1. Project design
The goal of my project was to create a data driven map of the characters in human stories and discover the archetypes around which they cluster. This would be useful both as a tool for story analysis (Joseph's Campbell's *A Hero with a Thousand Faces* was a boon to Hollywood) and as a model to inform automated story generation.

2. Tools
1. `gensim` for doc2vec and LDA
2. `spacy` for text parsing, tokenization, and lemmatization
3. `sklearn` for clustering, projection, and other utils (e.g. `CountVectorizer`)

3. Data
My data source was roughly 14,000 character articles from Wikipedia. I extracted these from a full Wikipedia dump by iterating through all the articles and checking each's category tags against a set of regex patterns.

Unfortunately there seems to be some disagreement about which characters should have stand alone articles and what those articles should contain. Fans of a character want to write the long, detailed biographies that would have been useful for this project, but Wikipedia's moderators discourage them from writing about characters as if they were people. As a result, many obscure comic book characters have longer articles then iconic ones like Darth Vader, whose article is mostly about the design and portrayal of the character in various media, not the events of his hypothetical life.

3.1. Data cleaning
I wanted to cluster characters based on their actions (kill, voyage, rescue) and attributes (nice, mean, sociopathic) and not their genre, so I wrote a script that tried to reduce an article to a set of clauses where each clause's subject was the subject of the article. It did this by walking up the parse tree of the matching subject to the verb and extracting all the tokens in that verb's subtree. To increase the number of extracted phrases, I also did some simple pronoun resolution by assuming that pronouns that followed the subject in a chain unbroken by another proper noun referred to the subject. I also replaced all other proper nouns and named entities with <ENITY> meta-tokens. The output for Bender's article is in appendix 1.

4. Algorithms
I used `gensim`'s implementation of doc2vec to train 50 dimensional document embeddings. I kept the embedding size relatively small because I was looking for abstract relationships and increased dimensionality increased training time without noticeably better results. The presented embeddings were trained for 5000 epochs.

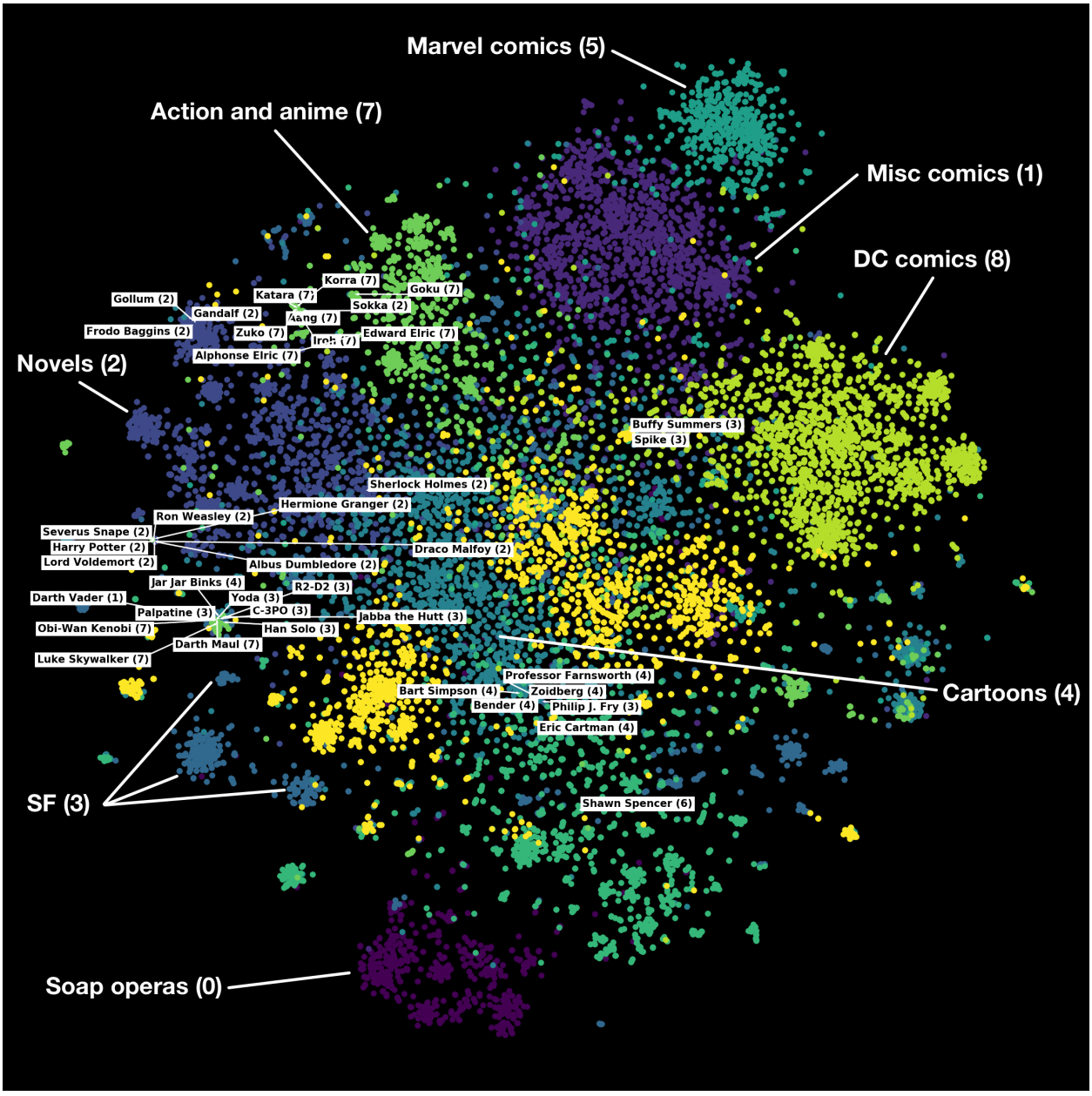
After training, I applied k-means clustering to the (scaled) embeddings. My elbow plot of inertia didn't have an elbow, so I used a set of about 20 benchmark characters to tune the number of clusters.

