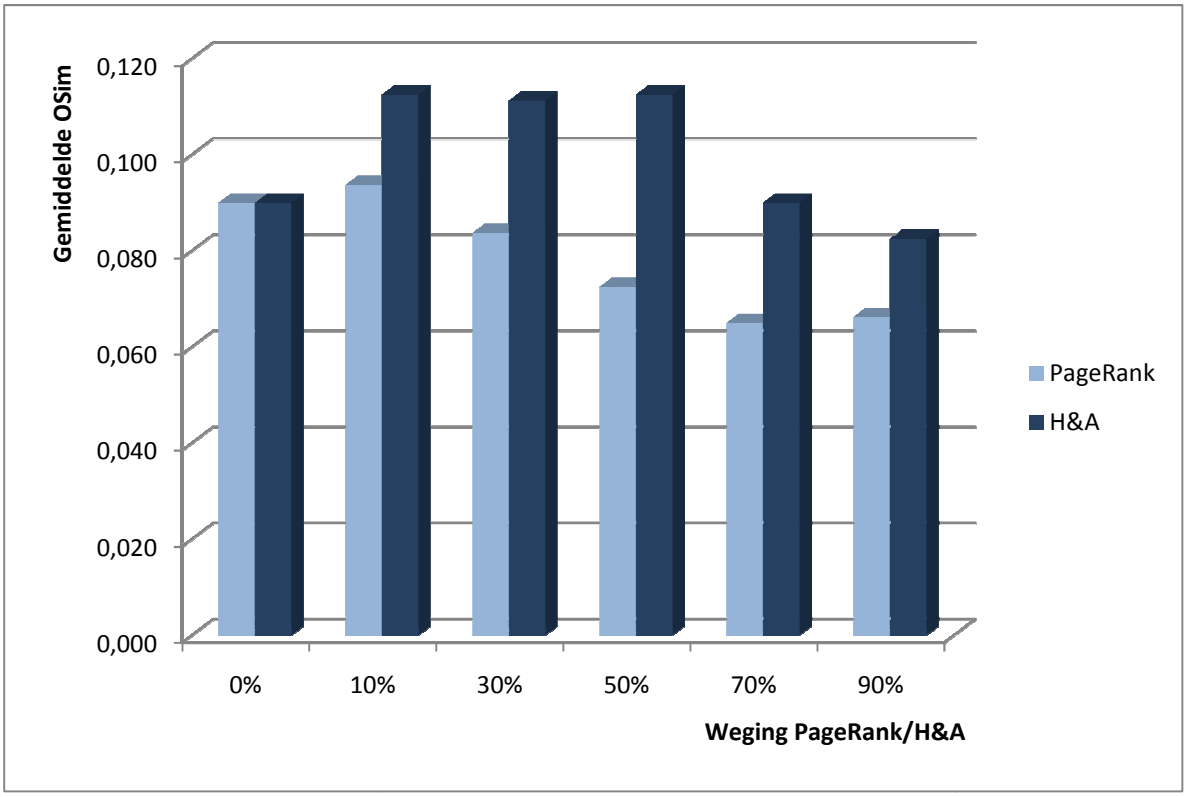
Met 50 resultaten worden de resultaten duidelijk beter:

Tekst-index & PageRank, 50 resultaten ('groot'), OSim & KSim												
Weging PageRank:	0%		10%		30%		50%		70%		90%	
	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim
alcoholism	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
architecture	0,10	0,60	0,100	0,600	0,10	0,600	0,14	0,511	0,18	0,500	0,22	0,513
bicycling	0,12	0,60	0,120	0,533	0,08	0,667	0,04	0,000	0,04	0,000	0,02	-
blues	0,04	0,00	0,040	0,000	0,04	0,000	0,04	0,333	0,04	0,333	0,04	0,333
buddhism	0,08	0,83	0,100	0,300	0,02	-	0,02	1,000	0,02	-	0,02	1,000
cheese	0,14	0,55	0,160	0,470	0,18	0,487	0,22	0,560	0,22	0,506	0,28	0,542
cruises	0,24	0,36	0,220	0,409	0,18	0,422	0,12	0,000	0,04	0,476	0,02	-
gardening	0,08	0,75	0,080	0,714	0,08	0,714	0,08	0,464	0,08	0,607	0,08	0,333
hiv	0,14	0,29	0,120	0,352	0,14	0,318	0,10	0,491	0,12	0,509	0,16	0,455
lipari	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
lyme	0,22	0,53	0,220	0,542	0,20	0,562	0,14	1,000	0,04	0,286	0,00	-
shakespeare	0,18	0,58	0,220	0,561	0,20	0,564	0,18	0,583	0,18	0,733	0,16	0,500
sushi	0,04	1,00	0,060	0,667	0,06	0,667	0,06	0,667	0,06	0,667	0,06	0,833
telecommuting	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
volcano	0,06	0,33	0,060	0,333	0,06	0,333	0,02	-	0,02	-	0,00	-
zener	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
Gemiddelde:	**0,090**	**0,534**	**0,094**	**0,457**	**0,084**	**0,485**	**0,073**	**0,510**	**0,065**	**0,462**	**0,066**	**0,564**

Zeker met de tweede methode:

Tekst-index & H&A, 50 resultaten ('groot'), OSim & KSim												
Weging H&A:	0%		10%		30%		50%		70%		90%	
	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim	OSim	KSim
alcoholism	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
architecture	0,100	0,60	0,10	0,73	0,10	0,80	0,41	0,61	0,12	0,71	0,16	0,51
bicycling	0,120	0,60	0,12	0,47	0,12	0,60	0,12	0,60	0,10	1,00	0,08	1,00
blues	0,040	0,00	0,06	0,33	0,06	0,33	0,06	0,33	0,04	0,67	0,04	0,67
buddhism	0,080	0,83	0,10	0,30	0,10	0,30	0,08	0,83	0,02	-	0,02	-
cheese	0,140	0,55	0,24	0,33	0,20	0,48	0,10	0,48	0,00	-	0,00	-
cruises	0,240	0,36	0,28	0,51	0,28	0,55	0,28	0,61	0,30	0,55	0,28	0,56
gardening	0,080	0,75	0,12	0,53	0,12	0,69	0,12	0,52	0,12	0,48	0,08	0,40
hiv	0,140	0,29	0,18	0,27	0,18	0,31	0,28	0,43	0,28	0,49	0,30	0,47
lipari	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
lyme	0,220	0,53	0,26	0,55	0,28	0,62	0,28	0,55	0,20	0,61	0,18	0,70
shakespeare	0,180	0,58	0,24	0,51	0,24	0,47	0,22	0,52	0,14	0,52	0,12	0,33
sushi	0,040	1,00	0,04	1,00	0,04	1,00	0,04	0,00	0,04	0,00	0,00	-
telecommuting	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
volcano	0,060	0,33	0,06	0,33	0,06	0,33	0,06	0,17	0,06	0,00	0,06	0,67
zener	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-	0,00	-
Gemiddelde:	**0,090**	**0,534**	**0,113**	**0,489**	**0,111**	**0,541**	**0,111**	**0,470**	**0,089**	**0,503**	**0,083**	**0,589**

In een grafiek is het verschil duidelijk te zien:



10 Reflectie

10.1 Conclusies

De voorgestelde combinatie van een tekst-index en het H&A-algoritme werkt significant beter wat betreft het vinden van autoriteiten op Twitter dan de tekst-index alleen, of de tekst-index in combinatie met PageRank. Het is interessant om te zien dat er zelfs onderwerpen in de testquery's waren die meer dan 20% overlap vertoonden met het voorgestelde vergelijkingsmateriaal van Listorious.

