Vrije Universiteit Brussel

Faculteit Wetenschappen
Departement Computerwetenschappen

Gevoelsanalyse in het Nederlands

Yannick Merckx

Bachelorproef

Rolnummer: 500294

Promotor: Yann-Michaël De Hauwere
Begeleiders: Maarten Deville
Peter Vrancx

Juni 2015

Samenvatting

Dank woord

Inhoud

Hoofdstuk 1

Introductie

Vandaag de dag beschikken we over een enorme hoeveelheid aan digitale informatie. Ook wordt deze hoeveelheid aan informatie iedere dag groter en groter. In deze *"Age of Big Data"* (Lohr (2012)) bestaat de uitdaging erin om uit deze grote hoeveelheid data, door middel van analyse bepaalde inzichten te krijgen.
Vele hebben dit probleem proberen aan te pakken, waarbij men zich vooral bezig hield met het brengen van structuur in deze grote dataset en zich voornamelijk concentreerde op onderwerpgebaseerde classificatie. Echter met de opkomst van sociale media, blogs, reviewsites is er een groeiende interesse ontstaan voor gevoelsanalyse. Het onderzoek dat hiernaar gebeurd, heeft ook de nodige aandacht van bedrijven en ook zij spelen hier een rol in.
Als we over gevoelsanalyse spreken dan refereren we naar het verwerken van natuurlijke taal om zo via tekstanalyse en Computationele taalkunde subjectieve informatie uit te tekst te kunnen halen. Volgend voorbeeld illustreert een tweet waarop men bijvoorbeeld gevoelsanalyse kan uitvoeren.

Figuur 1.1: een voorbeeldtweet voor gevoelsanalyse

Verschillende technieken zijn hier mogelijk als gevoelsanalyse om de subjectiviteit of opinie uit deze tweet te bepalen. Men kan zich baseren op het woord ślecht' en zo vast stellen dat de tweet een negatieve emotie voorstelt. Maar men kan zich ook baseren op eerder vastgestelde tweets en op basis hiervan een beslissing nemen. Echter stel dat de gegeven persoon juist de lotto had gewonnen en zich sarcastisch uitdrukte. Dit zijn problemen waar men vandaag de dag nog altijd niet uit is met gevoelsanalyse.

Nu bijna al het onderzoek dat de afgelopen jaren gebeurd is met betrekking tot gevoelsanalyse werd uitgevoerd op Engelse teksten en daarom is er zeer weinig te vinden over onderzoek met betrekking tot gevoelsanalyse op het Nederlands. Deels omdat het Engels een wereldtaal is en

het Nederlands niet, maar ook doordat de bedrijven die dergelijk onderzoek uitvoeren op het Nederlands hun onderzoek (resultaten) binnenshuis houden.
Voor deze bachelorproef kijken we of de bevindingen over gevoelsanalyse op het Engels gelijk toepasbaar zijn op het Nederlands en/of er verschillen zijn tussen het Nederlands en het Engels, waarbij men met een gevoelsanalyse rekening mee dient te houden.
Als eerste hebben we samen met de voorbereiding van de bachelorproef in het 1ste semester ons verder verdiept in de literatuur over Engelse gevoelsanalyse en maakte ons vertrouwd met de programmeertaal python, een van de programmeertalen bij uitstek die ons toelaat om experimentele analyses op te zetten. Vervolgens zijn we op zoek gegaan naar onze data. Voor het de gevoelsanalyse te kunnen vergelijken, moet men zowel over een Engelse dataset als een Nederlandse dataset beschikken. Door het vele onderzoek naar Engelse gevoelsanalyse zijn er voldoende dataset beschikbaar op het web. Echter Nederlandse datasets, gelabeld volgens gevoel, zijn heel moeilijk te vinden en dwingt ons om de data manueel te scrapen. Voor het scrapen moeten we zoals eerder vermelde opletten voor sarcasme. Dit sluit scrapingbronnen zoals Twitter en andere sociale media volledig uit. Gelukkig bieden reviewsites als `www.moviemeter.be` de oplossing en gebruiken we deze sites als bron voor te scrapen. In sectie 3.1 gaan we verder in op de scraping en verzamelde dataset. Na dat we de nodige kennis, vaardigheden en datasets hebben, bepalen we de technieken die we gebruiken voor de gevoelsanalyse en de vergelijking in onze experimentele analyse. Initiëel om goed te doorgronden wat er juist gebeurd tijdens deze technieken, implementeren we deze technieken zelf voor de experimentele analyse. Later bij het dooranalyseren, optimaliseren we de code door gebruik te maken van de bibliotheek sklearn (`http://scikit-learn.org/`), die ons toelaat om sneller een experimentele analyse uit te voeren.
Samengevat bespreken we in hoofdstuk 2 de technieken die we gebruiken voor de experimentele analyse met hun theoretische achtergrond.
Vervolgens gaan we in hoofdstuk 3 over naar de experimentele analyse.
In deze analyse vergelijken de resultaten van de technieken op Nederlandse en Engelse filmrecensies. Vervolgens concentreren we ons op de woordenschat en onderzoeken we de classificatie op basis van een Engels en Nederlands geannoteerde woordenlijst van gevoelens.
Afhankelijk van het positieve karakter van de resultaten gaan we nog iets dieper in Nederlandse gevoelsanalyse.
Na het experimentele analyse vormen we een conclusie in hoofdstuk 4 of de gevonden technieken voor gevoelsanalyse op het Engels al dan niet gelijk toepasbaar zijn op het Nederlands.

Hoofdstuk 2

Bouwstenen tekstanalyse

2.1 Overzicht

Voordat we aan onze gevoelsanalyse kunnen beginnen moeten er een paar belangrijke stappen doorlopen worden om tot de experimentele analyse te komen. Voor dit onderzoek moeten er drie belangrijke stappen doorlopen worden namelijke: De data, voorverwerking en het aanleren van het concept.

Als eerste stap *de data* lijkt misschien triviaal, maar dit is zeker niet het geval. Om een gevoelsanalyse uit te voeren moet men eerst over data beschikken, die bovendien juist gelabeld is volgens de subjectiviteit die men wil afleiden met de gevoelsanalyse. Meer over hoe we juist deze data verzameld voor dit onderzoek, vindt men in 3.1. Niet alleen het beschikken van de dataset is belangrijk, maar ook het juist en efficiënt voorstellen van de datasets is belangrijk om bij stil te staan. Meer hierover vindt men in de volgende sectie.

De volgende stap *voorverwerken van data* is eveneens belangrijk in het proces van gevoelsanalyse en wordt beschouwd in sectie 2.3. De sectie omvat technieken die we gebruiken tijdens de experimentele analyse in hoofdstuk 3 om onze data te optimaliseren en zo de classificatieprestatie te verbeteren.

Als laatste stap bepalen we het algoritme dat we gebruiken. Voor dit onderzoek is er gekozen voor zelflerende algoritmes die we een bepaald concept gaan aanleren, in het geval voor de experimentele analyse duidt dit op het onderscheiden van negatieve en positieve recensies. Meer over de leermethode vindt men in sectie 2.4.

Figuur 2.1: De drie bouwstenen voor tekstanalyse

2.2 Voorstelling dataset

De voorstelling van de data is een eerste element van het experiment waarmee men rekening moet houden. We kunnen bijvoorbeeld rauwe data meegeven aan het zelflerende algoritme of we kunnen de tekst omvormen naar een vector die het aantal voorkomens van ieder woord in de tekst bevat. Voor het experiment kiezen we het tweede voorbeeld, waarbij we een document voorstellen als een vector met daarin de woordfrequentie. Dit wordt de vector space methode genoemd en wordt door Turney et al. (2010) beschouwd als onderdeel van de oplossing voor de problematiek rond semantische analyse. Verder is deze voorstelling een populaire methode binnen het onderzoek naar gevoelsanalyse op het Engels en heeft dit zijn werking al aangetoond. Zie bijvoorbeeld Pang et al. (2002) en Maas et al. (2011a).

