(Role) Models

Wenqin Chen • Jordan Shedlock

Applied Natural Language Processing

Fall 2015

Project Goals

This project was inspired by a paper by Will Radford and Matthias Gallé [1], who used data from the Internet Movie Database (IMDb) to study the interaction of gender and types of film characters over time. They observe that, given the cultural influence of film, examining respective portrayals of male and female characters may provide insights on gender roles in the population at large. However, since qualitative examination of film portrayals has a great cost in time and effort, Radford and Gallé proposed using large datasets of IMDb credit lists (lists of roles in movies) to see how men and women are portrayed, how this changes over time, and how it corresponds to the actual distribution of professions in the population.

Their analysis was not without its problems. As Marti Hearst pointed out, main characters (who are probably the most significant in terms of cultural influence) are identified by name, rather than by role, which means that the analysis is biased towards minor characters. In addition, Radford and Gallé did not attempt to combine synonymous roles (e.g. “henchman” and “thug” or “police officer” and “police woman”). In addition to these, the authors specified some further shortcomings. Since their unit of analysis was a role in a single work, and these works could be films or television episodes, the data was heavily skewed in favor of television programs, which might have dozens or hundreds of separate records, each listing the same roles repeatedly. The authors also included all of the movies in the data set; since this includes movies from all over the world, it complicates their efforts to draw conclusions about cultural influence, since the creators and audiences of these films represent vastly different cultures (broadly speaking, this is true at any scale, but particularly so across countries and geographic regions). Moreover, some of the IMDb credits specify characters by name, others by role, and some by both. Finally, the dataset itself, which is to some extent crowdsourced, is biased (both in terms of observation and granularity of description) towards more recent films with large followings.

In our work, we set out to address some of these issues by extracting character roles from IMDb and Wikipedia summaries to identify major characters not just by name, but by character role (see below for a definition of our terms); by combining like character roles; and by limiting the analysis to films (as opposed to television episodes) produced in or with a major release in the United States.

The crux of our project, as it turned out, was extracting character role information from summaries and associating it with a name listed in the IMDb credits. Our starting point for this analysis is the (more or less) "official" lists of characters or roles for each movie, which we refer to as IMDb credit lists. The intuition, supported by empirical observation, was that major characters are often listed only by their given name ("character name," e.g. "Indiana Jones"). We contrast this with "character role," the role or function performed by that character in the movie. This may be a professional role (e.g. "archaeologist") or not (the credit list may contain the character name "CJ Cregg" which corresponds to the character role "press secretary," or the character name "Frank Costello" might relate to the character role "mobster" or "crime boss"). Using the IMDb credit lists, we mined plot summaries from IMDb and Wikipedia to extract name variants (character names that matched those contained in IMDb credit lists), then examined the text around them using regular expressions and part-of-speech chunking to find patterns that might identify the character role associated with that character name. We then processed this text to find candidate character roles, which could thus be associated with the IMDb credit (and, thus, a gender).

In sum, we developed a method for extracting character name variants from film summaries and associating them with IMDb credit lists, then extracting character roles associated with those names. **Our dataset contained ~34,000 movies; we found 114,922 character name variants in the summaries, for which we were able to extract 71,216 candidate character roles (2.1 roles per movie)**. A preliminary examination of these candidate character roles by gender suggests interesting patterns in the depiction of men and women in films. We addressed the television/film balance and cultural significance concern by limiting our analysis to American-made or -released films.

We evaluated our results by inspecting 10 films from a variety of genres. The movies are selected more or less at random from those known to the reviewer, deliberately including a variety of genres, and excluding sample used during development. First, we manually examined the collected IMDB and Wikipedia summaries (without prior knowledge of the machine tagged results) to see which characters were named in them and whether there was a character role associated with them. Then we checked that against the results produced by our algorithm to see whether these characters had been identified and matched with a character role within the summary by our algorithm. Of those that had been associated with a character role, we looked at how many were correct. **Overall, our algorithm achieved a 54% precision (proportion of correct, descriptive character roles among those that were matched), and 46% recall (character names from summaries correctly matched with IMDB credits).** The remaining ones either failed to match (despite the