To visualize the results, I used t-SNE to reduce the dimensionality from 50 to 2 and plotted the points with different per cluster colors.

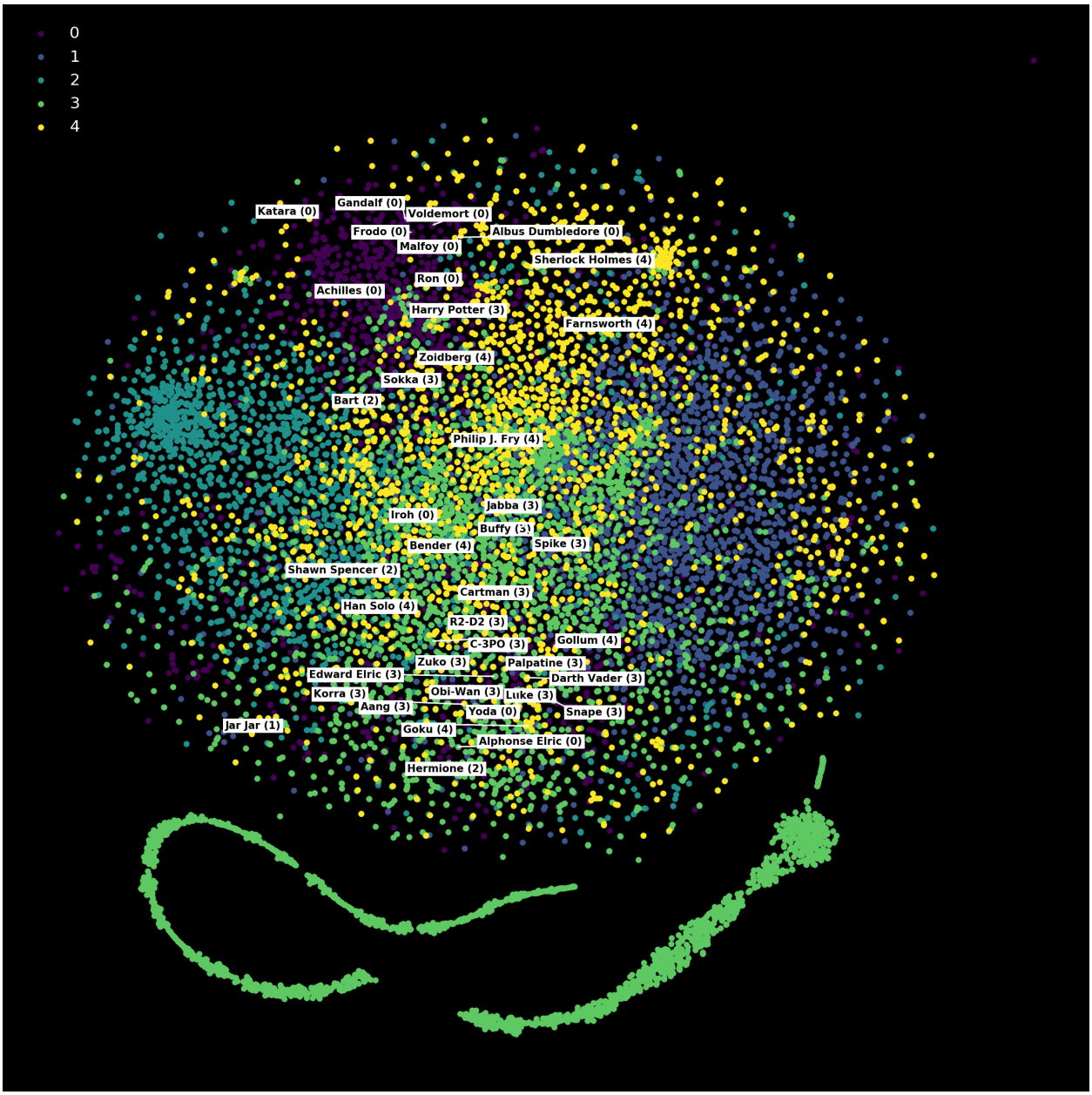
I also ran LDA on both the filtered articles (400 passes) and unfiltered articles (100 passes). In both conditions, I used a topic count of 20.