2.2.1 Vector Space Methode

De vector space methode (VSM) is een methode waarbij we een document als een vector voorstellen waarbij ieder element overeenkomt met een woord en zijn frequentie in het document. De elementen van de vector worden ook wel features genoemd. Als men concreet een document voorstelt kan men zeggen dat document j voorgesteld wordt door d_j met f_{ij} de frequentie van het woord w_i. Met de frequentie f_{ij} bedoelt men het totaal aantal voorkomens van het woord w_i in document j. Het aantal verschillende woorden in het document stelt men voor door n_w, wat eveneens de dimensie is van de vector. Het document j kan dus als volgt worden voorgesteld:

$$d_j = \begin{bmatrix} f_{1j} \\ f_{2j} \\ \vdots \\ f_{n_w j} \end{bmatrix}$$

Een belangrijk inzicht bij de vector space methode is dat een document voorgesteld wordt als een groep van woorden. Er wordt geen rekening gehouden met de volgorde waarin de woorden in het document voorkomen. Vaak ziet men ook dat de vector vaak ijl is en vanwege de grote hoeveelheid aan woorden in een document heel groot. Als we nu niet één document, maar meerdere documenten nemen en we zeggen dat het aantal documenten gelijk is aan n_d, resulteert dit in een matrix waarbij iedere kolom een document voorstelt.

$$\begin{matrix} \text{Documenten} \end{matrix}$$
$$D = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n_d} \\ f_{21} & f_{22} & \cdots & f_{2n_d} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n_w j} & f_{n_w 2} & \cdots & f_{n_w n_d} \end{bmatrix} Woorden$$

Deze matrix wordt een ***terms-documents matrix (TDM)*** genoemd. Wanneer men spreekt van een ***documents-terms matrix (DTM)***, spreekt men een getransponeerde terms-documents matrix. Een rij van een DTM stelt dan een document voor. In het experiment stellen we onze data voor aan de hand van een documents-terms matrix De voorstelling in een matrix geeft inzicht en biedt veel meer mogelijkheden om de data te analyseren. Bijvoorbeeld overeenkomstige woordfrequenties tussen twee documenten kan duiden dat documenten over hetzelfde onderwerp gaan of eenzelfde mening uitdrukken. In de praktijk is gebleken dat documenten vergelijken op basis van woordfrequentie niet altijd de gewenste resultaten oplevert en Pang et al. (2002) toont aan dat er ruimte is voor verbetering door middel van Pre-Processing technieken.

2.3 Technieken voor Pre-Processing

Zoals we in 2.2.1 al vermeldde kan pre-processing voor verbetering van de classifiers zorgen. De pre-pocestechnieken die we gebruiken in deze bachelorproef zijn al eerder gebruikt door Pang et al. (2002) en Wang & Wan (2011) en hadden een positief effect.

2.3.1 Bag of Words

De eerste techniek is Bag of Words. Dit is niet echt een pre-procestechniek, maar eerder een referentiepunt voor de andere pre-procestechnieken. Het steunt op het principe waarop de vector space methode zich baseert, waarbij ieder document wordt voorgesteld door zijn woordfrequenties. Het is de basistechniek die wordt uitgevoerd bij een gevoelsanalyse aan de hand van de VSM.

2.3.2 Verwijderen van stopwoorden

Wat men vaak ziet in het Nederlands, maar ook in taal algemeen, is dat er veel stopwoorden worden gebruikt. Stopwoorden als "klopt" en "eigenlijk" zeggen niet veel over teksten of ze nu positief of negatief zijn. Als een bepaald woord niet bijdraagt voor het algoritme kunnen we stopwoorden beschouwen als ruis in de dataset. Ruis vertroebelt het beeld van het concept dat we het algoritme willen aanleren en proberen we te elimineren. Daarom beschouwt men het verwijderen van stopwoorden en leestekens ook als een manier van pre-processing.
Onderstaande tabel geeft het gemiddeld aantal features weer voor dertig Engelse en Nederlandse datasets met 6000 recensies, wanneer wel of geen stopwoorden zijn verwijderd. Later in hoofdstuk 3.2 bekijken de invloed van deze techniek op de classificatieprestatie.

	Bag of Words	Verwijderen van Stopwoorden
Engels	39716	39593
Nederlands	32768	32668

Tabel 2.1: Gemiddeld aantal features bij Bag of Words en het Verwijderen van stopwoorden (op basis van 6000 samples/dataset en 30 datasets voor iedere taal)

2.3.3 Term weighting

Als we terugkijken naar de vector space methode, waarbij we enkel rekening houden met de woordfrequentie, kan men zeggen dat niet elk woord evenveel doorweegt. Een woord dat in alle documenten voorkomt biedt geen of minder waardevolle informatie, dan een woord dat zelden voorkomt. En hierop baseert term weighting zich. Het gaat een wegingsfactor introduceren. Ieder woord krijgt een gewicht toegewezen, dat weergeeft hoe belangrijk het woord is. Neem als voorbeeld een hoop recensies van de film "Pulp Fiction" en de woorden "Pulp" en "excellent". "Pulp" is een woord dat voorkomt in de titel van de film en komt ongetwijfeld in elke recensie voor. "Excellent" daarentegen is een woord dat enkel maar voorkomt wanneer de recensent de film fantastisch vond, het zal niet in elk document voorkomen en is waardevolle informatie. Term weighting zal dus bij dit voorbeeld "excellent" een groter gewicht toewijzen dan "Pulp". De kwantiteit van dit gewicht wordt vaak de **inverse document frequency (idf)** genoemd en wordt bepaald aan de hand van volgende formule:

$$w_i : idf_i = -log_2[P(w_i)]$$

met $P(w_i)$ de priori probability dat woord w_i voorkomt in het document.
De inverse document frequency geeft het algemeen belang van het woord w_i weer. Men kan dit benaderen door het logaritme te nemen van het aantal documenten waar w_i in voorkomt en het totaal aantal documenten. Een andere nuttige kwantiteit is de **term frequency** tf_{ij}. Deze geeft het belang weer van het woord w_i binnen in het document d_j en wordt als volgt genoteerd:

$$tf_{ij} = \frac{f_{ij}}{\sum_{i=1}^{n_w} f_{ij}}$$

tf_{ij} wordt berekend door de frequentie, het aantal voorkomens, van een woord w_i in document d_j te delen door de som van alle woordfrequenties in document d_j. Met deze twee kwantiteiten kan men een nieuwe begrip introduceren: de **tf-idf score**. Wat overeenkomt met het product van tf en idf.

$$\text{tf-idf score} = tf.idf_{ij} = idf_i.tf_{ij}$$

De tf-idf matrix bekomt men dan door alle woordfrequenties van het terms-document matrix te vervangen door de tf-idf score. Er bestaan nog uitbreiding op term weighting (**?**), maar voor het experimentele analyse houden we het bij de standaard tf-idf weighting.
Als kleine aanloop naar de experimentele analyse zien we met een eenvoudige proefje de kracht van tf-idf. Onderstaande tabel geeft enkele Nederlandse recensies weer met het woord of feature met de hoogste tf-idf voor die recensie na term weighting. De volledige recensies kan je lezen in bijlage X.

Tabel 2.2: My caption

Recensie	**Woord met hoogste tf-idf score**
in het begin dacht ik echt van.. wat is dit nou weer.. maar je wil hem PERSE afzien! kei mooie film!aanrader!	Kei
Geweldig verhaal! Aangrijpend. Ik heb deze film een stuk of 4 keer gezien, blijft indrukwekkend.****-sterren	Aangrijpend
Wat een geweldige film. De trilogy al 100 x gezien en het blijft goed. Ook vooral als je een de filosofie	Je
Coole film!	Coole
Voor mij een 5 sterren film. Ik had hem op dezelfde manier gemaakt. Inspirerende film	Inspirerende
Geweldige nagelbijtende oorlogsfilm,zo zie je ze jammer genoeg zelden,zien !!!!	nagelbijtende
Een prachtige en zeer meeslepende film. Het blijft ook erg boeiend, omdat ...	het
Ik vind dat dit een overgewardeerde film is, net als een heleboel andere Nederlandse films op deze site.	overgewardeerde

We zien hier al mooi resultaten zoals *nagelbijtende* en *overgewardeerde*, waar duidelijk de positieve aard van de recensie naar boven komt. Opvallend is dat bij vooral langere recensies nogal redelijk veel stopwoorden zoals *je* of *het* de hoogste tf-idf-score krijgen toegewezen. Het nog eens verwijderen van de stopwoorden voor de term weighting gaat mogelijks dit probleem verhelpen. Dit onderzoeken we verder tijdens de experimentele analyse in 3.2.

