



**KTH Computer Science
and Communication**

En jämförelse av matchningsmetoder för big data twitterbotar

MATTIAS FOLKE, MATFOL@KTH.SE
MITRA KHORSAND, MITRAK@KTH.SE

Examensarbete inom datalogi, grundnivå (DD143X)
Skolan för datavetenskap och kommunikation, KTH
Handledare: Anna Hjalmarsson
Examinator: Mads Dam

Referat

I denna kandidatuppsats redovisas hur chatbotar som använder en databas med information om tweets kan finna ett svar från ett existerande tweet sparad i databasen. För att sedan använda detta som svar till ett nyinkommet tweet. Två olika chatbotar implementeras och utvärderas för att hitta så relevanta svar som möjligt. Den ena bygger på att trigram av tweets matchas till varandra, och den andra bygger på att tweets semantiskt textklassificeras. Resultatet utvärderas därefter genom ett relevanstest och ett turingtest. Slutsatsen är att även om det inte går att statistiskt säkerställa något om prestationen av chatbotarna så antyder resultatet av utvärderingen att chatboten som bygger på semantisk textklassificering gjorde ett något bättre jobb än den som bygger på trigrammatchning.

Abstract

This bachelor thesis presents how chatbots that uses a database with information on tweets can find an answer to an unrelated tweet, written by a user, and use this as a response to a newly posted tweet. Two chatbots are implemented and evaluated to find the most relevant responses possible. The first is based on matching trigram of tweets, and the other is based on semantic text classification of tweets. The results are then evaluated through a relevance test and a Turing test. The conclusion is that although no statistically significance have been proven, the results of the evaluations indicates that the chatbot based on semantic text classification did a slightly better job than the one based on trigram matching.

Innehåll

Innehåll

Definitioner	1
Introduktion	1
Problemformulering	1
Hypotes	2
Bakgrund	2
Turingtest och chatbothistoria	2
Chatbotar idag	3
Big data	3
N-gram och trigram	4
Textklassificering	4
Metod	5
Kommunikation med Twitter	5
Databas	5
Tribot	6
Sembot	9
Utvärdering	12
Resultat	12
Turingtest	12
Relevanstest	15
Slutsats	15
Litteraturförteckning	19

Definitioner

- Semantisk textklassificering: klassificering av texters innebörd
- Körtweet: det inkomna tweetet som ska besvaras
- Modertweets: de tweets som finns i databasen
- Sembot: boten som använder sig av semantisk textklassificering
- Tribot: boten som använder sig av trigram
- Relevanta svar: hur stor betydelse användare anser att svar har.

Introduktion

Sociala nätverk som Twitter och Facebook används dagligen av extremt många användare och genererar massiva mängder data. Datan kan analyseras och användas till exempel för att bygga program som ger människolik respons på given input. Den typen av program kallas chatbotar och används mer och mer i kommersiella syften. Exempel på sådana chatbotar finnes i avsnittet bakgrund i denna uppsats.

En av de stora utmaningarna med att bygga en chatbot är att programmet skall kunna analysera input och producera output på ett så avancerat och sofistikerat sätt att mottagaren på andra sidan upplever att den konverserar med en riktig människa.

I uppsatsen redovisas hur chatbotar som använder en databas med information om tweets kan finna ett svar från ett existerande tweet sparad i databasen. För att sedan använda detta som svar till ett nyinkommet tweet. Två olika metoder implementeras och utvärderas för att hitta så relevanta svar som möjligt. Dels byggs en databastabell av trigram upp. Trigram är enheter bestående av tre på varandra följande ord, dessa beskrivs närmare i bakgrunden. Databastabellen används för att snabbt kunna grovsålla i databasen genom att matcha trigram av modertweets mot körtweetet. Dessutom har modertweetsen delats in i semantiska kategorier för att genom motsvarande kategorisering av körtweetet få ett annat urval ur databasen knutet till dess innebörd. Resultatet utvärderas därefter igenom ett relevanstest och en variant av ett turingtest.

Uppsatsen inleds med problemformuleringen, och en bakgrund som kort behandlar chatbotars utveckling till idag samt tre primära hjälpmedel som används för chatbotar idag. Därefter beskrivs hur implementationen av chatbotarna gjorts under rubriken metod. Där behandlas också hur utvärderingarna av botarna genomförts. Under resultatdelen redogörs utfallet av utvärderingarna och därefter står slutsatserna beskrivna. Sist återfinns referenser.

Problemformulering

Vid genomsökning av en stor existerande databas av tweets med målet att hitta ett besvarat tweet som liknar ett nyinkommet kan flera metoder användas. I detta projekt implementeras och testas två av dessa, trigram och semantisk textklassificering.

Därefter utvärderas hur människolika respektive hur relevanta svaren från de olika metoderna är.

Frågeformuleringen för projektet lyder:

Är trimapping eller semantisk textklassificering en lämpligare metod för att finna ett människolikt och relevant svar till ett nyinkommet tweet?

Hypotes

Användningen av semantisk textklassificering för att dela in tweets i ämneskategorier bör medföra att svaret som genereras har en semantisk koppling till det inkommande tweetet. Detta leder dock till att resultatens kvalitet är direkt beroende av klassificerarens kvalitet. Lösningen med trigram är jämförelsevis oberoende utav extern kod och baseras på garantin av att det kommer att finnas en syntaktisk knypunkt mellan tweetsen.

Vår hypotes är att det kommer existera en skillnad som antyder att semantisk textklassificering är en lämpligare metod. Hypotesen grundas i att ord har olika betydelse i olika sammanhang. Till exempel existerar uttrycket *it's raining cats and dogs*, som inte har något med katter och hundar att göra. Detta är en viktig aspekt som totalt ignoreras i trigram lösningen.

Bakgrund

I denna del av uppsatsen kommer den bakomliggande teorin till de tekniker som används för genomförandet av projektet att beskrivas. Dessutom presenteras ett urval av de produkter som redan finns på marknaden inom detta område.

Turingtest och chatbothistoria

1950 publicerade Alan Turing artikeln *Computing Machinery and Intelligence* [22] i vilken Turing belyste den filosofiska frågan "Kan maskiner tänka?" och Turingtestet introducerades. Turingtestet gjorde det möjligt för datorprogram att bli bedömda utifrån sin artificiella intelligens och sitt mänskliga beteende. Turingtestet genomförs genom att en chatbot och en människa båda försöker övertala en mänsklig domare att de är mänskliga genom att konversera med domaren via en terminal. Om domaren inte konsekvent är säker på vilken av dem som är människan så "vinner" boten testet. [7]

Chatbotens tidiga historia består till största delen utav introduktionen av ett par relativt primitiva chatbots skapade utifrån nyfikenhet i artificiell intelligens. De mest kända botarna var Eliza och Parry.

