Automatisk klassificering av åsikter

J E N S A R V I D S S O N
o c h S I M O N S T R Ö M

KTH Datavetenskap
och kommunikation

Examensarbete
Stockholm, Sverige 2012

Automatisk klassificering av åsikter

J E N S A R V I D S S O N
o c h S I M O N S T R Ö M

DD143X, Examensarbete i datalogi om 15 högskolepoäng
vid Programmet för datateknik 300 högskolepoäng
Kungliga Tekniska Högskolan år 2012
Handledare på CSC var Johan Boye
Examinator var Mårten Björkman

URL: www.csc.kth.se/utbildning/kandidatexjobb/datateknik/2012/
arvidsson_jens_OCH_strom_simon_K12009.pdf

Kungliga tekniska högskolan
Skolan för datavetenskap och kommunikation

KTH CSC
100 44 Stockholm

URL: www.kth.se/csc

1 Abstract

The Internet is full of opinions. Reading an opinionated text and judging the author's stance is something humans are generally expected to be able to do. In this report, the authors build and test a computer system able to read movie reviews and judge the author's opinion of the film, thereby being able to give numerical grades to any film that has a decent number of reviews. The authors try different machine learning algorithms to judge which is best suited for this task, and compare the results to numerical movie grades found on the Internet Movie Database. The algorithm Voted Perceptron comes out the winner, and the system built (named Judge Dr Film) manages to judge films quite close to their IMDb scores. The full source code can be found at `https://github.com/sajmoon/Dr-Judge-Film`.

2 Statement of collaboration

Projektet har en naturlig uppdelning eftersom det handlar om maskininlärning och insamling av data till maskininlärningen. Jens implementerade maskininlärningen och Simon datainsamlingen. Respektive person har även skrivit om de implementations- och bakgrundsdelar som tillhör de olika områdena. Simon har skrivit Möjliga förbättringar 8.2.1 och Möjliga Praktiska användningsområden 8.2.2, och Jens har skrivit resultatdelen.

Innehåll

Figurer

Tabeller

3 Introduktion

Internet är fullt av åsikter, och för en människa är det en enkel sak att läsa en recension eller ett blogginlägg och få en bild av författarens åsikt. För en dator är det dock en annan historia, och i den här uppsatsen skall vi undersöka huruvida ett system kan skapas som automatiskt kan avkoda filmrecensioner och översätta dem till ett numeriskt betyg för filmen. De åsikter som systemet undersöker kommer i första hand från korta kommentarer från Twitter, IMDb och Facebook, och systemet drar nytta av moderna maskininlärningsmetoder för att avgöra huruvida en kommentar är relevant (dvs att den faktiskt har att göra med en viss film), och vilken värdeladdning orden i kommentaren har (jämför meningen "Bästa filmen jag nånsin sett" gentemot "Schysst rulle!"). Utifrån dessa bedömningar skall systemet kunna beräkna ett numeriskt betyg på en film. Förhoppningen är att ett sådant system kan utvecklas för att täcka även andra problemdomäner. Man kan till exempel tänka sig ett liknande system för att göra politiska opinionsundersökningar, recensera restauranger eller i princip vad som helst som kan betygsättas. Målet med uppsatsen är helt enkelt att avgöra huruvida ett sådant system kan skapas, hur det ska vara byggt, och vad det kan användas till.

4 Inledning

4.1 Syfte

Syftet med denna uppsats och skapandet av systemet är att utröna i vilken grad en rättvisande och korrekt analys av åsikter i textform kan utföras automatiskt och huruvida extraktion av en relevant mängd data går att utföra ur redan existerande databaser av text på Internet. Att systemet är begränsat till ett specifikt område, film, är delvis för att begränsa problemområdet, men även för att det redan existerar lättförståeliga åsikter om filmer i form av numeriska betyg. Systemets effektivitet och kvalitet kan därmed mätas på ett tillfredsställande sätt. Ett välfungerande system av denna sort kan sedan rimligtvis användas inom andra problemdomäner för att "känna av" åsikter hos allmänheten, inom alla möjliga områden såsom politik, marknadsföring eller kanske huruvida fel låt vann?

4.2 Problemformulering

Att automatiskt kunna läsa, bedöma och betygsätta skrivet språk har många användningsområden utöver ren underhållning. Vi ska i den här rapporten presentera ett system som utnyttjar maskininlärningsmetoder för att automatiskt betygsätta filmer utifrån skrivna recensioner. Problemet är: går detta att göra på ett pålitligt, effektivt och användbart sätt?

5 Bakgrund

5.1 Datainsamling

Insamlandet av data är viktigt för utförandet av projektet. För att kunna göra en verklig analys av användare runt om i världen behöver vi hitta stora mängder data med åsikter. Det är inte alltid enkelt då mycket av användarkommentarer finns i exempelvis bloggar eller andra icke-sorterade källor som inte är enkla att tolka. Potentiellt är det svårt att specificera vilken film kommentarerna tillhör om ens texten handlar om film.

Möjliga källor är Twitter, IMDb,samt Rottentomatoes. Twitter är en mikroblogg-tjänst där den maximala längden för varje inlägg är 140 tecken.[12] IMDb och RottenTomatoes är tjänster med sammanställd information om många filmer. Dessa innehåller bland annat betyg, skådespelare, utgivningsår och mycket mer. Där finns även stora mängder användarrecensioner vilka ofta är långa texter med detaljerade kommentarer om filmen. Dessa kommentarer är länkade till en viss film vilket gör identifiering av film lättare. Tweets har hashtaggar (tecknet # följt av ett nyckelord) och både IMDb och RottenTomatoes har ett id-nummer till filmen.

Twitters inbyggda begränsning på 140 tecken gör tjänsten ultimat för sociala utrop men inte för hela recensioner. De utrop som kan vara relevanta är i stil med "Nu ska jag se världens bästa film #Alien". För att identifiera tweets kan hashtaggar användas, och det gäller att den är unik för filmen. Problem som kan uppstå är t.ex. att termen finns med i någon tweet men det inte alls handlar om filmen. Ett exempel är filmen "Up"[5] vars titel även är ett ganska vanligt ord i löpande text. Följande tweet är ett exempel på hur fel det kan bli:

> "If you're reading this it means you've already given **up** on interacting with your family today. Wise choice."[9]

Texten har inget med filmen "Up"[5] att göra. Hashtaggar är Twitters sätt att gruppera och kategorisera utropen med varandra som behandlar samma ämne.