2.3.4 Bigram Collocaties

Bigrams Collocaties is een techniek waarbij men op zoek gaat naar paren van woorden die een hoge waarschijnlijkheid hebben om samen voor te komen en een extra bron van informatie kunnen vormen. In het onderzoek van Pang et al. (2002) bleken bigrams niet voor een verbeterde prestatie te zorgen, al mag men de nuttigheid van bigrams niet onderschatten. Toch nemen we bigrams als een van de technieken,**?** toonde echter aan dat bigrams een nuttig kenmerk vormen voor het oplossen van woord zin ambiguïteit. Pang et al. (2002) merkt dan ook zelf op in zijn onderzoek dat bigram features mogelijks evenwaardig zijn met unigram features.
De bepaling van de informatieve waarde van de bigrams is gebaseerd op de frequentie van het bigram en de frequenties van de andere bigrams. Als men een overzicht krijgt over de frequenties introduceert men een metriek, die met behulp van de frequenties mogelijke verbanden kan blootleggen. Chi-kwadraat is zo'n metriek die er zich toe leent. De Chi-kwadraattoets is een statistische toets die het mogelijk maakt om de onafhankelijkheid tussen waarnemingen te onderzoeken. Bij Bigram Collocaties onderzoekt men via de Chi-kwadraattoets de afhankelijkheid tussen twee woorden. Hoe grotere de afhankelijkheid, hoe hoger de score.
Om een idee te krijgen hoeveel features er juist worden toegevoegd wanneer we gebruik maken van bigrams, hebben we onderstaande tabel opgesteld. De tabel geeft het aantal features weer wanneer men wel of geen bigrams gebruikt. De cijfers zijn gebaseerd op een gemiddelde van 30 datasets met 6000 recensies per dataset.

	Bag of Words	Bigrams
Engels	39716	493633
Nederlands	32727	270764

Tabel 2.3: Gemiddeld aantal features bij Bag of Words en Bigrams (Gemiddelde van 30 dataset met 6000 recensies/dataset)

In de tabel zien we duidelijk dat er een stevig aantal features wordt wanneer men bigrams mee in rekening brengt. Bij het Engels stijgt het aantal features met 1240% en bij het Nederlands met 820%, wat een enorme bron aan extra informatie voor het leeralgoritme kan betekenen. Of het een verbetering is voor gevoelsanalyse onderzoeken we verder tijdens de experimentele analyse in 3.2.

Chi-Kwadraattoets

De Chi-Kwadraattoets is een techniek uit de statistiek die gebruikt kan worden als een onafhankelijkheidstoets voor waarnemingen. De reden waarom we deze toets voor Bigram collocatie gebruiken is dat de toets parametervrij is. Wat wil zeggen dat er bij de start van de chi-kwadraattoets geen aanname over de populatie of het gemiddelde wordt verwacht. In deze sectie leggen we aan de hand van een voorbeeld uit hoe de chi-kwadraattoets juist deze afhankelijkheid bepaald.
Neem als voorbeeld het bigram *(heel , goed)*. Zoals bij iedere statistische test neemt men eerst een nulhypothese aan. Voor de chi-kwadraattoets is dit ook het geval. De toets neemt als nulhypothese aan dat beide woorden onafhankelijk van elkaar zijn en elkaars voorkomen niet beïnvloeden. Men vergelijkt de waargenomen frequenties van de woorden met de verwachte frequenties wanneer de woorden onafhankelijk zouden zijn. Als deze waarden te veel verschillen kan men de nulhypothese verwerpen en de alternatieve hypothese aannemen, namelijk dat de woorden afhankelijk zijn van elkaar.

Om de afhankelijkheid van woorden te bepalen, kijken we naar volgende gegevens:

- het aantal voorkomens van het woord in een bigram
- het aantal voorkomens van het woord in een bigram met het ander woord waar we de afhankelijkheid van onderzoeken
- het totaal aantal bigrams
- het aantal voorkomens van het ander woord in een bigram.

Als we voor het voorbeeld *(heel , goed)* bovenstaande gegevens in een kruistabel gieten krijgen we de volgende 2x2 tabel:

	w1= heel	w1 ≠ heel
w2 = goed	9 (heel goed)	7893 (bv. niet goed)
w2 ≠ goed	3632 (bv. heel slecht)	13498000 (bv. boeiende thesis)

We weten nu naar wat we moeten kijken bij het analyseren van de afhankelijkheid maar er mist nog een weging, een onderlinge verhouding tussen de kenmerken. De Chi-Kwadraatsom biedt hier de oplossingen en geeft die weging. De toetsingsgrootheid voor de Chi-kwadraattoets wordt gedefinieerd aan de hand van de volgende formule:

$$\chi^2 = \sum_{i,j} \frac{(O_i j - E_i j)^2}{E_i j}$$

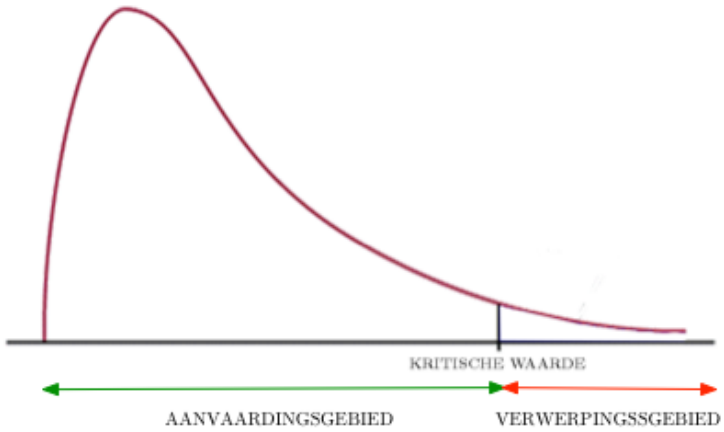
Waarbij $O_i j$ het aantal keer dat het paar (i, j) voorkomt. $E_i j$ stelt de voorspelde waarden voor als de woorden onafhankelijk moesten voorkomen
$E_i j$ wordt bepaald door volgende formule:

$$E_i j = \frac{O_i *}{N} + \frac{O_* j}{N} * N = \frac{O_i * O_* j}{N}$$

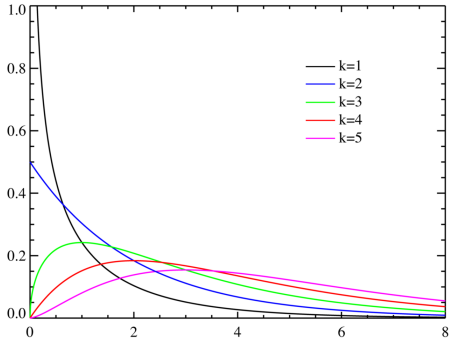
met $\frac{O_i *}{N}$ de marginale probabiliteit dat i als eerste deel van het bigram voorkomt en $\frac{O_j *}{N}$ de marginale probabiliteit dat j als tweede deel van het bigram voorkomt. N stelt het totaal aantal bigrams voor. Toegepast op het voorbeeld geeft dit voor het bigram "(heel , goed)":

$$E_1 1 = \frac{9 + 3632}{N} + \frac{9 + 7893}{N} * N \approx 0,0085$$

Als laatste onderdeel berekenen we de χ^2-score, bepalen we het aantal vrijheidsgraden en zoeken we de χ^2 distributie op met de berekende vrijheidsgraad. Stel dat het vooropgestelde betrouwbaarheidsinterval 95% bedraagt dan kunnen we de kritische waarde bepalen voor significantielevel α = 0,005. Als de berekende χ^2-score in het verwerpingsgebied ligt, kan de nulhypothese verworpen worden en kan het bigram beschouwd worden als afhankelijk.
Onderstaande afbeelding illustreert hoe de verwerping of aanvaarding van een nulhypothese juist in zijn werking gaat
Kort samengevat baseert de Chi-kwadraattoets zich op de afwijking tussen de geobserveerde frequentie en de verwachte frequentie. Hoe groter het verschil, hoe waarschijnlijker men de nulhypothese kan verwerpen. En dit is waar men zich bij Bigram Collocatie op gaat baseren.



Figuur 2.2: Illustratie eenzijdige-toets van een χ^2-distributie (Originele afbeelding: http://www.philender.com/courses/intro/notes3/xdist.gif)



Figuur 2.3: Chi-square distributies met K vrijheidsgraden (Bron: `http://upload.wikimedia.org/wikipedia/commons/2/21/Chi-square_distributionPDF.png`)

2.3.5 Best feature selection

Als we duizenden documenten verwerken, is het te voorspellen dat er enorm veel woorden algemeen voorkomen in de documenten, maar niet veel informatie bijdragen over het document zelf. Het is sterk vergelijkbaar met de voorgaande techniek in 2.3.2 bij het verwijderen van stopwoorden. Veel voorkomende features kunnen voor het document niet als iets identificerend dienen en zorgen voor ruis in de dataset. Daarom kan men verkiezen om deze low-information features te verwijderen zodanig dat men enkel de features overhoudt die echt iets zeggen over een document. Het bepalen van de informatiewinst kan gebeuren aan de hand van het aantal voorkomens in de verschillende klassen. Als een bepaalde feature voornamelijk in positieve documenten voorkomt en amper in negatieve documenten, kan men afleiden dat deze feature zeer informatief is omtrent positieve documenten. Als metriek om de informatiewinst te meten kan men wederom χ^2 uit 2.3.4 gebruiken. Chi-kwadraat laat ons namelijk toe om de correlatie tussen een bepaalde feature en de klassen te meten.