De volgorde van de rankings lijkt, gezien de KSim-waarden, in vergelijking met de referentie vrijwel geheel willekeurig: bij beide methoden, met elke wegingsfactor, komt er gemiddeld ongeveer 0,5 uit. Zie *Similarity measures* waarom dit betekent dat de volgorde willekeurig is.

De onderzoeksvraag:

> *Hoe kunnen we structurele patronen halen uit de onderwerpen waar tweeters over schrijven, op zo'n manier dat bepaald kan worden wie autoriteiten zijn op een bepaald gebied?*

kan dus nu gedeeltelijk beantwoord worden: de gebruikte structurele patronen zijn de tekst van de tweets, gecombineerd met welke tweeters naar elkaar verwijzen. Door verzamelde tweets samen te voegen per tweeter, deze te indexeren met een standaard tekst-index, en op te slaan welke tweeters naar elkaar verwijzen, is een geschikte database ontstaan waarmee query's kunnen worden uitgevoerd. Deze query's worden uitgevoerd door de resultaten van een standaard TF/IDF-algoritme te combineren met het Hubs and Authorities-algoritme. De resultaten zijn niet zaligmakend, maar voor een eerste opzet wel voldoende.

Zoals beschreven in hoofdstuk 7, werkt PageRank, waar het op het web wel goed blijkt te werken, niet goed op Twitter. Een verklaring zou kunnen zijn dat daar waar op het web de meeste webpagina's maar over 1 onderwerp gaan, tweeters vaak over meerdere onderwerpen iets schrijven (of retweeten).

10.2 Discussie

Dat het systeem niet hoger scoort kan op allerlei manieren verklaard worden:

- de belangrijkste is dat de lijsten van Listorious niet automatisch zijn samengesteld, maar handmatig. Omdat de methode van samenstellen van lijsten qua methode compleet verschilt met die van een geautomatiseerd systeem zoals het voorgestelde systeem, is het niet raar dat de verschillen in resultaten groot zijn. Dit geldt niet alleen voor de mate van overlap, maar ook de volgorde, en daarmee de KSim-waarden.
  Helaas is er geen ander – beter of beter vergelijkbaar – zoeksysteem gevonden waarmee de resultaten vergeleken kunnen worden.

- er is geen rekening gehouden met hoeveel volgers een tweeter heeft. Intuïtief zou je kunnen zeggen dat iemand die meer volgers heeft, waarschijnlijk een grotere autoriteit geniet.

- er is geen rekening gehouden met de inhoud van specifieke tweets waarin wordt verwezen: in dit onderzoek is er uit onder andere praktische redenen voor gekozen om alle tweets van één tweeter samen te voegen tot één document. Hiermee is de koppeling tussen wat er is gezegd en naar wie is verwezen verloren gegaan, terwijl dit juist een sterke indicatie zou kunnen zijn om aan te geven of een bepaalde tweeter ergens autoriteit in is.

- er zijn een aantal aannames gemaakt, en met een aantal variabelen uit tijdsinspanningssoverwegingen niet geëxperimenteerd. Er zit hier en daar zeer waarschijnlijk nog ruimte voor verbetering – waarbij overigens moet worden opgepast dat het systeem niet teveel wordt toegespitst op de testset.

10.3 Mogelijk vervolgonderzoek

Naast de al genoemde verklaringen waar in een eventueel vervolgonderzoek rekening gehouden kan worden, zijn er nog meer punten die meegenomen zouden kunnen worden:

- er is niet bepaald hoeveel pagerankstappen er zouden moeten worden uitgevoerd. Met de gebruikte dataset bleken 25 stappen genoeg om de waarden niet meer veel te laten fluctueren, maar uit (Page, Brin, Motwani, & Winograd, 1998) blijkt dat het benodigde aantal stappen afhangt van (vooral) de grootte van de graaf.

