Funny, not Funny

Walter Squires, Kyungchan Lim, and Autumn Henderson

I. INTRODUCTION

Joke theft, or joke poaching, has been prevalent in society as early as the end of the 19th century, where vaudeville relied on stock materials[1] and became more prominent in the 1970s, where more prominent figures such as Robin Williams have been singled out for stealing certain jokes or whole acts[2]. In today's society, joke theft is still rampant, but with the emergence of social media, recording, and information being at our fingertips, there is more visibility on a comedian and their acts, leaving them vulnerable to someone studying their act in great detail with the information readily available. With that being said, determining the originator of the joke can be difficult and can lead to financial loss for the originator as other comedians make money off of the originator's creations.

Written works such as books and scholarly articles have streamlined processes in the legal world when someone utilizes their productions without proper citations. Prosecuting thieves of stand-up works is more challenging. Copyright laws defend the expression of an idea, but not the idea itself, leading to cases where a comedian can tell a joke about a topic with slightly different wording and it is hard to prove that it was stolen. Likewise, there are "informal, network-based institutions" that, in the place of formal legal interventions, use social norms to "define exclusivity and guide adjudication of rights violations." Even with these institutions in place, they serve more as a deterrent and are "ineffective regarding the sanctioning of those that transgress property rights norms."[3]

II. OBJECTIVE

The goal of this study was to establish a system that, when given a joke and a comedian, can determine the likelihood that the comedian was the originator of that joke.

Originality can be defined in two ways: first, as "literary originality or originality of content", and second as "performance originality or originality of style"[4]. For the purpose of this study, both definitions will be applied when compiling and analyzing the data.

A joke can be broken down into three essential parts, each typically occurring in a systematic way: setup, punchline, and taglines. The setup usually poses a question or observation that provides an opportunity for the punchline to be given in response. Following that is the punchline itself, which is the climactic conclusion to the setup that is designed to make the audience laugh. Finally trails the tagline(s); they are optional and are any laugh lines after the original punchline relating back to that first set up, even if other setups and punchlines have come since.

This study developed a baseline tool that can be used to detect joke theft. There are "several online tools to check whether someone is trying to take any undue credit by using plagiarized content," but no equally plausible methods for stand-up comedy[5]. In addition, it will provide a data collection mechanism for compiling works of a comedian and the likelihood that the comedian is the originator of those works.

The motivation for such a study is multi-layered: Can a tool be created that can determine the originality of stand-ups with accuracy? How prevalent is joke-poaching? And who are the bad actors?

III. METHODS

A. Data Collection

Before developing a web scraper, it was necessary to find the best source of dataset to collect. There are plenty if not too many comedy scripts online; however there was certain criteria in the goal when looking for the dataset. Three main criterias took priority when designing the dataset: availability, variety, and format. Availability is important because the desire was to collect as much datasets as possible to try many different methods for analysis. If some datasets were only available by API, there may have been issues with limitations of number of calls allowed per day (or certain period of time); therefore, ideally, datasets were available directly on the website. Variety is important because the final end goal was to compare jokes from many different scripts by both the same performer as well as different performers. Some websites only had less than five scripts of jokes available. Format was also important because it was key to being able to automate as much as possible when collecting datasets and thus avoiding a potential bottleneck. The initial setup to clean up the format of the source website is necessary; however, if there was a need to modify every script when collecting because the format of the original dataset is different, it would require too many modifications to the scraper to adjust for the various formats. Choosing the right source of dataset took some time but it was worth the time spent because the website "Scraps from the Loft" offered availability, variety, and almost unified format. Collecting the raw data was done by using python. Python provides many different options to scrape data from the web, with the options utilized in this project being the "Request" and "Beautiful Soup" libraries, making data collection simpler than expected. Request library provides all of the data that is in the website. Beautiful Soup eliminates unwanted parts from the website.

After the transcripts were collected by the web scraper, they needed to be cleaned. The first step was to identify characters that should be treated as regular characters and then remove all special characters using regular expressions.

Afterwards, stop words were identified in order to determine where the end of sentences occurred. In addition, through trial and error and observations of the available transcripts, a set of joke indicators were identified for categorization. Many indicators that were originally developed were later discovered to not necessarily indicate that a joke had been told. Examples of this include "laughs" and "audience cheers", which initially one would discern would indicate jokes. This process showed initially not only how widely transcripts could differ, but how precise the model would need to be at categorizing jokes.