Eliza var ett av de första exemplen på applicerad NLP (Natural Language Processing). NLP är ett område inom språkteknologi som bygger på förståelse för människans språk. Eliza, som fortfarande går att konversera med på internet [11], bygger på ett koncept som många av dagens botar fortfarande tillämpar. Hon analyserar inputen från användaren genom att ta ut de mest centrala orden eller fraserna för

. BAKGRUND

att sedan använda förprogrammerade meningar för att ge ett relativt vettigt svar som output till användaren. Eliza är designad som en psykoterapeut som håller konversationer med ”patienter”.

Parry som introducerades efter Eliza var designad som en paranoid patient alltså som Elizas motsvarighet. Dessa två botar ställdes sedan mot varandra och hade intressanta och dessutom underhållande konversationer [3].

Chatbotar idag

Idag är de chatbotar som finns tillgängliga på nätet fortfarande förhållandevis omänskliga, det är lätt för en människa att upptäcka att den pratar med en robot. Detta är inte för att utvecklingen på något sätt gått långsamt, vetenskapen om artificiell intelligens har utvecklats enormt de senaste decennierna. Men mänskligt språk är något komplext och trots nya angreppsmetoder är det inte tillräckligt för att datorer ska uppfattas mänskliga av människor. Det är därför mycket svårt att skapa en chatbot som konverserar som en riktig människa.

Svårigheten för datorer att simulera mänskligt beteende är allmänt vedertagen inom området artificiell intelligens och beskrivs av Moravecs paradox [1]. Paradoxens huvudbudskap är att det är lätt för datorer att lösa avancerade matematiska problem men att de har svårt för att simulera enkla mänskliga beteenden som att hålla simpla konversationer eller uttrycka trivialt kroppsspråk.

Idag används chatbotar bland annat som supporthjälp av företag på hemsidor. Chatbotarna vägleder användaren genom hemsidan, svarar på användarens eventuella frågor och förser användaren med information av intresse [9]. Två exempel på sådana så kallade virtuella assistenter är it- och telekommunikationsbolaget Plusnets chatbot Jess [18] som svarar på användarens telefon och bredbands frågor och leksaksföretaget Toys R Us chatbot Emma [21] som hjälper och vägleder kunder online med ehandel.

Chatbotar används också som stöd för språkinläring. Användaren interagerar med boten för att öva på både skrift och uttal i diverse språk. Just nu utvecklas det sådana chatbotar för att lära ut engelska [10]. Ett exempel på en sådan chatbot är engelskspråkiga Meg [20] som konverserar med användaren om vardagliga ting för att bredda dess engelskakunskaper.

Big data

En teknik som blir mer och mer populär är big data, som baseras på att mycket data är effektivare än komplexa algoritmer. Dessutom har big data implementationer fördelen att de är snabba att applicera, är mer generella och går att automatiskt anpassa inkrementellt allt eftersom man får in mer data. Inom botvärlden används ofta en stor databas som fylls på allteftersom användare interagerar med tjänsten, en teknik som används av flera chatbotar som finns på marknaden just nu. Den mest kända av dessa är Cleverbot [4]. Cleverbot samlar löpande in data från sina användare för att uppdatera sin databas, detta gör att den hela tiden blir ”smartare” eftersom den får en större mängd data att använda sig av. Man kan uttrycka det som att den lär av sina användare.

N-gram och trigram

N-gram är en probabilistisk modell som ofta används i samband med markovmodeller. N-gram används för att dela upp någonting i mindre beståndsdelar, ofta för att därefter kunna förutspå vad som med störst sannolikhet inträffar efter tidigare tillstånd statistiskt sett. Man använder ofta n-gram inom språkteknologi men det används också flitigt inom andra områden som till exempel att räkna ut DNA-sekvenser inom biologin.

Trigram är en typ av n-gram där man genererar delar om tre objekt. Inom lingvistik kan dessa objekt t.ex. vara stycken som bryts upp i meningar, meningar som bryts upp i ord eller ord som bryts upp i bokstäver. Nedan finns exempel för att förtydliga.

Exempel trigram av ord:

Jag tycker att datorer är det roligaste som finns i livet.

1	Jag tycker att
2	tycker att datorer
3	att datorer är
4	datorer är det
5	är det roligaste
6	det roligaste som
7	roligaste som finns
8	som finns i
9	finns i livet

Exempel trigram av bokstäver:

Älskar livet

1	Äls
2	lsk
3	ska
4	kar
5	ar
6	r l
7	li
8	liv
9	ive
10	vet

Textklassificering

Under 1990-talet började klassificering av texter tillta i popularitet. Textklassificering genomfördes då både regelbaserat och med hjälp av inlärning och probabilistiska metoder. Ett exempel på tidiga regelbaserade klassificeringar är när epost började användas och behovet växte för spamfilter. Det var då lätt att leta efter specifika ord eller fraser som indikerade att det var spam. Ett problem med detta var dock att de som skickade ut spam ofta lyckades kringgå dessa regler. [5]

. METOD

Den enklaste probabalistiska klassificeraren var Bayes naiva klassificerare som utgick ifrån att alla element i inputen var oberoende av varandra [14]. Med hjälp av detta kunde man t.ex. bestämma om ett namn var ett kvinnonamn eller mansnamn. I ett första steg kunde man undersöka sista bokstaven i ett stort antal namn med bestämda genus. Därefter kunde programmet göra lämpliga gissningar, till exempel om den sista bokstaven var ett a så var sannolikheten 38 gånger större att det hörde till ett kvinnonamn. Detta kunde sedan utvecklas till att man lämnade det oberoende tankesättet och övergick till att titta på de två sista bokstäverna och fick ut en ännu bättre träffsäkerhet i gissningarna. [2]

Metod

En databas med tweets byggs och används för att skapa två chatbotar, tribot som använder sig av trigram, och sembot som är baserad på semantisk textklassificering. Dessa bygger båda på big data principen och de använder den gemensamma databasen för att svara på tweets. Matchningen mellan modertweets, de tweets som ligger i vår databas, och körtweet, tweetet vi söker svar till, kommer att ske enligt respektive sätt för botarna. Ifall svar ej går att finna för endera bot kommer den använda sig av standardsvaret "awesome!", svaret valdes då det kan tolkas både positivt och ironiskt av en människa. Detta är ett specialfall som enbart inträffar om inget av botarnas valda modertweets är besvarade utav en twitteranvändare.

