The Seinfeld Corpus: A Method for Generating A Humor Annotated Corpus and An Analysis of That Corpus

Ran Yad-Shalom, Yoav Goldberg
Computer Science Department
Bar-Ilan University
Israel
{ranyadshalom, yoav.goldberg}@gmail.com

Abstract

These old-fashioned sitcoms, in which the audience's laughter can always be heard after a joke, can be a prolific source for high quality data sets for humor detection research. In this work we present a corpus of 96 humor annotated "Seinfeld" screenplays, and the methods which we used to annotate the audience's laughter in the corpus, along with the timing of the laughter and the timing of the dialog.

While a lot of the research in computational humor deals with well defined problems like one-liners (Mihalcea et al, 2010), puns (Kao et al, 2016), double entendre (Kiddon er al, 2011) or tweets (Raz, 2012), we hope to inspire more works that focus on contextual humor, which is closer to the way humans perceive humor in the real world. An example of such work is Bertero et al, 2016, in which they used an LSTM network to predict the audience's laughter in the "The Big Bang Theory."

In this paper we present a fully automatic method for generating a humor annotated corpus given sitcom's video files, and we explain how we used this method to generate The "Seinfeld" Corpus. We also present a statistical analysis of the corpus in which we try to find patterns and tendencies in the data.

1. Corpus Creation and Annotation

In this section, we explain how we created our corpus of 96 annotated Seinfeld episodes. The corpus, along with the python code that we wrote to generate it, are all available on the project's repository[1].

1.1 Data Collection

The corpus is a hybrid of 3 data sources:

1. The screenplays, which contain the spoken dialog along with the speaking characters' names and scene descriptions. They are available for free at http://www.seinology.com/scripts-english.shtml.
2. The English subtitles, which contain the spoken dialog along with its timing. The Subtitles are available (for free) at https://www.opensubtitles.org.
3. The episodes' audio track from the DVDs/Blurays, which is the only resource that is not available for free. From the audio track we can extract the audience's 'laugh track', which will be used for humor annotation.

1.2 Extracting Laugh Cues from an Episode's Audio Track

This procedure consists of two phases. The first one is the separation of the audience's laughter (a.k.a 'the laugh track') from the rest of the audio, and the second one is extracting the times of laughter by measuring the amplitude of the laugh track. Alas, separation of audio signals that reside in the same recording is not a trivial task. That being said, in some Seinfeld episodes, it can be done very easily.

All of the dialog in Seinfeld is recorded in mono, while the audience is

```
# DONALD
975.851
What are you looking at?
977.179
977.252
Never seen a kid in a bubble before?
979.946
980.400
**LOL**
# GEORGE
983.158
Of course I have. Come on.
985.886
985.961
My cousin's in a bubble.
988.859
988.300
**LOL**
989.664
My friend Jeffrey's sister also,
bubble, you know...
993.256
993.200
**LOL**
993.402
I got a lot of bubble experience.
Come on.
996.062
```

Figure 1: The format of the corpus. This excerpt from The Bubble Boy episode.

[1] https://github.com/ranyadshalom/the_seinfeld_corpus

recorded in stereo[2]. We can take advantage of it, as we have two channels in which the speaking signal is identical, and the audience signal is not. Inverting the polarity of one side and then mixing the two channels in mono (a.k.a the 'Karaoke Effect'), causes a phase cancellation that eliminates the spoken dialog while leaving only the audience's recording. This method is not 100% accurate, since it leaves out the music which is also in stereo, and it performs poorly if the recording is muddled (thus making the peaks harder to measure) or if the episode has mixed Stereo/Mono audience recording (e.g. episode 6 in season 4 has some Mono laughs mixed in with the Stereo ones in the Diner scene at 4:00). However, we view these errors as noise in our dataset, since the majority of laughs extracted are correct.
Measuring the amplitude is done with a simple peak-finding method (a point t(i) is a peak iff t[i-1]<t[i]>t[i+1]). We've found that using the mean of the laugh track's dB levels as a threshold (peaks above this threshold are considered the audience's laughs) yields pretty accurate results.

1.3 Getting the Subtitles and Making Sure They Are In Sync
Opensubtitles offers free Xmlrpc API access to their subtitle database. However, for each Seinfeld episode, there are a variety of English subtitles to choose from, most of which are not in perfect sync with our episodes' video file. In order to automatically determine which of the subtitles are the most synchronized, we've developed a metric to measure the amount of synchronization between a subtitle file and an audio file (i.e. the extracted audio from the video file).
First, we claim that subtitles are in sync if and only if some of the subtitles' start time has a spike in the audio's dB levels and the some of the subtitles' end time has complete silence. The logic behind our intuition here is that in Seinfeld it is not uncommon to have total silence before a character starts speaking: this happens usually (but not exclusively) after a joke, when the actor has to wait for the crowd to stop laughing before she says her next line. Also, it is common to have total silence when a subtitle ends, since the actors tend to pause to take a breath between lines. We've found that as long as the subtitles get a score greater than 0.10 in our metric, they are synchronized enough to do the job.

$$0.6 \cdot \frac{|S'|}{|S|} + 0.4 \cdot \frac{|S''|}{|S|} > 0.10$$

S' is the group of subtitles that start with a volume rise, and S'' is the group of subtitles that end with silence. The python function that measures this metric is called 'is_in_sync' and it can be found in subtitle_getter.py.