2.3.6 Latent Semantic Analysis

Latent Semantic Analysis is een wiskundige techniek gebaseerd op statistische berekeningen, waar van aangetoond dat deze zeer nuttig is bij het analyseren van grote collecties tekstdata (Furnas et al. (1988)). Met LSA probeert men een notie te krijgen van de semantische informatie en meer bepaald het semantisch verband tussen woorden. Bijvoorbeeld als we zoeken naar documenten met het woord "economie", willen we ook documenten met "financiën" terugkrijgen. Voor LSA zijn twee woorden semantisch gerelateerd als ze gebruikt worden in dezelfde context. Met het concrete voorbeeld kunnen we zeggen dat er een semantisch verband is tussen twee woorden als ze vaak voorkomen in dezelfde documenten.
Merk op dat bij Latent Semantic Analysis het belangrijk is dat ieder woord naar één concept verwijst.
Analytisch wordt LSA toegepast door **Singular Value Decomposition (SVD)** toe te passen op de terms-documents matrix. SVD is een concept uit de lineaire algebra en zegt dat een matrix A opgesplitst kan worden als een product van matrixen namelijk

$$A = U\Sigma V^T$$

De reductie van de dimensie gebeurt aan de hand van volgend principe. Neem matrix A met rang r.

$$A = U\Sigma V^T = \underbrace{\begin{bmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_r\end{bmatrix}}_{\text{Kolommen } A} \underbrace{\begin{bmatrix}\mathbf{u}_{r+1} & \dots & \mathbf{u}_m\end{bmatrix}}_{\text{Nul } A^T} \begin{bmatrix}\sigma_1 & 0 & \dots & 0 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & \dots & \sigma_r & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 & 0 & \dots & 0 \\ \dots & & & & & & \\ 0 & 0 & \dots & 0 & 0 & \dots & 0\end{bmatrix} \begin{bmatrix}\mathbf{v}_1^T \\ \mathbf{v}_2^T \\ \dots \\ \mathbf{v}_r^T \\ \mathbf{v}_{r+1}^T \\ \dots \\ \mathbf{v}_n^T\end{bmatrix} \begin{matrix}\left.\vphantom{\begin{matrix}a\\b\\c\\d\end{matrix}}\right\} \text{Rijen } A \\ \left.\vphantom{\begin{matrix}a\\b\\c\end{matrix}}\right\} \text{Nul } A\end{matrix}$$

U is de unitaire matrix waarbij men $u_1, u_2, ..., u_r$ de linker singuliere vectors noemt. Deze stellen een document met zijn features voor. V^T is de geconjugeerde getransponeerde matrix van V. $v_1, v_2, ..., v_r$ noemt men de rechter singuliere vectors en stellen de woorden met hun features over alle documenten voor. Σ is een diagonaal matrix met singuliere waarden $\sigma_1, \sigma_2, .., \sigma_r$' op de diagonaal. De reductie van een terms-documents matrix naar een dimensie van K gebeurt door de hoogste K singuliere waarden te nemen in Σ met de overeenkomstige singuliere vectoren uit U en V. Doordat men de dimensionaliteit van de vectoren kan beperken door semantisch gelijkaardige woorden bijeen te voegen. Dit laat toe om een soort van context groepen te creëren en zo een zeker inzicht te krijgen in de dataset. Het is dan ook gebleken dat SVD toepassen een zeer nuttige eerste stap is bij text mining (**?**), omdat men nieuwe meer efficiënte features krijgt. De nieuwe features geven meer duidelijkheid en inzicht en kunnen dienen als input voor het zelflerende algoritme.

2.4 Leermethode

Voor het experiment hebben moeten we ook het algoritme bepalen dat de data gaat classificeren, ook wel classifier genoemd. Voor het algoritme gaan we beroep doen op de Machine learning en gebruik maken van supervised learning technieken. Deze technieken vereiseen dat men het algoritme eerst traint met een dataset die voorbeelden bevat over het concept dat we willen aanleren. De trainingsset bevat zowel de inputwaarden als de verwachte outputwaarde voor de input en men verwacht dat het algoritme hier verbanden in kan vinden zodanig dat het voor

willekeurige inputwaarden de juiste outputwaarde kan bepalen. Ye et al. (2009) toont echter aan dat supervised learning technieken een goede prestatie hebben bij gevoelsanalyse, terwijl dit niet het geval is bij unsupervised learning (Rothfels & Tibshirani (2010)).
Concreter kiezen we voor de Naive Bayes Classifier en de Decision Tree als supervised learning technieken voor het experiment. De Naive Bayes Classifier is een heel praktische aanpak voor bepaalde leerproblemen (Mitchell (1997)). Bijvoorbeeld onderzoekers Michie et al. (1994) tonen aan dat de prestatie van de Naive Bayes Classifier gelijkaardig of in sommige gevallen zelfs beter is dan andere leeralgoritmen, zoals beslissingsbomen en neurale netwerken onderzocht. Decision Trees zijn eveneens een populaire methode en werd ondermeer gebruikt door Zhang et al. (2008) voor een gevoelsanalyse op productrecensies en klanten feedback.

2.4.1 Naive Bayes Classifier

De Naive Bayes Classifier is gebaseerd op Bayesiaans redeneren. Bayesiaans redeneren is een aanpak die gevolgen trekt op basis van probabiliteit. Het is gebaseerd op de veronderstelling dat bepaalde hoeveelheden die ons interesseren probabilistisch verdeeld zijn en door te redeneren over die probabiliteit samen met de trainingsdata er optimale beslissingen kunnen genomen worden.

De werking van de Naive Bayes Classifier is volledig gebaseerd op probabiliteit. Neem als inputwaarden $x_1, x_2, x_3, ..., x_n$ en als de te voorspellen outputwaarde y_{res}. Nu moet de classifier voor de inputwaarden $x_1, x_2, x_3, ..., x_n$ de correct y_{res} voorspellen. Volgens het Bayesiaans redenering is, gebaseerd op $x_1, x_2, x_3, ..., x_n$, y_{res} de outputwaarde met de grootste waarschijnlijkheid. We kunnen dit neerschrijven als:

$$y_{res} = \arg\max_{y_i \in Y} P(y_i | x_1, x_2, x_3, ..., x_n)$$

Aan de hand van het Bayes theorema kunnen we dit herschrijven als

$$y_{res} = \arg\max_{y_i \in Y} \frac{P(x_1, x_2, x_3, ..., x_n | y_i) P(y_i)}{P(x_1, x_2, x_3, ..., x_n)}$$

Merk op $P(x_1, x_2, x_3, ..., x_n)$ is gelijk aan 1, aangezien dit gegeven is dus

$$y_{res} = \arg\max_{y_i \in Y} P(x_1, x_2, x_3, ..., x_n | y_i) P(y_i)$$

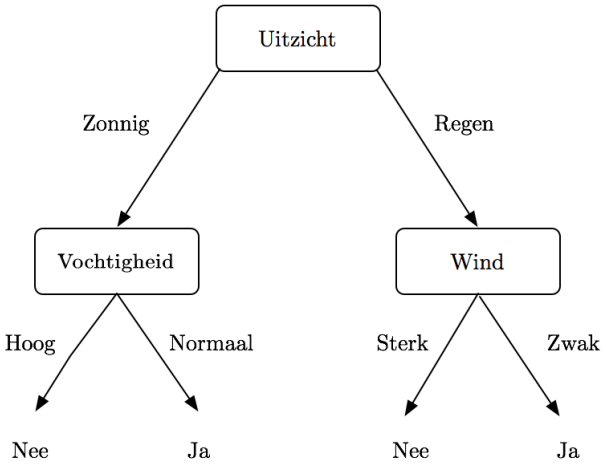
De twee componenten kunnen bepaald worden aan de hand van de trainingsset. $P(y_i)$ kunnen we bepalen door het aantal voorkomens van y_i in de trainingsset te tellen. $P(x_1, x_2, x_3, ..., x_n | y_i)$ is moeilijker af te leiden aan de hand van de trainingsset aangezien we meerdere voorkomens van $x_1, x_2, x_3, ..., x_n$ naar y_i moeten hebben om een goede schatting te kunnen maken. Indien we een heel grote trainingsset hebben is dit mogelijk, anders niet. Om dit toch te kunnen afleiden, gaat de Naive Bayes Classifier er van uit dat elke x_i uit $x_1, x_2, x_3, ..., x_n$ onafhankelijk is ten opzichte van de outputwaarde y_i. Wat betekent dat we het product van iedere probabiliteit kunnen nemen en $P(x_1, x_2, x_3, ..., x_n | y_i)$ kunnen herschrijven als $\prod_i P(x_i | y_i)$.
Voor het maken van voorspelling maakt het gebruik van probabiliteit, gebaseerde op de trainingsset en waar het aanneemt dat ieder feature onafhankelijk is tot de outputwaarde. Samengevat kunnen we dit schrijven als

$$y_{NBres} = \arg\max_{y_i \in Y} P(y_i) \prod_i P(x_i | y_i)$$

Ten slotte stellen we de verzameling van al deze probabiliteiten samen als de hypothese van de Naive Bayes Classifier.