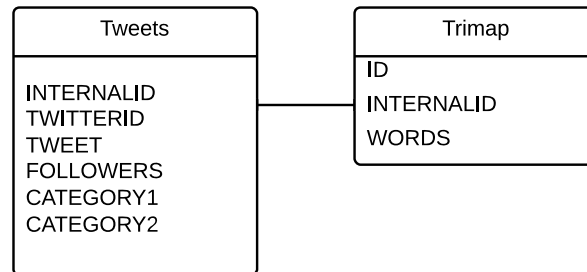
Kommunikation med Twitter

I arbetet används programmeringsspråket Java [16]. För att genomföra kommunikationen med twitter använde vi oss utav det öppna biblioteket Twitter4j [24]. I Twitter4j finns det fördefinierade metoder både för att hämta ut en specifik användare och dess tidslinje, samt om man är autentiserad publicera egna statusuppdateringar. Dock tillåter Twitters REST API max 350 anrop i timmen [23] vilket inte räcker långt för att genomföra data mining, det vill säga hämta ut information, som krävs för att fylla på databasen. Detta kringgås genom att använda Twitters strömnings API, där ett kontinuerlig flöde med information ifrån Twitter är tillgängligt utan begränsning.

Med Twitter strömnings API byggs vår databas upp för att möjliggöra sökningar med våra olika algoritmer. Dock måste båda våra algoritmers sista process, att hämta svar till ett tweet, använda sig av REST APIt där kvoten tyvärr fylls upp väldigt fort. Andra inom akademien har tidigare fått utökad kvot på APIt [6], men på grund av tidsbrist och långsam respons från Twitter löses detta genom att svar hämtas manuellt från Twitter med ett TwitterID.

Databas

En MySql [17] databas används för att hålla information om tweetsen. Databasen är uppdelad i två tabeller (se Figur: 1.1). Den första kallas tweets och används av båda botarna. Den andra är specifikt för tribot och benämns trimap. Dess primära funktion är att hålla samtliga uppdelningar av tre ord som står brevid varandra i något tweet.



Figur 1.1: Databas.

För att hålla databasen så relevant som möjligt genomförs en fyrstegsfiltrering innan tweets ifrån strömmen läggs in. Till att börja med kontrolleras att tweetet inte innehåller några länkar med hjälp av ett regex [8]. Därefter städas @personer, #saker och RT (Retweet) bort ifrån tweetet. Efter det kontrolleras att det är minst 3 ord kvar i tweetet, detta underlättar för det externa biblioteket langauge-detection [19] som bekräftar att tweetet är skrivet på engelska. Därefter anses tweetet vara klart att läggas in i databasen.

Totalt fylls databasen med 141 195 tweets som kontinuerligt samlas in under en treveckorsperiod.

Tribot

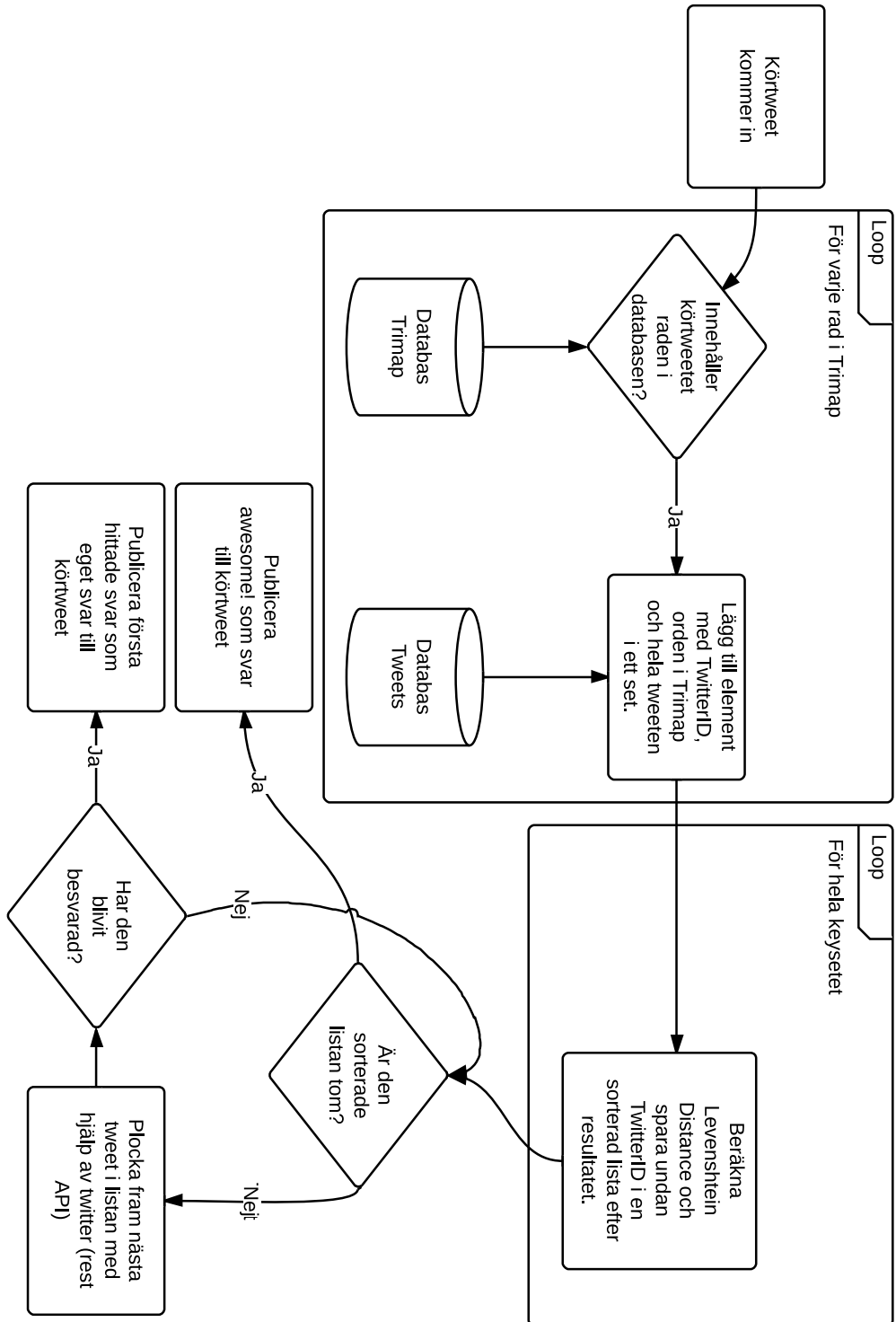
Tribot utgår ifrån databasens tabell trimap som innehåller trigram av ord ifrån alla tweets i databasen. När det kommer in ett nytt tweet, körtweetet, till tribot kommer hela trimap användas av boten som för varje rad i tabellen kollar om körtweetet innehåller de tre orden i följd. Om den innehåller ordföljden sparas TwitterID och orden undan, om inte går den vidare till nästa rad utan åtgärd. Detta visas i den första loopen i Figur: 1.2.

