

Automatisch definities verkrijgen van webbronnen.

Verslag van Mark Jansen s1253875
Information Science
Groningen University
In samenwerking met Olaf Woertel s1397117

Inleiding

Bij het lezen van wetenschappelijke teksten, vakliteratuur of simpelweg nieuwsberichten kom je nogal eens termen tegen waar je de betekenis niet van kent. Om de tekst dan toch goed te kunnen volgen, zul je de definities moeten opzoeken. Definities van zulke termen kun je over het algemeen vinden in Wikipedia [12] of gespecialiseerde woordenboeken.

Dit verslag zal uitweiden over een poging om Wikipedia te gebruiken als bron voor het automatisch verkrijgen van definities bij termen. Om het domein enigszins te beperken, zal het systeem zich met name richten op ICT termen. Wikipedia is een geschikte bron voor een dergelijk systeem, niet alleen vanwege de omvang (momenteel 1.858.050 artikelen in de Engelse versie en meer dan 250.000 artikelen in de Nederlandse versie), maar vooral ook vanwege de toegankelijkheid van de informatie. In de eerste plaats omdat de informatie van vrij beschikbaar is. Daarnaast hebben artikelen op Wikipedia allemaal min of meer dezelfde indeling, wat helpt bij het verkrijgen van de juiste informatie. Het doel van het systeem in dit onderzoek is dan ook deze informatie beschikbaar te maken als hulpmiddel bij het lezen van teksten. Definities kunnen dan op verzoek bij de tekst getoond worden.

Hoe nauwkeurig is dit systeem? Welke problemen zijn er? Is Wikipedia wel geschikt voor deze manier van informatievoorziening? Vragen als deze zullen beantwoord moeten worden bij het maken van een systeem als dit. Er zijn al verscheidene onderzoeken geweest naar het verkrijgen van definities uit verschillende bronnen. Vaak hebben deze ook een eigen domein waar ze zich op richten. Een paar van deze onderzoeken zullen worden besproken. Daarna zal uitgebreid worden ingegaan op de totstandkoming van dit systeem, de keuzes die zijn gemaakt en de problemen die zijn gevonden. Tot slot zullen een paar tests met betrekking tot nauwkeurigheid worden gedaan op teksten van verschillende bronnen en de resultaten daarvan worden besproken.

Literatuur

Definities vinden vereist nauwkeurige taalkundige regels en feilloze heuristieken. In eerste instantie ging men uit van handmatige aangemaakte patronen zonder gebruik te maken van machine learning technieken. Joho en Sanderson [1] halen *descriptive phrases* (dp) van *query nouns* (qn) uit teksten om definitie vragen zoals *Who is qn?* te beantwoorden. Patronen zoals '*dp* met name *qn*', Hearst [2] worden gebruikt om namen en hun omschrijving te vinden.

Hildebrandt et al. [3] maken gebruik van oppervlakte patronen om zoveel mogelijk relevante stukken informatie over een onderwerp uit teksten te halen. Een voorbeeld van een patroon die zij gebruiken voor extractie is NP_1 *be* NP_2 . Stukken tekst die niet met een lidwoord beginnen worden weggegooid om valse stukken eruit te halen. De stukken informatie uit alle teksten van een corpus worden vervolgens in een database opgeslagen. Definities kunnen vervolgens simpelweg met de relevante termen in de database opgezocht worden. Deze manier is vergelijkbaar met die van Fahmi en Bouma [4].

Fahmi en Bouma gaan er in eerste instantie vanuit dat een definitie onder andere een vorm van *zijn* bevat. Vervolgens proberen zij in de resultaten hiervan het aantal niet-definitiezinnen te verminderen. Dit doen zij als eerste met behulp van een simpel lexicaal filter, die bijvoorbeeld zinnen weghaalt waarvan het onderwerp begint met woorden als

oorzaak, gevolg, voorbeeld, probleem, resultaat, kenmerk, mogelijkheid, symptoom, teken etc. of als de zin het woord *geen* bevat. Om evaluatie- en trainingsdata te verkrijgen hebben ze handmatig 2500 zinnen geannoteerd als zijnde definitie, niet-definitie of onbeslist. Onbeslist betekent dat de zin kenmerken vertoont van een definitie, maar dit niet (volledig) is. Fahmi en Bouma experimenteren vervolgens met bag-of-words, bigrams, root forms, documenteigenschappen, syntactische eigenschappen en named entity tags, in combinatie met machine learning technieken, in hun poging om de beste middelen voor het verkrijgen van definities te vinden. Uit hun resultaten bleek dat met name het toevoegen van syntactische eigenschappen aan machine learning technieken de nauwkeurigheid erg vergrootte.

Het gebruik van machine learning technieken is te vinden in Miliaraki en Androutsopoulos [5] en Androutsopoulos en Galanis [6]. Zij gebruiken vergelijkbare patronen als Joho en Sanderson om trainingsattributen te verkrijgen. Sager en L'Homme [7] merken op dat de definitie van een term altijd tenminste een *genus* (categorie van de term) en *species* (eigenschappen van de term) moet bevatten. Blair-Goldensohn et al.[8] gebruiken machine learning en handmatig gemaakte lexicaal-syntactische patronen om zinnen te vinden die zowel een *genus* als een *species* gedeelte bevatten voor een gegeven term.

Fahmi en Bouma gebruiken voor hun corpus de Nederlandse versie van Wikipedia en hebben daaruit alle artikelen gehaald die voor hun onderzoeksdomein (het medische) van toepassing zijn. Uit hun geannoteerde zinnen blijkt dat een groot deel van de definities in hun bronmateriaal zich in de eerste zin van het betreffende artikel bevindt. Dit is te danken aan de manier waarop Wikipedia artikelen zijn gestructureerd en deze constatering gaat dus niet op voor andere, minder gestructureerde bronnen. In eerdere onderzoeken is er al wel vanuit gegaan dat definities zich over het algemeen aan het begin van een tekst bevinden. Liu et al. [9] maken ook voor een gedeelte gebruik van de structuren op webpagina's. Door gebruik te maken een combinatie van een zoekmachine, patroonherkenning en webpagina-structuren zoals headers (<h1> etc) en tekst-markup (etc) proberen zij onderwerpspecifieke concepten en definities te verzamelen vanaf het web.