Results

On the full tokenized article text, I was able to find coherent clusters:



On the filtered text (lemmatized extracted clauses, proper nouns, meta-tokens, and stop words removed), I was less successful:



The results of the LDA topic analysis were similar. I found clear topics in the unfiltered text for genre and fictional universe. In the filtered data set the topics were either indistinct or uninteresting, though one that appears to be about fratricide does stand out:

```
0.001*"when" + 0.001*"brother" + 0.001*"die" + 0.001*"in" +
0.001*"take" + 0.000*"say" + 0.000*"kill" + 0.000*"like" +
0.000*"name" + 0.000*"attempt"
```

The only place I found some possible evidence of the archetypes I was looking for was when looking directly at vector similarity:

```
1  d2v_model.docvecs.most_similar('Yoda')
```

```
[('Qui-Gon Jinn', 0.6633567810058594),
 ('Luke Skywalker', 0.6167474985122681),
 ('Optimus Prime', 0.6107190251350403),
 ('Scourge (Transformers)', 0.5825557708740234),
 ('Kanan Jarrus', 0.5685822367668152),
 ('Zuko', 0.5668112635612488),
 ('Gimli (Middle-earth)', 0.565727710723877),
 ('Mokujin', 0.5602509379386902),
 ('Albus Dumbledore', 0.5537698268890381),
 ('Beorn', 0.5510784387588501)]
```

```
1  d2v_model.docvecs.most_similar('Bender (Futurama)')
```

```
[('Michael Stivic', 0.6083042621612549),
 ('The Tunnelers', 0.6009865403175354),
 ('Eric Cartman', 0.5820137858390808),
 ('Sammy Seminole', 0.5690814256668091),
 ('Kayako Saeki', 0.5690186619758606),
 ('Zorak', 0.5530416369438171),
 ('Connor Walsh (character)', 0.5502055883407593),
 ('Stan Smith (American Dad!)', 0.5455904006958008),
 ('Meg Griffin', 0.5436952114105225),
 ('The Mooninites', 0.5256187915802002)]
```

What didn't work

- I failed to find the archetype clustering I'd hoped for in the filtered data. This might have happened because:
  1. Not enough, too much, or the wrong data was thrown out during filtering.
  2. I removed lemmas, stop words, and the meta-tokens when training the filtered dataset because that produced results on trial runs. Perhaps these contained important information.
  3. The archetypes don't exist or don't manifest in the dataset.
- I tried reducing dimensionality with PCA before clustering and visualization but didn't get a noticeable improvement.
- I tried other clustering methods: DBSCAN labeled all my benchmark characters as outliers; mean shift assigned them all to the same category; and spectral clustering didn't return results in an acceptable time frame even running on AWS.