En grovsällning bland tweetsen har i det tidigare skedet gjorts av tribot. Därefter görs en närmare undersökning för att försöka bestämma vilken utav alla kvarvarande tweets från databasen som mest liknar det nyinkomna tweetet. Detta görs med hjälp av Levenshtein Distance [13], som är en algoritm för att beräkna avståndet mellan två strängar baserat på tecken, för att få ett värde på hur långt ifrån varandra tweetsen är. Det värdet och twitterID sparas undan och sorteras för att skapa en prioriteringslista. Detta visas i den andra loopen i Figur: 1.2.

På grund av begränsningarna med Twitters strömmings API som nämnts i avsnittet "Kommunikation med Twitter" i denna uppsats så genomförs följande process manuellt. När svar hämtas från Twitter plockas det TwitterID som är först

. METOD

i prioritetslistan ut, med detta söks motsvarande tweet upp på Twitters hemsida. Eventuella svar som det tweetet fått blir då synliga. Om tweetet bara fått ett svar ges detta som botens slutgiltiga svar. Om tweetet fått flera svar ges det första i listan av svar som botens slutgiltiga svar. Har tweetet inte fått något svar alls genomförs samma procedur för nästa tweet i prioritetslistan. Om situationen att inga tweets i listan har ett svar uppstår ges standardsvaret "awesome!".



8
Figur 1.2: Tribot.

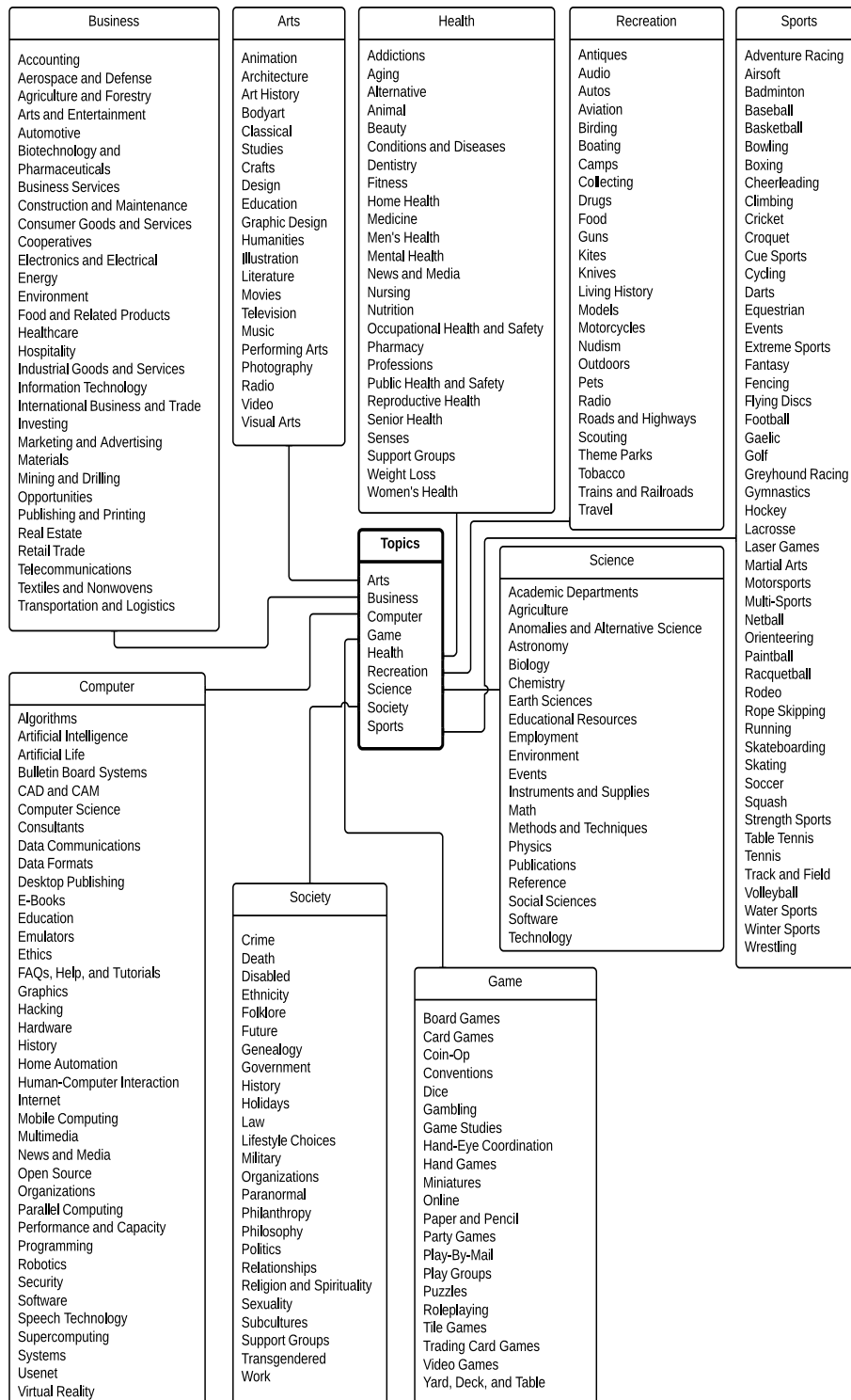
Sembot

Sembot bygger på att varje tweet kan klassificeras till en ämneskategori som t.ex. konst, hälsa och vetenskap. Genom att klassificera alla modertweets i vår databas samt körtweetet till ämneskategorier så kan vi lätt genomföra en grovsällning genom att enbart arbeta med tweets av samma ämneskaregori.