2.4.2 Decision Tree

Decision Trees of Beslissingsbomen zijn een van de meest gebruikte en praktische methode voor inductieve gevolgtrekking (Mitchell (1997)). De methode is robust met ruis op de data en houdt rekening met discrete klassen. De classifier gaat een beslissingsboom proberen op te stellen aan de hand van de trainingsdata. Na de training krijgt men een beslissingsboom die de hypothese moet voorstellen. Wanneer het getrainde algoritme onbekende data krijgt, gaat het inductief de output bepalen voor de inputwaarden. Men kan een beslissingsboom voorstellen als een disjuncte set van als-dan regels.
Onderstaande afbeelding is een voorbeeld van zo'n beslissingsboom die bepaald of het weer goed genoeg is om basketbal buiten te spelen. De bladeren van de boom stellen de verschillende outputwaarden voor. In dit geval zien we dat er een boom is opgesteld voor twee discrete klassen namelijk ja en nee. In de nodes staan testen beschreven die de het pad van de inputwaarden naar de outputwaarde bepalen. Merk op dat de bepaling altijd top-down gebeurd.



Figuur 2.4: Voorbeeld van een beslissingsboom

Hoofdstuk 3

Experimentele analyse

We willen voor dit onderzoek achterhalen of er verschillen zijn tussen Engelse en Nederlandse gevoelsanalyse. Hiervoor moeten we data verzamelen, data voorverwerken en vervolgens de data gebruiken om een concept aan te leren aan een zelflerende algoritme. In hoofdstuk 2 hebben we al deze bouwstenen in meer detail bekeken om nu tot de experimentele analyse te komen. In sectie 2.2 hadden we al de voorstelling van de dataset bepaald, Nu in sectie 3.1 gaan we dieper in op het verzamelen van de data voor deze experimentele analyse. Uiteindelijk is het beschikken over een goede dataset even belangrijk als het beschikken over goede technieken en evengoed een onderdeel van de experimentele analyse.
Deze analyse is opgedeeld in verschillende subanalyses om zo een optimaal beeld te krijgen over al dan niet de verschillen tussen Nederlandse en Engelse Gevoelsanalyse.
Voor de gevoelsanalyse gebruiken we de voorverwerkingstechnieken uit sectie 2.3 en de leermethoden uit sectie 2.4 en proberen we een onderscheid te maken tussen recensies met een negatieve of positieve opinie. De prestaties van de gevoelsanalyses beoordelen we in dit onderzoek op basis van de precisie waarmee negatieve en positieve recensies worden onderscheiden.

Als eerste analyse beginnen we in 3.2 met de verschillen te bekijken tussen Engelse en Nederlandse gevoelsanalyse op basis van de prestatie. Nadat we deze analyse hebben uitgevoerd, bekijken we in sectie 3.3 gevoelsanalyse met een andere eenvoudige en intuïtievere kijk en analyseren we de classificatie op basis van geannoteerde woordenlijsten met gevoelens.
Afhankelijk van het positief karakter van de voorgaande analyses gaan we nog iets dieper in op het Nederlands in sectie 3.4 en analyseren we de onderwerp-gevoeligheid van Nederlandse gevoelsanalyse.

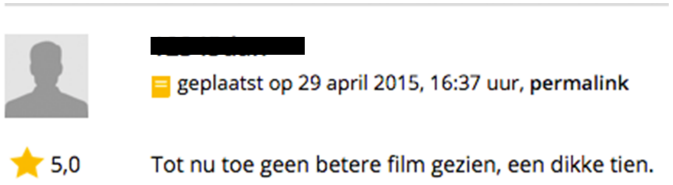
3.1 De Dataset

Het verzamelen van data lijkt misschien een triviaal onderdeel van heel de experimente analyse, maar dit is zeker niet het geval. Er moet heel verstandig en kritisch omgegaan worden bij het verzamelen van data voor gevoelsanalyse. Een eerste punt is sarcasme. Sarcasme is vandaag de dag nog altijd een onopgelost probleem (Liebrecht et al. (2013)) en is iets waar we rekening mee moeten houden als we de bron voor de verzameling van onze data selecteren. Sociale media zoals Twitter en dergelijke kunnen we dus voor onze gevoelsanalyse niet gebruiken. Een andere probleem is het labelen van de data, omdat we voor deze experimentele analyse supervised learning technieken gebruiken, is het heel arbeidsintensief om de data manueel te labelen.Echter Reviewsites bieden hier de oplossing. Deze sites laten gebruikers toe om omtrend een bepaald

product een recensie te posten en hierbij ook een score mee te geven. Door die score kunnen we tijdens het verzamelen van de data, de recensies ook automatisch labelen.

Uiteraard zijn er enorm veel reviewsites beschikbaar en stuiten we hier op enkele problemen. Men moet rekening houden met het aanbod. Om een zo goed mogelijk beeld te krijgen willen we in onze datasets een algemeen onderwerp in te brengen. Dit wil zeggen dat we niet in het wilde weg recensies kunnen scrapen van iedere reviewsite dat we tegenkomen, maar selectief te werk moeten gaan. Als eerste ingeving gingen we de oplossing zoeken bij webshops zoals Coolblue `http://www.coolblue.be/`, Tweakers `http://tweakers.net/` en Amazon `http://www.amazon.com/`. Op deze website kan men een enorme hoeveelheid aan productrecensies, ideaal dus voor onze gevoelsanalyse. Het probleem echter bij deze websites is dat de reviews vaak te specifiek zijn en mogelijks de analyses kunnen beïnvloeden, door bijvoorbeeld een bepaald model van beamer meteen als doorweegfactor voor een positieve recensie te beschouwen.
VOORBEELD VAN ZO'N SPECIFIEKE REACTIE (MISSCHIEN VAN DE VERSCHILLENDE WEBSITES)
Uiteindelijk hebben we de oplossing gevonden bij film-, muziek- en boekrecensies. Er al veel onderzoek gedaan naar Engelse gevoelsanalyse en filmrecensies zijn hier een populaire dataset. Dit maakt het voor ons mogelijk om Engelse datasets over te nemen uit eerder onderzoek. De Engelse dataset die we gebruiken in dit onderzoek is afkomstig uit een eerder onderzoek door Maas et al. (2011b). Al deze gebruikersrecensies zijn toen gescraped geweest van de website imdb (`http://www.imdb.com/`) en zijn dus filmrecensies.
Voor de Nederlandse gevoelsanalyse waren er geen datasets beschikbaar en moeten we deze scrapen. De websites `moviemeter.nl`, `boekmeter.nl` en `muziekmeter.nl` vormen de perfecte bron aan informatie om te scrapen. Ze bevatten allemaal toplijsten met films, boeken of muziekalbums waarop in grote aantallen gebruikers hun persoonlijke mening plaatsen.
Belangrijk om te vermelden is dat zowel bij het labelen van de Engelse als de Nederlandse dataset dezelfde voorwaarden werd gerespecteerd. Enkel hoog gepolariseerde recensies worden beschouwd in de dataset. Onderzoek rond polarisatie classificatie (Maas et al. (2011b)) ondersteund deze keuze. Een recensie wordt negatief gelabeld als het een score heeft van 4 op 10 of minder. Een positieve labeling wordt gegeven aan recensies met een score van 6 op 10 of meer.

