

Can we (artificially) understand Seinfeld's humor?

Dan Kufra (30219082-2 dank) [dan.kufra@mail.huji.ac.il]
Yuval Jacoby (30224707-7 yuvalja) [yuval.jacoby@mail.huji.ac.il]
Alon Netser (31160253-6, alonnetser) [alon.netser@mail.huji.ac.il]

February 26, 2019

Contents

1	Problem Description	2
1.1	Difficulties	2
1.1.1	Funny is not taken only from the text	2
1.1.2	Subjectivity of Funny	2
1.1.3	Context and Timing	2
2	Data	2
2.1	Introduction	2
2.2	Specs	3
2.3	Data Validation - no dataset is perfect in our world...	3
2.3.1	The first flaw: the laugh-tracks	4
2.3.2	The second flaw: the speaking character is sometime mislabeled.	4
2.4	Data Visualizations	4
2.4.1	Finding Meaningful Features for a Sentence	4
2.4.2	Word-Clouds	5
2.4.3	Characters Graph	5
3	Experiments	5
4	Examining the Results	5
5	Future Work	8
6	Conclusion	8

Abstract

Who among us isn't familiar with the hilarious sitcom "Seinfeld"? We all laughed from Jerry's stand-up scenes (YouTube), George Costanza ridiculous behaviour (YouTube), Kramer facial expressions and bursting into the room (YouTube), and Elaine pushing everyone away and shouting "Get Out!" (YouTube).

The main question we ask is - can we build a model to understand this humor like we do?

1 Problem Description

We want to build a model that “understands humor”, i.e. given the text of a scene in Seinfeld, will predict funniness for each sentences.

First, we validate the data. We find certain flaws and inaccuracies, and think about how to deal with them. Second, we analyze the data and visualize its content. Then, we visualize the connections between the different characters in the show, and build an interactive graph describing these relations.

1.1 Difficulties

There are many difficulties regarding this problem.

Here are a few:

1.1.1 Funny is not taken only from the text

When we watch Seinfeld and laugh, it is not only because of the text. Most of the times it is affected from the tone, and more generally the whole scene’s video. For example when Kramer enters the room demonstratively “in a funny way”. These aspects are not always obvious in the text itself.

1.1.2 Subjectivity of Funny

The laugh-track does not necessarily mean that this is funny. However, this issue is not that severe. Indeed, the laugh-track does not always indicates funniness, but the writers who decided to add the laugh-track at a certain point certainly thought that this might be funny. Therefore, learning when a laugh-track will occur is also interesting because it is the same as predicting if some human-being thought it might be funny.

Second, the accuracy of the laugh-track is quite good - most of the times there is indeed something funny going on. Moreover, the coverage of the laugh-track is quite close to 100%, i.e. there are (almost) no funny lines lacking a laugh-track.

Anyway, this issue is the reality and we can not change it. Even if we would label the sentences by ourself, it would still mean that some human-being (i.e. us) is thinking there are funny. It does not necessarily mean that another person will think it’s funny.

1.1.3 Context and Timing

The other (many) difficulties regarding natural-language-processing that were mentioned in the lecture also affect us (ambiguity, language is not static, etc).

Moreover, the context and timing are really important when understanding humor. When we watch an episode of Seinfeld we understand the overall topic, and remember events that occurred previously in the episode (and even in other episodes). In order to make the computer understand the humor as well, we must use a model that is capable of understanding context and “remember” important events from the past.

2 Data

2.1 Introduction

The data we use is basically Seinfeld’s subtitles, coupled with the speaking character and timing - both the the sentences said and the laugh-tracks.

We downloaded the dataset from GitHub, and we thank Ran Yad Shalom and Yoav Golberg for the great dataset they built [1].

2.2 Specs

The dataset contains 96 humor annotated “Seinfeld” screenplays, along with the timing of the laughter and the timing of the dialog.

R. Y. Shalom and Y. Golberg got the subtitles from opensubtitles.org, the scripts from seinology.com and used the audio tracks to extract the exact timing of the laugh-tracks. Furthermore, they used quite sophisticated techniques to align the subtitles with the exact timing, and attach the speaking character for every sentence using the scripts. You can further read about it in [1].