- bij het kiezen van 15 resultaten was er weinig overlap met Listorious. Dit is jammer, want in de praktijk blijken verreweg de meeste mensen niet voorbij een eerste zoekpagina te kijken (Zhang, Jansen, & Spink, 2009). Eigenlijk zou een zoeksysteem dat met de eerste paar resultaten meteen 'raak' is beter zijn. Er zou daarom specifiek onderzoek kunnen worden gedaan naar mogelijkheden om betere resultaten te krijgen bij de eerste paar hits. Overigens moet worden opgemerkt dat ook hier de volgorde van Listorious van invloed is: het gekozen rankingsvergelijkingsalgoritme kijkt alleen naar de eerste X van beide rankings, als dus Listorious het qua volgorde 'fout' heeft, en het gemaakte systeem doet het wel 'goed', dan scoort het systeem toch laag.

11 Bibliografie

Borodiny, A., Robertsz, G. O., Rosenthalx, J. S., & Tsaparas, P. (2002). *Finding Authorities and Hubs From Link Structures on the World Wide Web.*

Cheong, M., & Lee, V. (2009). *Integrating Web-based Intelligence Retrieval and Decision-making from the Twitter Trends Knowledge Base.* Hong Kong: ACM?

Craswell, N., Hawking, D., & Thistlewaite, P. (1999). Merging Results From Isolated Search Engines. *Proceedings of the Tenth Australasian Database Conference*, (pp. 189-200). Auckland.

Haveliwala, T. H. (2003, July/August). Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search. *IEEE Transactions on Knowledge and Data Engineering* , 784-796.

Kleinberg, J. M. (1999). *Authoritative sources in a hyperlinked environment.* New York: J. ACM.

Krishnamurthy, B., Gill, P., & Arlitt, M. (2008). A Few Chirps About Twitter. *Proceedings of the first workshop on Online social networks - WOSP '08* , 19-24.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* New York, NY: Cambridge University Press.

Metzler, D., Dumais, S., & Meek, C. (2007). Similarity Measures for Short Segments of Text. In *Advances in Information Retrieval* (pp. 16-27). Heidelberg: Springer Berlin.

Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The PageRank Citation Ranking: Bringing Order to the Web.* Palo Alto: Stanford University.

Siegler, M. (2010, September 14). *Twitter Hatches The New Twitter.com — A New Two-Pane Experience (Live).* Retrieved March 29, 2011, from TechCrunch: http://techcrunch.com/2010/09/14/twitter-event/

The Apache Software Foundation. (n.d.). *Apache Java Lucene.* Retrieved March 28, 2011, from Apache Lucene: http://lucene.apache.org/java/

The Apache Software Foundation. (n.d.). *Similarity (Lucene 3.0.3 API)*. Retrieved March 28, 2011, from Apache Lucene: http://lucene.apache.org/java/3_0_3/api/core/org/apache/lucene/search/Similarity.html

The JUNG Framework Development Team. (n.d.). *JUNG - Java Universal Network/Graph Framework*. Retrieved mei 09, 2011, from JUNG - Java Universal Network/Graph Framework: http://jung.sourceforge.net/

Twitter, Inc. (n.d.). *Twitter API Wiki / FrontPage.* Retrieved March 29, 2011, from Twitter API Wiki: http://apiwiki.twitter.com/

Twitter, Inc. (n.d.). *Twitter Help Center.* Retrieved March 30, 2011, from Twitter Help Center: http://support.twitter.com/groups/31-twitter-basics/topics/108-finding-following-people/articles/14022-how-to-find-people-on-twitter-twitter-search

Zhang, Y., Jansen, B., & Spink, A. (2009). Time series analysis of a Web search engine transaction log. *Information Processing & Management , 45* (2), 230-245.

12 Bijlage 1: technische implementatie systeem

12.1 Uitdagingen

Ondanks dat er slechts een deelverzameling van alle publieke tweets gebruikt is voor het onderzoek, bleek dit toch te resulteren in een vrij grote dataset: dagelijks bijna 1 gigabyte (zo'n 400.000 tweets) aan ruwe data. Een aantal bewerkingen hierop bleken vrij kostbaar, vooral omdat de dataset niet in het werkgeheugen van de gemiddelde computer past.