Lastly, Beautiful Soup was utilized in order to get specific format from the raw dataset. Specifically, the back-end of the website is made with different types of “scripts”, which are designed to divide parts of the website. Most of the scripts were stored in the same “scripts” within the website so once those were located, Beautiful Soup provided a function that eliminated certain parts of script to scrape.

There are many other methods to collect and analyze dataset from websites but the reason “Request” and “Beautiful Soup” were chosen was because they were powerful and easy to implement. There were some other methods that could be used to scrape from the web such as Scrapy in python, but a learning curve was associated with that library and Beautiful Soup provided control over the HTML portions without much modification. Likewise, Beautiful Soup was better suited for the text based datasets that were available.

B. Initial Observations on Comedian Modeling and Testing

Different performers have different types of speaking habits and word choices. By analyzing collected and extracted data, the model reflects how each performer has different habits and uses different word choices. Ideally, the model can identify a joke that was used in the construction of the model as belonging to its respective comedian with high probability.

Once confidence in the model was established, it was then tested against jokes from routines known not belong to the comedian on which the model was based. The goal in this case was for the model to be able to have high certainty that the given joke was not written by the modeled comedian.

The ultimate goal then, once these methods had been proven to be successful, was to model a comedian who has a reputation for joke poaching. In creating such a model, the hope was to be able to determine what jokes, if any which are attributed to them, were in fact poached from another non-credited comic.

C. Data Analysis

Routines typically follow a specific flow where the setup occurs first, then the punchline, and then finally the tagline. The model we build will utilize those specifications to determine how many jokes occur in a stand-up act as well as what types of jokes are delivered and the manner in which they are executed; in order to do so though, the raw data must be analyzed.

By going over the initial raw data, the following components were extracted:

- Laughs in the script
- Words that are used by performer

Due to the fact that both audience and performer can both laugh during the performance and both of them can be written in the script(raw data), differentiating between laughs from audience and performer was necessary; only laughs from the audience can be used to determine the presence of a punchline or tagline.

D. Processing Program

The first model used only the lengths of the component parts of the jokes and the total length of the joke itself in order to determine if a joke belonged to a comedian’s joke repertoire; this metric alone proved insufficient, and a qualitative approach was adopted instead: n-grams. Each comedian’s collection of jokes was broken up into n-grams up to the value of n for which there was not more than one occurrence. Then each joke to be analyzed was compared to these n-grams and was awarded points for the number of n-grams contained within the joke, with the number of points awarded being equal to $2\hat{}(n)$. The total number of points was then divided by the number of words in the joke so as to account for high scores based on length of the joke alone. This approach worked significantly better than using the lengths of the joke components; however it was further improved by allowing the number of standard deviations under the average similarity score of a joke to be factored in.

E. Assumptions

A setup can have one or multiple lines, whereas punchlines and taglines are one-liners. All three elements always occur in the same order: setup, punchline, tagline. The punchline and the tagline always make the audience laugh, and differentiating between the punchline and tagline involves evaluating when the most recent setup occurred and wherein the transcript the laughter prompt followed.

F. Roles

- Walter Squires
  - Develop methodology and milestones that the project should accomplish.
  - Ensure that other team members are on track to accomplish milestones as described in the timeline
  - Write algorithm for cleaning transcripts
  - Write algorithm for comparing transcripts
- Kyungchan Lim
  - Develop scraper for transcripts
  - Implement conversion of routines into aspects of the comedian that can be modeled
- Autumn Henderson

- Draft proposal and convert into IEEE format.
- Design final presentation
- Draft final paper

G. Timelines

- October 9th: Scrap stand-up site for transcripts
- October 23rd: Break routines up into components
- November 6th: Build comedian model (test accuracy against itself)
- November 18th: Final Presentation
- November 20th: Test model against jokes known not to belong to that comedian
- November 27th: Build a model for questionable comedian
- December 9th: Final Paper Due

IV. RESULTS

A. Preliminary Results

Measurements of interest on the quantitative side are the number of words in the setup, punchline and taglines (if present), as well as the number of tag lines a joke has if taglines exist and the total number of words in a joke. This was the first part of the model to be implemented, and even when testing a joke that was part of the dataset the model was made with, the percent error for any component of the joke was upwards of 45% with the exception of the setup component, which only had a 10% error rate.