Een ander aandachtspunt bij het vinden van definities zijn acroniemen. Acroniemen vormen ook een belangrijke barrière bij het begrijpen van vakliteratuur of wetenschappelijke teksten. De dynamische aard van dit onderdeel van taal, samen met feit dat er jaarlijkse vele nieuwe acroniemen bijkomen, zorgt ervoor dat er behoefte ontstaat naar een mogelijkheid om acroniemen te verkrijgen uit grote hoeveelheden tekst. Veel traditionele definities van acroniemen zijn beperkend zoals bij [10] "woorden gevormd door de initialen van andere woorden", en missen daardoor een behoorlijk aantal acroniemen (XML, DNA etc). Dit neemt niet weg dat een acroniem beschouwd kan worden als een systematische afkorting. Zahariev [11] doet een poging met behulp van *acronym-expansion matching* waarbij aan de hand van bepaalde patronen wordt geprobeerd het acroniem te matchen met woordengroepen in de teksten. Hoewel een vrij nauwkeurig resultaat wordt behaald, blijven er fouten bestaan. Gebrek aan morfologische bronnen en acroniemen die bestaan uit symbolen, zoals W3C voor 'World Wide Web Consortium', zijn problemen waar Zahariev tegen aanloopt.

Plan van aanpak

Vrijwel iedereen leest wel eens een krant of een ander tijdschrift. En vrijwel iedereen komt wel eens een woord tegen dat hij of zij niet begrijpt bij het lezen van een dergelijk tijdschrift. Je kunt vaak uit de context al opmaken wat het ongeveer betekent en het negeren terwijl je toch het artikel of de tekst begrijpt. In sommige gevallen kan dat niet of zou je wel wat meer over dat onderwerp te weten willen komen. Dan zul je al snel naar een encyclopedie of iets dergelijks moeten grijpen, maar erg handig is dit niet.

Tegenwoordig verschijnen veel teksten ook op het web. Vooral wetenschappelijke teksten worden vaak op deze manier beschikbaar gemaakt. Deze teksten zullen ook vaker

online gelezen worden. Het bovenstaande probleem bestaat echter nog steeds. Als je een onbekende term tegenkomt, zul je nog steeds opzoek moeten naar een definitie. In veel gevallen kun je dat meteen online doen, maar dit maakt het niet minder omslachtig. Het zou veel makkelijker zijn als je ter plekke een definitie kunt opvragen bij het betreffende woord, bijvoorbeeld met behulp van een popup. Dit is precies wat dit onderzoek probeert te doen.

Een systeem dat automatisch definities ophaalt van de woorden waarop geklikt wordt, moet uit een paar verschillende onderdelen bestaan. Grofweg zijn de volgende drie onderdelen te onderscheiden:

Teksten met onbekende termen. Er moet een tekst zijn, waarin de gebruiker kan aangeven van welk woord hij of zij de definitie van zou willen weten, door er bijvoorbeeld op te klikken. Er zijn verschillende manieren waarop dit gedaan zou kunnen worden. De mooiste manier zou natuurlijk om dit als onderdeel van bestaande programma's zoals Adobe Acrobat Reader te hebben, omdat deze toch al veel gebruikt worden om teksten online te lezen. Dit is echter niet praktisch, omdat het modificatie van bestaande programmatuur vereist. Een makkelijkere manier om de teksten klikbaar te maken is door ze om te zetten in HTML. Dit vereist echter wel dat de teksten in een browser geopend worden.

Bron van definities. Dit spreekt voor zich. Dit onderdeel is de plek waar definities vandaan gehaald kunnen worden als de gebruiker daar om vraagt. Ook dit onderdeel kan op meerdere manieren gerealiseerd worden. Zo kan er een database worden ingericht met daarin definitie na definitie die vervolgens kunnen worden uitgelezen als de gebruiker er om vraagt. Dit kan erg snel en nauwkeurig zijn, maar het is erg onderhoudsgevoelig en het vereist een enorme hoeveelheid gegevens. Heel anders is het idee om definities automatisch te verkrijgen uit al beschikbare teksten. Dit kan bijvoorbeeld door de eerder besproken manieren van machine learning en patroonherkenning. Voordeel van deze manier is dat je over heel veel gegevens beschikt. Het is echter veel moeilijker om met deze manier een hoge nauwkeurigheid te behalen.

Verwerkingsdeel. Dit is het onderdeel dat ervoor zorgt dat de definitie gezocht wordt bij de term die door de gebruiker is aangegeven, en laat het resultaat vervolgens zien aan de gebruiker. Hoe dit precies gerealiseerd wordt zal moeten afhangen van de hiervoor genoemde onderdelen, omdat de keuzes die daar gemaakt zijn aangeven wat de mogelijkheden zijn voor dit onderdeel. Als er bijvoorbeeld is gekozen voor patroonherkenning of machine learning dan zullen die onderdelen hier gerealiseerd worden.

Bouw van het systeem

Bronnen.

Al snel was duidelijk dat Wikipedia een geschikte bron voor ons systeem zou zijn. De informatie is vrij toegankelijk, en de URLs worden allemaal op dezelfde manier opgebouwd. Een URL van een Wikipedia pagina bestaat namelijk altijd uit *http://nl.wikipedia.org/wiki/* met daar achter de zoekterm. De zoekterm begint met een hoofdletter en spaties dienen te zijn vervangen door 'underscores'. Als je dus op zoek bent naar bijvoorbeeld informatie over Prins Willem Alexander, dan zou de URL naar Wikipedia er zo uit moeten zien:

http://nl.wikipedia.org/wiki/Prins_Willem_Alexander.

Een ander belangrijk kenmerk van pagina's van Wikipedia is de structuur waarmee ze zijn opgebouwd. In de meeste gevallen begint de pagina met een inleidende alinea, gevolgd door

een plaatje, enkele statistische gegevens en daarna meer informatie onderverdeeld in verschillende delen. In het geval van Prins Willem Alexander onder andere: biografie, huwelijk, functies en voorouders. Fahmi en Bouma merkten al op dat in vrijwel alle artikelen met een definitie, deze zich in de eerste alinea bevond.

Eén van de uitdagingen waar we voor stonden was het vinden van definities van acroniemen. In veel gevallen staat in Wikipedia de volledige term, dus er de definitie van vinden was geen punt. Probleem was echter het bovenstaande voordeel, want je kunt alleen de juiste informatie vinden als je de volledige term weet om in de URL te gebruiken. Er moest dus een manier komen om acroniemen in hun volledige term om te zetten, zodat deze daarna in Wikipedia opgezocht kunnen worden. De Techweb encyclopedie bood hier uitkomst [13]. Deze encyclopedie heeft termen specifiek voor ons domein (ICT) en heeft als belangrijk voordeel dat het kan omgaan met afkortingen. Deze genoemde punten zijn ook gelijk nadelen aan deze encyclopedie: het domein is te specifiek en de weergave van informatie lijkt niet aan echte standaards te voldoen. Voor het omzetten van afkortingen bleek het echter een uitkomst, omdat afkortingen gekoppeld worden aan hun volledige term in de metatags van de pagina.

Zoektermen.