1.4 Aligning the Subtitles with the Screenplay
All of the spoken dialog (along with its timing) is contained in the subtitles file. But we also need to know which character speaks - and when. The solution is to combine the subtitles and the screenplay. However, the text in the subtitles and the text in the screenplay do not always match one another, e.g. the line "You met her at the supermarket? How did you do that?" from the screenplay, is shortened in the actual episode to just "You met her at the supermarket? How?." Therefore a 'stupid' algorithm that matches the screenplay to the subtitles line-by-line is impractical.
We start by defining a matching score between two entities using a simple BoW metric as follows:

$$score(dialog\_line, S) = \frac{|DW \cap SW|}{|DW \cup SW|}$$

While S is a set of subtitles, and DW and SW are the sets of words from the dialog line and S respectively.
In order to find the best screenplay/subtitles match, we have to solve an optimization problem. Our optimization problem can be defined recursively as finding the i's that maximize:

$$match(D, S) = argmax_i\left(score\left(d_1, S_{[:i]}\right) + match\left(D_{[1:]}, S_{[i:]}\right)\right)$$

While [:i] means cutting the ordered set S i indices from the end, and [i:] means cutting the ordered set S i indices from the beginning.
After properly formulating the problem, we can solve it using a dynamic programming algorithm:

[2] Please note that not all episodes are in Stereo. The corpus is made out of all episodes from seasons 4,6,7,8 and 9 that were recorded in Stereo and thus are compatible with our method.

for i from 0 to $|D|$

  for j from 0 to $|S|$

    $match[\ D[-i:]\ ][\ S[j:]\ ] = argmax_k(\ score(\ D[0],\ S[:k]) + match(\ D[1:],\ S[k:])$

Please note that although the running time of this algorithm looks promising in theory (n^3), in practicality it fails miserably, e.g. for episode 8 in season 4, the number of repetitions of the innermost loop is |S|*|S|*|D| = 537*537*326 = 94,008,294. Using a pruning window of 15 for the variable k (which signifies the number of subtitles that can fit into one instance of dialog), we have managed to dramatically improve the running time and to render the algorithm practical. In that specific example, the number of repetitions in the innermost loop dropped to 537*15*326 = 2,625,930.

1.4 Bringing it all together

Once we have in-sync subtitles, and we have successfully aligned the subtitles and the screenplay, and have successfully extracted the times in which the audience laughs, now comes the time to combine them all into one coherent data file. For simplification, we assume that a laughter can only occur between 2 subtitles. The only exception to that is when 2 characters are speaking in the same subtitle (These cases are denoted with a dash ("-") in the subtitle), in which we allow a laughter to be annotated in the middle of the exchange.
We assume a 0.5 seconds delay between the joke delivery and the trigger of the audience's laughter (through trial and error, we have found 0.5 seconds to be a pretty good estimate).

1.5 A Fully Automated Procedure

Given a set of Seinfeld episodes in .mkv format, each must have its season and episode numbers in the filename (e.g. S04E14.mkv) we automatically generate corresponding annotated corpus files. We use ffmpeg to extract the audio from the .mkv file, and we use sox's 'oops' option to generate the laugh track from the audio file. The subtitles are downloaded through opensubtitles' Xmlrpc API (please note that you need a working API key for this to work), and the screenplay is fetched using a web crawler from Seinology. The python script that does all that is available in the project's repository[3].

2 Corpus Analysis

2.1 Laughter Distribution

Our corpus contains an average 152.79 laughs per episode, with a standard deviation of approximately 8.94. To people who know the show, it may not come as a surprise that the laughs are almost evenly distributed across episodes. Seinfeld is a comedy and it had to deliver on the laughs every single episode, or nobody would have watched it.

2.2 The Average Episode Laughter Distribution

Another interesting metric to look at is the laughter distribution across an average episode in the corpus:

[3] https://github.com/ranyadshalom/the_seinfeld_corpus/blob/master/corpus_creation/create_corpus.py

