The Automated Script Reviewer

Katherine Schinkel
University of Virginia
Data Science Institute
kms6bn@virginia.edu

Marcus Rosti
University of Virginia
Data Science Institute
mer3ef@virginia.edu

ABSTRACT

This paper discusses the application of text mining techniques and regression analysis to two 90s sitcoms as a means to predict the shows' viewership and rating. We collected ratings from IMDb and viewership from Wikipedia's reported Nielsen ratings. We use tfidf as predictors to our regression model.

Keywords

Text Mining; Regression Analysis; Machine Learning; Automated Systems

1. INTRODUCTION

TV shows cost an enormous amount of money. For instance just to film and develop a pilot, Priceecomics says it costs $2 million to develop a typically 30 minute comedy or over $5 million to make an hour long drama and even that is a low estimate.[1] That's a substantial investment into a project that may never be seen by the public. In the same article they say 98% of scripts fail to be made into pilots and of those, around half get a full season and an even smaller percentage make it past one season.

This creative and expensive process touches a number of people starting from pitching ideas for shows to actual filming the season. The step we focus on has to do with the script screening and development process. A production company will field many ideas and make a call for scripts based on those ideas. A producer must then screen through scripts and rate them only on their intuition or even some cognitive bias towards the writer. A producers time is valuable but on top of that the chosen script must maximize the chance of developing into a hit show.

Our research addresses automating the script review process or at least providing assistance to the reviewer via an automated system. We apply the text mining ideas of text preprocessing and feature engineering to predict the continuous values of IMDb rating and Nielsen viewership. To motivate those two values, a producer cares about two things when developing a show. On the less import end (monetarily wise), the show should be a high quality show, so the script needs to be well written and should receive a high IMDb value when it airs. But more importantly the show needs to be viewed by many millions people to drive advertising or generate more subscriptions in the case of HBO or Netflix. Thus again, our model needs to be able to predict with some certainty these values.

We used two 90s Sitcoms, both being widely viewed and rated, to at least validate that our method could accurately model that response. Given that these shows have proliferated that culture we expect to see at least come predictive power to our approach and conclude that there is a correlation between the script and the audience response.

[1]http://priceonomics.com/the-economics-of-a-hit-tv-show/

CS Text Mining Spring '16

DOI: 10.1145/1235

2. LITERATURE REVIEW

Researchers have explored sentiment analysis of text thoroughly. Bo Pang and Lillian Lee explored a multiclass approach to categorizing movie reviews as either positive, average and negative. They used support vector machines and a one versus all separation in order to classify them [6]. That research expanded a traditional binary classification into a multivalued problem.

In a separate field, Saxena explored applying a data mining techniques to screen resumes as part of a job application. In this case, the author explains that as the internet has become more popular, people are able to apply to many more jobs thus online job postings become flooded with resumes. To parse through these, they should apply several information retervial techniques but this paper focused on skills extraction, finding unique features and dealing with the specific setting of the resume. They had mixed results and concluced that more research would be required to find a more confident parser [8].

Lastly, Pivotal, a data sciences research company, attempted a script reviewer with success. They extracted information from scripts that included speaker and scene delegation, for which the did extensive feature engineering. For example, they were able to distill the personalities of speakers via topic modeling and that feature engineering and include this with their linear model [**?**]. However, the authors did not reveal more intimate details of their modeling and approach.

3. METHOD OVERVIEW

Our model follows a standard pipelined approach and employs 10 fold cross validation to justify the results.

3.1 Data Parsing

stemming tokenizing stopward removal

The two response variables we used were again the IMDb mean rating for the episode and the Nielsen rating for viewership rating. For the IMDb rating, we paired the episode with its corresponding rating. The rating of the episode should stand on its own since it could come days weeks or years after the actual episode airs. Our claim here is that the rating is independent of the timing of the episodes air. However, for the Nielsen rating, it would not make sense to pair the episodes viewership with the actual episode. Consider someone watching the show. That person has already come to the decision to watch the show independent of the script because it is unfolding before them. We, instead, map the next episodes viewership with that episode; moreover, Script_i corresponds to Viewership_{i+1}. We model it that way because if a person were to watch an episode, enjoy watching it because of the script and then be more likely to watch the next one. In this case, we capture the temporal aspect of the airing.