Aangezien besloten was om termen te beperken tot het ICT domein, was het noodzakelijk om de bronteksten in eerste instantie ook te beperken tot dit domein. Dit heeft twee voordelen: a) de zekerheid van zoektermen en b) een gevarieerd aanbod aan termen om het systeem mee te ontwikkelen en te testen. Als bronteksten werden nieuwsberichten en reviews van zowel hardware.info als tweakernet gebruikt. Deze teksten zijn Nederlandstalig en bevatten een reeks aan ICT termen. De geselecteerde teksten zijn omgezet in HTML, waarbij de termen handmatig van anchor (<a>) tags werden voorzien.

De anchor tags vervullen een tweetal taken. In de eerste plaats markeren ze de termen. Dit is vooral van belang om onderscheid te kunnen maken tussen ‘gewone’ woorden als *de*, *het*, *huis* en *lopen* aan de ene kant, en ‘moeilijke’ woorden als *computer*, *CPU* en *front side bus* aan de andere kant. Daarnaast zorgt de anchor tag ervoor dat de term klikbaar wordt en het daarmee mogelijk wordt er een Javascript *event* aan te koppelen. Als de gebruiker nu een term aanklikt, wordt het verwerkingsdeel van het systeem geactiveerd. Daarbij wordt ook de aangeklikte term meegenomen naar het verwerkingsdeel.

Verwerkingsdeel.

De zoekterm wordt aan het verwerkingsdeel meegegeven zoals het in de tekst voorkomt. In de meeste gevallen is de zoekterm dan niet geschikt om te gebruiken voor het ophalen van een Wikipedia-pagina. Daarom begint ons systeem met het vervangen van de eerste letter van de zoekterm door een hoofdletter. Ook worden alle spaties vervangen door een underscore. In de meeste gevallen zou dit voldoende moeten zijn om de term te kunnen gebruiken. Er zijn echter uitzonderingen.

Het kan voorkomen dat een term in zijn meervoudsvorm in de tekst staat, zoals *processors*. Als deze term dan wordt gebruikt in de URL, zal er geen resultaat gevonden worden, terwijl deze er wel is. Om dit probleem te omzeilen worden er naast de oorspronkelijke zoekterm ook varianten gebruikt. Deze varianten zijn niets anders dan de zoekterm zonder meervouds *-s* of *-en*. Als de zoekterm deze kenmerken van zichzelf al niet had, dan gebeurt er niets mee. Tenslotte is er nog een variant die er vanuit gaat dat de oorspronkelijke zoekterm volledig uit hoofdletters bestaat, en deze dus omzet naar de variant die begint met een hoofdletter en voor de rest uit kleine letters bestaat. Van elke zoekterm komen er dus vier varianten. In het geval van bijvoorbeeld *processors* zijn dit:

zoekterm 1: Processors

zoekterm 2: Processor
zoekterm 3: Processors
zoekterm 4: Processors

Merk op dat zoekterm 3 en 4 hetzelfde zijn als zoekterm 1, omdat deze niet eindigt op –en of volledig uit hoofdletters bestaat.

Alle varianten van de zoektermen worden in een array geplaatst. Deze worden stuk voor stuk doorgegeven aan een xmlhttprequest, totdat de juiste pagina is gevonden of totdat alle varianten zijn geprobeerd. De xmlhttprequest is het onderdeel dat de zoekterm combineert met het standaard URL gedeelte en vervolgens Wikipedia verzoekt die pagina terug te geven. Als Wikipedia de gezochte pagina gevonden heeft, zal deze in zijn geheel worden teruggestuurd naar ons systeem. Het zal regelmatig voorkomen dat een term niet op Wikipedia te vinden is. In dat geval zal er een popup verschijnen met het bericht dat die definitie niet gevonden kon worden.

De pagina die Wikipedia bij een gevonden term terugstuurt is gelijk aan de pagina zoals die te vinden zijn op de website zelf. Het systeem zou deze pagina rechtstreeks terug kunnen geven aan de gebruiker, zodat deze zelf meer kan lezen over dat onderwerp. Het doel is echter om alleen een korte definitie van de term terug te geven. De pagina zal dus geparsed moeten worden, waarbij alleen informatie overblijft die voor ons relevant is. Het kan echter ook voorkomen dat Wikipedia een doorverwijspagina teruggeeft. Een voorbeeld daarvan is *netwerk*. De vele verschillende vormen van netwerken zorgen ervoor dat Wikipedia niet een eenduidig antwoord kan geven op wat een netwerk is en geeft daarom de mogelijkheid om een specifiekere vorm te kiezen.

Om te beginnen doet ons systeem twee dingen met de pagina die het binnenkrijgt. In de eerste plaats kijkt het naar de titel van de pagina en slaat deze op. Daarnaast wordt ook de volledige tekst opgeslagen. Dit kan doordat in de code van pagina vaste kenmerken staan die aangeven waar deze onderdelen zich bevinden. Bovendien wordt de opmaak van deze opgeslagen stukken weggegooid, zodat er geen vreemde code in komt. De tekst wordt nu doorzocht op voorkomens van het woord ‘doorverwijspagina’, om zo onderscheid te kunnen maken tussen ‘informatiepagina’s’ en ‘doorverwijspagina’s’.

Als de pagina een doorverwijspagina blijkt te zijn dan wordt daaruit het relevante deel geselecteerd en daarna weergegeven aan de gebruiker met behulp van een popup. Het relevante deel is het deel waarin staat ‘<zoekterm> kan verwijzen naar:’ met daaronder een lijst van varianten op de zoekterm. Voordat dit wordt weergegeven wordt nog wel even gekeken of de zoekterm niet volledig uit hoofdletters bestaat. Als dit het geval is dan gaat het waarschijnlijk om een acroniem en zal eerst nog een poging gedaan worden om deze te disambigueren met behulp van de Techweb encyclopedie. Er is een poging gedaan om ons systeem automatisch een keuze te laten maken op een doorverwijspagina, maar die was niet altijd even succesvol. Hoe dit werkt, en waarom het niet standaard in het systeem zit, wordt verderop in dit verslag uitgelegd.

Een term ophalen van Techweb gaat op vrijwel dezelfde manier als het ophalen van de Wikipedia teksten. Ook Techweb heeft de mogelijkheid om artikelen te benaderen door de zoekterm in de URL te zetten. Met behulp van een xmlhttprequest wordt het gezochte artikel opgehaald, waarna ons systeem het relevante stuk tekst uit de pagina haalt en de rest weggooit. Dit gebeurt op dezelfde manier als bij Wikipedia teksten: de bron van de pagina heeft kenmerken die aangeven waar de relevante tekst gevonden kan worden. Het relevante stuk is in dit geval een regel tekst waarin tussen haakjes de volledige term van het acroniem staat. Er wordt gecheckt of de letters overeenkomen met de gevonden tekst, en of het langer is dan twee letters, om er zeker van te zijn dat we met de goede term te maken hebben. Als dit

voltooid is, wordt de oorspronkelijke zoekterm vervangen door deze nieuwe geëxpandeerde vorm, en start het proces opnieuw.