5.2 Data mining

Termen data mining avser processen då en dator lär sig att läsa ut innebörden av till exempel en mening eller annan form av mänsklig text. Texter, till skillnad från regelrätta betygsättningar eller facebook-"gillaknappen", är ofta tvetydiga. Det kan därför vara svårt även för en människa att förstå exakt vad motparten ville förmedla. Målet med applikationen är att ur varje enstaka kommentar förstå hur mycket personen gillar filmen ifråga. Dessa kommentarer är ofta korta meningar, men kan även vara längre recensioner från användare, men de måste alltid tydas. På en skala ett till tio, hur skulle

du betygsätta meningen “Filmen är ganska bra”? Eller “Filmen var bra, men jag somnade”. Ingen av dessa har självklara översättningar till ett betygssystem. Somnade personen för att det faktiskt inte var så kul, eller var han/hon trött?

5.3 Maskininlärning

Maskininlärning är en samlingsterm för studier av algoritmer som grundar sitt beteende på mönster i insamlad data. Enkelt uttryckt kan man säga att en maskininlärningsalgoritm ‘studerar’ data, och använder det den lärt sig av tidigare data för att bedöma ny data, på ett liknande sätt vi människor gör. Algoritmen kan utföra förutsägelser och bedömningar om information den inte sett förut, och kan därför sägas inneha artificiell intelligens, det område som den akademiska disciplinen maskininlärning faller under. Det finns en mångfald sätt att implementera inlärning hos ett program, och de delas in i ett antal subtyper baserade på hur inlärningen går till: övervakad, oövervakad, semiövervakad, förstärkande, transduktion, och multi-task-inlärning. I arbetet med denna uppsats har vi begränsat oss till att arbeta med algoritmer som faller under kategorin övervakad inlärning.

Övervakad inlärning Övervakad inlärning innebär att ett program lär sig känna igen mönster utifrån givna träningsexempel. Klassificeringsalgoritmer som arbetar på detta sätt används för att klassificera objekt med bestämda attribut. För att den skall fungera måste den matas med etiketterad data, så att den kan bygga upp ett register över vilka attribut som är vanliga hos vilka objektsklasser. Om algoritmen matas med följande träningsdata:
plommon: storlek=3,färg=lila,smaskighet=7;
äpple: storlek=6,färg=röd,smaskighet=4;
apelsin: storlek=8,färg=orange,smaskighet=6;
och liknande instanser som alla har värden snarlika dessa, kommer algoritmen till slut att förknippa en frukt som har högt smaskighetsvärde och liten storlek med plommon, och därför förmodligen klassificera en jordgubbe som ett plommon. Algoritmen bygger alltså en matematisk funktion från indata till önskad utdata utifrån de exempel den tränas med. Ny indata som matas in i funktionen producerar således en kvalificerad gissning om vilken klass indatan tillhör. Denna funktion kallas för en klassificerare, då dess uppgift är att klassificera data. Klassificerare finns i ett stort antal varianter, som bygger på olika matematisk grund, och de algoritmer som utnyttjar övervakad inlärning är generellt döpta efter sin klassificerare.[14] Vi har i denna rapport använt oss av två olika klassificerare som båda använder övervakad inlärning, Naive Bayes och Support Vector Machines.

5.3.1 Klassifikationsalgoritmer

Naive Bayes-algoritmen Naive Bayes-algoritmen används ofta för att klassificera dokument, och den gör detta genom att se ett dokument som en samling ord(bag-of-words) utan inbördes förhållande till varandra. För varje dokumentklass (exempel på dokumentklasser kan vara negativa och positiva recensioner, nöjes-, sport- eller nyhetsartiklar etc) har varje ord en viss sannolikhet att dyka upp, och denna sannolikhet räknas ut med hjälp av träningsdatan på följande sätt: Ponera att vi har tillgång till 100 filmrecensioner där hälften har etiketten positiv, och hälften negativ. Anta sedan att ordet "överdrivet" dyker upp i 10 av de positiva, och 30 av de negativa recensionerna. Detta ger en sannolikhet på $1/5$ för att ordet överdrivet dyker upp i ordklassen positiv, och $3/5$ att det dyker upp i ordklassen negativ. Sannolikheten för att ett helt dokument är av klassen positiv är sedan helt enkelt de multiplicerade positiv-sannolikheterna för varje ord, vilka sedan normaliseras för en total sannolikhet av 1. För att fortsätta exemplet ovan, ponera att en recension består av meningen: "Den var överdrivet bra". Efter träning på ett antal dokument är sannolikheterna för att dessa ord dyker upp i klassen positiv följande:

$$den = 9/10, var = 8/10, \ddot{o}verdrivet = 1/5, bra = 7/10$$

Motsvarande sannolikheter för klassen negativ är:

$$den = 9/10, var = 8/10, \ddot{o}verdrivet = 3/5, bra = 2/10$$

Algoritmen kallas för naiv, eftersom den antar oberoende mellan attribut (i det här fallet de olika orden). Vi antar detta och multiplicerar helt enkelt sannolikheterna:

$$P(positiv) = 9/10 * 8/10 * 1/5 * 7/10 = 0.1008$$

$$P(negativ) = 9/10 * 8/10 * 3/5 * 1/10 = 0.0432$$

Vi normaliserar dessa så den totala sannolikheten uppgår till 1 och erhåller:

$$P(positiv) = \frac{0.1008}{0.1008 + 0.0432} = 0.7 = 70\%$$

$$P(negativ) = \frac{0.0432}{0.1008 + 0.0432} = 0.3 = 30\%$$

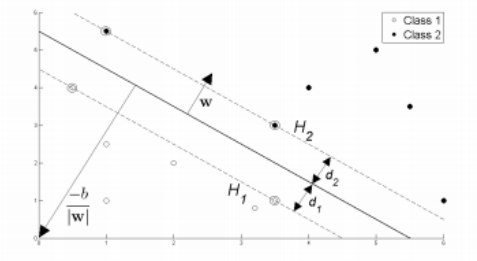
Sannolikheten att recensionen är positiv är alltså större än att den är negativ, och vi klassar den som sådan, vilket i det här fallet visade sig vara korrekt.[16][8]

