

Replikidentifiering av rollfigurer från tv-serien The Simpsons med N-Grams

Anton Kindestam antonki@kth.se

Rasmus Ansin ransin@kth.se

17 december 2015

Innehåll

1	Bakgrund	1
2	Utgångspunkt	2
3	Metod	2
3.1	Inläsning av data	2
3.2	Generering av N-Gram	2
3.3	Stopplistor	3
3.4	Precision, Återkallning, F_1 -mått	3
3.5	Korsvalidering	4
3.6	Poängfunktioner	4
4	Resultat	4
4.1	Vanliga n-gram	4
4.2	Resultat av korsvalidering	5
4.2.1	Baseline	5
4.2.2	$n = 2$	5
4.2.3	$n = 3$	7
4.2.4	Slumpmässig gissning	7
5	Utvärdering	8

1 Bakgrund

Vi har valt att analysera repliker av The Simpsons-avsnitt för att skapa en klassificerare som kan klassificera vilken yttring som tillhör vilken Simpsons-rollfigur. Vi gör detta genom att skapa en n-gram-databas över yttringar kopplade till rollfigurer. För en given mening generar vi dess n-gram, och sen väljer vi den rollfigur för vilken denna n-gram-sekvens är sannolikast.

2 Utgångspunkt

Vårt korpus består av en samling med manuskript från den amerikanska tv-serien The Simpsons vilka vi fann på webbsidan <http://www.simpsoncrazy.com/scripts>. Vår plan är att använda detta korpus för att generera en n-gram-databas per rollfigur, med vilka vi kan förutspå sannolikheterna för att en viss yttring skulle tillhöra just en specifik rollfigur, genom summering av förekomst-frekvenserna av test-meningens n-gram för respektive rollfigurs n-gram-databas.

3 Metod

3.1 Inläsning av data

Manuskripten från `simpsonscrazy.com` har följande form:

ACT ONE

"The Simpsons Christmas Special" appears on screen. The episode begins with Homer, Marge and Maggie arriving at Springfield Elementary School. They are late for the schools' Christmas show.

MARGE

(Angry) Oh, careful, Homer!

HOMER

There's no time to be careful, we're late.

They enter the hall. A class is singing "Oh, Little Town of Bethlehem".

MARGE

Sorry, excuse me, pardon me, sorry.

Vi sparar dessa i ett råtextformat och gör en första filtrering av dessa rader genom att ta bort ACT-taggar och radera ord och meningar som skrivs inom parenteser.

Den något modifierade filen läses sedan in rad för rad till ett Pythonscript. Vi plockar ut de rader som följs av en rad bestående av enbart versaler (ett namn).

Varje rad vi läser in filtreras ytterligare genom att alla tecken som inte är av typen `[a-zA-Z']` tas bort. Resultatet är nu av det här slaget:

```
$ python read_lines_from_file.py 01_01_raw.txt
repliker = {
    "MARGE": ["oh careful homer"], ["sorry excuse me pardon me sorry"],
    "HOMER": ["there's no time to be careful we're late"]
}
```

3.2 Generering av N-Gram

Från replikerna bildar vi en uppsättning n-gram, för olika n-värden, för varje rollfigur. I rapporten presenterar vi främst resultat för $n = 2$ men även för $n = 3$.

```

grams = {
  2:{
    "MARGE": ["$ oh", "oh careful", "careful homer", "homer $", "$
              sorry", "sorry excuse", "excuse me", "me pardon",
              "pardon me", "me sorry", "sorry $"],
    "HOMER": ["$ there's", "there's no", "no time", "time to", "to
              be", "be careful", "careful we're", "we're late",
              "late $"]
  }
  3:{
    "MARGE": ["$ oh careful", "oh careful homer", "careful homer $",
              "$ sorry excuse", "sorry excuse me", "excuse me pardon",
              "me pardon me", "pardon me sorry", "me sorry $"],
    "HOMER": ["$ there's no", "there's no time", "no time to", "time
              to be", "to be careful", "be careful we're",
              "careful we're late", "we're late $"]
  }
}

```

Notera att vi använder tecknet \$ i n-gram för att signalera start och stopp av replik.

3.3 Stopplistor

Vi är intresserade av n-gram som är särskiljande. För att vanligt förekommande n-gram inte ska räknas plockar vi bort dem med hjälp av en stopplista. Vi skapade en första stopplista genom att lista de vanligast förekommande n-grammen sett över alla rollfigurers n-gram. Till denna stopplista lades sedan de 100 vanligaste engelska n-grammen till, från Corpus of Contemporary American English (COCA)¹. Genom att exkludera dess n-gram ur n-gram-databasen och ur meningar innan klassificering hoppas vi förbättra precisionen hos klassificeraren.

3.4 Precision, Återkallning, F_1 -mått

För att kunna utvärdera hur väl vår klassificerare fungerar så använder vi måtten precision, återkallning och beräknar ett F_1 -mått.