Vinden van de definitie.

In eerste instantie was het de bedoeling om met ons systeem een simpele vorm van patroonherkenning te gebruiken voor het vinden van de juiste informatie op een pagina. Vormen als 'is', 'is de' of 'is een' in combinatie met de zoekterm is een goede indicatie dat de zin waar dit in voorkomt de definitie van de zoekterm bevat. Al snel bleek echter dat het selecteren van de eerste zin waarin de zoekterm voorkwam net zo succesvol, zo niet beter was. Dit komt door de Wikipedia structuur waarbij de eerste alinea een inleiding op het onderwerp van de pagina omschrijft. Door de tekst te splitsen op punten in de tekst werden de verschillende zinnen gescheiden. Dan was het een kwestie van de eerste regel waarin de zoekterm voorkwam aan de gebruiker laten zien.

Hoewel dit een goede manier was om de locatie van de definitie te bepalen, bleek het selecteren van de zin alleen niet voldoende te zijn om de betekenis over te brengen. Bovendien kwam het voor dat zinnen niet volledig waren, omdat ook afkortingen met een punt, zoals een persoonlijke titel ('dr.') als zinseinde werden gezien door het systeem. Om deze problemen op te lossen is besloten de gehele eerste alinea te selecteren als 'definitie' van de zoekterm. De redenering hierachter is dat het beter is iets teveel informatie te geven dan niet genoeg.

Een alinea selecteren gebeurt aan de hand van twee varianten van de zoekterm. In eerste instantie de oorspronkelijke zoekterm. Deze wordt opgezocht in de tekst, en de eerste alinea waarin deze term voorkomt wordt aan de gebruiker getoond. Het kan echter zo zijn dat Wikipedia de zoekterm doorverwijst naar een pagina over die term, maar waarbij een iets andere term in de tekst gebruikt wordt. Een goed voorbeeld hiervan is *harddisk*. Tijdens het bouwen van het systeem werd er bij deze term een lege popup teruggegeven. Dit gaf aan dat er wel degelijk een positief antwoord kwam van Wikipedia, maar dat de term niet was gevonden in de tekst. Uit onderzoek bleek dat er op de pagina alleen werd gesproken over *harde schijf* en niet over de zoekterm, waardoor er dus niets kon worden teruggegeven. Dit is de reden voor de tweede variant van de zoekterm. Als Wikipedia een pagina teruggeeft, wordt ook de titel die bovenaan het artikel staat gebruikt als zoekterm om de juiste alinea te vinden. Deze titel is namelijk altijd overeenkomstig met de tekst, zelfs als de oorspronkelijke zoekterm werd doorverwezen naar die pagina. Als er een doorverwijzing op deze manier heeft plaatsgevonden, dan zal dat in de popup worden aangegeven op de volgende wijze: '<titelzoekterm> is hetzelfde als <oorspronkelijke zoekterm>'.
'

Zowel de definities als de overige berichten teruggegeven door middel van een popup. Dit is niet meer dan een nieuw venster op een kleiner formaat. De dimensies ervan kunnen in het systeem worden aangepast. De huidige dimensies (350 bij 300 pixels) voldoen voor de meeste definities. Een enkele keer zal het voorkomen dat een gebruiker moet scrollen om de gehele weergegeven tekst te kunnen lezen. In de titel staat de zoekterm waarmee de definitie was gevonden. Dit kan dus ook de titel van de Wikipedia-pagina zijn in plaats van de oorspronkelijke zoekterm. Binnen het venster worden zowel de titel als de zoekterm herhaald, als er een verschil is.

Zoals eerder vermeld is er een poging gedaan om bij doorverwijspagina's het systeem een keuze te laten maken tussen de onderwerpen, en vervolgens dat onderwerp aan de gebruiker te tonen. Het idee hierbij was dat de doorverwijspagina extra informatie geeft bij de links naar de verschillende opties, en dat die extra informatie gebruikt zou kunnen worden om te bepalen welke link gevolgd zou moeten worden om bij de juiste pagina uit te komen.

Om dat te bereiken wordt de doorverwijspagina bij het binnenkomen gesplitst op regels, zodat elke keuzemogelijkheid apart in een array komt te staan. Vervolgens worden de

regels één voor één doorlopen en gematcht op zowel de zoekterm als een klein aantal ICT gerelateerde termen, zoals: computer, digi, ict, en informati-. Dit zijn de woorden die veel voorkomen bij de keuze die voor ons interessant is. Als er een optie wordt gevonden waarbij de zoekterm en één van deze woorden samen voorkomen, dan wordt de url gelezen die bij deze optie hoort. Daaruit wordt de nieuwe zoekterm gehaald, waarna het hele proces zich herhaalt met deze nieuwe zoekterm (pagina zoeken, ophalen, parsen enz.). Als geen van deze woorden voorkomt bij de zoekterm wordt alsnog de doorverwijspagina aan de gebruiker getoond.

Ons systeem maakt momenteel geen gebruik van dit onderdeel. In plaats van proberen door te verwijzen, laat ons systeem altijd de keuze pagina zien. De automatische selectie had wisselende resultaten. In een aantal gevallen koos het systeem de juiste optie om daarna een goede definitie van de gezochte term te geven. In andere gevallen was er geen optie die overduidelijk in de richting van de ICT variant wees, waardoor het systeem niets anders kon dan de lijst weergegeven. Beide kwamen ongeveer even vaak voor en zijn allebei prima wat betreft resultaten. De gebruiker kan immers altijd zelf nog aangeven wat er waarschijnlijk bedoeld wordt. Dit werd echter teniet gedaan door de ontdekking van termen die ongerelateerde definities kregen. Voorbeeld hiervan was *accu* waarbij in de popup kwam te staan: '*Registergeheugen* is hetzelfde als *accu*', met vervolgens een uitleg over het registergeheugen.

Bij het nalopen van de stappen van ons systeem bleek dat er meerdere keren doorverwijspagina's was gegaan, waarbij steeds een keuze werd gemaakt, ook wanneer die weinig meer te maken had met de oorspronkelijke zoekterm. Bovendien stond de definitie van de keuze die bij het voorbeeld *accu* was gemaakt op een doorverwijspagina. Hieruit bleek dat de structuur van Wikipedia toch niet zo gestructureerd was als we graag hadden gewild. Dat levert een probleem op met de betrouwbaarheid van ons systeem. Daarom is het verstandiger de gebruiker zelf de keuzes te laten maken wanneer er meerdere mogelijkheden voor de zoekterm blijken te zijn.

