Scoren op Twitter

Een studie naar event detectie met behulp van tweets

Mart Busger op Vollenbroek
Rijksuniversiteit Groningen
Faculteit der Letteren - Informatiekunde
Juni 2015

Scoren op Twitter

Een studie naar event detectie met behulp van twitter

Mart Busger op Vollenbroek
Rijksuniversiteit Groningen
Faculteit der Letteren - Informatiekunde
Juni 2015

Voorwoord

Deze scriptie is geschreven door Mart Busger op Vollenbroek, Pre-Master student aan de Rijksuniversiteit Groningen. De scriptie is geschreven als afsluiting van de Pre-Master Information Science en dient daarmee als voorbereiding voor de Master Information Science. Met behulp van de verkregen vaardigheden en opgedane kennis tijdens het afgelopen jaar heb ik naar deze scriptie toegewerkt. Aan het begin van het jaar had ik geen ervaring met programmeren of andere informatiekunde-gerelateerde kennis, omdat mijn vorige opleiding, Bedrijfseconomie, totaal verschillend is ten opzichte van Informatiekunde. Aan het begin dit collegejaar werden mij de basisbegrippen van het programmeren aangeleerd en daar is tijdens de rest van het collegejaar op voort gebouwd. Voornamelijk de vakken als Information Retrieval waar aandacht besteed werd aan tekst classificatie en dergelijke waren achteraf erg leerzaam en nuttig voor de scriptie.

De scriptie zelf was verdeeld over een heel semester en waar de eerste periode stond voor oriëntatie en het onderwerp eigen maken, stond de tweede periode in het teken van daadwerkelijk onderzoek doen, programmeren en de uiteindelijke scriptie schrijven. Tijdens het semester was er iedere week een moment waar zaken besproken werden als de voortgang van de scriptie, keuze van onderwerp, maar er werd ook algemene feedback gegeven en waar nodig individueel zaken besproken.

De begeleiders voor mijn scriptie waren prof. dr. Johan Bos en Malvina Nissim. Middels deze weg wil ik beiden bedanken voor de ondersteuning en begeleiding bij mijn onderzoek en het schrijven van mijn scriptie.

Ten slotte wil ik u, de lezer, veel plezier wensen bij het lezen van mijn scriptie.

Mart Busger op Vollenbroek

Juni 2015

Samenvatting

Inhoudsopgave

Inleiding 1
Onderwerp 1
Relevante literatuur 2
Vervolg onderzoek 3
Methode 4
Dataverzameling 4
Dataverwerking 4
Annotatie 5
Classificatie 5
Resultaten 6
Discussie 7
Conclusie 8
Literatuurlijst I
Boeken I
Artikelen I
Bijlagen II
Code II
1. tweetfilter.py II

Inleiding

Onderwerp

Twitter is een grote bron van informatie voor wijd scala aan onderwerpen. Het is daarom een lastige opgave om de juiste informatie te kunnen vinden en om irrelevante informatie te filteren. Het vinden van die informatie of gebeurtenissen (events) wordt Event Detection genoemd. Het is lastig om een eenduidige definitie van Event Detection te geven, omdat het een vrij ambigu begrip is. Het is daarom van belang om eerst duidelijk te hebben wat precies een event is en wat het detecteren inhoudt. Allereerst kan een event opgedeeld worden in een main event en meerdere daarbij horende sub events. Een voorbeeld in het kader van dit onderzoek is om een voetbalwedstrijd als main event te zien en gebeurtenissen als doelpunten of kaarten tijdens een wedstrijd als sub events te zien. Dit onderzoek probeert om de zogenaamde sub events te scheiden van een main event en ze in kaart te brengen. Een event, zowel main events als sub events dienen te voldoen aan drie eisen:

- Er is sprake van een entiteit, dit kan van alles zijn zoals personen of gebeurtenissen
- Er is sprake van een afgebakende tijdsperiode of een bepaald moment
- Er is sprake van een gebeurtenis met de entiteit tijdens die tijdsperiode of op een bepaald moment

In het geval van de voetbalwedstrijd geldt de wedstrijd als een main event omdat het aan deze drie eisen voldoet. De wedstrijd zelf is de entiteit, er is sprake van een afgebakende tijdperiode en tijdens de wedstrijd is er sprake van gebeurtenissen zoals doelpunten, gele kaarten en wissels. Er is echter wel sprake van een zekere vorm van van ambiguïteit wanneer er een scheiding gemaakt wordt tussen main events en sub events, het kan namelijk op verschillende niveaus en er is nog een andere manier om een voetbalwedstrijd te zien. Een voetbalwedstrijd is namelijk vaak onderdeel van een toernooi of een competitie, welke op zichzelf ook geïdentificeerd kunnen worden als main event. Om het toernooi als voorbeeld te gebruiken: dit is op zichzelf een entiteit tijdens een afgebakende periode waarin gebeurtenissen plaatsvinden. Tijdens dit onderzoek zal echter uitgegaan worden van eerste opvatting, waarbij een voetbalwedstrijd als main event gezien wordt en de gebeurtenissen tijdens de wedstrijd als sub events gezien worden. Er wordt met dit onderzoek antwoord gegeven op de volgende onderzoeksvraag:

Hoe kunnen met behulp van tweets sub events geïdentificeerd worden en van een main event onderscheden worden tijdens sportwedstrijden?