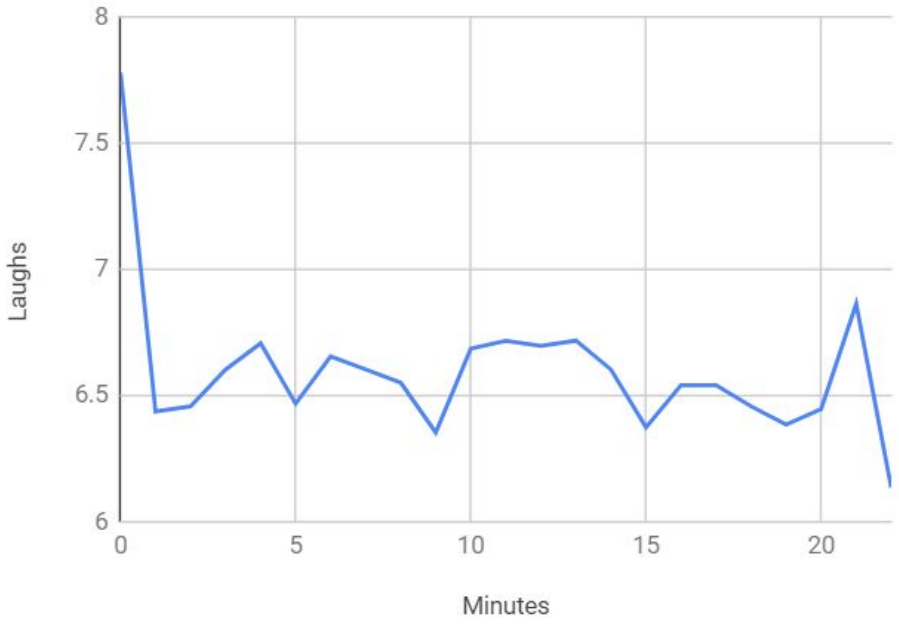

Figure 2: Distribution of laugher across an average episode.

The spike in the beginning could be attributed to the opening monolog. What's interesting to look at is the bump in the middle (minute 10 to 14) where, storytelling-wise, it is common to find the story's "turning point". And since in the comedy genre, the story is all about absurdism, the turning point is usually where the absurd hits a new high. For example, in the episode "The Puffy Shirt," in which Jerry is forced into wearing a ridiculous shirt on national television, the 10th minute is the first time the audience actually sees the Puffy Shirt, along with Jerry's realization that he actually agreed to do it. In "The Soup Nazi," during the 10th minute, Jerry and Sheila break up because the Soup Nazi bans her from the soup shop, to which Jerry responds by renouncing himself from her in order to maintain a good relationship with the Soup Nazi. Elaine sums it up in the 11th minute by telling Jerry that "he chose soup over a woman."
The spike in the end (20th minute) could be explained either as the laughs produced the story's conclusion or by the ending monolog.

2.3 Number of Laughs Across The Show's Seasons

Another interesting metric to look at is the average laughs per episode across seasons. Did the show get funnier in time? Did Larry David's departure in the end of season 7 affect the quality of the show? If the number of laughs is an indicator for the show's quality, it seems that the opposite is true.

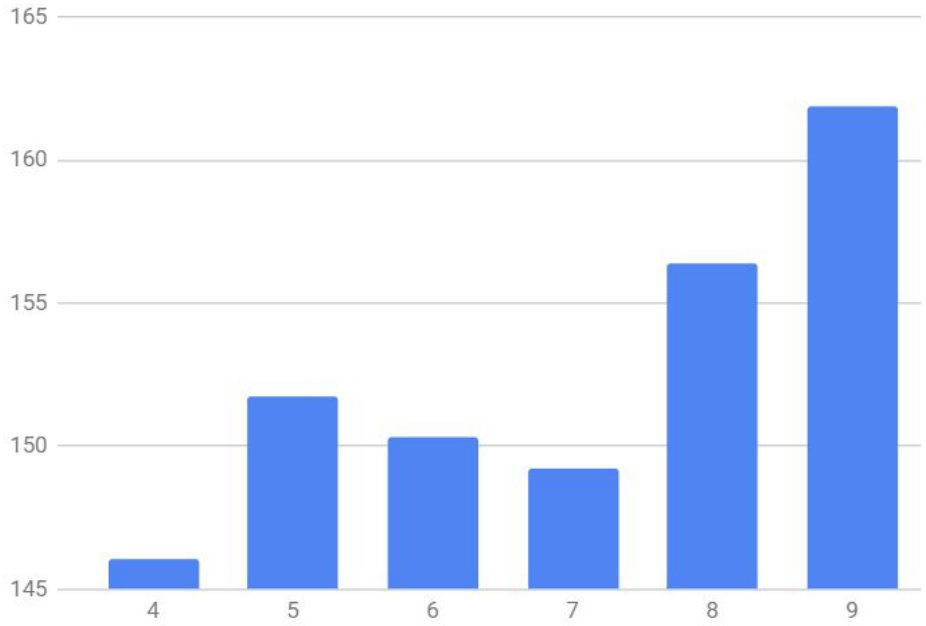

Figure 3: Average laughs per episode across seasons in the corpus.

The show's last 2 seasons were the ones in which the audience laughed more, in average. Which is in line with the claim that on its last 2 seasons, the show became faster paced with a new writing staff, according to the Seinfeld wikipedia page. Also, these seasons did not contain Jerry's monologues, as opposed to seasons 4, 5, 6 and 7.

2.4 Funniest Characters

Out analysis of the corpus can provides us with insights on the amount of funniness of each character. We say that a character generates laughter if the laughter is preceded by the character's speech. While every audience member might have her favourite character, the analysis shows that the character that generates the most laughs (36% of them) is Jerry, with George not far behind (26.6%).