Overige aanpassingen.

Het systeem is gemaakt voor het Nederlands en met de Nederlandse versie van Wikipedia in gedachten. Er wordt gebruik gemaakt van Nederlandse teksten en de zoekterm zijn daarom ook in het Nederlands. Vergeleken met de Engelse versie van Wikipedia is de Nederlandse versie nogal klein (ongeveer 1.880.000 artikelen in de Engelse tegenover ongeveer 316.000 artikelen in de Nederlandse). Aangezien de nauwkeurigheid van het systeem in het vinden van definities sterk afhankelijk is van de bron, en dan vooral het aantal artikelen in die bron, leek het ons van begin af aan al duidelijk dat het systeem nauwkeuriger zou zijn als het gemaakt was voor Engelse teksten en de Engelse versie van Wikipedia. Daarom is besloten het systeem ook te maken voor het Engels.

Veel veranderingen zijn er niet nodig om het systeem om te zetten naar een andere taal. De enige noodzakelijke wijziging is dat het systeem naar de Engelse Wikipedia verwezen moet worden. Daarnaast zijn natuurlijk de teksten met de zoektermen en alle berichten omgezet naar het Engels.

Om het systeem minder afhankelijk te maken van menselijke invloeden hebben we ook besloten om de manier te veranderen waarop zoektermen worden aangemerkt in de tekst. In eerste instantie was dit min of meer willekeurig, omdat dit handmatig gebeurde en het dus totaal afhankelijk is van de persoon die de teksten markeert of een woord mee wordt genomen of niet. Ook is deze manier erg tijdrovend. Lange teksten kunnen vele tientallen termen bevatten die allemaal een link moeten hebben. Gelukkig hebben de links allemaal dezelfde

vorm en is het dus niet moeilijk om dit automatisch te laten gebeuren. Het probleem zit in het selecteren van de woorden die een link moeten krijgen.

De oplossing hiervoor was het aanleggen van een woordenlijst. Zodra de gebruiker een tekst aanklikt, gaat ons systeem de tekst doorlopen op zoek naar woorden die in de woordenlijst voorkomen. Als er een woord gevonden is, wordt het vervangen door hetzelfde woord met een definitielink. Woorden die niet in de woordenlijst voorkomen worden met rust gelaten. Voordeel van deze manier is dat de beheerder van het systeem nu alleen teksten beschikbaar hoeft te maken voor de gebruiker, waarna het systeem de rest doet. Bovendien is er consistentie in de aangemerkte woorden. Vóór het automatiseren van de termselectie werd alleen de eerste keer dat een woord in de tekst voorkwam aangemerkt, nu zal een term elke keer worden omgezet. Er zitten echter ook nadelen aan deze manier. Zo is het systeem merkbaar langzamer, omdat elke tekst geparst moet worden als het wordt geselecteerd. Dit kan eventjes duren voor lange teksten.

Het grootste nadeel is echter dat het systeem staat of valt bij de omvang van de woordenlijst. Als het systeem een hele reeks woorden niet klikbaar maakt dan kan de gebruiker nog de definities niet vinden en heeft het systeem dus niet veel nut. Om ons systeem een degelijke woordenlijst te geven, hebben wij gezocht naar verzamelingen van ICT termen. Dit bleek een lastige klus, omdat dergelijke verzamelingen veelal niet groot zijn en de verschillende verzamelingen vaak dezelfde woorden bevatten. Met behulp van websites als *webopedia.com* hebben we echter meer dan 5800 woorden weten te verzamelen. Dit is verre van voldoende, maar een goed begin om ons systeem mee te testen.

Tests en resultaten

Het testen van het systeem zal in drie delen uiteenvallen, namelijk het testen van de Nederlandse versie, de Engelse versie en de ‘automatische versie’. De automatische versie is de versie waarbij teksten door het systeem van *anchors* worden voorzien in plaats van de beheerder. De andere twee maken gebruik van handmatige gekenmerkte teksten. Er zal worden gekeken naar het aantal keer dat er een tekst wordt teruggegeven, waarbij het niet uitmaakt of de tekst in de popup ook daadwerkelijk de gezochte definitie is. Ons systeem kan dit immers niet valideren. Er zal ook worden gekeken naar de verschillen tussen de versies.

Nederlandse versie

Voor het testen van de Nederlandse versie zijn vijf teksten gehaald van de website *computable.nl* [14]. Alle vijf de teksten zijn ICT gerelateerde nieuwsberichten met een gezamenlijke lengte van ongeveer 1450 woorden. In deze teksten werden in totaal 97 woorden van anchors voorzien. Hierbij zitten onder andere bedrijfsnamen, technische termen en productnamen. Voor de resultaten werd onderscheidt gemaakt tussen vier mogelijke reacties van het systeem, te weten:

- definitie – het systeem kwam met een mogelijke definitie voor de zoekterm
- keuzelijst – het systeem kwam met een disambiguatie-keuzelijst voor de zoekterm
- niets gevonden – het systeem was niet in staat informatie te vinden voor de zoekterm
- onbekend – het systeem kwam met een onbekend resultaat of error

Om de nauwkeurigheid van dit systeem te bepalen werden keuzelijsten echter beschouwd als een correct antwoord, omdat dit de gebruiker duidelijke mogelijkheden geeft om meer informatie te vinden over de zoekterm. ‘Onbekende’ reacties zijn daarentegen juist meegenomen bij foutieve resultaten, simpelweg omdat de gebruiker hier niets mee kan.

Dit zijn de resultaten voor de vijf Nederlandse teksten:

definities: 40
keuzelijsten: 7
niets gevonden: 50
onbekend: 0

Er is nog aardig wat variatie in het soort termen dat het systeem vindt, hoewel dit natuurlijk grotendeels ligt aan wat er aan informatie te vinden is op Wikipedia. Als de term er niet in voorkomt, dan kan het logischerwijs ook geen definitie teruggeven. Zo werden bijvoorbeeld de termen ‘*scrollen*’ en ‘*brandsnelheid*’ niet gevonden. Dit kan ook niet omdat hier geen pagina’s voor zijn.

Woorden als ‘*business intelligence*’ (“Business Intelligence (BI) staat voor het verzamelen van informatie binnen de eigen handelsactiviteit....”) en ‘*upgrades*’ (“Bij computers is een upgrade het vervangen van oude hardware of software door...”) leveren echter geen enkel probleem op voor het systeem. Ook zijn er resultaten waarbij er wel een definitie wordt gegeven, maar is dit niet wat je zou verwachten, bijvoorbeeld *brander* (“...is een schip dat gevuld wordt met teer en explosieven, dan met opzet aangestoken...”). Keuzelijsten zijn in de meeste gevallen ook wat je ervan zou verwachten. Een voorbeeld hiervan is bijvoorbeeld ‘*programma*’, dat in de popup een keuze aanbiedt tussen *computerprogramma*, *televisieprogramma*, *radioprogramma* of *programma* als tijdsplan. Termen die het meeste fout gaan zijn echter bedrijfs- en productnamen, en termen waarbij er verschillende spellingsvarianten zijn. Met bovenstaande resultaten heeft het Nederlandse systeem een respons van 48,5%.