Due to the technic they used to build the data (using the fact that the dialogs were recorded in mono, and the laugh-track in stereo) we have the episodes starting at season 4 episode 6. This is because the previous episodes were not recorded this way.

There are 46,497 sentences in total, associated with several properties:

- ‘character’: The speaking character.
- ‘txt’: The text.
- ‘start’: Start time (in seconds).
- ‘end’: End time (in seconds).
- ‘is_funny’: Whether a laugh occurred after this sentence or not.
- ‘laugh_time’: If this sentence is funny, this is the duration of the laugh (in seconds). Note that if the sentence was not funny, this is set to NaN.
- Various meta-data about the episode, such as ‘season’ (season’s number), ‘episode_num’, ‘episode_name’, ‘total_lines’, ‘global_episode_num’ (in the whole dataset).
- Useful features of the sentence, such as ‘num_words’, ‘length’ (in seconds), ‘line_num’ (in the episode), ‘avg_word_length’ (in letters).

For example, here are 3 lines from our dataset (meta-data such as season, episode, etc omitted).

character	txt	start	end	is_funny	laugh_time
SUSAN	I told you to take the offer.	199.003	201.469	F	
GEORGE	Look, I had nothing to do with this. It wasn't my decision.	201.539	205.7	F	
GEORGE	It was Jerry. Jerry told me. I'm the creative guy.	206.044	209.341	T	208.3

2.3 Data Validation - no dataset is perfect in our world...

Remark. This section was done in an interactive Jupyter Notebook named ‘Data_Cleaning.ipynb’. You are more than welcome to take a look!

The first task was analyzing the dataset and validate it. We watched several episodes and looked at the dataset at the same time.

We saw that the text is pretty accurate, except minor changes the man (or machine) who translated did. These changes are not really an influence to our task. For example, “You met her in the supermrket. How did you do that?” is shorten to “You met her in the supermrket. How?”.

The timing of the talking are also pretty accurate, and by reading the paper of Ran Yad-Shalom [1] who created this dataset he addressed this issue specifically and payed extra attention to take several subtitles and choose the one that is best aligned with the audio. His job here is pretty impressive.

2.3.1 The first flaw: the laugh-tracks

The laugh tracks are not perfect. It's mostly labeled correctly, but there are some false-positives. Also, the timing of the laughs are not measured perfectly. It's quite understandable, because in the show it's also not so obvious when the laugh start and when it ends. However, because most of the time it's actually labels correctly, we treat this as acceptable noise in our dataset and continue with our work.

2.3.2 The second flaw: the speaking character is sometime mislabeled.

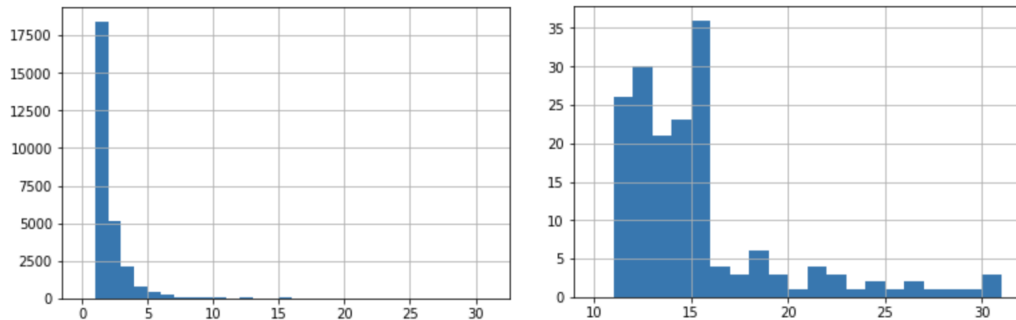
We noticed that the speaking characters sometime isn't correct. We found an attribute that might be the cause for mistakes in the speaking character labels - one character speaking a lot of sentences in a row. It seems that while building the dataset, some characters labels "get stuck" on the same character (possibly wrongly).