För detta används uClassify [12], ett gratis ramverk som dels innehåller ett antal olika färdiga textklassificerare men också tillåter att man bygger egna klassificerare kring deras ramverk. uClassify testas med ett mindre antal slumpade tweets och presterar där på en nivå som får anses acceptabel relativt det ringa tidsspänn som projektet förfogar över. Textklassificerarna som används i denna uppsats finns förtydligade med namn och kategorier i Figur 1.3. Dessa textklassificerare är uppbyggda med Open Directory Project (ODP) som är en struktur för att länka ord till ordkategorier [15].

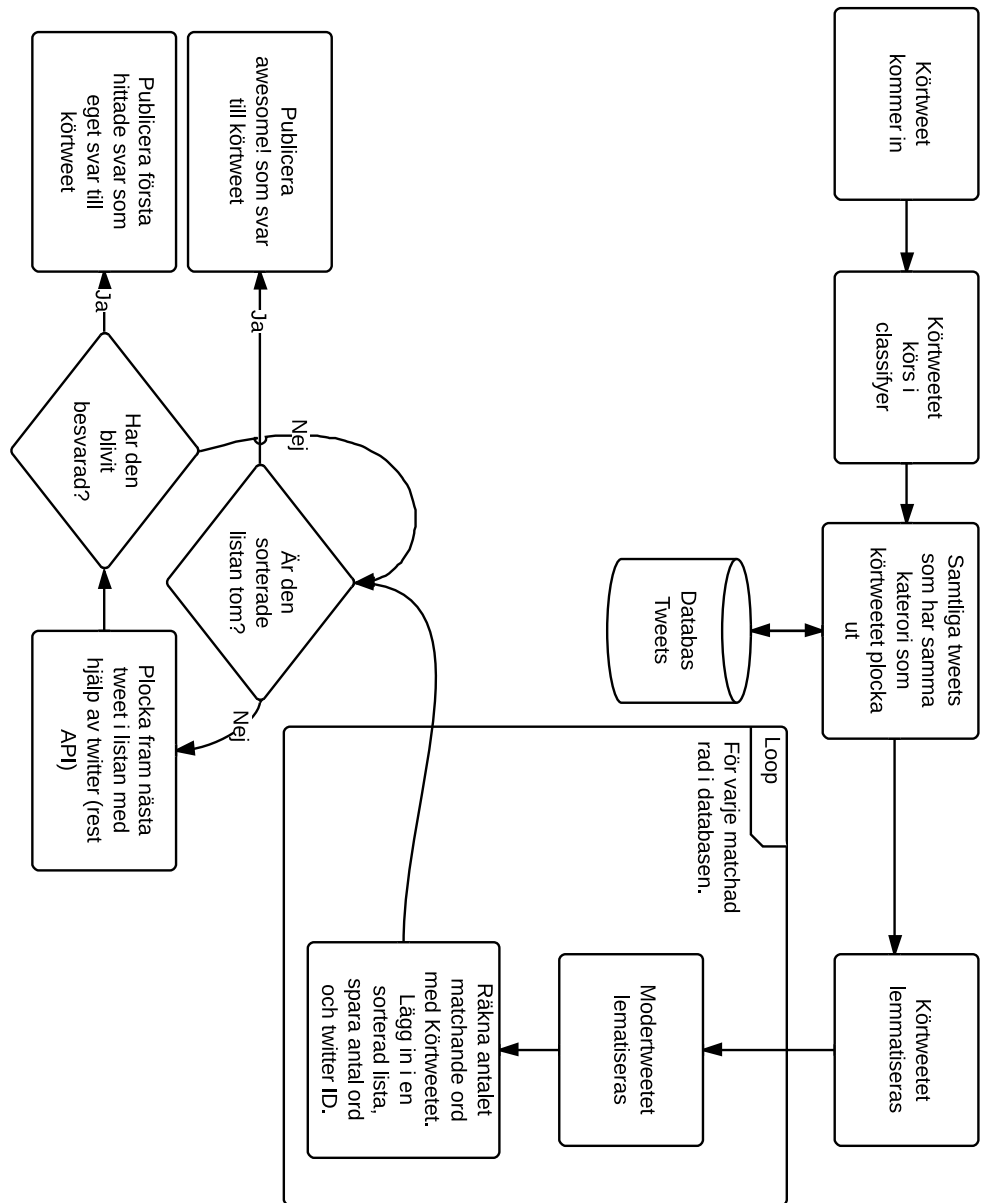
Sembot genomför klassificering av ett tweet i två steg. Först klassificeras tweetets huvudkategori med hjälp av Topics. Därefter används resultatet i den för att göra ytterligare en klassificering för att bestämma underkategori. Hur kopplingen mellan kategorier ser ut illustreras i Figur 1.3. Resultatet blir att varje tweet kan sparas in i databasen med både en huvudkategori och en underkategori.

När sembot tar emot körtweetet är första åtgärden att klassificera det och dess underkategori sparas. Därefter plockar sembot ut samtliga tweets i databasen med samma underkategori. Vid undantagsfallet då det inte finns några modertweets inom underkategorin används istället körtweetets huvudkategori. Sedan processeras de kvarvarande modertweetsen och körtweetet genom att dess ord görs om till lemman, vilket är respektive ords grundform. Därefter matchas körtweetets lemman mot var och ett av modertweetsens lemman. Antalet överensstämmande lemman sparas undan tillsammans med TwitterID i en prioritetslista. Därefter genomförs samma process som beskrivs i det avslutande stycket i Tribot (Se ovan) för att hitta ett svar. Hela processen beskrivs i form av ett flödesschema i Figur: 1.4.



Figur 1.3: uClassify schema.

. METOD



Figur 1.4: Sembot.

Utvärdering

Två tester genomförs för att utvärdera botarna, ett turingtest och ett relevanstest. Anledningen till att dessa tester görs är att botarna ska analyseras på både hur människolika och relevanta dess svar är. Turingtestet är ett vedertaget sätt att utvärdera chatbotar och det är det självklara valet för att testa människolikhet. Människolikhet är en mycket viktig egenskap hos chatbotar, det behövs för att botar över huvudtaget skall kunna förstås av människor och därmed kunna vara användbara. Men relevansen i chatbotars svar har också stor betydelse för användbarheten eftersom att människolikhet inte direkt kan försäkra relevans. Det är denna relation mellan människolikhet och relevans som har lett till att utvärderingen gjorts med både Turingtest och relevanstest.