Engelse versie

Voor het testen van de Engelstalige versie van het systeem zijn ook vijf teksten verzameld. Net als in de Nederlandse versie zijn dit ICT gerelateerde nieuwsberichten, in dit geval afkomstig van *computerworld.com* [15], *techworld.com* [16] en *itworld.com* [17]. De teksten hebben een gezamenlijke lengte van ongeveer 1650 woorden, waarin 118 woorden van anchors werden voorzien. Ook hierbij zitten onder andere bedrijfsnamen, technische termen en productnamen. Voor de resultaten werd wederom onderscheid gemaakt tussen de vier mogelijke reacties van het systeem.

Dit zijn de resultaten voor de vijf Engelse teksten:

definities: 48
keuzelijsten: 30
niets gevonden: 35
onbekend: 5

Opvallend is het verschil tussen de Engelse en de Nederlandse versie met betrekking tot de hoeveelheid keuzelijsten. Hier is echter een logische verklaring voor. In het Engels zijn veel termen die van toepassing zijn op computers en computerprogramma’s afgeleid van woorden die te maken hebben met kantoor en kantoorartikelen (*office*, *desktop*, *file*). Wikipedia maakt onderscheid tussen de termen middels de disambiguatiepagina’s. In het Nederlands is dit niet nodig, omdat er gewoon twee verschillende benamingen voor zijn. Vandaar deze toename in het aantal gevonden keuzelijsten.

Wat betreft het soort resultaten is er niet veel verschil met de Nederlandse versies. Van de gevonden definities zijn er zowel correcte (*laser mouse*) als incorrecte (*mouse*) teksten teruggegeven, maar deze staan min of meer los van woordsoorten. Net als in het Nederlandse gaan vooral productnamen nog vaak fout, hoewel dit bij het Engels niet ligt aan de beschikbaarheid, maar aan vele spellingsverschillen. Als de brontekst het anders schrijft dan

Wikipedia kan het systeem er niets mee. Voorbeeld hiervan is afstandsbediening. Deze term kwam voor in twee van de teksten, maar was in één geschreven als *remote-control* in plaats van *remote control*. Deze laatste wordt gevonden, de eerste niet. Opvallend is ook het aantal ‘onbekende’ resultaten. Deze kwamen in de Nederlandse versie amper voor, in ieder geval niet in de testteksten, maar wel in de Engelse versie. In alle ‘onbekende’ gevallen ging het om een herhaling van de zoekterm of een variant. Hoewel het helemaal duidelijk is, lijkt dit te komen door de manier waarop plaatjes zijn gebruikt op de betreffende pagina’s. Met bovenstaand resultaat heeft het Engelse systeem een respons van 66,1%.

Automatische versie.

Om de respons van de automatische versie te testen is gebruik gemaakt van dezelfde Engelse teksten als voor de Engelse versie. Dit om het verschil aan te kunnen geven met de handmatige gemarkeerde teksten. Het systeem markeerde in deze vijf teksten in totaal 125 woorden. Dit zijn echter maar 62 unieke termen, de overige zijn herhalingen. De automatische versie moet dus nog een flinke uitbreiding van de woordenlijst krijgen om vergelijkbare hoeveelheden termen te krijgen als met handmatig gemarkeerde teksten.

Een ander belangrijk verschil zit in de manier waarop combinatietermen worden gemarkeerd. In de automatische versie wordt momenteel gewerkt met de eerste match, wat in de meeste gevallen betekent dat de kortste variant wordt aangemerkt. In één van de teksten wordt *remote control* meerdere keren als twee losse termen aangemerkt. Dit is op zich geen probleem, omdat beide termen in Wikipedia voorkomen en daarom ook gevonden worden. Het is voor de tekst echter relevant om de combinatie gebruiken, zeker omdat dit een erg frequente term is. Bovendien zal de definitie van *remote control* meer van toepassing zijn dan de combinatie van de delen.

Dit zijn de resultaten voor de vijf teksten:

definities: 31

keuzelijsten: 20

niets gevonden: 5

onbekend: 6

Dit geeft de automatische versie van ons systeem een respons van 82,3%. Hoe komt het dat de automatische versie zo veel beter is dan de andere twee versies? Deze versie verschilt alleen van de andere versies met de manier waarop termen worden aangemerkt. Aangezien ook de teksten hetzelfde zijn, ligt de enige verklaring voor het verschil in de termen die zijn gebruikt. De automatische versie maakt geen gebruik van combinatietermen, en deze zijn juist foutgevoeliger. Zie het voorbeeld van *remote control* tegenover *remote-control* in de resultaten van de Engelse versie.

Een ander belangrijk verschil tussen de zoektermen van de verschillende versies is de hoeveelheid bedrijfs- en productnamen. Deze konden in de meeste situaties niet gevonden worden. De woordenlijst bevat deze termen over het algemeen niet en deze kunnen nu dus ook geen foutieve resultaten opleveren. Met het uitbreiden van de woordenlijst zullen deze termen toch toegevoegd moeten worden, wat de respons zal drukken totdat Wikipedia de artikelen hierover ook ter beschikking heeft.

Iets wat terugkwam in alle testen was het ‘verdwijnen’ van resultaten. Om nog onbekende redenen worden resultaten niet consistent teruggegeven. Zo kon het voorkomen dat een term de ene dag gevonden kon worden, en de volgende dag kon het systeem niet met een resultaat komen voor dezelfde term. In beide gevallen was de zoekterm exact hetzelfde en was de Wikipedia-artikel gewoon aanwezig en toegankelijk via de browser. Tijdens het ontwikkelen zijn verschillende tests gedaan om te zien waar het aan kan liggen, en hierbij

werd ons systeem uitgesloten als de oorzaak. Er lijken dingen verkeerd te gaan in de communicatie met Wikipedia.

Er is een demoversie van de automatische versie van het systeem online beschikbaar. Zie Appendix A voor meer informatie over deze demoversie en enkele screenshots.