In testing the qualitative accuracy of this model, a percentage of the extracted jokes were used for training, and a smaller percentage for checking the accuracy of the model. When using 95% of the jokes for training, the model was, on average, able to correctly identify the remaining 5% of the jokes 74% of the time. Using between 90% and 85% of the jokes for training resulted in average accuracy around the 70% mark, with accuracy dipping down to 62% accurate when only 80% of the jokes for training. Table 1 summarizes their applicability to the results generated from the final iteration of the program.

B. Final Results

The accuracy of the model being tested against different random percentages of the training data can be seen in Table 1.

Amount of Random Training Data used in Test	n-grams Success Rate on Average	n-grams and STD Success Rate on Avg
5%	74%	94%
10%	71%	95%
15%	70%	95%
20%	62%	94%

TABLE I

TABLE 1 DEPICTS THE SUCCESS RATE OF USING N-GRAMS ON AVERAGE AND BOTH THE N-GRAMS AND STANDARD DEVIATION SUCCESS RATE ON AVERAGE WHEN DIFFERENT PERCENTAGES OF TRAINING DATA ARE UTILIZED.

Although the model was relatively successful in detecting jokes belonging to its own comedian, it was less effective at classifying jokes from other comedians as such. When testing the primary model against jokes that were known not to belong to them, the false attribution rate was no less than 21%, with the rate reaching nearly that of self identification in some cases; while including the standard deviation into the model's decision regarding if a joke belonged to the modeled comedian helped with self identification, it significantly increased the number of false positives as well, as evident in Table 2. It is possible that n-grams are not as effective for classifying someone's speech as originally thought, or that words in jokes may pull from a lexicon that is already fairly inbreed and there is a limit to how "original" a joke can be. On the other end of the spectrum, there are likely other ways to build these models which may be more effective than what was implemented.

Model Tested Against	n-grams False Positive Rate	n-grams and STD False Positive Rate
Iglesias	56%	89%
Papa	21%	80%
Regan	76%	94%
Chappelle	80%	95%

TABLE II

TABLE 2 DEPICTS THE POSITIVE RATE AND THE FALSE POSITIVE AND STANDARD DEVIATION RATE OF USING N-GRAMS TO MODEL OTHER ACTORS. FOR CHAPELLE, THE RATES REFLECT THE MODEL'S ABILITY TO MATCH THE ACTOR TO ITSELF.

V. LIMITATIONS & ISSUES

There were initial issues with ways some of the special characters were encoded in transcripts, namely ellipses. This slowed down the development of the data cleaner to a degree and raised the alarm that future complications with encodings could arise from other transcript formats.

Only using transcripts that had audience laugh descriptions was both a benefit and detriment; it allowed for easily-identifiable jokes, but greatly limited the number of comedians that could be modeled. Dave Chapelle has five transcripts making up his model, whereas all other comedians only had two transcripts that could be utilized. As a result, there was a limit in the ability to model other comedians, namely the potential bad actors that were the original targets.

VI. DISCUSSION

Data collection was successful but limited due to the wide variety of transcript formatting. This meant that there were less transcripts available than initially planned, further limiting the model's strength. The web scraper was successful in gathering the information needed, but further work would need to be accomplished in regards to data cleaning as well as categorization. This could be possible utilizing different machine learning techniques to train the model's ability to categorize without cues and even clean data with different characters and languages.

Utilizing n-grams in order to develop the model largely resulted in less-than-ideal metrics with regards to its ability to predict whether works belong to an actor. This is not to suggest that it is not possible to build an algorithm that can match better, but rather that a combination of more data and a different algorithm(s) could substantially aid in better key performance indicators. Addressing the algorithm's ability to categorize lines would be a giant step towards the former, giving way to more options for the latter.

VII. POSSIBLE IMPLICATIONS

A. Robin Williams and Others

Robin Williams has been targeted for being a notorious joke-stealer by other comedians. Ideally this model was to be able to identify jokes that he stole from other comedians, but it has not yet reached that level of accuracy. Once a model is built that can do this, an end-goal would be to utilize the model to determine, with a specific accuracy, the extent with which Robin Williams, and other popular thieves, stole jokes. This can have a career-impacting affect; actors could be subjected to the public opinion and be black-listed, lose money, or worse. Or, actors could potentially confess their sins and turn over a new leaf, potentially leading to creative, new content.