Als deze methode werkt, kan er gekeken worden naar mogelijkheden om het programma uit te breiden naar andere (team)sporten. Op deze manier zou er ook een realtime systeem onderworpen kunnen worden dat eventueel meldingen geeft wanneer er iets gebeurd tijdens een wedstrijd om zo supporters of gebruikers van het systeem up to date te houden houden met de stand. Tevens wordt er met dit onderzoek gekeken naar mogelijkheden van tekstanalyse op basis van zeer korte teksten in grote hoeveelheden zoals tweets.

Relevante literatuur

Er zijn vorige onderzoeken geweest naar event detection in de twitter stream, Weng et al. (2011) hebben bijvoorbeeld het Event Detection with Clustering of Wavelet-based Signals (EDCoW) systeem ontwikkeld. Dit systeem is getest tijdens de verkiezingen in Singapore in 2011 en hoewel het niet test op relatief korte events zoals een voetbalwedstrijd, kaart het wel een van de belangrijkste problemen aan bij het detecteren van events op twitter: de overload aan (irrelevante) data. Bij conventionele event detection, waar vaak gebruik gemaakt wordt van nieuws artikelen of wetenschappelijke artikelen, zijn de artikelen vaak aan elkaar gerelateerd en zijn ze nagenoeg allemaal relevant. Op twitter worden echter veel tweets gepubliceerd die niet informatief zijn en daardoor irrelevant voor onderzoek. Het is van groot belang een manier te vinden om deze tweets te filteren.

Corney et al. (2014) hebben een systeem voor event detectie tijdens voetbalwedstrijden ontwikkeld dat op basis van frequentie van woorden in tweets event probeert te detecteren. Dit doen ze met behulp temporal document frequency - document inverse frequency (df-idft) scores, een variant van de veelgebruikte term frequency - inverse document frequency (tf-idf) scores. Tf-idf scores worden berekend door te kijken hoe vaak een woord voorkom in een document vergelijken met hoe vaak het in alle documenten voorkomt om zo te bepalen hoe uniek een woord is voor het document. Een hoge term frequency en een lage inverse document frequency geven een hoge tf-idf score terug en daarmee is de kans groter dat een woord uniek is.

Vervolg onderzoek

Na deze korte inleiding zal het onderzoek verder gaan met de beschrijving van de methode. In dat hoofdstuk zal aandacht worden besteed aan de dataverzameling, verwerking van de data en verschillende manieren waarop er met de data gewerkt is. Tevens zal in de methode uitleg gegeven worden over de werking van het programma dat de uiteindelijke detectie van events moet doen. Na het methode hoofdstuk zullen de resultaten gerapporteerd worden in een apart hoofdstuk en vervolgens zullen de resultaten verklaard en besproken worden in het discussie hoofdstuk. Ten slotte zal in het conclusie hoofdstuk een beknopte eindconclusie gevormd worden op basis van de resultaten en zullen er aanbevelingen gedaan worden voor verder onderzoek.

Methode

Dataverzameling

De data voor het onderzoek is gekomen uit de corpus van tweets van de Rijksuniversiteit Groningen welke te vinden is in de /net directory op de Karora server. In de Twitter corpus van de universiteit worden alle Nederlandstalige tweets opgeslagen, geordend op jaar, maand, dag en uur. Ieder uur is een uniek gecomprimeerd bestand in de map van de bijbehorende maand en iedere maand is een map in de map van het bijbehorende jaar. Een voorbeeld van een pad naar de tweets van 2 tot 3 uur 's middags op 15 mei 2014 is als volgt: /net/corpora/twitter2/Tweets/2014/05/20140515:14.out.gz.