Precision är den andel av de fall som klassificerades positivt som var rätt. Återkallning är den andel av de positiva exemplen i våra data som fångades av vår klassificerare. F_1 -mått är det harmoniska medelvärde av precision och återkallning, och används eftersom det är svårt att jämföra två olika utvärderingsmått.

$$\begin{aligned}
 \text{precision} &= \frac{\text{sanna positiva}}{\text{sanna positiva} + \text{falska positiva}} \\
 \text{återkallning} &= \frac{\text{sanna positiva}}{\text{sanna positiva} + \text{falska positiva}} \\
 F_1 &= 2 \cdot \frac{\text{precision} \cdot \text{återkallning}}{\text{precision} + \text{återkallning}}
 \end{aligned}$$

¹http://www.ngrams.info/download_coca.asp

3.5 Korsvalidering

Korsvalidering går ut på att man delar upp sitt dataset i ett testset och ett träningsset. Man brukar dela upp det så att träningssetet innehåller all data förutom den som tillhör testsetet. Man roterar sedan det som ingår i testsetet och träningssetet så att all data i det samlade datasetet har klassificerats av klassificeraren, utan att det dataset man klassificerar ingår i träningsdatan när man klassificerar.

I vårt projekt har vi gjort korsvalidering på avsnittsnivå. Vi tar bort ett avsnitt från alla avsnitt vi har och tränar klassificeraren på återstående data. Sen klassificerar vi det borttagna avsnittet och mäter hur väl klassificeraren kan hitta vem som sa vad i det borttagna avsnittet, jämfört med namnen som står i manuset.

3.6 Poängfunktioner

För att koppla en specifik mening till en viss rollfigur så har en funktion som beräknar ett poängvärde för varje rad och rollfigur skapats. Den rollfigur som ger högst poäng för en viss rad kopplas ihop med den raden.

```
function calculate_points_for_ngrams(all_ngrams, ngram, name):  
    return all_ngrams.count(ngram)  
  
function calculate_score_for_line(all_ngrams, line, name):  
    points ← 0  
    for each ngram n in line:  
        points ← points + calculate_points_for_ngram(all_ngrams, n, name)  
    return points
```

Vi använder även en bättre variant, där vi tar hänsyn till hur "unikt" varje n-gram är hos en rollfigur.

Om ett visst n-gram förekommer X gånger hos rollfigur A, Y gånger hos rollfigur B och Z gånger hos rollfigur C får ett n-gram poängen $\frac{X}{(Y+Z)}$ för rollfigur A, $\frac{Y}{(X+Z)}$ för rollfigur B och $\frac{Z}{(X+Y)}$ för rollfigur C.

```
function calculate_score_for_line_improved(all_ngrams, line, name):  
    ngrams ← all_ngrams without those associated with character name  
    points ←  $\frac{\text{calculate\_score\_for\_line}(\text{all\_ngrams}, \text{line}, \text{name})}{\text{calculate\_points\_for\_ngram}(\text{ngrams}, \text{n}, \text{name})}$   
    return points
```

4 Resultat

4.1 Vanliga n-gram

Homer i don't (33), mr burns (29), in the (24), oh i (23), going to (22), this is (22), i can't (21), out of (20), to the (20), do you (18), if you (17), have to (17), i was (17), do do (16), have a (16)

Bart are you (12), i don't (10), this is (10), to be (9), i know (8), can i (7), i got (7), go to (7), dad i (7), in the (7), going to (7), to the (6), on the (6), quit it (6), the bus (6)

Marge i don't (16), are you (16), in the (12), you can (11), of the (11), going to (11), have to (10), needs braces (9), and i (9), lisa needs (9), to be (9), have a (8), go to (8), this is (8), a little (7)

Lisa i don't (9), i am (9), i think (9), a little (8), in the (7), but i (7), and i (6), dad i (6), be a (6), dad you (6), all right (6), this is (6), mr burns (6)

Burns in the (9), to be (8), going to (8), i want (8), do you (6)

Homer's brain the pudding (9), eat the (9), pudding eat (8)

4.2 Resultat av korsvalidering

4.2.1 Baseline

Om man bara gissar på Homer, som har flest repliker, får vi 42% korrekta gissningar. 42% av alla repliker tillhör alltså Homer.

4.2.2 $n = 2$

without score function

Correct guesses: 896 (33.9%), incorrect guesses: 1748

Rows: Correct name, Columns: Guessed name

Confusion Matrix:

	HOMER	LISA	BURNS	BART	MARGE
HOMER	343	172	129	220	246
LISA	46	100	50	74	102
BURNS	27	23	66	17	54
BART	77	118	42	142	112
MARGE	42	79	60	58	245

	Precision	Recall	F1Score
HOMER:	0.641	0.309	0.417
LISA:	0.203	0.269	0.231
BURNS:	0.190	0.353	0.247
BART:	0.278	0.289	0.283
MARGE:	0.323	0.506	0.394
avg:	0.327	0.345	0.314

Using stoplist with length: 187

Correct guesses: 882 (33.4%), incorrect guesses: 1762

Rows: Correct name, Columns: Guessed name

Confusion Matrix:

	HOMER	LISA	BURNS	BART	MARGE
HOMER	315	193	119	234	249
LISA	53	106	41	75	97
BURNS	21	27	64	19	56
BART	71	129	38	157	96
MARGE	46	88	45	65	240