Support Vector Machines Support Vector Machines, förkortat SVM, används i maskininlärning för att känna igen mönster när man klassificerar data[15]. SVM kan används för att klassificera data som binära, dvs den ger indatan en av två klasser, samt linjärt separerbara[2]. Att de är binära skulle t ex kunna vara recensioner av filmer där de antingen är positiva eller negativa. Vi återkommer till exakt definitionen av separationslinjen lite senare.
För ett träningsset L, har varje instans x_i ett antal attribut. Dessa attribut kan vara ord, eller färg och storlek, beroende på vad vi vill klassificera efter. x_i är en användarrecension och den tilldelas ett värde y som kan vara 0(negativ) eller 1(positiv) vilket beskriver om den är bra eller dålig. Dimensionen vilken vi definierar med R^d är beroende på antalet ord i varje recension.

$$\{x_i, y_i\} where\ i = 1...L; y_i \in \{0, 1\}, x \in R^d$$

Antag att vi ska klassificera äpplen, vilka har två attribut, Färg och Storlek. Då har vi två dimensioner. Med hjälp av dessa attribut ska det klassificeras hur goda respektive äpple ser ut att vara. Vi kan då representera datan grafiskt med hjälp av följande figur av Tristan Fletcher[2]: I Figur 5.3.1 har

Figur 1: Support Vectors med tillhörande hyperplan.



två axlar vilka motsvarar de två attributen vi har, t ex färg och storlek. Figuren är baserad på träningsdata där vi vet vilken klass respektive x_i. För varje ny $x_{(i+1)}$ vi vill se vilken klass den tillhör och därför behöver vi en separationslinje. Separationslinjen, eller ett hyperplan om objekten har fler attribut än två, kalas H och skiljer de två klasserna åt. Hyperplanen, H1 och H2, är de plan som utgörs av två x_i som är närmast H kallade Support Vectors[2]. d1 och d2 är avståndet från H till H1 respektive H2 och har det gemensamma namnet SVM margin[2]. För att hitta det H som ligger längs ifrån alla Support Vectors vill vi maximera SVM Margin. Med hjälp av denna linje, eller hyperplan, kan vi klassificera den riktiga datan beroende på vilken sida den hamnar. Detta ger oss vår Support Vector Machine.

Voted Perceptron En Perceptron är en sorts artificiellt neuronnätverk som kan utföra binär klassificering. Funktionen som används är ett antal numeriska vikter som indatan multipliceras med (vektormultiplikation). Dessa

vikter kalibreras med hjälp av träningsdata för att funktionen ska klassificera indata i enlighet med träningsdatans klassificeringar. Skillnaden mellan Voted Perceptron och en enkel Perceptron är att Voted Perceptron lagrar fler än en uppsättning vikter, och ger varje uppsättning vikter (viktvektor) en viss beslutstyngd (ett värde oberoende av vilket värde själva vikterna har, dvs deras 'vanliga' tyngd) som reflekterar hur väl de presterar på träningsdatan. När funktionen ska klassificera en ny instans, klassificerar alla lagrade viktvektorer instansen, och deras klassificering räknas som en 'röst' där viktvektorns beslutstyngd avgör hur mycket dess röst är värd. Den sammanlagda viktade majoritetsrösten bland alla lagrade viktvektorer avgör vilken klass instansen klassas som.[3]

5.3.2 WEKA

WEKA är mjukvara skriven i språket Java som innehåller implementationer av en mängd maskininlärningsalgoritmer och verktyg för att visualisera och analysera data med hjälp av dessa. Förutom de många inlärningsalgoritmerna finns verktyg för stemming, dvs tekniker för att reducera ord i en text till dess grundformer för att underlätta igenkänning, och analysverktyg för att testa olika algoritmers beteende och effektivitet med olika sorters data. Ramverket stöds av ett API som tillåter att implementationerna används i eget skrivna Javaprogram. WEKA är skapat av Machine Learning Group vid Waikato-universitetet på Nya Zeeland, och distribueras med helt öppen källkod under Creative Commons ShareAlike-licens, vilket innebär att den får användas fritt, även för kommersiella syften, och dessutom modifieras under förutsättning att modifikationen släpps under samma licens.[4]

5.4 Jämförelser och bedömningsmetoder

5.4.1 Precision och recall

Precision och recall är två statistiska metoder för att med hjälp av exempel som på förhand är klassificerade av en pålitlig källa(t.ex. en människa) avgöra hur väl en klassificerare klassificerar objekt tillhörande en klass.[13] En klassificerare kan, givet ett objekt, ge ett svar per möjlig klass som tillhör en av fyra kategorier: falsk negativ(fn), falsk positiv(fn), sann positiv(tp, true positive) och sann negativ(tn, true negative). Falsk och sann innebär i den här meningen att svaret som gavs var korrekt, och positiv eller negativ anger vad klassificeraren antog att svaret var, dvs dess förutsägelse. För att illustrera detta köper vi ett grönt äpple och ställer frågan till vår klassificerare: är det här ett grönt äpple? Svarar klassificeraren "ja", är svaret sant positivt, om "nej", falskt positivt. Frågar vi om äpplet är rött, är svaret "ja" falskt positivt, och "nej" sant negativt. Måttet precision definieras som

$$\frac{tp}{tp + fp}$$

och är alltså ett mått på korrektheten när ett objekt har klassificerats(dvs. om klassificeraren säger att äpplet är rött, hur stor är sannolikheten att den har rätt?), och recall som:

$$\frac{tp}{tp + fn}$$

vilket mäter klassificerarens förmåga att klassificera objekt ur en given klass. Tp är alla de korrekt klassade instanserna av en given klass, och $tp + fn$ är det totala antalet objekt av klassen.[16]

5.4.2 IMDb-jämförelse

IMDb Internet Movie Database är en gigantisk databas med all tänkbar information rörande film och tv-serier, från regissörer och skådespelare till manusförfattare och assistenter medverkande i närapå alla filmer som någonsin spelats in(åtminstone i västvärlden). På IMDbs webbtjänst imdb.com ges även möjlighet för användare att lämna kommentarer och sätta betyg på filmer, och IMDb är idag i särklass den största filmsajten på internet. [7] Jämförelse Varje film på IMDb har ett betyg, som är grundat på medelvärdet av användarnas röster. Detta betyg är i skalan 1.0-10.0, och det är även denna skala som denna rapports system bedömer filmer efter. En jämförelse mellan IMDb-betyget och systemets betyg fungerar som en bekräftelse på att systemets bedömning är någorlunda korrekt, då de i många fall speglar en omröstning gjord av tusentals användare.