Conclusie

Het is zeer goed mogelijk om met behulp van Wikipedia een leeshulp te maken. De artikelen zijn redelijk goed gestructureerd, waardoor het verkrijgen van tekst hieruit niet al te moeilijk is. De opbouw van de inhoud is gebeurd ook veelal op dezelfde manier, met een definitie of verkorte uitleg van het onderwerp aan het begin van de pagina. Het aanroepen van de pagina's is ook hetzelfde voor alle onderwerpen. Dit maakt ook meteen duidelijk wat de belangrijkste problemen zijn bij het verkrijgen van definities. Als een artikel niet bestaat, dan is het natuurlijk niet mogelijk een definitie terug te geven. Ook de schrijfwijze van de definities is hierbij van invloed. Wat dit betreft is de automatische versie het beste, omdat deze ongebruikelijk spellingsvormen ver het algemeen niet in de woordenlijst heeft.

Verschillende onderdelen van het systeem zijn nog voor verbetering vatbaar. In eerste plaats is zal de woordenlijst flink uitgebreid moeten worden om meer woorden in de teksten te kunnen aanmerken. Bovendien zal er een optie ingebouwd moeten worden waarbij de langste match wordt gebruikt, dus *remote control* in plaats van *remote* en *control*. Daarnaast zullen er nog meer varianten van de zoektermen moeten komen die spellingsvarianten afvangen. Denk daarbij bijvoorbeeld aan verbindingsstreepjes of verledentijdsvorm. Ook het selecteren van de tekst die moet worden teruggegeven is nog voor verbetering vatbaar. Momenteel wordt alleen nog de eerste alinea geselecteerd. Hoewel dit over het algemeen de gewenste informatie geeft, moet het selecteren nauwkeuriger kunnen. Dan zal er ook meteen een oplossing worden gevonden voor de gevallen met een 'onbekend' resultaat, omdat dit hier waarschijnlijk mee te maken heeft.

Het onderdeel dat zich met de acroniemen bezighoudt kan ook nauwkeuriger gemaakt worden, maar het is niet duidelijk hoeveel dit gaat uitmaken. In tegenstelling tot Wikipedia is de Techweb encyclopedie lang niet zo consequent in de manier waarop de tekst is opgebouwd, waardoor niet altijd een volledige term wordt teruggegeven. Tot slot zal er ook nagedacht moeten worden over een oplossing voor de keuzelijst. Het automatisch doorlinken bleek niet betrouwbaar te werken. In plaats van doorlinken zou het misschien een goed idee zijn om het systeem een item te laten kiezen en deze een andere kleur te geven volgens dezelfde criteria.

Met bovenstaande verbeteringen kan het systeem heel degelijke resultaten geven, waarbij de nauwkeurigheid vrijwel alleen zal afhangen van de beschikbaarheid van de artikelen op Wikipedia. Merk hierbij wel op dat er geen rekening is gehouden met de inhoud van de informatie. Het is immers niet aan ons systeem om te verifiëren of de verkregen informatie ook daadwerkelijk correct is. Dit is misschien een idee voor toekomstig onderzoek.

References

1. H. Joho and M. Sanderson. 2000. Retrieving descriptive phrases from large amounts of free text. In *CIKM*, pages 180.186.
2. Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of the 14th COLING*, pages 539.545, Nantes, France.
3. W. Hildebrandt, B. Katz, and J.J. Lin. 2004. Answering definition questions with multiple knowledge sources. In *HLT-NAACL*, pages 49.56.
4. Fahmi, Ismail and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In Roberto Basili and Alessandro Moschitti, editors, *Proceedings of the EACL workshop on Learning Structured Information in Natural Language Applications*, Trento, Italy.
5. S. Miliaraki and I. Androutsopoulos. 2004. Learning to identify single-snippet answers to definition questions. In *20th International Conference on Computational Linguistics (COLING 2004)*, pages 1360. 1366, Geneva, Switzerland. COLING 2004.
6. I. Androutsopoulos and D. Galanis. 2005. A practically unsupervised learning method to identify single-snippet answers to definition questions on the web. In *Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT-EMNLP 2005)*, Vancouver, Canada.
7. Juan C. Sager and M.C. L'Homme. 1994. A model for definition of concepts. *Terminology*, pages 351. 374.
8. S. Blair-Goldensohn, K. McKeown, and A.H. Schlaikjer. 2004. Answering definitional questions: A hybrid approach. In *New Directions in Question Answering*, pages 47.58.
9. B. Liu, C.W. Chin, and H.T. Ng. 2003. Mining topic-specific concepts and definitions on the web. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 251.260, New York, NY, USA. ACM Press.
10. J. Sykes, Ed., *The Concise Oxford dictionary of current English*, 6th ed. Oxford University Press, 1976.
11. M. Zahariev, Efficient Acronym-Expansion Matching for Automatic Acronym Acquisition, Simon Fraser University, Burnaby BC, Canada, 2004
12. <http://www.wikipedia.org/>
(link correct op 18 augustus 2007)
13. <http://techweb.com/encyclopedia/>
(link correct op 18 augustus 2007)
14. <http://www.computable.nl/>

(link correct op 18 augustus 2007)

15. <http://www.computerworld.com/>
(link correct op 18 augustus 2007)

16. <http://www.techworld.com/>
(link correct op 18 augustus 2007)