presence of descriptive or associative language around the names), matched incorrect or irrelevant words (e.g. a salesman associated with “bar”), or included a single correct term and a large number of incorrect ones (other characters’ names and roles, or other non-descriptive language). For some reason, both in this sample and in other empirical observations, our algorithm seemed to perform well for crime movies and Disney movies. The best-performing movies from this sample were *The French Connection* (81% of character names in summary were matched with credits from the IMDb credit list, and 78% of those were correctly described), Disney’s *Aladdin* (62.5% matched, 50% correct), and *The Life Aquatic with Steve Zissou* (40% of roles matched, 91.6% correctly identified). At the other end of the spectrum, *Saving Private Ryan* had only 50% of roles matched and only 17% correctly, and *A Clockwork Orange* with only 25% matched and none correct.

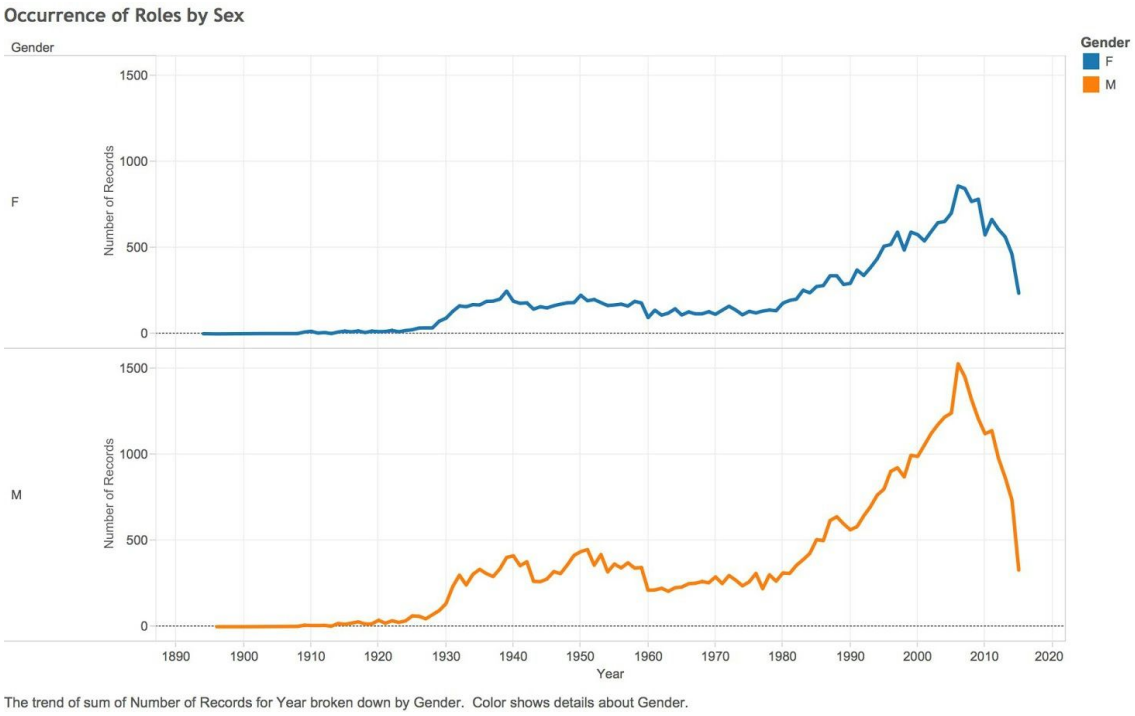
Movie	**Characters Named in Summary**	**# matched with role**	**% matched with role**	**# correct**	**% correct**
Glengarry Glen Ross	11	2	18.18181818	1	50
Saving Private Ryan	12	6	50	1	16.66666667
Aladdin	8	5	62.5	2.5	50
All the President's Men	15	5	33.33333333	5	100
Roman Holiday	4	2	50	1	50
The Bonfire of the Vanities	6	3	50	1.5	50
The French Connection	11	9	81.81818182	7	77.77777778
The Life Aquatic	15	6	40	5.5	91.66666667
The Bridges of Madison County	4	2	50	1	50

A Clockwork Orange	8	2	25	0	0

Evaluation details at:
https://drive.google.com/file/d/0BzlzBMSKz9RgdDJvazVnLXRkWGs/view?usp=sharing