6 Utförande

6.1 Implementation av datainsamlingsmodulen

Datainsamlingsmodulen skrevs i språket Ruby. Implementationen inleddes med att hämta hem så kallade tweetsfrån Twitter. Twitter säger sig ha ett API för nedhämtning av tweets[12] och denna fungerade, men returnerade ett väldigt litet antal tweets. Till Ruby finns ett gem kallat “twitter"vilket underlättar hanteringen av Twitter i applikationer skrivna i Ruby.[10] Detta gem är även den begränsad av Twitters API, vilket har hindrat författarna från att använda den. Däremot fick applikationen ut fler träffar när en egen metod utvecklades för detta syfte. Metoden gör en sökning på Twitter, vilken returnerar ett JSON-objekt med 100 stycken tweets per sida. Sökningen görs med en GET-förfrågan till adressen `http://search.twitter.com/search.json?q=%23test&page=32&rpp=100` där parametern query är hashtaggen, page mellan 1 och 32 och rpp[1] är satt till max vilket är 100 stycken. Resultatet returneras i ett JSON-objekt vilket sparas ner. Parsning"[2] av JSON finns

[1] Responses per page
[2] tolkning

i ett gem kallat JSON. Då både IMDb och den likartade tjänsten Rottentomatoes är baserade runt filmer så har varje film ett unikt id och kan referera till. För IMDb nås en enskild film enklast genom www.imdb.com/ttid och tillhörande recensioner finns under /reviews. Dessa tjänster är väldigt lika varandra men då IMDb har fler recensioner valde författarna att enbart använda IMDb och Rottentomatoes valdes bort. I IMDb fall vet vi att recensionen som vi hittat tillhör just denna film. Inte bara film titel utan utgivnings år finns tillgängligt för utökad kontroll. Filmen "Världarnas krig" är en film där det finns minst en nyinspelning som då skiljer sig i utgivningsår.[6] Att extrahera data ur denna tjänst visade sig dock svårare än vi trott. IMDb har inget allmänt API som tillåter access till datan de har samlat in. Det gem till Ruby vilket planerats användas visade sig inte heller ha funktionen att insamla recensioner. Denna funktionalitet implementerade författarna till i en lokalversion av det tilläggspaketet. Lösningen blev att använda sig av deras ordinarie hemsida där IMDb gör stor del av informationen tillgänglig. För varje film på IMDb, vilka alla har ett id i formen av ttnummer, betydde det att adressen *www.imdb.com/tt{nummer}/reviews* behövde läsas in. Från den sidan kunde därefter de 10 första användarrecensionerna läsas in. Efterföljande recensioner följer med parametern ?step=10 för att specificera vilken recensionen att visa. Samtidigt är html-sidan skrivet utan css-klasser eller andra regelbundna kännetecken, utan består endast av en följd html paragrafer, <p>. Även om IMDb-gemet som användes och modifierades saknade recensionsinformation så hade den mycket av den andra informationen som behövdes inklusive betyg, utgivningsår samt titel. Den data som hämtades från både Twitter och IMDb sparas ner var för sig i en textfil för att sedan behandlas av applikationen.

6.2 Implementation av maskininlärningssystemet

Maskininlärningsdelen av vårt system, med projektnamnet Judge, är skrivet i Java med hjälp av ramverket WEKAs Java-API[1]. Det utnyttjar ett relativt simpelt konsolgränssnitt, där användaren får välja alternativ från en meny med hjälp av tangentbordet. De alternativ som presenteras är att välja varifrån Judge ska hämta recensionsdata, välja olika typer av träningsdata, vilken klassificerings-algoritm som ska användas, och alternativet att starta bedömningen av de recensioner som är valda. Det finns även ett alternativ för att evaluera den valda algoritmens prestanda. Datan som använts för träning av algoritmerna är tagen från en datamängd kallad Polarity Dataset, sammansatt av forskare från Cornell University. Polarity Dataset innehåller 1000 recensioner som klassats som positiva, och 1000 negativa, och dessa har vi delat upp i två separata mängder med 500 positiva och 500 negativa i vardera mängd.[11] Den ena mängden har använts för träning, och den andra för test och evaluering av algoritmer. WEKAs Java-API sköter mycket av den tunga implementationen i Judge, framför allt formatering av data,

klassificering och att ta fram statistik för evaluering från resultaten, vilka utförs med ett antal metodanrop.[4]

6.2.1 Betygssättning

Varje film som ska bedömas har en egen mapp, som innehåller all data som ska klassificeras i form av varsin fil för varje enskild åsikt (dvs. en fil per recension). Den algoritm som är vald i Judge klassificerar in all denna data i klassen negativ eller positiv, och räknar varje fil som en enskild instans. Judge räknar sedan ihop antalet positiva recensioner, och medelvärdet av dessa utgör basen för filmens betyg enligt: Betyg = Positiva recensioner/antal recensioner * 10. Ett medianbetyg räknas dessutom ut, och multipliceras även den med tio för att följa IMDbs betygssystem.

6.3 Interaktion mellan komponenterna

De två komponenterna i Judge Dr. Film, datainsamlingsmodulen och bedömningsmodulen Judge, är separata men beroende av varandra. För att bedöma en film krävs att datainsamlingsmodulen startas först för att samla in data, och därefter att Judge startas och ges sökvägen till denna data för att bedöma den. Systemet består alltså inte av ett stort integrerat program, utan av dessa två separata delar, något som gör att de båda kan användas för andra syften och i andra sammanhang, till exempel för bedömningar inom andra områden eller någon annan analys av recensionsdata.

6.4 Testning

De kriterier som använts vid testning av algoritmer att använda för systemet, på träningsdatan och på vår betygssättningsmetod är höga värden på måtten precision och recall, och huruvida slutbetyget på ett urval av filmer ligger rimligt nära dessa filmers betyg på IMDb. I de inledande faserna av implementationen gjordes ett snabbt test av alla klassificerar-implementationer som finns tillgängliga i WEKA, för att utröna vilka som var kompatibla med vår data, och vilka som kunde tränas på vår träningsdata inom rimliga tidsgränser på en vanlig kontorsdator. Av ursprungligen 50 algoritmer sållades antalet ner till de 3 olika algoritmer som fungerade, kunde köras på rimlig tid och visade bäst resultat mätt i precision och recall. Dessa algoritmer var Naive Bayes, Support Vector Machines och Voted Perceptron, och det är med dessa som evalueringen av systemet har utförts.