	Precision	Recall	F1Score
HOMER:	0.623	0.284	0.390
LISA:	0.195	0.285	0.232
BURNS:	0.208	0.342	0.259
BART:	0.285	0.320	0.302
MARGE:	0.325	0.496	0.393
avg:	0.327	0.345	0.315

with score function

Correct guesses: 906 (34.3%), incorrect guesses: 1738

Rows: Correct name, Columns: Guessed name

Confusion Matrix:

	HOMER	LISA	BURNS	BART	MARGE
HOMER	393	105	103	181	328
LISA	76	55	29	76	136
BURNS	36	17	40	18	76
BART	113	53	29	141	155
MARGE	76	45	42	44	277

	Precision	Recall	F1Score
HOMER:	0.566	0.354	0.436
LISA:	0.200	0.148	0.170
BURNS:	0.165	0.214	0.186
BART:	0.307	0.287	0.297
MARGE:	0.285	0.572	0.380
avg:	0.305	0.315	0.294

Using stoplist with length: 187

Correct guesses: 911 (34.5%), incorrect guesses: 1733

Rows: Correct name, Columns: Guessed name

Confusion Matrix:

	HOMER	LISA	BURNS	BART	MARGE
HOMER	398	109	104	189	310
LISA	79	57	30	81	125
BURNS	40	21	42	20	64
BART	120	61	30	149	131
MARGE	83	46	44	46	265

	Precision	Recall	F1Score
HOMER:	0.553	0.359	0.435
LISA:	0.194	0.153	0.171
BURNS:	0.168	0.225	0.192
BART:	0.307	0.303	0.305
MARGE:	0.296	0.548	0.384
avg:	0.304	0.318	0.297

4.2.3 $n = 3$

Eftersom vi inte har någon stopplista för trigram så har vi inga resultat med stopplista att visa för $n = 3$.

without score function!

Correct guesses: 951 (36.0%), incorrect guesses: 1693
Diagonal Sum: 951 (0.359682), Non-Diagonal Sum: 1693 (0.640318)
Rows: Correct name, Columns: Guessed name

	HOMER	LISA	BURNS	BART	MARGE
HOMER	652	105	99	130	124
LISA	189	71	16	49	47
BURNS	93	16	37	15	26
BART	273	64	36	72	46
MARGE	236	45	43	41	119

	Precision	Recall	F1Score
HOMER:	0.452	0.587	0.511
LISA:	0.236	0.191	0.211
BURNS:	0.160	0.198	0.177
BART:	0.235	0.147	0.180
MARGE:	0.329	0.246	0.281
avg:	0.282	0.274	0.272

with score function!

Correct guesses: 983 (37.2%), incorrect guesses: 1661
Diagonal Sum: 983 (0.371785), Non-Diagonal Sum: 1661 (0.628215)
Rows: Correct name, Columns: Guessed name

	HOMER	LISA	BURNS	BART	MARGE
HOMER	690	84	63	132	141
LISA	202	64	17	44	45
BURNS	101	16	28	10	32
BART	298	42	24	70	57
MARGE	248	31	28	46	131

	Precision	Recall	F1Score
HOMER:	0.448	0.622	0.521
LISA:	0.270	0.172	0.210
BURNS:	0.175	0.150	0.161
BART:	0.232	0.143	0.177
MARGE:	0.323	0.271	0.294
avg:	0.290	0.272	0.273

4.2.4 Slumpmässig gissning

Här har vi istället för att använda klassificeraren gissat slumpmässigt på en av de 5 rollfigurerna för varje gång där vi normalt hade anropat klassificeraren. Vi kan se att vår algoritm åtminstone har bättre resultat än slumpen.

Correct guesses: 550 (20.8%), incorrect guesses: 2094

Rows: Correct name, Columns: Guessed name

Confusion Matrix:

	HOMER	LISA	BURNS	BART	MARGE
HOMER	233	216	222	218	221
LISA	72	74	82	81	63
BURNS	40	26	43	44	34
BART	81	95	86	97	132
MARGE	92	94	86	109	103

	Precision	Recall	F1Score
HOMER:	0.450	0.210	0.286
LISA:	0.147	0.199	0.169
BURNS:	0.083	0.230	0.122
BART:	0.177	0.198	0.187
MARGE:	0.186	0.213	0.199
avg:	0.209	0.210	0.193

5 Utvärdering

Vårt korpus är för litet för uppgiften. Det visar sig att det var svårare än vi räknat med att få tag i avskrift av Simpsons-avsnitt på nätet. Det visar sig även att Homer har flest repliker så vårt korpus är snedbalanserat och tenderar att föredra Homer². Antalet korrekt klassificerade repliker är alltså ett dåligt mått, då en gissning på Homer för alla repliker kommer ge över ca 42% korrekta svar. Vi kan dock trösta oss med att vår klassificerare i alla fall ger bättre resultat än att gissa slumpmässigt.

²<http://www.vikparuchuri.com/blog/figuring-out-which-simpsons-character-is-speaking/>