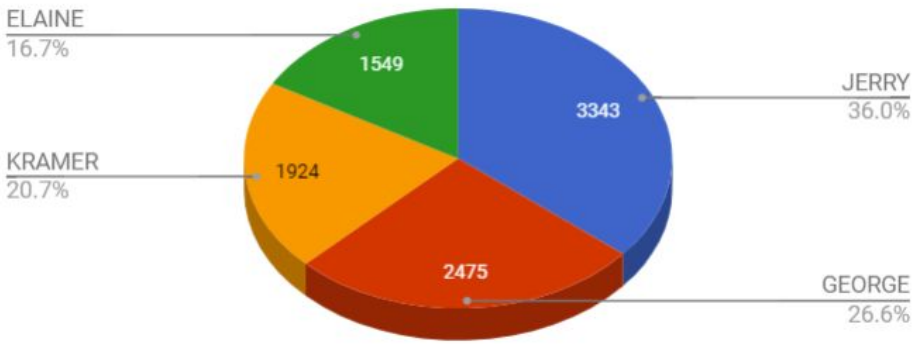

Figure 4: Percentage & number of laughs each character directly triggers across the corpus.

2.5 Indirect Laughter

Sometimes the humor does not come from the character that delivers the punchline. Sometimes the humor comes from the character being spoken to. For example, in the episode "The Marine Biologist," Jerry tells George, "that at this point, she's under the impression that you're a, uh... marine biologist." at which the audience bursts into laughter. But really, they are laughing at George's situation more than at Jerry's. So we came up with a metric to measure that. A laughter is

considered to be generated by a character indirectly if the character spoke before the character that spoke before the laughter. Jerry generates most laughters in this metric as well, but Elaine is getting closer to George at Kramer's expense.

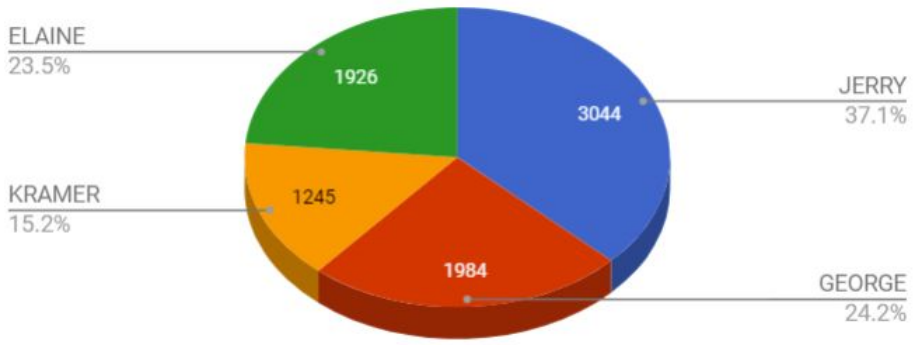

Figure 4: Same as above, but for the character that indirectly triggered the laugh, i.e. the character that spoke before the character that triggered the laughter.

2.4 Funniness Measure

The major problem with the previous metrics is that they are susceptible to the number of lines each character has. Jerry may generate the most laughs because he has the most lines. So to better measure who is the funniest character, we have to divide the number of laughs by the total number of lines a character has. We name this measure "funniness."

$$funniness(\ character) \ = \frac{|\ funny\_lines(character)\ |}{|\ lines(character)\ |}$$

According to this measure, Kramer is the funniest character - he even manages to generate more laughs in total than Elaine, despite the fact that he has less lines than her. Following are George and Jerry, very close to each other in their funniness measure, and last is Elaine.

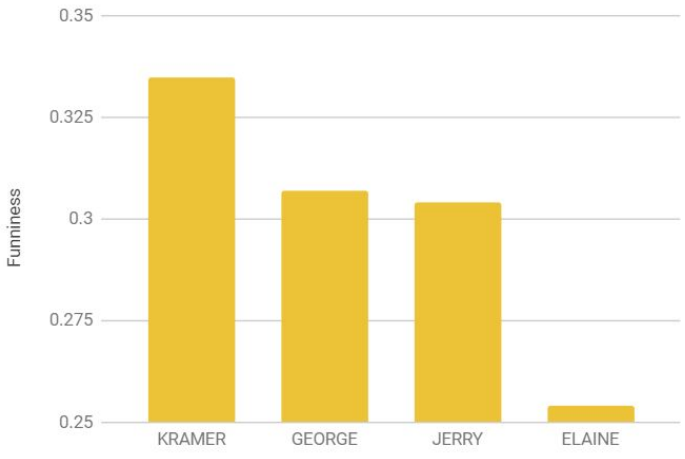

Figure 5: Direct Funniness measure across the corpus.