7 Resultat

Systemet testades mot ett antal slumpmässigt valda filmer, på datan från Twitter och IMDb. Nedan redovisas resultaten i form av tabeller som visar

jämförelserna mot IMDb-betyg, och tabeller som visar mått på precision och recall. Varje test utfördes med en av de tre utvalda algoritmerna Naive Bayes, Support Vector Machines och Voted Perceptron, och jämförelser mellan dessa redovisas vid sidan om tabellerna.

7.1 Twitterdata

Trots en stor datamängd från Twitter gav datan undermåliga testresultat, och datakällan bedömdes som opålitlig. I samtliga tester av Twitter-recensioner gav systemet betyg som skiljde sig markant från IMDb-betygen, och mätvärden på Precision och Recall var även de låga nog att utesluta en fungerande klassificering. Författarna valde därför att fokusera på IMDb-recensionerna och resultatet som erhölls av dessa, men en bild av Twitter-resultatet medtogs för illustrationssyfte. I tabell 1, 2 och 3 visas systemets bedömning av Twitter-recensioner om ett litet urval filmer med alla tre algoritmerna, och i samtliga fall visas ett markant fel. Både medelvärdet, medianen, och deras motsvarande fel relativt IMDb-betyget visas nedan. Anledningen till felet som uppstod av datan från Twitter berodde på att tweets med en films hashtagg visade sig relativt sällan tillhörde just filmen utan var helt orelaterade och därför blev resultatet missvisande.

Tabell 1: Bedömning av Twitter-data med Naive Bayes

Titel	IMDb-betyg	Medelvärde	Median	Medelfel	Medianfel
Chaplin (1992)	7.3	4.1	4.0	3.17	3.27
Dazed and...	7.6	2.4	2.3	5.21	5.34
Dummy (2002)	6.7	2.1	2.2	4.57	4.47
Forrest Gump (1994)	8.7	5.2	5.0	3.51	3.73
Hobo with a Shot...	6.2	2.5	2.9	3.71	3.26
In Time (2011)	6.5	5.5	5.8	1.02	0.67
Inception (2010)	8.8	6.6	8.7	2.24	0.08
The Fifth Element	7.5	3.2	1.9	4.25	5.6
The Shawshank...	9.2	2.6	3.8	6.59	5.45
Zoolander (2001)	6.4	3.2	3.3	3.15	3.13
Totalt:				3.7	3.5

Tabell 2: Bedömning av Twitter-data med Support Vector Machines

Titel	IMDb-betyg	Medelvärde	Median	Medelfel	Medianfel
Chaplin (1992)	7.3	0.0	0.0	7.3	7.3
Dazed and... (1993)	7.6	0.0	0.0	7.57	7.57
Dummy (2002)	6.7	0.0	0.0	6.66	6.67
Forrest Gump (1994)	8.7	0.0	0.0	8.69	8.68
Hobo with a Shot...	6.2	0.2	0.2	6.0	5.96
In Time (2011)	6.5	0.0	0.0	6.5	6.5
Inception (2010)	8.8	0.0	0.0	8.8	8.8
The Fifth Element	7.5	0.0	0.0	7.5	7.5
The Shawshank...	9.2	0.0	0.0	9.2	9.2
Zoolander (2001)	6.4	0.0	0.1	6.36	6.33
Totalt:				7.5	7.5

Tabell 3: Bedömning av Twitter-data med Voted Perceptron

Titel	IMDb-betyg	Medelvärde	Median	Medelfel	Medianfel
Chaplin (1992)	7.3	2.5	2.1	4.8	5.22
Dazed and...	7.6	2.1	1.9	5.46	5.68
Dummy (2002)	6.7	2.5	2.5	4.21	4.22
Forrest Gump (1994)	8.7	3.2	3.3	5.52	5.36
Hobo with a Shot...	6.2	3.0	2.4	3.17	3.85
In Time (2011)	6.5	6.2	7.5	0.34	1.0
Inception (2010)	8.8	3.9	3.1	4.91	5.72
The Fifth Element	7.5	1.8	1.7	5.71	5.83
The Shawshank...	9.2	2.2	1.6	6.97	7.64
Zoolander (2001)	6.4	2.5	2.1	3.9	4.28
Totalt:				4.5	4.9

7.2 IMDb-data

7.2.1 Precision och Recall

Resultaten i tabellerna 4, 5 och 6 avspeglar tester gjorda med 1000 på förhand klassade recensioner, med värden på precision och recall för de två möjliga klasserna positiv och negativ. Medelvärdet av mätvärdena för de två klasserna avspeglar den totala prestandan för varje algoritm. Enligt dessa mätvärden presterade Naive Bayes bäst, tätt följd av Support Vector Machines, och sämst prestanda gavs av Voted Perceptron.

Tabell 4: Precision och recall för Naive Bayes

Klassificering	Precision	Recall
Negativ	0.8	0.82
Positiv	0.81	0.79
Viktat medelvärde:	0.8	0.8

Tabell 5: Precision och recall för Support Vector Machines

Klassificering	Precision	Recall
Negativ	0.78	0.78
Positiv	0.78	0.78
Viktat medelvärde:	0.78	0.78

Tabell 6: Precision och recall för Voted Perceptron

Klassificering	Precision	Recall
Negativ	0.75	0.78
Positiv	0.77	0.73
Viktat medelvärde:	0.76	0.76

7.2.2 IMDb - jämförelse

Resultaten från testerna utförda med samtliga IMDb-recensioner på ett slumpmässigt urval av filmer presenteras i tabell 7, 8 och 9 i Appendix 2. Tabellerna visar filmens IMDb-betyg, systemets median- och medelvärdesbedömning, och det motsvarande felet (skillnaden mellan bedömningen och IMDb-betyget) för varje bedömning. I detta test hade Voted Perceptron det minsta felet, vilket har ett medelvärde på ca. 1.10. Naive Bayes presterar något bättre med ett medelfel på 1.75, och Support Vector Machines har relativt stora fel på 3.1.