There are some issues we would have liked to investigate if we pursued this project further. First, like Radford and Gallé, we did not group like roles together, so this still leads to the under-representation of roles with multiple synonyms. We might be able to draw stronger conclusions by using Word2Vec or Wordnet to associate synonymous roles, possibly by clustering them using vectors of context words. Second, we would like to improve our algorithm to achieve greater recall and precision. The most immediate way to do this would be to improve the regular expressions and chunking grammar with more patterns, in order to catch candidate character roles that are currently missed, and reduce spurious ones. However, there is a limit to what can be found with patterns, and as David Bamman and Ramakrishna Akella pointed out during the project showcase, a more productive approach would be to use machine learning techniques to examine the context of character name references and infer their roles on that basis. Third, we would attempt to clean up the IMDb credit lists further. Because credits are listed by actor (and each can play more than one role), and because there is no standard format, credits may include multiple character names ("Sir Lancelot/Black Night"), an associated character name and role ("Allison's Maid Joanne"), or multiple character roles. These may be delineated by a hyphen, slash, comma, or not at all. Commas may denote either a name-role split, or a surname-given name split. We tried to resolve this by splitting up compound credits into names and roles, but were unable to do successfully and consistently.

Looking at the roles we extracted, 45,700 are male and 25,516 are female, numbers which (as our classmate Anubhav Gupta pointed out during the showcase) should be taken into account in examining the data. Overall, male and female roles seemed to be subject to the same trends in moviemaking and representation: the number of roles per year seems to vary more or less equally for each. However, men are much more highly represented than women (see figure below).



The breakdown of extracted roles by gender was also interesting. Family and relationship roles dominated the top spots for both genders, but beyond that, there was a strong trend towards professional and leadership roles for men, and traditionally feminine, family-defined, or service roles for women.

Top 30 roles by gender over time as extracted by our algorithm from summaries

Men	**Women**
1. Friend	1. Wife
2. Brother	2. Daughter
3. Son	3. Girlfriend
4. Father	4. Sister
5. Man	5. Mother
6. Husband	6. Friend
7. Boyfriend	7. Girl
8. Detective	8. Woman

9. Agent	9. Actress
10. Partner	10. Singer
11. Boss	11. Student
12. Owner	12. Secretary
13. Friends	13. Widow
14. Boy	14. Niece
15. Officer	15. Friends
16. Captain	16. Assistant
17. Leader	17. Nurse
18. Director	18. Teacher
19. Lawyer	19. Reporter

20. Student	20. Sweetheart
21. Professor	21. John
22. Manager	22. Aunt
23. Attorney	23. Agent
24.Uncle	24. Neighbor
25. Reporter	25. Star
26. Assistant	26. Cousin
27. Cousin	27. Teenager
28. Lieutenant	28. Waitress
29. Producer	29. Partner
30. Businessman	30. Mistress

The temptation to draw conclusions about the effects or representativeness of American movies is very strong, but it is important to remember that there are limitations to this data. The problems with observation and level-of-description bias remain, as does the problem of synonymous roles. The fact that we did not successfully split up IMDb movie credits into multiple roles, or roles and name where applicable. There are also

multiple extracted roles for a single IMDb credit, which can bias the representation. When we tried to examine trends in top roles across decades, we were unable to see any meaningful trends, due to the sparsity of data for early films, the predominance of family roles, and the imbalance of movie data across years. Finally, recognizing the limits of our algorithm (based on the recall and precision abilities mentioned above), we cannot claim that these extracted roles are necessarily truly representative.

Description of Data

We used movie data from Wikipedia and IMDB, and name data from the Social Security Agency and Kantrowitz et al. Citations for datasets are in the Works Cited section.

Data cleaning was a big task for our project.

- We started with the IMDB role dataset, which contained movie titles, year, reviews, venue (TV or film), summaries, and credit lists.
- we then associated the IMDB actors and actress datasets with the IMDB credit list roles that they played. This allowed us to infer the genders of the roles.
- Then, we refined the quality of our dataset by filtering out TV shows, non-USA films, and titles with fewer than 10 IMDB reviews. We exploited the fact that TV episodes were listed in the IMDb files in a slightly different format, which made it possible to identify them. We also used a separate IMDb list that identified movies by country, and used this to filter our initial dataset, retaining only the movies listing a US release.
- Then, we matched the movies in Bamman's Wikipedia dataset with our IMDB dataset, combining duplicate titles into single entries. At this point, we have a dataset of US movies with at least 10 IMDB reviews, with their role credits, role genders, and IMDB and/or Wikipedia reviews. This amounted to about 34,000 files.
- In addition, we took the SSA first name dataset and the CMU surname dataset, and combined them into one set of names.