We found that this is often when some character is speaking more that 10 sentences in a row. Indeed, this is quite rare for the same character to speak more than 10 sentences in a row. There are exceptions - for example when a person is talking over the phone, or when Jerry is doing his stand-up scenes.

To do this, we collected all these cases, and analyzed them statistically. We saw that the mean sentences-in-a-row is around 1, and that 99% of the times it is below 8.

Our solution was to mark these speaking characters as 'unknown', but keep the other attributes (such as text, duration, funniness, etc).

DOES IT?



(Left) Number of sentences in-a-row.

(Right) Number of sentences in-a-row, bigger then 10 (the "tail").

2.4 Data Visualizations

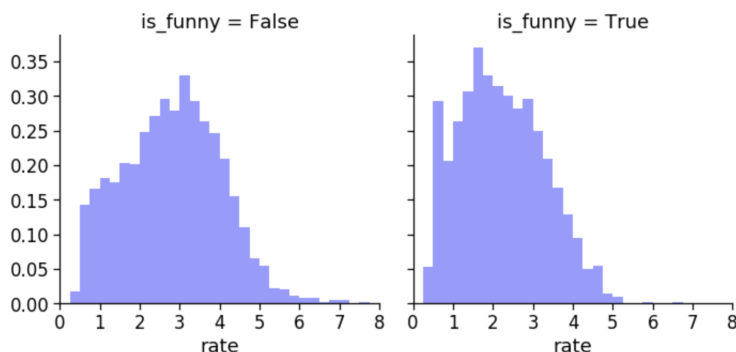
Remark. This section was done in an interactive Jupyter Notebook named 'Visualizations.ipynb'. You are more than welcome to take a look!

We tried to visualize several aspects of the data, in order to get more insight about it and get ideas about how to solve it.

2.4.1 Finding Meaningful Features for a Sentence

We wanted to see if there are differences between funny and not-funny sentences in several aspects, such as length (in seconds), number of words in the sentence, speech-rate (words per second), etc. We saw that the distributions of the length / #words is slightly different between funny and not-funny sentences. However, these differences are not significant, as the standard deviations is also pretty high. We found the the length in seconds and the length in #word behave different, and the speech-rate seems also like a good feature (see the figure below).

Our conclusions were to add these features to the sentence. We saw that the results actually improved when we gave them these features.



Elaine's speech-rate

2.4.2 Word-Clouds

Another nice visualizations we did was to analyze the speaking characters. One thing was to make a word-cloud for each character, as well as to separate for funny/not-funny sentences and see if we can see any difference. We can see that there are many common frequent words between funny/not-funny sentences. However, we can see some interesting issues. For example, in Jerry's not-funny sentences the word 'Elaine' is more common. Can we learn that when Jerry says 'Elaine' in a sentence it is less likely to be funny?

2.4.3 Characters Graph

Another nice visualization was the characters-graph, containing the different characters and their connections. Each edge's weight is proportional to the amount of times the characters spoke to each other. One can see perfectly the "community" of the 4 main characters (Jerry, George, Kramer and Elaine), as well as other secondary characters (such as George parents Monty and Helen).