12.1.1 Sorteren

Een interessant voorbeeld hiervan is het samenvoegen van alle tweets per tweeter, om op die manier alle tweets van één tweeter als document te kunnen zien. Het sequentieel door alle data heenlopen en per tweet te bekijken of er van de bijbehorende tweeter al eerder een tweet was gevonden, en zo ja de teksten ervan samen te voegen leek behoorlijk tijdsintensief. Door goede indexen is zoeken eventueel nog vrij snel uit te voeren, maar het steeds aanpassen van kleine stukjes tekst op de harde schijf duurt lang. De oude tekst moet eerst worden ingelezen, en daarna de nieuwe tekst geschreven. Dit kost daarmee ruwweg 2 schijfrotaties, en met een moderne harde schijf van 7200 toeren per minuut is dat ongeveer 16 milliseconden. Met de verzamelde dataset kost dat in totaal minimaal 2 dagen. Het leek daarom handiger om de tweets te sorteren op tweeter, en dit resultaat sequentieel te lezen: op deze manier heb je alle tweets van één tweeter bij elkaar en kan je in 1 keer een document ervan bouwen. Harde schijven zijn orden van grootte sneller in sequentieel lezen en schrijven, dan random lezen en schrijven.

Het sorteren van tweets was echter weer een uitdaging op zich. De oplossing leek in een soort merge sort: sorteer steeds een aantal tweets dat nog wel in het geheugen past, en sla dit op als een los bestand. Doe dit met alle tweets, zodat er een relatief klein aantal bestanden (zeg 50) overblijft die elk op zich gesorteerd is. Vervolgens kunnen deze bestanden tegelijkertijd sequentieel worden doorlopen, waarbij steeds de 'laagste' tweet wordt opgepakt en wegschrijven naar één resultaatbestand. Door voor deze 50 bestanden een relatief grote geheugenbuffer te nemen hoeft er maar af en toe een leesactie gedaan te worden. Hetzelfde geldt voor het resultaatbestand: ook hier wordt met een geheugenbuffer gewerkt, om zo maar af en toe te hoeven schrijven.

In twee fasen zijn hiermee alle tweets gesorteerd: in de praktijk kon op deze manier, zonder veel geoptimaliseerd te hebben, ongeveer 12 GB aan tweets in 20 minuten gesorteerd worden, waarbij de meeste tijd ging zitten in het parseren van de ruwe tweets, en niet het sorteren, lezen of schrijven.

12.1.2 Doorzoeken

De gekozen aanpak voor het zoeken naar autoriteit ging uit van een variant van PageRank. PageRank zelf geeft alleen per document aan hoe belangrijk deze is in vergelijking met andere documenten. Om er daadwerkelijk nog een zoeksysteem van te maken, dat wil zeggen: reageren op een query, moesten de tweets ook nog doorzoekbaar zijn. De keuze hierin is om dit zelf uit te programmeren, of een al bestaand systeem te gebruiken.

Omdat het niet om het belangrijkste onderscheidende onderdeel van het onderzoek gaat, en het dus niet zeer specifiek aangepast hoeft te worden hieraan, is een standaardsysteem een goede keuze.

Een veelgebruikt standaardsysteem is Apache Java Lucene (The Apache Software Foundation), dat zichzelf aanprijst als een "*high-performance, full-featured text search engine library written entirely in Java.*". (The Apache Software Foundation) beschrijft het geheel als een systeem dat het Boolean Model van de Information Retrieval combineert met het Vector Space Model: documenten die voldoen aan de query aan de hand van het Boolean Model, worden gerankt met behulp van het Vector Space Model.

12.2 Pseudocode

De uiteindelijke opzet van het systeem is, in pseudocode, als volgt.