- I tried other filtering methods like simply removing named entities and truncating the articles at certain sections.

Appendix 1. Clauses extracted from Bender's article

1. according to the character 's backstory , <SUBJECT> was built in <ENITY> , <ENITY> .
2. <SUBJECT> , a high-tech industrial metalworking robot , was built in <ENITY> at <ENITY> , a manufacturing facility of <ENITY> 's <ENITY> in <ENITY> , <ENITY> .
3. unlike most other robots , <SUBJECT> is mortal and , according to <ENITY> 's calculations , may have <ENITY> years to live .
4. after reporting that defect to his manufacturer , <SUBJECT> barely escapes death from a guided missile and a robot death squad dispatched by mom in order to eliminate him and effectively take the defective product off the market .
5. <SUBJECT> had a job at the metalworking factory , bending steel girders for the construction of suicide booths .
6. <SUBJECT> has an apartment in the " <ENITY> .
7. although the pair enjoy living together , <SUBJECT> is sometimes portrayed as manipulating his guileless friend .
8. in the series ' early episodes , <SUBJECT> is shown preferring to occupy smaller areas of their apartment , like the closet , referring to them as " cozy " , although in later episodes <SUBJECT> is shown to have his own individual bedroom , like <ENITY> .
9. <SUBJECT> hates magnets and has a near-pathological fear of electric can openers , as magnets cause <SUBJECT> to uncontrollably start singing folk songs when near his head ; magnets shut off his inhibition , causing <SUBJECT> to reveal his secret ambition to be a folk singer .
10. that <SUBJECT> is a " round peg in a square hole .
11. in <ENITY> , <SUBJECT> states that <SUBJECT> flipped a coin to decide his color , ending up with foghat gray rather than gold .
12. in <ENITY> , <SUBJECT> shows the kids a black-and-white mug shot of himself taken after his arrest for theft .
13. in " <ENITY> ' " , <SUBJECT> is shown trying to join a basketball team and makes <SUBJECT> taller by simply extending his legs .
14. in <ENITY> , <SUBJECT> claims <SUBJECT> also has a nose , but <SUBJECT> chooses not to wear it .
15. <SUBJECT> was designed specifically for the relatively simple task of bending straight metal girders into various angles .
16. when <SUBJECT> reboots .
17. presumably either <SUBJECT> has a separate " catchphrase drive " or the majority of the catchphrases are also pornographic .
18. <SUBJECT> is powered by alcohol-based fuels , which <SUBJECT> can convert into an electrical power source sufficient to operate not only himself , but also small household appliances plugged into his power receptacle .
19. <SUBJECT> is a classic narcissist .
20. being the show 's breakout character , <SUBJECT> has made several cameos in different episodes of the <ENITY> , another series by <ENITY> .
21. within the <ENITY> , <SUBJECT> has appeared in episodes <ENITY> , " <ENITY> vs. <ENITY> vs. <ENITY> " , " <ENITY> : impossible " and <ENITY> .
22. in the <ENITY> crossover episode <ENITY> , <SUBJECT> travels back in time in <ENITY> to <ENITY> to kill <ENITY> , whose dna is tied to the creatures rampaging in <ENITY> .
23. <SUBJECT> ends up befriending <ENITY> before learning that the creatures are in fact <ENITY> 's genetic offspring .
24. once the crisis is averted , <SUBJECT> goes into shutdown mode in the <ENITY> ' basement .

25. <SUBJECT> was still resting inert in the <ENITY> ' basement as of <ENITY> episode <ENITY> in which his empty body cavity was used to store the family 's cash nest-egg .
26. <SUBJECT> also makes a background cameo appearance in the <ENITY> episode .
27. <SUBJECT> has cameo appearances in several <ENITY> episodes .