Description of Algorithms

Before going into the details, here is an overview our three main steps:

1. **Associate character roles with IMDB role credits**
    a. Extract character name variants from summaries using Stanford NER tagger and SSA/CMU name dataset

i. We wanted to use character names as anchor points for our data extraction patterns. So our first task was to extract valid character names. We did this by first NER tagging the summaries. The tagged results had a decent coverage. But since our goal at this stage is to maximize recall (we would filter them later), we checked the NER tagged summaries against the SSA/CMU name dataset, and tagged more /PERSON entities.

ii. The NER tagger operated at the word level. However, many movie characters have names consisting of multiple words. In addition, we observed that some multi-word characters names were partially tagged as /PERSON, and partially tagges as /O (other). To effectively group these words together as one single character, we first grouped consecutive /PERSON words, and then, if there were consecutive capitalized /O words immediately following a /PERSON word, we grouped them with the /PERSON words. In the way, we can get a valid character name that may have been tagged something like Alison/PERSON Ann/PERSON Hunter/O.

iii. Before settling with the above tagging strategy, we initially tried lifting out character names from the IMDB credit roles using part of speech and regular expression, but this attempt turned out unsuccessful due to the short lengths of role credit phrases. We then tried tagging summaries using nltk.chunk.ne_chunk, but found it lower in accuracy compared to the Stanford tagger, and the returned tree structure was harder to postprocess using regex.

b. Filter extracted name variants against IMDB role credits, and associate variants with their respective IMDB credit roles

i. Having maximized the recall of character name extraction, we moved on to improve its precision. Since the IMDB credit roles hold the most comprehensive information on movies' character names, we wanted to filter the extracted name variants against names in the credit roles. we decided to use a bag-of-words approach for the filtering task, considering the difficulty of actually parsing out names from the credit roles, as discussed in the previous step.

We first tokenized both the IMDB credit roles and the extracted character names. We put all the words for each name in its separate bag. We then searched the bags of extracted character names against those of IMDB credit roles. If there was no match,

we discard the extracted name. If there was at least one match, we associate the extracted name with the most-matched IMDB credit role. If there was a tie in the matches, we chose the credit role with the fewest tokens.

Using this matching algorithm, we effectively discarded extracted director names, actor names, location names, deity names etc, and associated the valid character name variants with their credit roles.

c. Extract roles associated with IMDB credit roles using regular expressions and POS chunking
    i. We defined a set of regular expressions to capture phrases that we felt would associate character name variants with character roles, e.g. “<name> is a <role>,” “<name>, a <role>,” “<role> <name>,” “<name>, his/her/their <role>.” We ended up with 6 regular expressions in all. Running over the list of IMDb credits and associated character name variants, we retrieved all of the matches for these expressions.
    ii. We tagged each of the retrieved phrases with parts of speech, then used a chunk parser to pull out patterns of words that would be informative: sequences of words ending in nouns, adjectives followed by nouns, strings of proper nouns, or proper nouns preceded by other nouns. We used “chinks” to exclude prepositions, adverbs, “to,” determiners, and possessives, since we felt that these would not include character roles. The chunks were returned as candidate character roles and placed in the dictionary with the IMDb credit role.
    iii. We struggled with precision and recall in both the regular expression and chunking stages, since no word/punctuation sequence unambiguously refers to a single relationship, and we captured spurious roles such as other characters’ names, cities, companies, or common haunts. The regular expressions also worked across sentence boundaries at times, capturing words that were not related. However, refining them often led to a loss of information, so it was a difficult balance to strike.