12.2.1 Verzamelen dataset

```
Stel in welke tweets moet worden worden verzameld.
filterwoorden = "egypt tunesia mubarak jan25 tahrir qatar cairo palastine freeegypt
      tunisia tunesia alcoholism architecture bicycling blues buddhism cheese
      cruises gardening hiv lipari lyme shakespeare sushi telecommuting volcano
      zener d66 pvda cda groenlinks sgp sp vvd";

Start de tweetverzamelaar, en laat deze een aantal weken draaien.
dataset = twitter.filterApi(filterwoorden);
```

12.2.2 Indexeren dataset

```
Sorteer de dataset op twitternaam (en verder chronologisch).
gesorteerdeDataset = sorteer(dataset, {twitternaam, datum});

Lees de nu gesorteerde dataset sequentieel in, en voeg alle tweets van één tweeter
samen. Hierbij wordt ook opgeslagen naar welke tweeters een tweeter heeft verwezen.
timelines = voegTweetsSamenTotTimelines(gesorteerdeDataset);

Stop voor elke tweeter de timeline in de tekst-index-database Apache Lucene.
Hierbij wordt de documentlengte artificieel aangepast zodat deze overeenkomt met de
vermoedelijke hoeveelheid tweets van de tweeter die niet door het filter heen
kwamen.
tekstIndex = creeerTekstIndex(timelines);

Maak een PageRank-graaf van elke tweeter (een tweeter wordt gerepresenteerd als een
knoop, een verwijzing als een kant).
pageRankGraaf = creeerPageRankGraaf(timelines);

Maak een PageRank-database door het PageRank-algoritme uit te voeren op de
PageRank-graaf. Het resultaat is een database van tweeters waarbij elke tweeter een
PageRank-waarde heeft.
pageRankDatabase = pageRank(pageRankGraaf);
```

12.2.3 Queryen methode 1 (PageRank)

```
Vraag de tekst-index om maximaal 300 documenten (tweeters) die voldoen aan de
ingegeven query.
hitlijstMetDocumentScores = tekstIndex.query(ingegevenQuery, 300);

Vraag voor elk van de gevonden tweeters hun PageRank-waarde op.
hitlijstMetPageRankWaarden =
      pageRank.geefPageRankWaarden(hitlijstMetDocumentScores);

Normaliseer beide lijsten naar waarden tussen 0 en 1.
hitlijstMetDocumentScores = normaliseer(hitlijstMetDocumentScores);

hitlijstMetPageRankWaarden = normaliseer(hitlijstMetPageRankWaarden);

Voeg de scores en waarden samen om zo tot een nieuwe ranking te komen. Hierbij
kunnen verschillende gewichten aan de beide rankings gehangen worden.
```

```
hitlijstMetSamengevoegdeScores = voegRankingsSamen(
       {hitlijstMetDocumentScores, 0.7}, {hitlijstmetPageRankWaarden, 0.3});

Geef de 50 resultaten met de hoogste scores weer, op volgorde van score.
printBesteResultaten(hitlijstMetSamengevoegdeScores, 50);
```

12.2.4 Queryen methode 2 (H&A)

```
Vraag de tekst-index om alle documenten (tweeters) die voldoen aan de ingegeven
query.
hitlijstMetDocumentScores = tekstIndex.query(ingegevenQuery);

Gebruik de eerste paar als 'seed' (root set) voor het H&A-algoritme.
rootSet = hitlijstMetDocumentScores.geefDeBesteTweeters(30);

Maak de base set: neem de root set, en voeg daar alle tweeters aan toe die naar
deze root set verwijzen, en alle tweeters waarnaar de root set verwijst.
baseSet = rootSet + rootSet.getPredecessors() + rootSet.getSuccessors();

Voer het HITS-algoritme uit op de base set, en haal vraag van elke tweeter de
autoriteitswaarde op.
hitlijstMetAutoriteitsWaarden =
       voerHitsAlgoritmeUit(baseSet).geefHitlijstMetAutoriteitsWaarden();

Normaliseer beide lijsten naar waarden tussen 0 en 1.
hitlijstMetDocumentScores = normaliseer(hitlijstMetDocumentScores);

hitlijstMetAutoriteitsWaarden = normaliseer(hitlijstMetAutoriteitsWaarden);

Voeg de scores en waarden samen om zo tot een nieuwe ranking te komen. Hierbij
kunnen verschillende gewichten aan de beide rankings gehangen worden.
hitlijstMetSamengevoegdeScores = voegRankingsSamen(
       {hitlijstMetDocumentScores, 0.7}, { hitlijstMetAutoriteitsWaarden, 0.3});

Geef de 50 resultaten met de hoogste scores weer, op volgorde van score.
printBesteResultaten(hitlijstMetSamengevoegdeScores, 50);
```