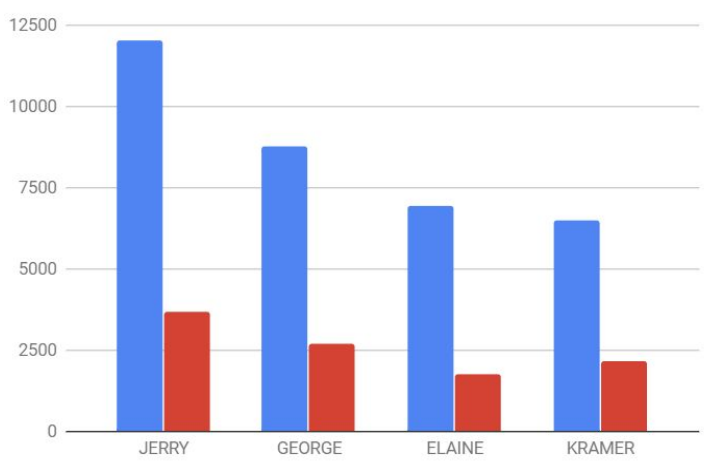

Figure 6: Direct Lines vs. Laughs across the corpus. Blue indicates the total number of lines, while red indicates the number of funny lines.

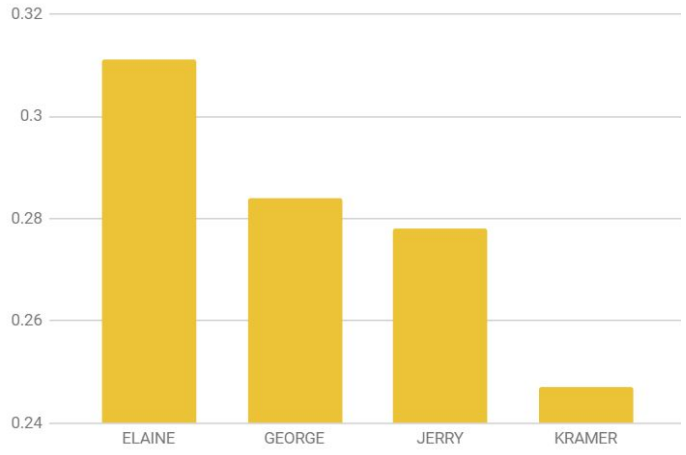

Figure 7: Indirect Funniness measure across the corpus.

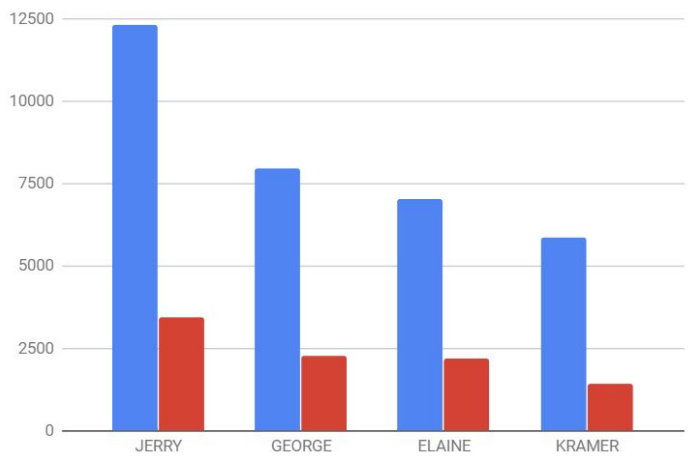

Figure 8: Indirect Lines vs. Laughs across the corpus. Blue indicates the number of indirect lines, while red indicates the number of indirect funny lines.

But what about the indirect case? We compute indirect funniness as follows:

$$indirect\_funniness(character) = \frac{| indirect\_funny\_lines(character) |}{| indirect\_lines(character) |}$$

Indirect funny lines are all the lines where the character spoke before the character that generated the laughter, and indirect lines are all the opportunities that a character has to generate indirect laughter, i.e. all the lines delivered by other characters when she was the last to speak.
Interestingly enough, in the indirect case Elaine and Kramer switch places. It turns out of all the characters, Elaine is the most likely to help make other characters in her scenes generate laughters, while Kramer does not (he generates laughs by himself). Also, the gap between George and Jerry is a little broader in the indirect funniness measure with George being a little more indirectly funny than Jerry.

2.4 Clustering of the Funniness Vectors

For each episode in our corpus, we generated a funniness vector of 4 coordinates that corresponded the four main characters. After normalizing the vectors, we used a K-means clustering algorithm to divide them into 4 clusters:

Cluster 1				
Episode Name	JERRY	GEORGE	ELAINE	KRAMER
S04E07 The Bubble Boy	0.2936	0.2829	0.1442	0.2793
S04E20 The Junior Mint	0.3319	0.2381	0.099	0.331
S05E02 The Puffy Shirt	0.3363	0.2624	0.1953	0.2059
S05E03 The Glasses	0.298	0.2308	0.1886	0.2826
S05E06 The Lip Reader	0.2794	0.2288	0.2248	0.267
S05E09 The Masseuse	0.3136	0.2281	0.1738	0.2845
S05E11 The Conversion	0.2897	0.2862	0.1541	0.2699
S05E12 The Stall	0.3026	0.2356	0.2142	0.2477
S05E13 The Dinner Party	0.3281	0.2366	0.165	0.2702
S05E17 The Wife	0.2991	0.1886	0.2021	0.3102
S05E21 The Hamptons	0.2869	0.2074	0.2073	0.2984
S06E09 The Secretary	0.3303	0.1863	0.173	0.3105
S06E20 The Doodle	0.2771	0.2226	0.2001	0.3002
S07E10 The Gum	0.2858	0.2438	0.2256	0.2448
S07E12 The Caddy	0.2909	0.247	0.218	0.2441
S07E20 The Calzone	0.2612	0.2579	0.1969	0.2839
S08E01 The Foundation	0.3024	0.2624	0.2049	0.2304
S08E03 The Bizarro Jerry	0.3265	0.2852	0.218	0.1703
S08E14 The Van Buren Boys	0.2915	0.2901	0.1819	0.2365
S08E15 The Susie	0.2875	0.2156	0.2357	0.2612
S08E20 The Millennium	0.2725	0.2472	0.1958	0.2845
S08E21 The Muffin Tops	0.3042	0.2399	0.1691	0.2867
S09E03 The Serenity Now	0.3137	0.1996	0.2275	0.2592
S09E12 The Reverse Peepho	0.2619	0.2373	0.219	0.2818
S09E17 The Bookstore	0.2984	0.2882	0.1935	0.22
S09E19 The Maid	0.2842	0.2507	0.2018	0.2633

Cluster 2				
Episode Name	JERRY	GEORGE	ELAINE	KRAMER
S04E11 The Contest	0.2274	0.2049	0.1777	0.39
S04E17 The Outing	0.1717	0.2301	0.2706	0.3276
S04E19 The Implant	0.2017	0.2548	0.2305	0.3131
S04E22 The Handicap Spot	0.2605	0.2271	0.1341	0.3783
S05E05 The Bris	0.2227	0.2359	0.2203	0.3211
S05E07 The Non-Fat Yogurt	0.2013	0.2113	0.2258	0.3616
S06E08 The Mom and Pop Store	0.2572	0.2205	0.1979	0.3244
S06E12 The Label Maker	0.2495	0.239	0.1792	0.3324
S06E19 The Jimmy	0.2475	0.2443	0.1518	0.3564
S06E21 The Fusilli Jerry	0.2124	0.2111	0.2331	0.3435
S07E02 The Postponement	0.212	0.2237	0.1899	0.3744
S07E05 The Hot Tub	0.207	0.2411	0.1849	0.367
S07E08 The Pool Guy	0.2143	0.274	0.1387	0.373
S08E08 The Chicken Roaster	0.261	0.1982	0.2371	0.3037
S09E05 The Junk Mail	0.2562	0.2275	0.2001	0.3162
S09E10 The Strike	0.1929	0.2592	0.2128	0.3351
S09E11 The Dealership	0.1914	0.2705	0.1657	0.3724
S09E13 The Cartoon	0.2293	0.1996	0.1985	0.3726
S09E16 The Burning	0.2702	0.2004	0.1953	0.334

Cluster 3				
Episode Name	JERRY	GEORGE	ELAINE	KRAMER
S04E06 The Watch	0.2171	0.3119	0.182	0.289
S04E08 The Cheever Letters	0.2722	0.2897	0.1559	0.2822
S04E15 The Visa	0.2342	0.2715	0.1749	0.3194
S04E16 The Shoes	0.2457	0.2927	0.1811	0.2806
S05E01 The Mango	0.2677	0.3093	0.1993	0.2237
S05E04 The Sniffing Accountan	0.2601	0.2864	0.1199	0.3337
S05E08 The Barber	0.2174	0.2973	0.2216	0.2638
S05E10 The Cigar Store Indian	0.2335	0.2722	0.188	0.3063
S05E14 The Marine Biologist	0.2665	0.3042	0.113	0.3162
S06E18 The Doorman	0.2461	0.3187	0.1701	0.2652
S06E23 The Face Painter	0.2428	0.3191	0.1661	0.272
S07E01 The Engagement	0.2081	0.2641	0.2352	0.2926
S07E06 The Soup Nazi	0.2421	0.2682	0.2122	0.2775
S07E09 The Sponge	0.2557	0.2701	0.1832	0.291
S07E16 The Shower Head	0.2561	0.2602	0.2018	0.2819
S08E02 The Soul Mate	0.2444	0.3258	0.1892	0.2406
S08E04 The Little Kicks	0.2405	0.3257	0.1648	0.2689
S08E11 The Little Jerry	0.2021	0.3222	0.2072	0.2685
S08E13 The Comeback	0.2246	0.3378	0.1728	0.2649
S08E18 The Nap	0.1898	0.2876	0.2183	0.3043
S09E02 The Voice	0.2037	0.3486	0.155	0.2926
S09E04 The Blood	0.1777	0.3488	0.1788	0.2946
S09E06 The Merv Griffin Show	0.2428	0.264	0.2049	0.2883
S09E14 The Strongbox	0.2027	0.2718	0.2186	0.3069
S09E15 The Wizard	0.2139	0.2698	0.221	0.2953