De relevante data voor dit onderzoek zijn de wedstrijden van het Nederlands Elftal op het Wereldkampioenschap Voetbal 2014 in Brazilië. Dit waren zeven wedstrijden, en volledigheidshalve is er gekozen om voor iedere wedstrijd 3 uur aan tweets te verzamelen. Dit zijn in totaal 21 bestanden voor 21 uur aan tweets. Omdat de bestanden gecomprimeerd zijn en de tweets opgeslagen zijn in het Json formaat, moeten ze eerst gedecomprimeerd worden en moet de data er op de juiste manier uitgehaald worden. Dit gebeurt met een commando, waarmee de output ook naar een tekstbestand geschreven wordt gescheiden op tab. Dit commando is voor het voorgaande voorbeeld als volgt: zcat /net/corpora/twitter2/Tweets/2014/05/20150515:14.out.gz | /net/corpora/twitter2/tools/tweet2tab -k id date user text | grep -P "^[^\t]+\t[^\t]+" > 20140514.txt. Met dit commando wordt de relevante informatie van de tweets naar een tekstbestand geschreven. Voor de andere uren op dezelfde dag kunnen de tweets toegevoegd worden aan het tekstbestand door twee vishaken te gebruiken in het commando in plaats van een.

Dataverwerking

Nadat de data allemaal verzameld is, moet deze gefilterd worden. Dit gebeurt met het tweetfilter.py programma, de code van dit programma staat in bijlage 1. Dit programma leest het tekstbestand in waar de tweets naartoe geschreven zijn en filtert daar, afhankelijk van de wedstrijd, de tweets op bepaalde woorden. In het geval van de wedstrijd Spanje - Nederland op 13 juni 2014 worden er alleen tweets uitgehaald waar of 'Spanje' of 'Nederland' of '#spaned ' in voor komt. Vervolgens wordt er gekeken naar de gebruikers die de tweets geplaatst hebben, voor dit onderzoek zijn tweets van sites als voetbalzone of nosvoetbal niet

van belang en deze worden dan ook niet mee genomen. Ten slotte word er gekeken of er een van de woorden ‘goal’, ‘doelpunt’, ‘kaart' en dergelijke voorkomt in de tweet, dit om het grote aantal irrelevante tweets alvast deels te beperken. De uiteindelijke output wordt naar een bestand geschreven, wederom gesplitst op tab. Afhankelijk van wat er met de data gaat gebeuren wordt er een 0 of een ? aan het eind van de regel geplaatst.

Annotatie

Wanneer de data voor het eerst gebruikt wordt en er nog geen data om te trainen beschikbaar is, moet het print statement met de 0 gebruikt worden. Hierdoor wordt er aan het eind van iedere regel een 0 geprint, waarna begonnen kan worden met de annotatie.

Voor de rest van het onderzoek is meer data nodig om meerdere wedstrijden te voorspellen, dus er moet een training set gecreëerd worden. Deze training set wordt door een classifier gehaald, waarmee het voorspellend vermogen van de classifier berekend wordt. Bij het annoteren is het van belang dat duidelijk is waarvoor je annoteert, zodat je classifier op dezelfde manier ‘denkt’ als jij. Voor dit onderzoek is het van belang dat tweets geannoteerd worden als ‘relevant' (1) of ‘irrelevant' (0). Omdat de verwachting was dat er meer irrelevante tweets tussen zitten er achter iedere tweet een 0 geplaatst die veranderd kan worden in een 1 wanneer een tweet relevant is. In het geval van dit onderzoek is het van belang dat er gesproken wordt over een doelpunt, gele kaart of rode kaart.

Classificatie

Resultaten

Discussie

Conclusie

Literatuurlijst

Boeken

Artikelen

Corney, D., Martin, C., & Göker, A. (2014). Spot the ball: Detecting sports events on Twitter. In Advances in Information Retrieval (pp. 449-454). Springer International Publishing.

Weng, J., & Lee, B. S. (2011). Event Detection in Twitter. ICWSM, 11, 401-408.

Bijlagen

Code

1. tweetfilter.py

```
import sys

def main(argv):
        tweetfile = argv[1]
        rawTweetData = open(tweetfile, encoding='utf-8')
        for line in rawTweetData:
                tweetID, tweetDate, tweetUser, tweetText = line.split('\t')
                tweetUser = str(tweetUser)
                tweetText = str(tweetText)
                invalidUsers = ['sport1nl', 'nosvoetbal', 'voetbalpings', 'voetbalzonenl',
                                'voetbalprimeur', 'by433', '433live', '443nl' 'foxsportslive',
                                'nu_sportslive', 'livesports_hd', 'livefootball']
                if tweetUser.lower() not in invalidUsers:
                        conditions = ['chili', 'nederland', '#nedchi']
                        eventTypes = ['doelpunt', 'goal', 'rode', 'gele', 'rood', 'geel', 'kaart']
                        if any(condition in line for condition in conditions) and any(eventtype
                        in line for eventtype in eventTypes):
                                print(tweetID + "\t" + tweetDate[11:19] + "\t" +
                                tweetText[:-1] + "\t" + "0", file=sys.stdout) #for annotating
                                print("{}\t{}\t{:>140}\t?".format(tweetID,  tweetDate[11:19],
                                tweetText[:-1])) #for readying file for classifier


if __name__ == '__main__':
        main(sys.argv)
```