För att utvärdera hur respektive bot presterar i relevans och människolikhet tas ett tvådelat webbformulär fram. Den första delens mål är att spegla människolikhet. Detta görs genom att 20 slumpmässiga tweets som har blivit besvarade tas ifrån twitter. Dessa processeras därefter av respektive bot och kompletteras med den manuella svaruthämtning som beskrivs i det sista stycket i avsnittet Tribot (Se ovan). Formulärets 20 punkter består av ett originaltweet följt av tre svarsförslag. Ett genereras av respektive bot och det tredje tas från den användare som besvarat originaltweetet. När deltagare genomför testet kan de enbart välja ett av dessa som de anser vara svarsförslaget som troligast är originalsvaret. Efter varje fråga får deltagarna välja hur de kom fram till sitt val. Alternativen är uteslutningsprincipen, magkänsla eller annat. Om användaren valt annat får den en textruta med möjlighet att ytterligare kommentera sitt val.

Den andra delen består utav ytterligare elva tweets med ett svarsförslag från respektive bot. Uppgiften besvaras till skillnad ifrån föregående del genom att dra en slider till ett värde man finner lämpligt på en skala mellan noll och etthundra. Detta för att med relevans som utgångspunkt kunna jämföra de bägge botarna.

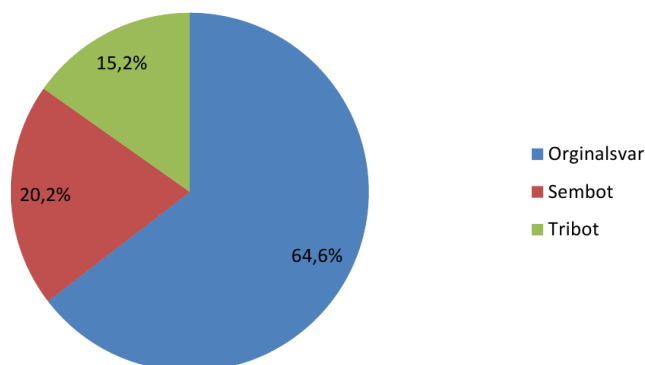
Resultat

Utvärderingen besvarades av 23 personer och samtliga av dessa slutförde hela utvärderingen.

Turingtest

Resultatet av turingtestet redovisas översiktligt med ett diagram som visar testdeltagarnas sammanställda svarsfördelningen mellan sembot, tribot och originalsvaret, se Figur: 1.6. Ur figuren utläses att av alla svar som deltagarna givit har 64,6% tillfallit originalsvaret, 20,2% till sembot och 15,2% till tribot.

. RESULTAT



Figur 1.5: Sammanställd svarsfördelning.

Nedan presenteras exempel från testet som visar då botarna fått dåligt respektive bra resultat på specifika tweets.

Exempel då botarna fått bra resultat:

ex 1: tweet: "irritated, pissed, angry, & annoyed."

sembot: "Parents, problems, friends, fakes, love, crush, dating, responsibility, drama, heartache, failures, stress,"

sembot fick 14 av de 23 svaren.

ex 2: tweet: "Gunna cuddle into bed and act like I'm not insecure."

tribot: "same :/"

tribot fick 11 av de 23 svaren

Exempel då botarna fått dåligt resultat:

ex 1: tweet: "Recipe of the day a lovely Crostini of mozzarella, chilli & lemon super simple and delicious"

sembot: "she was saying rt if you dont judge someone on their sexual preferences tho."

sembot fick 1 av de 23 svaren.

ex 2: tweet: "This makes me so happy honestly the fact that you're dating Morgan is perfect because it gives me an excuse to hang out with you"

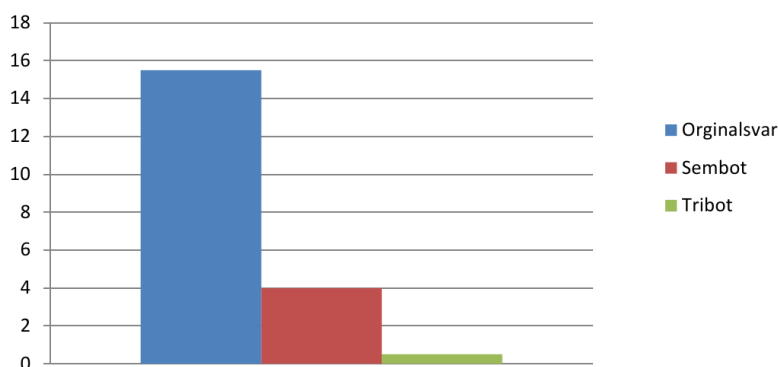
tribot: "hey some of us Voted already....was there by 5.30am!!"

tribot fick 0 av de 23 svaren

Den del som visade på människolikhet visade väldigt stora variationer i kvaliteten på svaren som botarna genererade. Sembot lyckades t.ex. få över 69% av användarna att tro att dess svar var originalsaret på en specifik fråga. Å andra sidan var det bara drygt 4% av deltagarna som trodde att sembots svar var originalet när den presterade som sämst. Tribot presterade något sämre sett till specifika frågor och hade ett förtroende av deltagarna på 0% i sitt sämsta fall och lyckades bara uppnå

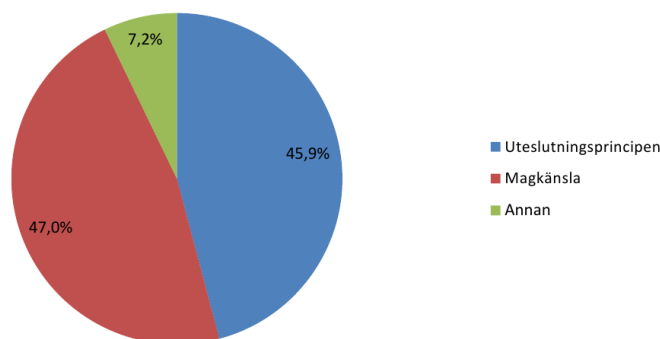
ett deltagarförtroende på strax under 48% som bästa prestation. De riktiga svaren var stabilare på en högre nivå och extremfallen låg på 17% respektive 91%.