Later in 3.2 gaan we de grenskeuze nog beter analyseren, door de prestaties te vergelijken, wanneer er een hogere polarisatie wordt doorgevoerd.



Figuur 3.1: Een voorbeeld van een positieve commentaar op `moviemeter.nl`

Alle Nederlandse recensies zijn afkomstig van de "All Time Top 250"-toplijst op de betreffende website. Onderstaande linkertabel geeft het aantal verzamelde Nederlandse recensies van ieder onderwerp weer, waarbij een onderscheid wordt gemaakt tussen positief en negatief. Analoog wordt dit in de rechtertabel voor de Engelse recensies weergegeven.

	Positief	Negatief
Filmrecensies	197358	17978
Muziekrecensies	15197	3019
Boekrecensies	146	3719

Tabel 3.1: Aantal verzamelde Nederlandse recensies

	Positief	Negatief
Films	197358	17978

Tabel 3.2: Aantal verzamelde Engelse recensies

Wat meteen opvalt is dat het aantal verzamelde positieve boekrecensies heel klein is tegen over de andere recensies. Later bij het gebruik van deze dataset in 3.4 zullen we hier rekening mee moeten houden.

Om nog een beter inzicht te krijgen over de dataset geven onderstaande tabellen nog wat extra statistieken weer over de datasets.

	Positief	negatief
Filmrecensies	60	75
Muziekrecensies	89	105
Boekrecensies	58	61

Tabel 3.3: Gemiddeld aantal woorden voor een Nederlandse recensie

	Positief	Negatief
Filmrecensie	229	228

Tabel 3.4: Gemiddeld aantal woorden voor een Engelse recensie

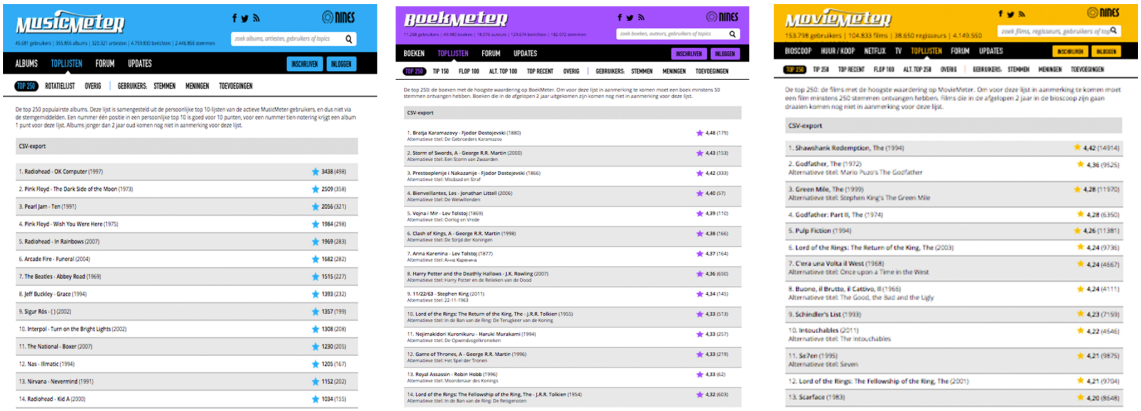
Uit de tabel kunnen we afleiden dat de verzamelde Engelse filmrecensies gemiddeld veel langer zijn. De mogelijke invloed op de prestatie van dit gegeven, onderzoeken we verder in 3.2.

	Positief	Negatief
Filmrecensies	2,64%	7,41%
Muziekrecensies	7,44%	12,52%
Boekrecensies	10,29%	25,39%

Tabel 3.5: Percentage woorden van het totaal aantal woorden in de Nederlandse dataset dat uniek is.

	Positief	Negatief
Filmrecensies	4,39%	4,41%

Tabel 3.6: Percentage woorden van het totaal aantal woorden in de Engelse dataset dat uniek is.



Figuur 3.2: de “All Time Top 250”-toplijsten op de websites

In 3.2 voeren we ook een analyse uit zonder gebruik te maken van een dataset om uit te leren, maar classificeren we de gegeven recensies op basis geannoteerde woordenlijsten van gevoelens. Als bron voor deze woordenlijsten hebben we het *Opinion lexicon* gebruikt, dat voor het eerst werd samengesteld door Hu & Liu (2004). Deze woordenlijsten bestaan uit een lijst met negatieve en een lijst met positieve woorden. De lijsten bevatten in totaal ongeveer 6800 woorden en zijn enkel in het Engels verkrijgbaar. De Nederlandse woordenlijsten hebben we verkregen door de Engelse lijsten te vertalen met behulp van Google vertalen.

Onderstaande tabel geeft weer hoe de woordenlijsten zich tegenover elkaar verhouden.

Tabel 3.7: Aantal woorden in iedere woordenlijst

	Positief	Negatief
Engels Woordenlijsten	2006	4783
Nederlands Woordenlijsten	2006	4647

We zien dat er een klein verlies van woorden is bij de negatieve Nederlandse woordenlijst. Dit komt door de vertaling van het Engels naar het Nederlands.

3.2 Engelse gevoelsanalyse versus Nederlandse Gevoelsanalyse

Als eerste analyse vergelijken we de Engelse gevoelsanalyse met de Nederlandse gevoelsanalyse. Als datasets gebruiken we de Nederlandse en Engelse filmrecensies, besproken in sectie 3.1. De filmrecensies stellen we voor aan de hand van de Vector Space Methode uit 2.2.1, waarbij iedere filmrecensie wordt voorgesteld als een vector met zijn woordfrequenties. Als Classifiers gebruiken we de Naive Bayes Classifier en de Decision Tree, beide werden besproken in sectie 2.4. Bij de analyse vergelijken we ook alle voorverwerkingstechnieken uit 2.3 en zelf combinaties hier van. Deze analyse is zodanig opgesteld dat we al de resultaten van de verschillende classifiers met een specifieke voorverwerkingstechniek naast elkaar kunnen leggen en de prestaties kunnen vergelijken voor wanneer men een Engelse of Nederlandse dataset gebruikt.

Concreter is de gevoelsanalyse die we in dit onderzoek uitvoeren, het correct kunnen onderscheiden van positieve en negatieve filmrecensies. de vergelijking wordt dan telkens gemaakt op basis van de prestaties van de analyses. De prestatie wordt beoordeeld op basis van de precisie waarmee de classifier de recensies classificeert. De precisie die we opnemen in onze resultaten voor een classifier wordt bepaald door het gemiddelde te nemen van 30 runs. Bij iedere run wordt er een ongetrainde classifier getraind met een trainingsset en wordt de precisie getest door het classificeren van de testset. In dit experiment bestaat iedere trainingsset uit 6000 filmrecensies en testset uit 2000 filmrecensies. Ook zorgen we er telkens voor dat zowel de trainingsset als de testset willekeurig en gebalanceerd samengesteld worden. Dit wil zeggen dat de datasets telkens voor de helft uit positieve en de andere helft uit negatieve recensies bestaan en wanneer men deze willekeurig zou classificeren, men een precisie baseline van 50% krijgt.

Zoals eerder vermeld gebruiken we voor onze analyse de verschillende voorverwerkingstechnieken uit 2.3. We hebben ook de vrijheid genomen om verschillende voorverwerkingstechnieken te combineren en hier de resultaten van te bekijken.
Onderstaande tabellen geeft de belangrijkste resultaten weer van de gevoelsanalyses. In bijlage A vindt men de volledig tabel met de resultaten.
Bag of words (zie rij 1) gebruiken we in deze tabel als baseline om de invloed van de andere technieken te vergelijken. Om een overzicht te krijgen hebben we in de tabellen de resultaten die beter presteren dan Bag of Word vet gedrukt.