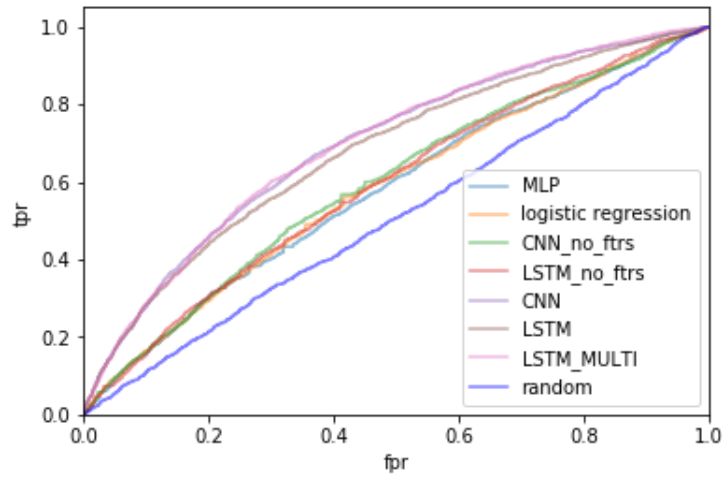
3 Experiments

TODO

4 Examining the Results

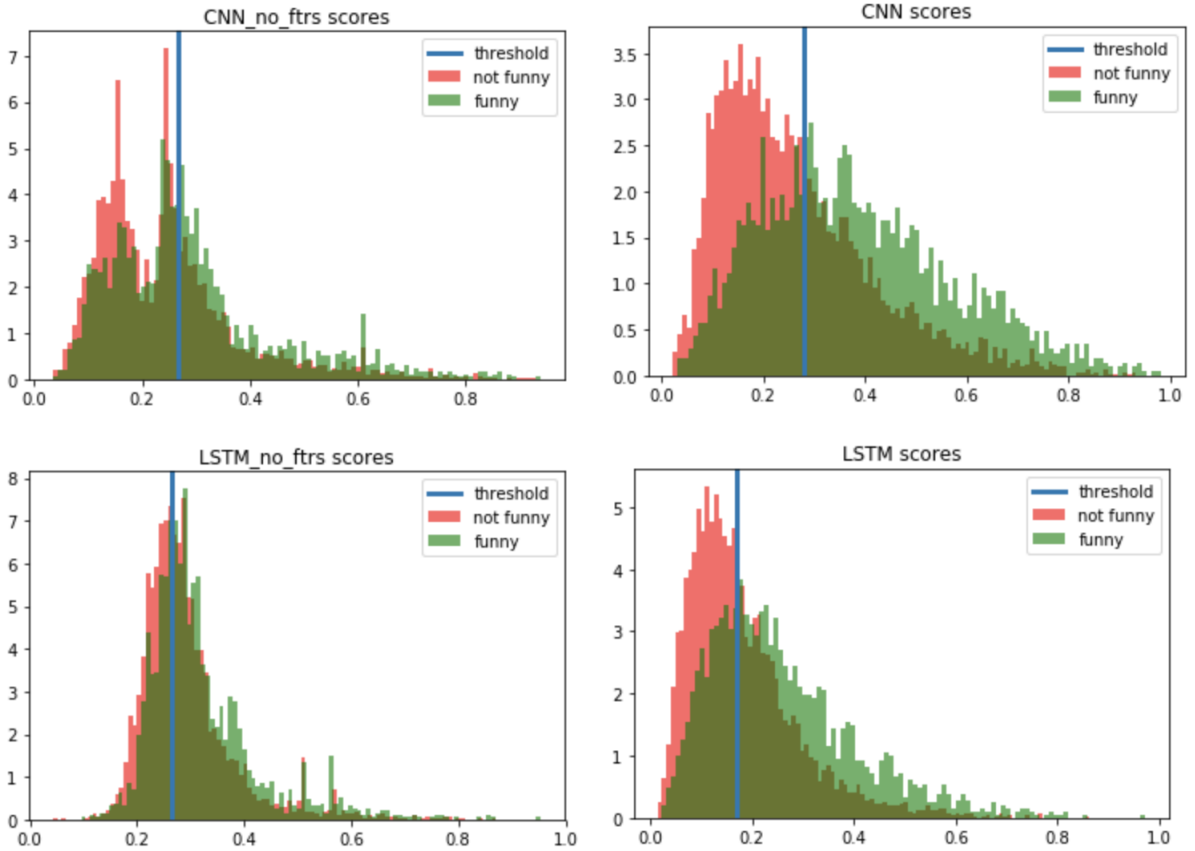
Remark. This section was done in an interactive Jupyter Notebook named 'Analyze_Results.ipynb'. You are more than welcome to take a look!

After we finished training the different models we built, we turned to analyze their performance. We looked visually at some of the results, and plotted several measures.