Efter sammanställning av deltagarnas svar framkom det att användarna valde originalsvar i 75% av fallen. Detta illustreras i Figur: 1.6. Den sammanställda svarsfördelningen visar också på att användarna i stor utsträckning valt originalsvar och det finns en antydning att sembot presterat något bättre än tribot. Genomförandet av ett oparat t-test har dock visat att detta samband inte är statistiskt säkerställt, $t(38) = 0.974$, $p = 0.336$.



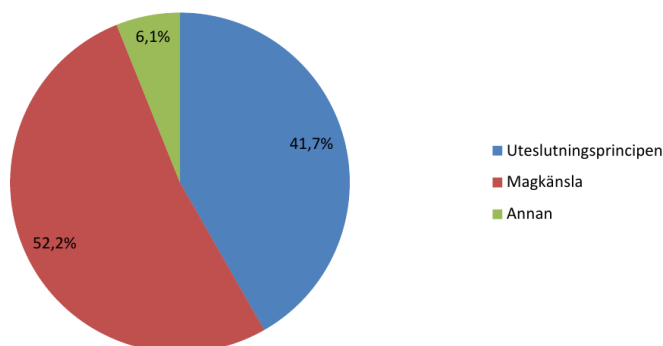
Figur 1.6: Antal frågor där majoriteten av deltagarna bedömde att ett visst svar var originalsvar.

Sammanställningen av anledningarna till deltagarnas svar visade på att magkänsla och uteslutningsprincipen var ungefär lika vanlig anledning och att det tredje valet, annat, stod för mindre än en tiondel av valen. Detta visas i Figur: 1.7. Det kan också jämföras med anledningarna som deltagarna har hänvisat till i fallen då de trots att ett av botarnas svar var originalet. Då har magkänslan haft ett större inflytande än i det genomsnittliga fallet. Detta illustreras också i Figur: 1.8.



Figur 1.7: Sammanställd anledning till svar.

. SLUTSATS

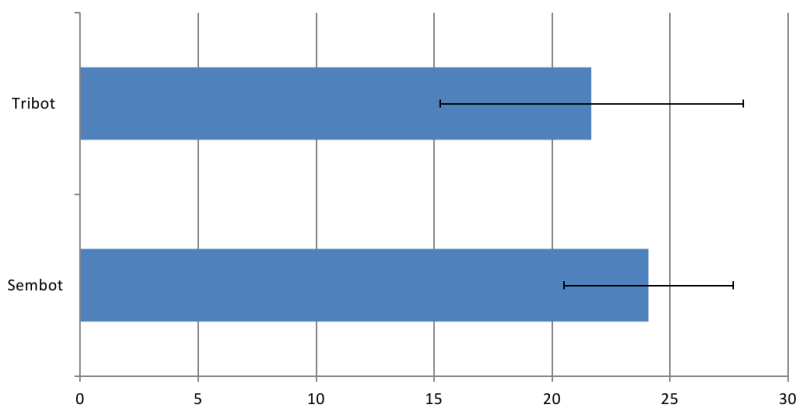


Figur 1.8: Anledning i fallen då deltagarna trott att ett av botarnas svar var originalet.

Relevanstest

Ett operat t-test genomfördes för att jämföra de båda botarnas resultat. Sembot fick högre relevansbetyg av användarna ($M = 24.09$, $SD = 11.87$) än vad tribot fick ($M = 21.67$, $SD = 21.33$), $t(20) = 0,321$, $p = 0,745$. Skillnaden är inte statistiskt säkerställd. Botarnas resultat finns grafiskt representerade i Figur: 1.9.

Även sammanställningen av relevansdelen i utvärderingen visade på stora skillnader från fråga till fråga där tribot uppnådde över 65 i sitt bästa fall men var nere på 5 i sitt värsta. Sembot fick dock ett jämnare resultat med drygt 7 i värsta fallet och 38 i det bästa. Värt att tillägga är dock att sembot enbart hade ett värde under 10 i sammanställning medans motsvarande siffra för tribot var fyra stycken.



Figur 1.9: Genomsnittliga relevansresultat med standardfel.

Slutsats

Till att börja med kan man bekräfta att resultatet motsvarade hypotesen väldigt bra även om botarna inte presterade speciellt bra i något utav testerna. Detta var dock också förväntat med tanke på den begränsade tid som har erbjudits detta projekt.

Det är lätt att se att ingen av botarna i den nuvarande utformningen skulle ha någon rimlig chans att klara av ett turingtest tillämpat genom twitter. Det som dock kan ses som en god indikator på att botarna klarade sig bättre än något som var helt slumpgenererat var att det vanligaste svarsalternativen på hur man kommit fram till sitt val var magkänsla snarare än uteslutningsprincipen vilket hade tytt på att de genererade svaren hade varit helt orlaterade. Man kan styrka detta resonemang med siffrorna som visar på att folk litat mer på sin magkänsla när de har gissat på svaret från en av botarna.

Relevanstestet visade på att deltagarna inte tyckte att svaren som botarna genererade var speciellt relevanta. Även om det inte gick att statistiskt säkerställa något om prestationen så antyder resultatet av den här studien att sembot gjorde ett något bättre jobb än tribot.

För att få bättre resultat finns det många potentiella förbättringsfaktorer. För att kunna få ett statistiskt säkerställt resultat skulle det definitivt behövs fler deltagare i undersökning. Dessutom marknadsfördes utvärderingen enbart på facebook och där primärt gentemot dataloger på KTH, det är därför troligt att deltagarna inte var ett representativt urval ur populationen vilket är något som bör beaktas om man ska göra vidare studier inom området.

I grund och botten tror vi fortfarande att angreppssättet kan vara intressant för framtiden och att sociala medier som sparar konversationslik data kan vara en potentiell framtid för att skapa mer relevanta botar. Den kanske mest uppenbara förbättringspunkten för båda botarna vore självklart en större databas med moder-tweets. Med den utformning som sembot har idag skulle detta dock kunna vara svårt på grund av att tiden som läggs ner för lematisering redan i dagläget är rätt lång, ungefär 15 min. Denna tidsfördröjning är inte något större problem när enstaka tweets ska besvaras men den kan ställa till problem vid regelbunden användning. Detta problem skulle man dock kunna gå runt igenom att uppdatera databasen så att den också håller den lematiserade strängen i minnet vilket gör att beräkningarna kan skjutas från när svaren genereras till när databasen fylls på. Dessutom har vi identifierat vad vi anser vara en allvarlig brist i sembot att den inte plockar bort ord när ur sökningen när det har blivit använda. Detta ledde i vissa fall till att tweets som egentligen låg långt ifrån varandra ansågs vara lika.