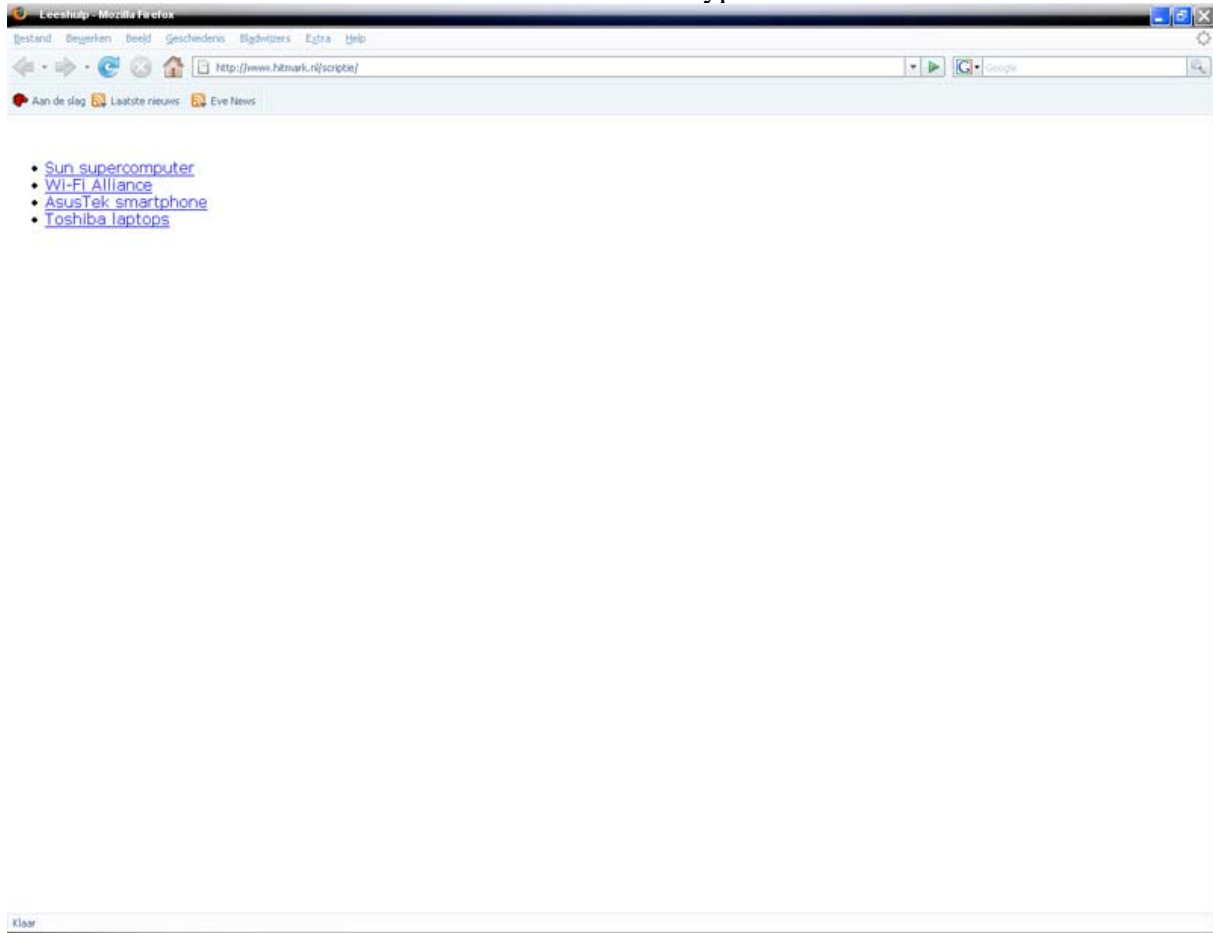
17. <http://www.itworld.com/>
(link correct op 18 augustus 2007)

Appendix A

Hieronder enkele screenshots van de online demo van de automatische versie van het systeem. Deze versie is te vinden op: <http://www.hitmark.nl/scriptie>. Het systeem vereist dat de Java Runtime Environment (JRE) is geïnstalleerd. Deze is te vinden op de Sun Microsystems website: <http://www.sun.com/download>. Het systeem is gebouwd en getest met Mozilla Firefox en zou probleemloos moeten werken met deze browser en de JRE.

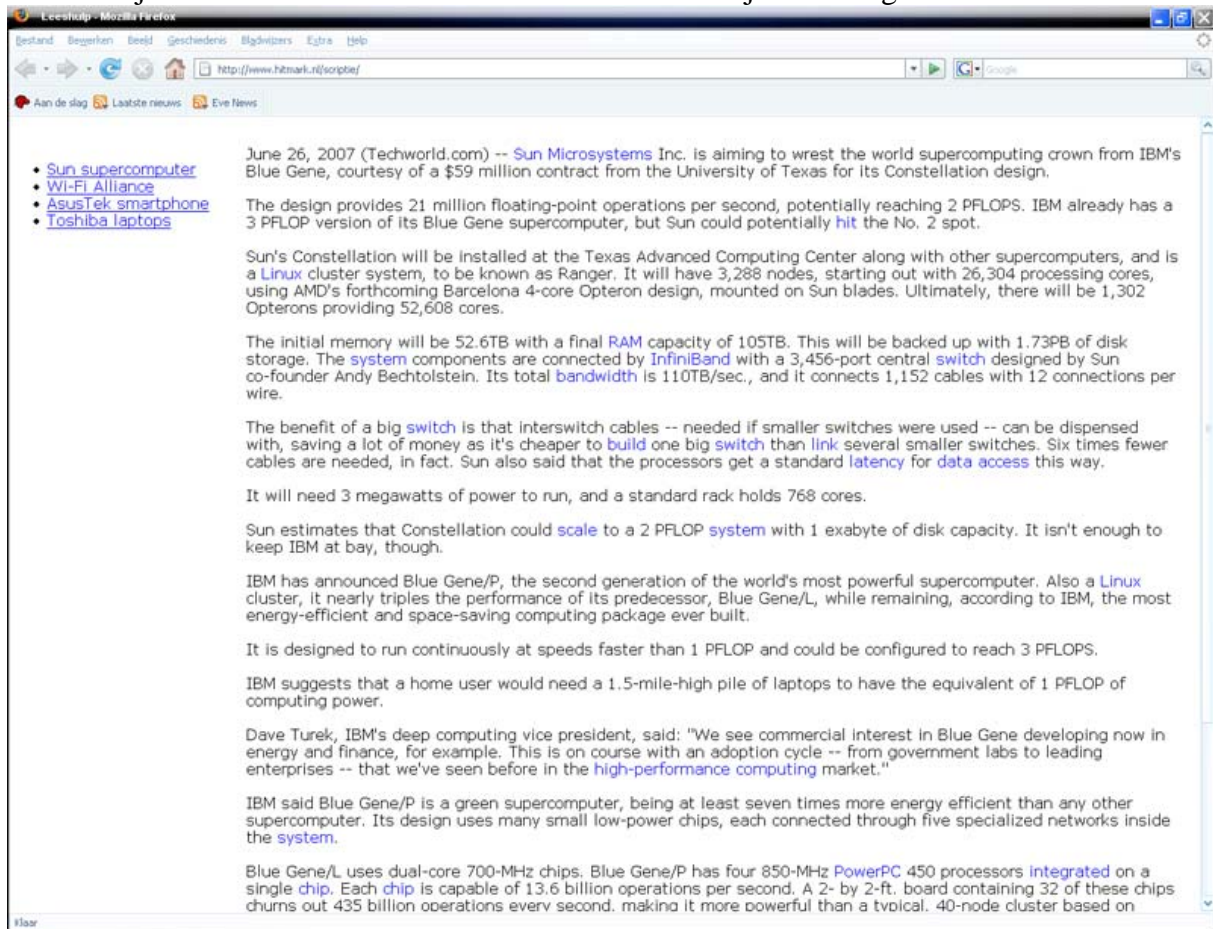
Screenshot 1

Het systeem kan op simpele wijze op een website worden geïntegreerd. Dit is een screenshot van de demoversie. Aan de linkerkant bevinden zich hyperlinks naar enkele voorbeeldteksten.



Screenshot 2

Wanneer op een link geklikt wordt, gaat het systeem de tekst parsen met behulp van de woordenlijst. Het kan even duren voordat de tekst verschijnt met de gevonden woorden.



Screenshot 3

Als vervolgens op een woord geklikt wordt, verschijnt er een popup met het resultaat. In dit geval is er geklikt op RAM.