8 Analys

8.1 Tolkning av resultat

Resultaten från testerna av Twitter-datan var oanvändbara, och visar att datamängden är opålitlig. Detta grundar sig i att Twitter används främst för kommunikation, och många av de tweets som samlats in inte ger en åsikt om filmerna de talar om, utan framför allt är meddelanden om ATT man ser filmen i fråga, vilket ger slumpmässiga resultat vid bedömningen. Resultaten från de två andra mätningarna visar tvetydiga resultat. I testerna av precision och recall hamnar Naive Bayes klart först, men när det kommer till IMDb-jämförelsen halkar den efter Voted Perceptron, som visar slående bra resultat i dessa tester. Skillnaden kan bero på förekomsten av recensioner i

IMDb-datan som är neutrala eller har en varierande ståndpunkt och som rent slumpmässigt bedöms olika av algoritmerna, men Voted Perceptron har ett mycket konsekvent mindre fel. Värt att nämna är även att de två filmer där Voted Perceptron har störst fel, The Boondock Saints och Bridesmaids(vilka även de andra två algoritmerna ger ett markant lägre betyg än IMDb gör), har ett stort antal recensioner med frågeställningar om betyget på filmen. Detta upptäcktes efter en slumpmässigt vald läsning av recensionsdatan från dessa filmer. De frågar saker som: "I am shocked how this movie is rated so well", "This is basically a overrated comedy" och "I had high hopes. I heard from many people what a good movie this was. Boy, were they wrong". Den stora skillnaden mellan IMDb-betyget(som är framtaget av många fler röstningar än det fanns recensioner) och recensionsbetyget kan avspegla att dessa är filmer som filmfantaster, av vilka det antas finns en större andel av bland recensenterna, inte uppskattar, men allmänheten uppskattar. Att många av de med stort fel är komedier som av filmälskare kan anses vara klyschiga verkar stödja denna tes. Just denna egenskap hos IMDb-datan, att de användare som sätter betyget inte nödvändigtvis är samma som de som skriver recensioner, gör att felet relativt IMDb-betygen inte är ett absolut mått på hur bra klassificeraren gjorde sitt jobb. Detta verkar speciellt gälla komedier och filmer som inte framställs som seriösa eller välproducerade (i denna kategori faller även parodier och filmer som medvetet är klyschiga eller icke-originella).

8.2 Slutsatser

De sammanlagda resultaten givna av systemet när Voted Perceptron används visar att det går att skapa ett system som det beskrivet i problemformuleringen. Systemet ger resonabla bedömningar av filmer, och bedömningar av fler än 2000 recensioner (vilket får anses vara en stor korpus) tar under tio sekunder att köra vilket får anses en resonabel körtid. Pålitligheten är den faktor som uppvisar störst brister, då betygets precision kan bero på filmens typ och genre. Betygen blir dock tillräckligt bra för att författarna ska anse systemet som lyckat. Systemet skulle även med få modifikationer kunna anpassas till andra problemdomäner, tack vare dess öppna natur.

8.2.1 Möjliga förbättringar

Implementationen visade sig vara lyckad men ett antal förbättringsmöjligheter visade sig senare. De algoritmerna som användes kunde vi ha modifierat eller skrivit själva. Då hade vi kunnat optimera dessa för den indata vi hade. Ett annat alternativ kan vara att modifiera indatan att passa algoritmen. Vissa av de recensioner författarna använde sig av visade sig vara korta, och innehöll till exempel information om att recensionen innehöll avslöjande om handlingen i filmen. Detta är onödigt och kan ha påverkat resultatet i en-

staka fall. Flera källor skulle också kunna förbättra resultatets träffsäkerhet. Problemen uppstår i hur man får tag på dessa. Att hitta inlägg i bloggar som behandlar en specifik film är svårt att filtrera ut men resultatet kan bli mer exakt av ett större urval av recensioner och kommentarer. För att göra detta kan det dock behövas väldigt mycket manuellt arbete i att formatera och filtrera ut data som anses vara relevant. De algoritmer författarna har använt sig av är alla binära i den meningen att de betygsätter kommentarer som "positiva" eller "negativa". Detta är inte ultimat då det resulterande betyget skall placeras på en 10-gradig skala. Att då betygsätta varje inlägg på en mer finkornig skala än "bra" och "dålig" kan ge mer precisa resultat. Programmet kan utvecklas att klara flera problemområden som att exempelvis generera betyg på bilar, öl eller serveringsställen. Implementationen klarar i nuläget av detta men ny träningsdata kan komma behövas som är specifik för uppgiften. Positiva och negativa termer för recensioner om filmer och åsikter/recensioner om bilar kan skilja sig märkbart och träningsdatan för algoritmen måste då anpassas.

8.2.2 Möjliga praktiska användningsområden

De praktiska användningsområdena för en applikation som denna är mycket stora. Allt som folk har åsikter om går att bedöma, och då även om saker det inte nödvändigtvis finns en sammanställning på nätet motsvarande IMDb. Allt i från att betygsätta politiker inför ett val, bilar att köpa etc. Som tidigare nämnt så behövs det då anpassad träningsdata för området som behandlas. Ytterligare ett problem som har diskuterats tidigare är problemet att hitta relevant data att klassificera. IMDbs recensionsdel är välanpassad då de grupperas per film men för många av de möjliga praktiska användningsområdena så saknas denna möjlighet. För exempelvis politiker behövs många olika källor och tolkning av dessa för att få allmänhetens bild av en politiker eller ett parti. Metoden innebär en spännande utvecklingsmöjlighet för en ny typ av snabba maskinsamplingar huvudsakligen från nätet, som kan ha många användningsområden i politisk analys, marknadsundersökningar för produkter eller tjänster med mera. Det kan potentiellt innebära en utökad möjlighet för snabbt analysera trender och liknande ur den enorma mängd data som genereras varje timme på nätet. Detta är inte huvudämnet för vårt arbete, och måste värderas och utvecklas vidare av samhällsvetare och marknadsanalytiker, men fördelarna med att ha ett system som bedömer politiker direkt efter ett uttalande är att det kan ge en betydligt snabbare form av opinionsmätning än den traditionella.

9 Referenser

Referenser

[1] Weka java api. http://weka.sourceforge.net/doc.stable/. Besökt 2011-04-11.

[2] Tristan Fletcher. Svm explained. http://www.tristanfletcher.co.uk/SVMExplained.pdf, 2012. Besökt 2012-04-04.

[3] Yoav Freund and Robert E. Shapire. Voted perceptron. http://cseweb.ucsd.edu/~yfreund/papers/LargeMarginsUsingPerceptron.pdf. 1999.

[4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update.

[5] IMDb. Up. http://www.imdb.com/title/tt1049413/. Besökt 2012-04-01.

[6] IMDb. War of the worlds. http://www.imdb.com/find?s=all;q=war+of+the+worlds. Besökt 2012-04-01.

[7] IMD.com. http://www.imdb.com. Besökt 2012-04-11.