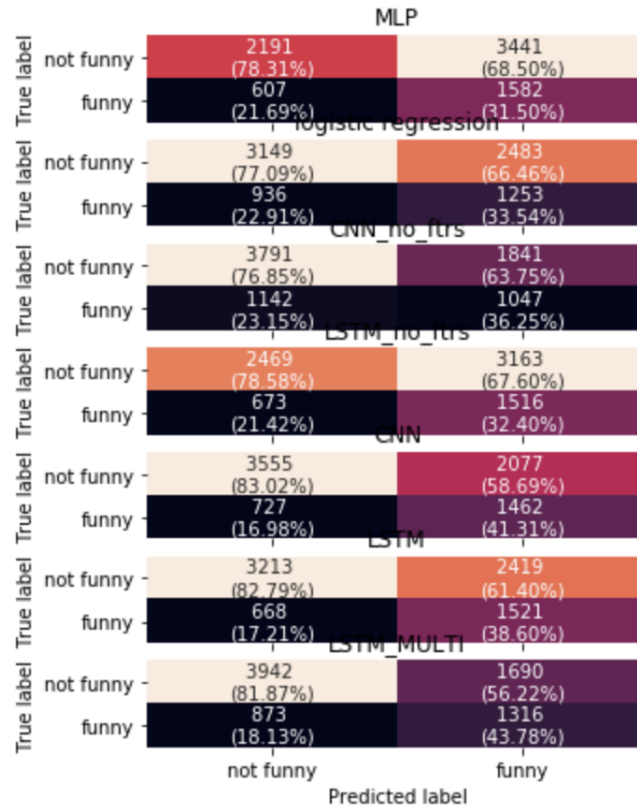
ROC-curves

We saw that the features we added helped the models greatly. Here are the scores for the CNN and LSTM models, with/without the additional features.



Scores with/without the additional features.

Numerically, here are the confusion matrices for the different models:



Confusion matrices

It's not perfect, but it definitely learned something. We must remember that our data is quite noisy, so nothing can do it perfectly.

We tried to split to the 4 main characters and see if the performance are different when we look at certain character. We found that there are minor changes, but overall they're all quite the same. You can see for yourself in the notebook.

Furthermore, we wanted to look at some of the results visually.

Here is the most severe false-positive (i.e. the model thought it's funny but it doesn't). We also add some context (5 sentences before the allegedly 'funny' sentence).

LSTM score is 0.86

character	txt	is_funny
ELAINE	"I'm very impressed"?	False
ELAINE	You mean "pressed" because it's a dry cleaner?	False
JERRY	Yeah, see? That's why I hate it.	False
JERRY	So come on, you wanna go?	False
ELAINE	Well, what about the sleeping arrangements in the cabin?	False
JERRY	Well... Same bed, and underwear and a T-shirt.	False

"Severe" FP

Here are the severe false-negatives (i.e. the model thought it's funny but it doesn't):

```

LSTM score is 0.03
character                               txt  is_funny
  ALAN      I hope we can get  past all this.    False
  ALAN              Oh, past? We're way past.    False
  ALAN              So you have a big head.      False
  ELAINE              So what?                   True
  ALAN  Goes well with the bump  in your nose.    False
  ELAINE              What?                      True

LSTM score is 0.031
character                               txt  is_funny
  ELAINE              I spoke to Alan.           False
  ELAINE  You know, I told him  I didn't wanna see him anymore.  False
  ELAINE              Called me "big head."       True
  JERRY              [SCOFFS] Big head?          False
  JERRY              It's almost a compliment.    True
  ELAINE  It's one of the nicest things  anyone's ever said to me.  True

```

“Severe” FN

These mistakes don’t seem that severe. By looking at it (without checking the label) one can think it’s label is the opposite. It’s shows how difficult this task is.

5 Future Work

TODO

6 Conclusion

TODO

References

- [1] Ran Yad-Shalom and Yoav Goldberg, The Seinfeld Corpus: A Method for Generating A Humor Annotated Corpus and An Analysis of That Corpus. Computer Science Department, Bar-Ilan University, Israel, 2017.