Ett av tribots största problem i utvärderingen var att den ofta inte hittade svar från ett modertweet och därför fick returnera standardsvaret. Orsaken till detta problem är antagligen att trigram är för stora för de tweets vi behandlat, som i vissa fall är ganska små. Detta i kombination med att antalet tweets i databasen är för litet leder till att trigram från modertweets inte går att matcha mot körtweetet. Problemet skulle kunna undvikas genom att istället för att använda trigram dela upp modertweets i bigram, det vill säga i delar med två ord. Det skulle leda till fler matchningar mellan modertweets och körtweetet och därmed finna fler potentiellt besvarade modertweets. Problem som uppstår vid ett sådant byte är att det krävs mer hårddiskutrymme och att processeringstiden för tribot blir längre.

För tribot finns det också en förbättringspotential utöver större databas och byte från trigram till bigram i att kunna försöka låta algoritmen anpassa sig mindre efter tweetsens längd. Detta hade eventuellt kunnat anpassats genom att modifiera Levensthein Distance så att längre sammanhängande stycken premieras.

. SLUTSATS

Avslutningsvis så kan vi konstatera att vi inte kunna ge ett statistiskt säkert svar till vår frågeställning men att resultaten i den här studien pekar emot att semantisk textklassificering skulle kunna vara en lämpligare sökalgoritm än trigram.

Avslutningsvis så kan vi konstatera att vi inte kunna ge ett statistiskt säkert svar till vår frågeställning men att resultaten i den här studien pekar emot att semantisk textklassificering skulle kunna vara en lämpligare sökalgoritm än trigram.

Litteraturförteckning

- [1] Agrawal Kush. *Study of the phenomenon of the Moravec's paradox*. [Website, Digital Paper]. [Visited 5 mars 2013]. Hittas på: <http://arxiv.org/ftp/arxiv/papers/1012/1012.3148.pdf>. [C2010].
- [2] Bird Steven, Klein Ewan, Loper Edward. *Natural Language Processing with Python*. [Website, Digital Book]. [Visited 5 mars 2013]. Hittas på: <http://nltk.googlecode.com/svn/trunk/doc/book/ch06.html>.
- [3] Bloggen computer minds. *When the machines talked to each other*. [Website]. [Visited 5 mars 2013]. Hittas på: <http://www.computableminds.com/post/chatbot/eliza/parry/skyнет/human-language/natural-language>. [c2010].
- [4] Carpenter Rollo. *Cleverbot*. [Website]. [Visited 5 mars 2013]. Hittas på: <http://cleverbot.com/>. [C2012].
- [5] Chiarella Joseph, O'Brien Jason. *An Analysis of Spam Filters*. [Website, Bachelor Essay]. [Visited 5 mars 2013]. Hittas på: <http://web.cs.wpi.edu/~claypool/mqp/spam/mqp.pdf>. [c2003].
- [6] Chu Zi, Gianvecchio Steven, Wang Haining, Jajodia Sushil. *Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?* [Website, Digital Paper]. [Visited 5 mars 2013]. Hittas på: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6280553&tag=1>. [C2012].
- [7] Gene Loebner Hugh. *Computing Machinery and Intelligence*. [Website]. [Visited 5 mars 2013]. Hittas på: <http://www.loebner.net/Prizef/TuringArticle.html>.
- [8] Goyvaerts Jan. *Regular Expression*. [Website]. [Visited 5 mars 2013]. Hittas på: <http://www.regular-expressions.info/>. [c2013].
- [9] Hypergrid Buissnes. *Paper: Chatbots now smarter, more connected, and more emotional*. [Website]. [Visited 5 mars 2013]. Hittas på: <http://www.hypergridbusiness.com/2010/11/paper-chatbots-now-smarter-more-connected-and-more-emotional/>. [c2010].

LITTERATURFÖRTECKNING

- [10] Japan today. *Chatbots teach English conversation*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.japantoday.com/category/lifestyle/view/chatbots-teach-english-conversation>.
- [11] Joseph Weizenbaum. *Eliza - A Rogerian therapist*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.chatbots.org/chatbot/eliza/>. [c2013].
- [12] Kågström Jon, Karlsson Roger, Kågström Emil. *uClassify*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.uclassify.com/Default.aspx>.
- [13] Levenshtein Vladimir. *Levenshtein distance*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.levenshtein.net/>.
- [14] McCallum Andrew, Nigam Kamal. *A Comparison of Event Models for Naive Bayes Text Classification*. [Website, Digital Paper].[Visited 5 mars 2013].Hittas på: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.65.9324&rep=rep1&type=pdf>.
- [15] Netscape. *dmoz Open Directory Project*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.dmoz.org/>. [C2013].
- [16] Oracle Corporation. *Java*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.java.com/en/>.
- [17] Oracle Corporation. *MySQL - The world's most popular open source database*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.mysql.com/>. [C2013].
- [18] Plusnet plc. *Plusnet Virtual Assistant - Jess*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.plus.net/help/>. [].
- [19] Shuyo Nakatani. *Language Detection Library for Java*. [Website].[Visited 5 mars 2013].Hittas på: <https://code.google.com/p/language-detection/>. [C2011].
- [20] SpeakGlobal, Ltd. *Meg the English Teacher*. [Website].[Visited 5 mars 2013].Hittas på: http://www.speakglobal.co.jp/member/bots/meg_trial.php. [C2013].
- [21] Toys R Us. *Toys R Us Virtual Assistant - Emma*. [Website].[Visited 5 mars 2013].Hittas på: <http://www.toysrus.co.uk/info/help.jsf>. [C2013].
- [22] A.M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.
- [23] Twitter. *REST API Rate Limiting in v1.1*. [Website].[Visited 5 mars 2013].Hittas på: <https://dev.twitter.com/docs/rate-limiting/1.1>. [C2013].
- [24] Twitter4J. *Twitter4J an unofficial Java library for the Twitter API*. [Website].[Visited 5 mars 2013].Hittas på: <http://twitter4j.org/en/index.html>.