[8] George H John and Pat Langley. Estimating continuous distributions in bayesian classifiers. http://anthonypenniston.com/grammarchecker/export/121/articles/bayes-continuous.pdf, 1995. Besökt 2012-04-02.

[9] Twitter: The Dark Lord. https://twitter.com/#!/Lord_Voldemort7/status/188978288922853377.

[10] John Nunemaker, Wynn Netherland, and Erik Michaels-Ober nd Steve Richert. Twitter gem. https://github.com/jnunemaker/twitter.

[11] Bo Pang and Lillian Lee. Polarity dataset 2.0. http://www.cs.cornell.edu/people/pabo/movie-review-data/, 2004.

[12] Twitter. www.twitter.com. Besökt 2012-04-11.

[13] Wikipedia. Precision and recall. http://en.wikipedia.org/wiki/Precision_and_recall. Besökt 2012-04-01.

[14] Wikipedia. Supervised learning. http://en.wikipedia.org/wiki/Supervised_learning. Besökt 2012-04-02.

[15] Wikipedia. Support vector machine. `http://en.wikipedia.org/wiki/Support_vector_machine`. Besökt 2012-04-04.

[16] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques 2nd ed.* Morgan Kaufmann, second edition, 2005.

A Appendix 1 - Ordlista

Beskrivning	
Ruby	Ett modernt skriptspråk, `http://www.ruby-lang.org/en`
Hashtagg	Ett nummertecken (#) följt av ett nyckelord, används av Twitter för att gruppera tweets.
Twitter	Social tjänst online för att sprida sina åsikter.
Tweet	Ett inlägg på Twitter, även känt som socialt utrop
IMDb	Katalogtjänst på internet där mycket information om filmer samlas
RottenTomatoes	Filmrecensionstjänst på Internet, `rottentomatoes.com`
IMDb-Data	Recensioner hämtade från IMDb.
Gem	En tilläggsmodul till Ruby.
HTTP GET Request	Anrop för att hämta hem en sida från Internet.
API	Application Programming Interface, Ett gränssnitt som låter andra program utnyttja ett programs funktioner
Java	Generellt programspråk

B Appendix 2 - Tabeller

Tabell 7: Bedömning av IMDb-data, Naive Bayes

Titel	IMDb-betyg	Medelvärde	Median	Medelvärdesfel	Medianfel
Alien (1979)	8.5	7.2	7.3	1.27	1.24
American Pie (1999)	6.9	3.6	3.5	3.27	3.4
Back to the Future	8.5	7.0	8.1	1.55	0.39
Batoru rowaiaru	7.8	6.3	6.3	1.46	1.55
Bridesmaids (2011)	6.9	3.3	3.1	3.59	3.78
Chaplin (1992)	7.3	8.3	8.3	1.01	1.03
Crazy, Stupid, Love	7.5	5.7	6.3	1.75	1.18
Dazed and Conf...	7.6	4.8	4.4	2.85	3.21
Dr. Jekyll and Mr...	7.7	7.7	7.1	0.01	0.58
Dummy (2002)	6.7	7.6	7.5	0.92	0.8
Eternal Sunshine...	8.4	7.4	7.7	1.02	0.73
Fight Club (1999)	8.8	7.0	7.1	1.85	1.73
Forrest Gump (1994)	8.7	7.5	7.6	1.22	1.12
Goodfellas (1990)	8.8	8.0	7.9	0.77	0.88
Green Lantern	6.0	5.6	4.9	0.43	1.11
He Got Game	6.8	7.5	8.1	0.7	1.28
Hobo with a Shot...	6.2	3.9	3.8	2.27	2.39
Hot Shots! (1991)	6.5	1.8	1.5	4.66	5.0
In Time (2011)	6.5	6.5	6.3	0.01	0.23
Inception (2010)	8.8	6.8	6.6	2.03	2.2
Joan of Arc	6.3	7.2	6.9	0.89	0.59
John Carter (2012)	7.0	6.8	7.1	0.17	0.08
Kick-Ass (2010)	7.9	4.7	4.8	3.19	3.1
Melancholia (2011)	7.3	6.5	6.6	0.83	0.68
Men in Black II	5.7	2.0	2.1	3.71	3.61
Midnight in Paris	7.8	7.7	7.8	0.12	0.0
Mirror Mirror (2012)	5.6	7.1	7.6	1.5	2.05
Orgazmo (1997)	6.0	2.5	2.1	3.49	3.9
Resident Evil (2002)	6.5	3.2	2.7	3.34	3.81
Schindler's List (1993)	8.9	8.5	8.6	0.38	0.26
Serbuan maut	8.4	5.4	5.0	2.98	3.4
The Artist (2011)	8.3	8.3	8.2	0.0	0.13
The Big Year (2011)	6.0	6.0	7.2	0.0	1.17
The Boondock...	7.8	2.8	2.8	4.97	4.95
The Claim (2000)	6.5	7.6	8.3	1.12	1.75
The Fifth Element	7.5	5.9	6.5	1.56	1.03
The Girl with the...	8.0	7.1	6.8	0.94	1.17
The Hangover (2009)	7.8	4.4	4.9	3.4	2.87
The Matrix (1999)	8.7	5.3	5.3	3.42	3.36
The Million Dollar...	5.6	4.9	5.3	0.67	0.26
The Muppets (2011)	7.6	6.4	6.4	1.16	1.17
The Shawshank...	9.2	9.0	9.2	0.17	0.03
The Wire (2002)	9.6	7.1	7.1	2.54	2.46
We Bought a Zoo	7.2	8.2	7.5	0.99	0.26
Wrath of the Titans	6.3	4.4	3.0	1.93	3.3
Zoolander (2001)	6.4	3.6	3.3	2.8	3.07
Totalt:				1.7	1.8