Nr	**{Title}**	**{Precisie Naive Bayes Classifier}**	**{Precisie Decision Tree}**
1	Bag of Words	85,74%	69,06%
2	Best Feature selection on Bag of Words (max features)	67,79%	**69,43%**
3	Best Feature selection on TFIDF (max features)	74,90%	**69,79%**
4	Bigram Collocaties	**89,23%**	**69,41%**
5	LSA on Bag of Words (max features)	63,11%	62,07%
6	LSA on TFIDF (max features)	78,98%	**71,54%**
7	Term Weighting	**86,75%**	**69,76%**
8	Verwijderen van stopwoorden	**86,62%**	**69,45%**
9	Verwijderen van stopwoorden + Best feature selection on Bag of Words (max features)	74,43%	**69,36%**
10	Verwijderen van stopwoorden + Best feature selection on TFIDF (max features)	74,94%	**69,47%**
11	Verwijderen van stopwoorden + Bigram Collocaties	**89,23%**	**69,51%**
12	Verwijderen van stopwoorden + Bigram Collocaties + Term Weighting	**89,29%**	**69,44%**
13	Verwijderen van stopwoorden + LSA on Bag of Words (max features)	54,88%	68,66%
14	Verwijderen van stopwoorden + LSA on TFIDF (max features)	73,58%	**75,50%**
15	Verwijderen van stopwoorden + Term Weighting	**87,41%**	**69,60%**

Tabel 3.8: Resultaten experiment op Engelse recensies

Nr	**{Title}**	**{Precisie Naive Bayes Classifier}**	**{Precisie Decision Tree}**
1	Bag of Words	70,51%	59,34%
2	Best Feature selection Bag of Words	58,86%	**59,45%**
3	Best Feature selection on TFIDF (max features)	59,53%	**59,35%**
4	Bigram Collocaties	70,20%	**59,35%**
5	LSA on Bag of Words	54,84%	57,53%
6	LSA on TFIDF (100 features)	63,15%	58,58%
7	Term Weighting	69,40%	58,83%
8	Verwijderen van stopwoorden	70,35%	56,82%
9	Verwijderen van stopwoorden + Best feature selection on Bag of Words (max features)	60,76%	56,74%
10	Verwijderen van stopwoorden + Best feature selection on TFIDF (max features)	59,18%	56,44%
11	Verwijderen van stopwoorden + Bigram Collocaties	**70,63%**	56,80%
12	Verwijderen van stopwoorden + Bigram Collocaties + Term Weighting	**70,66%**	56,58%
13	Verwijderen van stopwoorden + LSA on Bag of Words (100 features)	53,74%	57,23%
14	Verwijderen van stopwoorden + LSA on TFIDF (max features)	60,15%	59,24%
15	Verwijderen van stopwoorden + Term Weighting	**70,54%**	56,55%

Tabel 3.9: Resultaten experiment op Nederlandse recensies

Wat meteen opvalt als we de resultaten bekijken in tabel **??** en **??** is het algemeen beter presteren van de technieken op de Engelse dataset. Tabel **??** geeft het verschil in prestatie aan tussen het Engels en het Nederlands. We zien dat de prestatie op de Engelse dataset gemiddeld 13% beter

presteert bij de Naive Bayes Classifier en 10% beter bij de Decision Tree.

Title	Verschil in Precisie Naive Bayes Classifier	Verschil in Precisie Decision Tree
Bag of Words	14,72%	16,05%
Best Feature selection Bag of Words	8,78%	12,18%
Best Feature selection on TFIDF (max features)	15,36%	10,44%
Bigram Collocaties	16,25%	10,41%
LSA on Bag of Words	27,27%	10,37%
LSA on TFIDF (100 features)	19,83%	10,69%
Term Weighting	11,79%	10,89%
Verwijderen van stopwoorden	31,79%	11,92%
Verwijderen van stopwoorden + Best feature selection on Bag of Words (max features)	35,55%	12,22%
Verwijderen van stopwoorden + Best feature selection on TFIDF (max features)	-2,56%	12,61%
Verwijderen van stopwoorden + Bigram Collocaties	18,60%	12,61%
Verwijderen van stopwoorden + Bigram Collocaties + Term Weighting	13,67%	12,61%
Verwijderen van stopwoorden + LSA on Bag of Words (100 features)	15,09%	12,48%
Verwijderen van stopwoorden + LSA on TFIDF (max features)	-15,67%	12,11%
Verwijderen van stopwoorden + Term Weighting	3,93%	5,63%
Gemiddeld verschil	**13,40%**	**10,83%**

Tabel 3.10: Verschil in precisie tussen het Engelse en het Nederlands (Eng - NL)

Nu in tabel **??** uit 3.1 zien we dat de Engelse dataset gemiddeld meer woorden heeft dan de Nederlandse dataset. Dit kan mogelijks een positieve invloed hebben op de classificatie, aangezien hoe meer woorden, hoe meer informatie betekent voor de classifier en het zo beter kan classificeren. Om dergelijke stelling te kunnen onderbouwen voeren we een kleine extra analyse uit. We voeren opnieuw een gevoelsanalyse uit op beide datasets, enkel beperken we het aantal woorden per recensie voor zowel de Engelse als de Nederlandse dataset tot 60 woorden. Als referentie gebruiken we de best presterende combinatie van voorverwerkingstechniek en classifier. Dit is de Naive Bayes Classifier met als voorverwerkingstechniek *Verwijderen van stopwoorden + Bigram Collocaties + Term Weighting* (zie rij 12).
resultaten van het experiment + subanalyse van het experiment
Ook is het interessant om te zien naar het verschil in de prestatie van de classifiers voor een bepaalde voorverwerkingstechniek. Bij rij 12 in tabel **??** zien we bijvoorbeeld een verschil van bijna 20 %. Na een kleine subanalyse waarbij we kijken naar de recensies waarbij de ene classifier de recensie juist classificeert en de andere fout, zien we dat....

Resultaten hier + wat blabla
Verder zien we ook dat de prestatie voor beide talen een grotere spreiding heeft bij de Naive Bayes Classifier dan bij de Decision tree. Voor het Engels vallen de resultaten van de Naive Bayes Classifier binnen een interval van 36% en bij de Decision Tree heeft dit interval een lengte van 14%. Voor het Nederlandse we hetzelfde verschijnsel. De resultaten van de Naive Bayes classifier vallen binnen een interval van 18% en bij de Decision tree is dit 2%.

Als we nu als referentie de prestatie van Bag of Words nemen en kijken hoe de andere technieken presteren tegenover Bag of Words, zien we dat voor de Naive Bayes Classifier de pre-processing technieken: Bigram collocaties, Term Weighting en het verwijderen van stopwoorden positief naar voren komen. Alleen of in combinatie hebben ze een positieve invloed op de prestatie. Dit het geval voor beide talen, al is deze bevinding bij het Engels overtuigend aanwezig en bij het Nederlands eerder minimaal. Opmerkelijk is dat de combinatie van de drie pre-processing technieken bij beide talen als best presterende techniek naar boven komt. Opnieuw is bij het Nederlands dit verschil minimaal ten op zichten van de andere combinaties.

Voor de Decision tree springt de prestatie van /textitVerwijderen van stopwoorden + LSA + Term weighting in het oog met 75% als beste resultaat. Bij het Nederlands springt deze techniek er niet uit en is de prestatie zelfs minder goed als Bag of Words, al is het verschil miniem en hoort het nog steeds bij de betere resultaten van de Decision Tree.

Algemeen kunnen we zeggen dat de trends die we zien bij het Engels zich ook voordoen bij het Nederlands. De technieken werken ook op het Nederlands, als men weet dat in eerder onderzoek Pang et al. (2002) aantoont dat een human-based classifier resultaten haalt van ongeveer 58% tot 64% op Engelse filmreviews, kunnen we stellen dat de technieken goede prestaties halen op Nederlandse reviews, met als de Naive Bayes Classifier de best presterende van de twee leermethoden.

3.3 Classificatie op basis van geannoteerde woordenlijsten

Om meer inzicht te krijgen over de verschillen in de gevoelsanalyse bij de twee talen, voeren we nog een tweede experiment uit. Bij dit experiment gaan we heel eenvoudig en intuïtief te werk. We kijken hoe de classificatie verloopt, wannneer we enkel geannoteerde woordenlijsten met gevoelens in beschouwing nemen. Er wordt voor iedere recensie gekeken, hoeveel woorden van de recensie voorkomen in de positieve lijst en hoeveel in de negatieve. De lijst met de meest overeenkomstige woorden geeft aan of de recensie positief of negatief moet worden geclassificeerd. Voor de woordenlijsten gebruiken we de eerder vermeldden woordenlijsten uit 3.1. De classificatie zelf testen we op de Engelse en Nederlandse filmrecensies.
Onderstaande tabel geeft de resultaten van de classificatie weer met als precisie het gemiddelde van 30 runs en een testset van 2000 samples random en gebalanceerd samengesteld.