13 Bijlage 2: een tweet

Zoals beschreven in *5.2 Tweet* bevat een tweet, zoals gestuurd door Twitters api's, relatief veel gegevens. Naar keuze kan je deze ontvangen in XML- of JSON-formaat, waarbij JSON een veel compacter en sneller te parseren formaat is. Een voorbeeld van een tweet, in JSON-formaat:


```
{
    "contributors": null,
    "user": {
        "listed_count": 4,
        "contributors_enabled": false,
        "verified": false,
        "profile_sidebar_border_color": "829D5E",
        "notifications": null,
        "profile_use_background_image": true,
        "favourites_count": 0,
        "created_at": "Sat Apr 18 08:15:07 +0000 2009",
        "friends_count": 254,
        "profile_background_color": "4c2c2a",
        "location": "On KPFK radio Mondays at 7pm. ",
        "profile_background_image_url":
                  "http://a1.twimg.com/profile_background_images/101052867/grub19.j
                  pg",
        "followers_count": 110,
        "profile_image_url":
                  "http://a1.twimg.com/profile_images/1014642754/No_H8_9250_sheri_n
                  ormal.jpg",
        "description": "Equality advocate & communications professional w/ a few
                  higher ed degrees, seeks brevity & clarity in less than 160
                  characters. ",
        "show_all_inline_media": false,
        "geo_enabled": false,
        "time_zone": "Pacific Time (US & Canada)",
        "profile_text_color": "3E4415",
        "screen_name": "SheriLunn",
        "follow_request_sent": null,
        "statuses_count": 543,
        "profile_sidebar_fill_color": "99CC33",
        "protected": false,
        "url": null,
        "id_str": "32831355",
        "is_translator": false,
        "profile_background_tile": true,
        "name": "Sheri Lunn",
        "lang": "en",
        "id": 32831355,
        "following": null,
        "utc_offset": -28800,
        "profile_link_color": "D02B55"
    },
    "retweeted": false,
    "in_reply_to_user_id_str": null,
    "geo": null,
    "in_reply_to_status_id": null,
    "text": "RT @GOOD: In Vermont, a growing interest in #bicycling is reducing
                  traffic, creating jobs, and improving quality of life http:
                  //su.pr/2qkPrW",
    "created_at": "Tue Feb 15 19:21:49 +0000 2011",
    "source": "u003Ca href="http://www.echofon.com/"
                  rel=\"nofollow\"\u003EEchofon\u003C\/a\u003E",
    "in_reply_to_user_id": null,
    "truncated": false,
    "entities": {
        "user_mentions": [
            {
```

```
                "indices": [
                    3,
                    8
                ],
                "screen_name": "GOOD",
                "name": "GOOD ",
                "id_str": "19621110",
                "id": 19621110
            }
        ],
        "urls": [
            {
                "indices": [
                    121,
                    140
                ],
                "url": "http://su.pr/2qkPrW",
                "expanded_url": null
            }
        ],
        "hashtags": [
            {
                "indices": [
                    44,
                    54
                ],
                "text": "bicycling"
            }
        ]
    },
    "coordinates": null,
    "place": null,
    "favorited": false,
    "id_str": "37592435647455232",
    "retweet_count": 0,
    "in_reply_to_screen_name": null,
    "id": 37592435647455232,
    "in_reply_to_status_id_str": null
}
```