Tabell 8: Bedömning av IMDb-data, Support Vector Machines

Titel	IMDb-betyg	Medelvärde	Median	Medelvärdesfel	Medianfel
Alien (1979)	8.5	4.8	4.8	3.66	3.72
American Pie (1999)	6.9	4.4	4.0	2.53	2.92
Back to the Future	8.5	7.2	8.6	1.26	0.15
Batoru rowaiaru	7.8	4.2	4.0	3.6	3.78
Bridesmaids (2011)	6.9	1.5	1.7	5.45	5.2
Chaplin (1992)	7.3	6.7	7.0	0.6	0.3
Crazy, Stupid, Love	7.5	4.7	4.9	2.84	2.63
Dazed and Conf...	7.6	4.7	4.5	2.86	3.09
Dr. Jekyll and Mr...	7.7	4.8	4.1	2.89	3.63
Dummy (2002)	6.7	5.9	4.3	0.82	2.38
Eternal Sunshine...	8.4	5.4	5.6	2.98	2.81
Fight Club (1999)	8.8	5.1	5.3	3.7	3.49
Forrest Gump (1994)	8.7	6.1	6.1	2.59	2.63
Goodfellas (1990)	8.8	5.6	6.0	3.21	2.84
Green Lantern	6.0	3.1	3.1	2.89	2.87
He Got Game	6.8	5.5	4.5	1.3	2.28
Hobo with a Shot...	6.2	4.2	4.0	1.96	2.22
Hot Shots! (1991)	6.5	4.4	4.2	2.08	2.33
In Time (2011)	6.5	2.8	3.1	3.7	3.43
Inception (2010)	8.8	4.2	4.1	4.65	4.71
Joan of Arc	6.3	3.9	4.0	2.39	2.26
John Carter (2012)	7.0	4.2	3.4	2.75	3.57
Kick-Ass (2010)	7.9	2.2	2.0	5.68	5.9
Melancholia (2011)	7.3	2.8	3.1	4.45	4.23
Men in Black II	5.7	2.5	2.8	3.2	2.87
Midnight in Paris	7.8	5.5	5.6	2.35	2.16
Mirror Mirror (2012)	5.6	3.6	2.6	2.02	2.95
Orgazmo (1997)	6.0	4.0	4.4	1.99	1.65
Resident Evil (2002)	6.5	2.6	2.5	3.89	3.99
Schindler's List (1993)	8.9	4.6	5.0	4.31	3.9
Serbuan maut	8.4	5.4	6.4	3.01	2.04
The Artist (2011)	8.3	5.3	5.6	3.03	2.67
The Big Year (2011)	6.0	3.6	3.5	2.38	2.52
The Boondock...	7.8	2.0	2.1	5.83	5.7
The Claim (2000)	6.5	3.6	2.9	2.92	3.64
The Fifth Element	7.5	4.4	4.7	3.09	2.82
The Girl with the...	8.0	4.3	4.3	3.69	3.69
The Hangover (2009)	7.8	2.4	2.9	5.37	4.87
The Matrix (1999)	8.7	4.2	4.2	4.46	4.46
The Million Dollar...	5.6	3.8	5.0	1.82	0.64
The Muppets (2011)	7.6	4.0	4.0	3.56	3.61
The Shawshank...	9.2	3.8	4.4	5.4	4.82
The Wire (2002)	9.6	4.4	5.3	5.24	4.34
We Bought a Zoo	7.2	5.6	6.6	1.57	0.58
Wrath of the Titans	6.3	3.4	3.1	2.89	3.16
Zoolander (2001)	6.4	4.4	4.2	2.01	2.23
Totalt:				3.1	3.1

Tabell 9: Bedömning av IMDb-data, Voted Perceptron

Titel	IMDb-betyg	Medelvärde	Median	Medelvärdesfel	Medianfel
Alien (1979)	8.5	7.6	7.8	0.88	0.75
American Pie (1999)	6.9	4.8	4.5	2.07	2.44
Back to the Future	8.5	9.3	9.2	0.84	0.69
Batoru rowaiaru	7.8	7.0	6.2	0.82	1.64
Bridesmaids (2011)	6.9	3.9	4.1	3.03	2.79
Chaplin (1992)	7.3	7.2	7.8	0.14	0.53
Crazy, Stupid, Love	7.5	6.5	6.3	1.0	1.25
Dazed and Conf...	7.6	6.1	5.9	1.52	1.66
Dr. Jekyll and Mr...	7.7	6.4	5.1	1.28	2.62
Dummy (2002)	6.7	6.6	6.8	0.08	0.12
Eternal Sunshine...	8.4	7.0	7.3	1.39	1.1
Fight Club (1999)	8.8	7.6	7.6	1.17	1.18
Forrest Gump (1994)	8.7	7.6	7.5	1.13	1.16
Goodfellas (1990)	8.8	8.4	8.5	0.41	0.32
Green Lantern	6.0	6.4	6.4	0.45	0.37
He Got Game	6.8	6.7	6.4	0.1	0.36
Hobo with a Shot...	6.2	4.9	4.3	1.3	1.88
Hot Shots! (1991)	6.5	6.1	5.7	0.42	0.83
In Time (2011)	6.5	6.8	6.9	0.25	0.43
Inception (2010)	8.8	7.5	7.6	1.31	1.2
Joan of Arc	6.3	6.5	7.3	0.22	0.96
John Carter (2012)	7.0	6.8	6.9	0.2	0.1
Kick-Ass (2010)	7.9	5.9	5.7	2.03	2.17
Melancholia (2011)	7.3	6.6	6.5	0.74	0.81
Men in Black II	5.7	4.5	4.9	1.25	0.83
Midnight in Paris	7.8	7.3	6.9	0.54	0.91
Mirror Mirror (2012)	5.6	7.8	8.5	2.18	2.93
Orgazmo (1997)	6.0	5.3	4.7	0.74	1.32
Resident Evil (2002)	6.5	5.1	5.1	1.38	1.44
Schindler's List (1993)	8.9	8.3	8.4	0.57	0.54
Serbuan maut	8.4	7.2	6.8	1.21	1.58
The Artist (2011)	8.3	7.8	7.5	0.55	0.77
The Big Year (2011)	6.0	6.4	7.4	0.42	1.39
The Boondock...	7.8	4.6	4.4	3.2	3.39
The Claim (2000)	6.5	6.4	6.5	0.09	0.01
The Fifth Element	7.5	6.4	6.7	1.09	0.75
The Girl with the...	8.0	5.7	5.7	2.31	2.27
The Hangover (2009)	7.8	5.8	5.5	2.04	2.33
The Matrix (1999)	8.7	7.1	7.1	1.57	1.63
The Million Dollar...	5.6	5.4	5.6	0.24	0.04
The Muppets (2011)	7.6	7.3	7.5	0.34	0.12
The Shawshank...	9.2	9.0	9.0	0.23	0.24
The Wire (2002)	9.6	7.2	6.9	2.36	2.68
We Bought a Zoo	7.2	8.3	8.3	1.05	1.11
Wrath of the Titans	6.3	5.8	6.0	0.48	0.3
Zoolander (2001)	6.4	5.5	6.2	0.94	0.21
Totalt:				1.0	1.2