2. **Associate IMDB credit roles with genders** (refer to data cleaning for description)
3. **Associated character roles with character genders**

a. Organized the extracted roles and gender for each IMDB credit role under the same dictionary entry with the character name as the key. This made it easy for gender-role associations.

iPython notebook link:
http://nbviewer.ipython.org/54cce534cba20a746f2d
GitHub link:
https://github.com/jwqchen/NLP_movie_roles
Link to Data:
https://drive.google.com/a/berkeley.edu/file/d/0BzlzBMSKz9RgRWVNRHpfbHlXLUE/view?usp=sharing
Project Flowchart:
View on Next Page

Role Info

Source: IMDB
Eg: {{'role': 'Alison Hunter', 'gender': F},
{'role': 'Alison's Maid Joanne', 'gender': F}}

Summaries

Source: IMDB, Wikipedia
Eg: 'Alison Ann Hunter is a lawyer who works in Jackson. She recently hired a maid, Joanne, for housekeeping..."

SSN Name Database

Stanford NER Tagger

NER Tagged Corpus:
Alison/PERSON Ann/PERSON Hunter/O is/O a/O lawyer/O who/O works/O in/O Jackson/LOCATION ./O

Tagger Enhancement:
1. Group consecutive /PERSON together
e.g. 'Alison Ann'
2. Extend group to capitalized words after the last /PERSON
e.g. "Alison Ann Hunter"

Turn Each Role into A Bag of Words

Names?

['Alison',
~~'Jackson',~~
'Joanne']

Selection Rules:
1. # of matches > 0
2. The highest # of matches
3. If tied, the lowest # of words in role_bag

{'Alison Hunter':
{'role_bag': ['Alison', 'Hunter']},

'Alison's Maid Joanne':
{'role_bag': ['Alison', 's', "Maid", 'Joanne']}

Names!

['Alison',
'Joanne']

Collect Names in One Place

{'Alison Hunter': {'role_bag': ['Alison', 'Hunter'],
"names_in_sum": ['Alison', Alison Hunter", "Hunter"]},

'Alison's Maid Joanne':
{'role_bag': ['Alison', 's', "Maid", 'Joanne'], **"names_in_sum": ['Joanne', "Joanne Gunther"]}**
}

Regular Expression + POS Chunkers

Extraction Steps:
1. Extract role-related text using regex.
e.g. 'Alison is a ([^.])+?\.'

2. POS tag the extracted text, then chunk by POS pattern.
e.g. ^<.*>*?(<NN.*>)

3. Extract the noun from the extracted POS parse tree using regex.
e.g. '[Hh](is|er) ([A-Za-z]+),? ' + char

Final Result!

{'Alison Hunter': {'gender': 'F', role_bag': ['Alison', 'Hunter'], "names_in_sum": ['Alison', Alison Hunter", "Hunter"],
'roles_in_sum': ['lawyer', 'wife']},

'Alison's Maid Joanne': {'gender': 'F', role_bag': ['Alison', 's', "Maid", 'Joanne'], "names_in_sum": ['Joanne', "Joanne Gunther"],
'roles_in_sum': ['maid', 'mother']}}

Team Member Contributions

Task	Wenqin	Jordan
Data Cleaning	80	20
POS and NER Tagging	100	-
Name Extraction	100	-
Name Filtering and Alias Association	100	-
Regex and Chunking	20	80
Result Evaluation	-	100
Result Analysis	-	100
Presentation and Writeup	50	50

Works Cited

[1] W. Radford and M. Gallé. "Roles for the Boys?" Mining Cast Lists for Gender and Role Distributions over Time. *WWW 2015 Companion*, May 18-22, 2015, Florence, Italy, pp. 323-329.

[2] D. Bamman, B. O'Connor, and N. Smith. Learning Latent Personas of Film Characters. ACL 2013, Sofia, Bulgaria, August 2013 (dataset).

[3] "Alternative Interfaces." Internet Movie Database [IMDb]. Accessed from http://www.imdb.com/interfaces on November 15, 2015 (dataset).

[4] M. Kantrowitz. Name Corpus: List of Male, Female, and Pet names. CMU Artificial Intelligence Repository. March 29, 1994. Accessed from http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/nlp/corpora/names/other/family.txt on December 3, 2015 (dataset).

[5] National Data on the relative frequency of given names in the population of U.S. births where the individual has a Social Security Number. Social Security

Administration. Accessed from https://www.socialsecurity.gov/OACT/babynames/limits.html on December 3, 2015 (dataset).