Cluster 4				
Episode Name	JERRY	GEORGE	ELAINE	KRAMER
S04E09 The Opera	0.275	0.212	0.2811	0.2318
S04E10 The Virgin	0.2359	0.2221	0.2694	0.2726
S04E12 The Airport	0.2881	0.2536	0.2304	0.2279
S04E13 The Pick	0.2587	0.3161	0.2203	0.2049
S04E18 The Old Man	0.1562	0.3453	0.2986	0.1999
S04E21 The Smelly Car	0.2839	0.2574	0.2611	0.1975
S05E15 The Pie	0.2631	0.2759	0.2385	0.2224
S05E16 The Stand-In	0.2321	0.2628	0.3066	0.1986
S05E20 The Fire	0.2252	0.2247	0.2623	0.2878
S06E06 The Gymnast	0.242	0.2811	0.229	0.2479
S06E07 The Soup	0.2734	0.2199	0.3163	0.1904
S06E13 The Scofflaw	0.2261	0.1968	0.276	0.3011
S06E17 The Kiss Hello	0.2562	0.2675	0.2511	0.2253
S06E22 The Diplomats Clul	0.2495	0.2944	0.238	0.2181
S07E03 The Maestro	0.2	0.2963	0.2686	0.2351
S07E04 The Wink	0.3225	0.2265	0.2696	0.1814
S07E07 The Secret Code	0.2481	0.244	0.2733	0.2347
S07E17 The Doll	0.2674	0.2367	0.2667	0.2293
S08E05 The Package	0.2923	0.2247	0.2475	0.2354
S08E06 The Fatigues	0.239	0.2583	0.2824	0.2203
S08E09 The Abstinence	0.2442	0.245	0.2733	0.2375
S08E10 The Andrea Doria	0.2521	0.1934	0.3008	0.2538
S08E12 The Money	0.2509	0.2529	0.2416	0.2546
S08E19 The Yada Yada	0.2676	0.1873	0.2775	0.2676
S09E09 The Apology	0.2175	0.2786	0.2704	0.2335
S09E18 The Frogger	0.2571	0.1951	0.2577	0.2901

Figure 9: Funniness vectors clustering - the green means maximum funniness (in that episode) while red means minimum.

The patterns are quite interesting. Cluster 1 is without a doubt Jerry's cluster, while Cluster 2 is Kramer's. Cluster 3 seems to be episodes which are equally held by the duo Kramer and George, as they are both usually funnier than their counterparts, taking the lead spot interchangeably. Cluster 4 is where Elaine's funniest episodes are, but also the most balanced one out of all clusters.

2.6 Funniness of Words

We'd like to see if there are trigger words for humor in Seinfeld. To do so, we have created a funniness measure for words:

$$word\_funiness = P(laugh|word) \cdot min\left(\frac{occurrences(word)}{\mu(word\_occurrences)}, 1\right)$$

The left part of this metric is straightforward - the number of the occurences of the word in funny lines divided by the number of all occurrences. The right part is for normalization of the score of words that rarely occur in the corpus. Also, all words have been lemmatized by nltk's WordNetLemmatizer.

According to this metric, the word which is most likely to trigger laughter in Seinfeld is "shut." A closer look at the corpus reveals that this must be due to various instances of the expression "shut up," that creates the comedic effect. "Up" will not be a funny word since its appearance in non-funny lines lowers its funniness score. Also the fact that the words "toilet" and "stink" are up there in the top 10 shows that talking about toilet or bad smells is, at least 50% of the time, funny.

For a more general analysis of the trigger words, we divided the top 30 of them into LIWC categories. The leading category was Affect: "interesting," "killing," "hope," "damn" and "bastard." The runner up was the Anger/Negemo category: "killing," "damn," and "bastard." The following 3 categories were tied: Physical/body words ("face," "body," "foot"), Present words ("hope," "drive," "speak") and Social words ("speak," "baby," "chicken").

Another interesting result is funny word #20: "baby", with 0.48 probability of a laughter following it. What's so funny about the word 'baby?' It turns out that a lot of the "baby" occurrences are George saying things like "I'm back, baby!" There is even a line by Jerry where he says to Elaine "Oh, that's gold, baby." to which she replays "Baby? Are you doing George now?"
This led us to believe that it is reasonable to assume that trigger words are character-dependent. So the next thing we did, is measured trigger-words for each character.

ALL CHARACTERS	
shut	0.583
nut	0.553
killing	0.533
body	0.514
toilet	0.513
beat	0.513
stink	0.51
hang	0.502
war	0.502
pop	0.502
kept	0.502
full	0.5
ball	0.493
hope	0.489
sometimes	0.488
either	0.486
mickey	0.486
light	0.485
drive	0.48
baby	0.479
face	0.476
damn	0.475
york	0.475
sponge	0.471
bastard	0.471
speak	0.471
chicken	0.47
foot	0.469
interesting	0.468

Figure 10: Word funniness.