	Precisie
Engels recensies	67,43%
Nederlands recensies	1,16%

Tabel 3.11: Classificatieprecisie aan de hand van woordenlijsten

Voor het tweede experiment gooiden we het over een andere boeg en werkt we met met geannoteerde woordenlijsten van gevoelens. Het resultaat voor Engelse recensies met 68% is goed, maar voor het Nederlands met 1% kunnen we zeggen dat de classificatie methode niet werkt. We moeten echter kritisch zijn en rekening houden met een paar dingen. Een eerste element waar we rekening mee moeten houden is de oorsprong van de woordenlijsten. De woordenlijsten zijn samengesteld op basis van Engelse recensies en het verlies in de vertaling naar het Nederlands kan een mogelijk effect hebben op de Nederlandse classficatie. Al zien we in tabel xxx dat dit verlies beperkt wordt tot 2,5%. Een andere invloed zijn de leenwoorden. De woorden uit de Engelse woordenlijst kunnen juist vertaald zijn door Google translate, maar kunnen onnatuurlijk overkomen in het Nederlands. Bijvoorbeeld het positieve woord *cool* wordt vertaald door Google translate als *koel*, wat in het Nederlands helemaal niet wordt gebruikt als positief woord. Ook de Engelse woordenschat om zich positief uit te drukken kan helemaal anders zijn dan die van het Nederlands. Als laatste heeft men ook internetslang en uitgesmeerde woorden zoals *sssaaaaiiii* die niet in rekening worden gebracht. In verder onderzoek kan men dergelijke invloeden vermijden door eigenhandig een Nederlandse geannoteerde woordenlijst met gevoelens samen te stellen en deze te gebruiken voor het classificeren van de Nederlandse reviews

3.4 Onderwerpgevoeligheid van Nederlandse Gevoelsanalyse

Nu we weten welke methode goed presteert op het Nederlands en welke niet, kunnen we er nog iets dieper ingaan op Nederlandse gevoelsanalyse. De voorgaande experimenten zijn altijd uitgevoerd op filmrecensies en hadden goede prestaties. Het is interessant om eens te kijken of de voorgaande technieken onderwerp gevoelig zijn of niet. Concreet voor dit experiment onderzoeken we enkel de beste presterende techniek uit 3.2, namelijk de Naive Bayes Classifier in combinatie met Term weighting en het verwijderen van stopwoorden. We kijken hoe deze techniek presteert wanneer we het trainen en testen met recensies over hetzelfde onderwerp en hoe het presteert met een verschillend. Als datasets nemen we film- , muziek en boekrecensies. De prestatie van de classifiers is telkens de gemiddelde classificatieprecisie van 30 runs, waarbij de trainingsset uit 6000 samples bestaat en de testset uit 2000 samples.
Onderstaande kruistabel met classificatieprecisies vat de belangrijkste resultaten van het experiment samen. De volledige resultaten vindt men in bijlage B. Merk op dat men hier ook de controle op over- of onderfitting vindt. Over- en onderfitting zijn symptomen bij machine learning waarbij men de classifier over of ondertrained. Voor de volledigheid hebben we deze grafieken in de bijlage toegevoegd.

	Films	**Muziek**	**Boeken**
Films	70,66%	61,00%	56,25%
Muziek	62,07%	82,62%	56,47%
Boeken	65,87%	61,46%	71,76%

Tabel 3.12: Kruistabel van alle classificatieresultaten uit **??** en **??** met de kolommen het onderwerp van de trainingsset en de rijen het onderwerp van de testset.

Als laatste hebben nog de confusion matrixen van het experiment. Een confusion matrix geeft weer hoeveel recensies er juist en fout geclassificeerd zijn.

	P	**N**
P'	43%	6%
N'	18%	31%

Tabel 3.13: Gemiddelde confusion matrix in percent voor een Naive Bayes Classifier, waar trainings- en testset over hetzelfde onderwerp gaan

	P	**N**
P'	32%	18%
N'	21%	29%

Tabel 3.14: Gemiddelde confusion matrix in percent voor een Naive Bayes Classifier, waar trainings- en testset over een verschillend onderwerp gaan

		voorspelde waarde	
		p	**n**
eigelijke waarde	**p′**	Waar Positief	Vals Negatief
	n′	Vals Positief	Waar Negatief

Tabel 3.15: Illustratie van de confusion matrix

Hoofdstuk 4

Conclusie

Literatuur

Bullinaria, J. A. (2004). *Bias and variance, under-fitting and over-fitting.* http://www.cs.bham.ac.uk/~jxb/NN/l9.pdf. (Accessed: 2014-27-05)

Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A. & Lochbaum, K. E. (1988). Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th annual international acm sigir conference on research and development in information retrieval* (pp. 465–480).

Goodness-of fit test, a nonparametric test. (z. j.). http://www2.cedarcrest.edu/academic/bio/hale/biostat/session22links/basics.html. (Accessed: 2015-05-23)

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J. & Tibshirani, R. (2009). *The elements of statistical learning* (Dl. 2) (nr. 1). Springer.

Hu, M. & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining* (pp. 168–177).

Landauer, T. K., Foltz, P. W. & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes*, *25*(2-3), 259-284. Verkregen van http://dx.doi.org/10.1080/01638539809545028 doi: 10.1080/01638539809545028

Latent semantic analysis (lsa) tutorial. (z. j.). http://www.puffinwarellc.com/index.php/news-and-articles/articles/33-latent-semantic-analysis-tutorial.html?showall=1. (Accessed: 2014-15-11)

Liebrecht, C., Kunneman, F. & van den Bosch, A. (2013). The perfect solution for detecting sarcasm in tweets# not.

Liu, M. & Yang, J. (2012). An improvement of tfidf weighting in text categorization. *International Proceedings of Computer Science and Information Technology*, 44–47.

Lohr, S. (2012). The age of big data. *New York Times*, *11*.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. (2011a). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1* (pp. 142–150).

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y. & Potts, C. (2011b, June). Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies* (pp. 142–150). Portland, Oregon, USA: Association for Computational Linguistics. Verkregen van http://www.aclweb.org/anthology/P11-1015

Manning, C. D., Raghavan, P. & Schütze, H. (2008). *Introduction to information retrieval* (Dl. 1). Cambridge university press Cambridge.

Manning, C. D. & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press.

Mantrach Amin, H. B. M. S., Nicolas Vanzeebroeck. (z. j.). *Machine learning course ulb: Text mining*. `https://ai.vub.ac.be/sites/default/files/textmining2011.pdf`. (Accessed: 2014-15-11)

Martineau, J. & Finin, T. (2009). Delta tfidf: An improved feature space for sentiment analysis..

McKinney, W. (2012). *Python for data analysis: Data wrangling with pandas, numpy, and ipython*. "O'Reilly Media, Inc.
.

Michie, D., Spiegelhalter, D. J. & Taylor, C. (1994). *Machine learning, neural and statistical classification*.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, *45*.

Ng, A. (z. j.). *Machine learning course*. `https://class.coursera.org/ml-005/lecture/preview`. (Accessed: 2014-15-11)

Paltoglou, G. & Thelwall, M. (2010). A study of information retrieval weighting schemes for sentiment analysis. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1386–1395).

Pang, B. & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, *2*(1-2), 1–135.

Pang, B., Lee, L. & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the acl-02 conference on empirical methods in natural language processing-volume 10* (pp. 79–86).

Pedersen, T. (2001). A decision tree of bigrams is an accurate predictor of word sense. In *Proceedings of the second meeting of the north american chapter of the association for computational linguistics on language technologies* (pp. 1–8).

Petitpierre, D. & Russell, G. (1995). Mmorph-the multext morphology program. *Multext deliverable report for the task*, *2*(1).

Rothfels, J. & Tibshirani, J. (2010). Unsupervised sentiment classification of english movie reviews using automatic selection of positive and negative sentiment items. *CS224N-Final Project*.

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *44*(1.2), 206–226.

Turney, P. D., Pantel, P. et al. (2010). From frequency to meaning: Vector space models of semantics. *Journal of artificial intelligence research*, *37*(1), 141–188.

A tutorial on clustering algorithms. (z. j.). `http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/index.html`. (Accessed: 2015-01-11)

Wakade, S., Shekar, C., Liszka, K. J. & Chan, C.-C. (2012). Text mining for sentiment analysis of twitter data. In *International conference on information and knowledge engineering (ikeÕ12)* (pp. 109–114).

Wang, L. & Wan, Y. (2011). Sentiment classification of documents based on latent semantic analysis. In *Advanced research on computer education, simulation and modeling* (pp. 356–361). Springer.

Ye, Q., Zhang, Z. & Law, R. (2009). Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications*, *36*(3), 6527–6535.

Zhang, C., Zuo, W., Peng, T. & He, F. (2008). Sentiment classification for chinese reviews using machine learning methods based on string kernel. In *Convergence and hybrid information technology, 2008. iccit'08. third international conference on* (Dl. 2, pp. 909–914).