Text Mining South Park for Characteristic Phrases

Kaylin Walker • February 5, 2016

Introduction

Quickly after Matt Stone and Trey Parker launched South Park, an adult animated series, in 1997, it became known for its use of crude language and dark satire of current events. Stone and Parker, the primary writers, producers and voice actors, have given the series a unique voice and personality. Each episode opens with the disclaimer:

> "All characters and events in this show—even those based on real people—are entirely fictional. All celebrity voices are impersonated. . . ..poorly. The following program contains coarse language and due to its content it should not be viewed by anyone."

The series follows four main characters (Stan, Kyle, Cartman and Kenny), though it has an extensive ensemble cast of recurring characters. This analysis serves to analyze their speech to determine which words and phrases are distinct for each character. Since the series uses a lot of running gags, common phrases should be easy to find.

The programming language R and packages XML, RCurl, tm, RWeka and stringr were used to scrape South Park episode transcripts from the internet, attribute them to a certain character, break them into ngrams and identify characteristic ngrams for each character. Full scripts on Github.

Method & Summary Statistics

Transcripts of the first 132 episodes of South Park (season 1 through season 9, episode 7) were scraped from unstructured text on the Internet Movie Script Database using the XML package. The series has run for 19 seasons and 267 episodes, but to my knowledge the remaining transcripts are not available online.

I was able to assign a speaker to each line by splitting the html at `<b>` tags that contained only uppercase text and were shorter than 40 characters. Each line was followed by a blank line, so I used their index to create a starting and stopping point for text to be attributed to a speaker. From there, I used the tm package to pre-process the text (to lowercase, remove punctuation, numbers and white space; remove stop words for unigrams and bigrams, but left them in for tri-, 4- and 5-grams) and form a corpus, which contained more than 18,000 unique words spoken more than 211,000 times. Reducing the sparsity brought that down to about 2,300 words spoken 172,000 times. Processing the text reduced it further to:

ngram.size	total	unique
1	138248	656
2	115310	1620
3	18524	577
4	3482	176
5	1550	119

23 characters with the most words were retained, and the remaining 1781 speakers combined into one "all others" category so as not to lose the text.

Table 2: Number of Words by Character

speaker	words	speaker	words	speaker	words
cartman	22872	jimmy	1673	wendy	842
stan	16420	sharon	1659	ms..choksondik	776
kyle	14212	counselor.mackey	1557	kenny	764
butters	4511	announcer	961	terrance	696
mr..garrison	4440	chris	933	mephesto	614
chef	3873	mayor	907	reporter	551
randy	3551	ms..cartman	875	narrator	502
jimbo	2173	jesus	855	all.others	52031

Log Likelihood

Each corpus was analyzed to determine the most characteristic words for each speaker. Frequent and characteristic words are not the same thing - otherwise words like "I", "school", and "you" would rise to the top instead of unique words and phrases like "professor chaos", "hippies" and "you killed kenny."

Log likelihood was used to measure the unique-ness of the ngrams by character. Log likelihood compares the occurrence of a word in a particular corpus (the body of a character's speech) to its occurrence in another corpus (all of the remaining South Park text) to determine if it shows up more or less likely that expected. The returned value represents the likelihood that the corpora are from the same, larger corpus, like a t-test. The higher the score, the more unlikely.

The **chi-square test** (χ^2), or goodness-of-fit test, can be used to compare the occurrence of a word across corpora.

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where O = observed frequency and E = expected frequency.

However, flaws have been identified: invalidity at low frequencies (Dunning, 1993) and over-emphasis of common words (Kilgariff, 1996). Dunning was able to show that the **log-likelihood statistic** was accurate even at low frequencies:

$$2 \sum O_i * ln(\frac{O_i}{E_i})$$

Which can be computed from the contingency table below as $2 * ((a * log(\frac{a}{E1}) + (b * log(\frac{b}{E2}))$, where E1 = $(a + c) * \frac{(a+b)}{(c+d)}$, and E2 = $(b + d) * \frac{(a+b)}{(c+d)}$.

Table 3: Basic Framework

Group	Corpus.One	Corpus.Two	Total
Word	a	b	a+b
Not Word	c	d	c+d
Total	a+c	b+d	N=a+b+c+d

Table 4: An Example with Log Likelihood 101.7

Group	Cartmans.Text	Remaining.Text	Total
'hippies'	36	5	41
Not 'hippies'	28170	144058	172228
Total	28206	144063	172269

Computed:
$E1 = 28206 * (41/172269) = 6.71$ & $E2 = 144063 * (41/172269) = 34.28$
$LL = 2 * [36 * log(36/6.71) + 5 * log(5/34.28)] = 101.7$

Based on the overall ratio of the word "hippies" in the text, $41/172269 = 0.00023$, we would expect to see hippies in Cartman's speech 6.71 times and in the remaining text 34.28 times. The log likelihood value of 101.7 is significant far beyond even the 0.01% level, so we can reject the null hypothesis that Cartman and the remaining text are one and the same.

Only ngrams that passed a certain threshold were included in the log likelihood test; for unigrams, 50 occurrences, for bigrams, 25, for tri-grams, 15, 4-grams, 10 and 5-grams 5. Each ngram was then compared to all speakers, including those who said it 0 times (using 0.0001 in place of 0 to prevent breaking the log step of the formula). If the number of times the speaker said the word was less than expected, the log likelihood was multiplied by -1 to produce a negative result.

For this analysis, a significance level of 0.001 was chosen to balance the number of significant ngrams with significance. 1.31% of ngrams were found to be significantly characteristic of a given character.

Table 5: Log Likelihood Significance Levels

Level	Critical.Value	p.Value	Percent.Sig
5%	3.84	0.05	4.95
1%	6.63	0.01	2.55
0.1%	10.83	0.001	1.31
0.01%	15.13	0.0001	0.84

The results were then filtered to include each word two times or less: once for the speaker most likely to say it (highest LL) and once for the speaker least likely to say it (lowest LL).

Ranking

Finally, the results were ranked using the formula

$$LL * ngram.length$$

Ranking was used to condense the range of log likelihoods (-700 to 1000+). The ranking formula includes ngram length because longer ngrams appear fewer times in the text, leading to lower log likelihoods, but carry more semantic meaning.

References

Dunning, T. (1993) *Accurate Methods for the Statistics of Surprise and Coincidence.* Computational Linguistics, 19, 1, March 1993, pp. 61-74.
Kilgarriff. A. (1996) *Why chi-square doesn't work, and an improved LOB-Brown comparison.* ALLC-ACH Conference, June 1996, Bergen, Norway.