JERRY		GEORGE		ELAINE		KRAMER	
anybody	0.684	whatever	0.765	susie	0.625	done	0.769
glass	0.66	piece	0.651	move	0.615	walk	0.667
sound	0.615	called	0.647	shut	0.6	eat	0.645
found	0.611	stay	0.625	head	0.556	pig-man	0.645
run	0.609	costanza	0.6	than	0.5	enough	0.643
either	0.6	took	0.588	face	0.5	guess	0.632
wife	0.588	stupid	0.588	business	0.486	mama	0.615
easy	0.588	toilet	0.586	top	0.486	crazy	0.615
while	0.583	baby	0.579	walk	0.486	only	0.609
open	0.571	yada	0.577	away	0.471	away	0.6
than	0.565	house	0.56	stupid	0.471	real	0.591
waiting	0.556	hand	0.56	again	0.469	chicken	0.588
lost	0.55	used	0.545	everybody	0.467	own	0.571
everyone	0.55	mother	0.531	more	0.459	face	0.571

Figure 11: Word funniness for each character.

The results are presented in Figure 10. Although there are a few similarities between general trigger words and character's trigger words, it is not surprising to see that each character has an almost entirely different set of trigger words.
It's interesting to see that Jerry's trigger words do not seem very unique or telling. In Seinfeld's Wikipedia page, Jerry is described as the "voice of reason" amidst the general insanity generated by the people in his world. So it may come as no surprise that out of all the character's trigger words, his seem to be the least eccentric.
Other characters have more interesting results. For example, one of George's trigger words is his own last name. If George says "costanza," there is a 60% chance that it will be followed by a laughter. A closer look at the corpus reveals that some of those funny moments are when George introduces himself, or when he talks about himself in third body ("I put these glasses on a rock, you know what jumps into people's minds? Costanza"). The same goes for Elaine's "shut," - it is funny when she tells people to "shut up."
Some of these trigger words come exclusively from one episode, where they are used repeatedly for comedic effect. For example, Elaine's word "susie" is exclusively from an episode called "The Susie," where Elaine is mistakenly called Susie by a fellow worker. George's trigger word "yada" is only used in an episode called "The Yada Yada." Kramer's "pig-man" is from an episode called "The Bris."

2.7 Episodes That Have Joke Words

From the previous section we conclude that some episodes have "joke-words," i.e. trigger words that are specific to those episodes, which are part of the episode's ongoing joke. A word is considered an episode's joke word if:

$$P(funny\_word \in EP \mid funny\_word) \cdot min\left(\frac{occurrences\_in\_EP(funny\_word)}{\mu(all\_funny\_words\_occurrences)}, 1\right) > 0.7$$

This measure is almost identical to the funniness measure, but instead of calculating the probability that a word is funny and normalizing, we calculate the probability that a funny word is in a specific episode and normalize to weep out non-consequential words.
Not all episodes have joke words, but these are the ones we've found:

Episode	Joke Word	Joke Word Funniness
S04E07 The Bubble Boy	bubble	0.707
S05E05 The Bris	pig-man	0.707
S06E08 The Mom and Pop Store	jon	0.707
S06E19 The Jimmy	jimmy	0.846
S07E03 The Maestro	maestro	0.771
S07E10 The Gum	lloyd	0.737
S08E15 The Susie	susie	0.835
S08E19 The Yada Yada	yada	1
S09E03 The Serenity Now	serenity	0.899

Figure 12: Episodes that have joke words.

The fact that most of the joke words are in the episodes' titles is very telling, but even when the joke word is not in the episode's title, it is a central element in the episode's comedic mosaic. For example, in "The Bris," even though the episode is about a bris, Kramer is obsessed with a "pig-man" that he saw in the hospital, only to find out, by the end of the episode, that the man is not actually a pig-man. In "The Mom and Pop Store," George buys a subpar car for the sole reason that its previous owner was famous actor Jon Voight.

3 Conclusion and Future Work

In this paper, we described the methods that we used in order to generate The Seinfeld Corpus, which consists of 96 humor annotated "Seinfeld" screenplays. Every moment of audience laughter is denoted as ``LOL`` and its corresponding timing is above it (the timing refers to the peak of the laugh, i.e. the loudest point). The timing of the dialog is above and below each line. We purposely kept all the timing information in the corpus - timing is crucial when it comes to humor, it may prove helpful in the task of humor recognition.

The corpus can be useful in further pushing and developing new humor detection techniques. Building a classifier that successfully differentiates a funny line from an unfunny one, in the context of the episode, could prove a very interesting future work.

Another interesting future work could be the generation of more annotated corpora from other sitcoms using our method, especially since we've observed that Friends, The Big Bang Theory and How I Met Your Mother all have Stereo recorded laughs, which makes laugh extraction using our method a viable option.

We also did analysis of the data, trying to find patterns and tendencies in it. We've found Kramer to be the funniest character, with the highest probability that a line of his will trigger the audience's laughter, while Elaine is the least funny of all characters. Conversely, if laughter is measured indirectly, Elaine is the funniest character while Kramer is the least funny. We've also found that there are laughter trigger words across the whole corpus and character specific trigger words as well. Also, some episodes have joke-words, which are trigger words specific to those episodes. And lastly, by clustering the funniness data we've discovered that there are recurring patterns in the character's funniness balance across episodes.

An interesting future work could be to further the analysis with regard to semantic features (are some subjects funnier than others?) or timing (length of punchlines, length of silence before a punchline is delivered).