B. Legal Ramifications

As stated in the Introduction, lawsuits brought against bad actors for joke-poaching are complicated. This model has the potential to provide a systematic way to prosecute thieves. For example, say the model determines with statistically significant accuracy that a comedian replicated 85 percent of an originator's work, where a threshold of 80% is considered prosecutable; then litigation becomes simplified in a way unseen in history.

VIII. FUTURE RESEARCH

The final version of this model depends on transcripts that are structured in such a way that the program can easily parse them to determine where the setup, punchline, and taglines occur. Expectations were that cues existed that indicated what part of the joke the comedian was telling. In addition, many special characters were not cleaned well by Beautiful Soup, such as ellipses and music notes, and more straightforward documents would have expedited the sanitization process. Because transcripts are not formatted the same, there is potential to enhance the model's ability to process differently-formatted transcripts.

Ideally, either this algorithm is further developed or a machine-learning algorithm is created that can fully analyze any type of transcript and accurately categorize the lines. Being able to handle the line type either without the typical cue or the cue being in a different format would diversify the transcript portfolio. Furthermore, a more-advanced model could handle any special characters and a variety of encodings to maximize transcripts available. This may involve utilizing another library outside of Beautiful Soup or implementing algorithms that can reformat a document with ease.

These algorithms could be expanded even further to include multiple languages. The model currently handles standard English, but the possibility exists that a comedian steals a joke from another that did not tell it in English. Or even evade future joke detection by utilizing a different language while telling the joke. This may be more complicated but can be made less so by finding accommodating libraries that can handle translations. Developing the model on this level not only aids in capturing more poaching, but also opens up the opportunity for expanding databases on English-speaking comedians as well as those who speak a different language.

With regards to storing data collected, databases could be constructed utilizing the program that stores an originator and their transcripts as well as specific data about those transcripts, such as how many jokes it consisted of, how many lines of setups and taglines there were for stand-up acts, how much those works resemble other works, and other data as needed. Those databases can then have various other applications unrelated to this study.

IX. CONCLUSION

Joke theft has been prevalent for decades, but because of the nature in which one can modify a joke even the slightest and tell it in a different way, it is hard to pinpoint. There are no real legal ramifications if one steals a joke, like there are when novels are plagiarized, and while the court of public opinion can hold a comedian accountable, it is often hard to discern where the joke actually originated.

The model built aimed to be able to take written transcripts of comedians and compare those transcripts to others with the end goal of determining joke poaching with statistically significant results. This development was hindered by the variety of transcripts out there as well as the accessibility of them. For improvements on this model, it is suggested that more transcripts are collected to develop a more-robust machine-learning algorithm. In order to do that, the way in which the data is collected and cleaned will need several enhancements to tackle different formats of the transcripts.

The overarching goal of developing a baseline tool for detecting plagiarism was accomplished. Given more time and resources for sculpting the algorithm, the model will be on its way to be able to aid in the prosecution of a joke poacher, either by the court of opinion or court of law.

REFERENCES

[1] Library of Congress, "Bob Hope and American Variety: Bits 'I&' Sketches," Library of Congress. [Online]. Available: https://www.loc.gov/exhibits/bobhope/bits.html. [Accessed Sep. 26th, 2021].

[2] R. Zoglin, Comedy at the Edge: How Stand-up in the 1970s Changed America, Bloomsbury USA, 2003.

[3] P. Reilly, "The Weakness of Sanctioning in Norms-Based Property Systems: An Investigation of Joke Theft," Academy of Management Proceedings, vol. 2015, no. 1, November, 2017, [Online]. Available: Academy of Management, https://journals.aom.org/doi/abs/10.5465/ambpp.2015.15659abstract. [Accessed Sep. 26th, 2021]

[4] G. Pate, “Whose Joke Is It Anyway? Originality and Theft in the World of Standup Comedy,” Theatre Journal, vol. 66, no. 1, March, 2014, [Online]. Available: The Johns Hopkins University Press, https://www.press.jhu.edu/. [Accessed Sep. 25, 2021].

[5] RK Dewan ‘I&’ Co, “‘Joke Theft’: Not a joke,” Lexology. [Online]. Available: https://www.lexology.com/library/detail.aspx?g=5d3d7764-098f-45b1-83ef-f3d2e2e5abe0. [Accessed Sep. 26th, 2021]