3.2 Regression Analysis

linear regression because it's response is continuously defined and because of it's simple nature

3.3 Model Validation

We followed a testing procedure based on 10 fold cross validation. In this procedure we compared our regression model, built on each training fold, with the mean of the response, again on the training fold. We use our model to predict the response on the test set and compare it to the mean of the response on the training set. Then using a two sample t test to make the claim that our model exceeds the performance of the mean.

$$t^* = \frac{\bar{Y}_1 - \bar{Y}_2}{\sqrt{s_1^2/N_1 + s_2^2/N_2}} \tag{1}$$

For our claim to be valid we would have to show that our model outperforms the mean at a significance level of $\alpha = .05$ and a t value of 1.812.

4. ANALYSIS

5. FUTURE WORK

6. THE *BODY* OF THE PAPER

Typically, the body of a paper is organized into a hierarchical structure, with numbered or unnumbered headings for sections, subsections, sub-subsections, and even smaller sections. The command `\section` that precedes this paragraph is part of such a hierarchy.[2] LaTeX handles the numbering and placement of these headings for you, when you use the appropriate heading commands around the titles of the headings. If you want a sub-subsection or smaller part to be unnumbered in your output, simply append an asterisk to the command name. Examples of both numbered and unnumbered headings will appear throughout the balance of this sample document.

Because the entire article is contained in the **document** environment, you can indicate the start of a new paragraph with a blank line in your input file; that is why this sentence forms a separate paragraph.

6.1 Type Changes and *Special* Characters

We have already seen several typeface changes in this sample. You can indicate italicized words or phrases in your text with the command `\textit`; emboldening with the command `\textbf` and typewriter-style (for instance, for computer code) with `\texttt`. But remember, you do not have to indicate typestyle changes when such changes are part of the *structural* elements of your article; for instance, the heading of this subsection will be in a sans serif[3] typeface, but that is handled by the document class file. Take care with the use of[4] the curly braces in typeface changes; they mark the beginning and end of the text that is to be in the different typeface.

You can use whatever symbols, accented characters, or non-English characters you need anywhere in your document; you can find a complete list of what is available in the *LaTeX User's Guide*[5].

6.2 Math Equations

You may want to display math equations in three distinct styles: inline, numbered or non-numbered display. Each of the three are discussed in the next sections.

6.2.1 Inline (In-text) Equations

A formula that appears in the running text is called an inline or in-text formula. It is produced by the **math** environment, which can be invoked with the usual `\begin. . .\end` construction or with the short form `$. . .$`. You can use any of the symbols and structures, from α to ω, available in LaTeX[5]; this section will simply show a few examples of in-text equations in context. Notice how this equation: $\lim_{n\to\infty} x = 0$, set here in in-line math style, looks slightly different when set in display style. (See next section).

6.2.2 Display Equations

A numbered display equation – one set off by vertical space from the text and centered horizontally – is produced by the **equation** environment. An unnumbered display equation is produced by the **displaymath** environment.

Again, in either environment, you can use any of the symbols and structures available in LaTeX; this section will just give a couple of examples of display equations in context. First, consider the equation, shown as an inline equation above:

$$\lim_{n\to\infty} x = 0 \tag{2}$$

Notice how it is formatted somewhat differently in the **displaymath** environment. Now, we'll enter an unnumbered

[2]This is the second footnote. It starts a series of three footnotes that add nothing informational, but just give an idea of how footnotes work and look. It is a wordy one, just so you see how a longish one plays out.

[3]A third footnote, here. Let's make this a rather short one to see how it looks.

[4]A fourth, and last, footnote.

Table 1: Frequency of Special Characters

Non-English or Math	Frequency	Comments
Ø	1 in 1,000	For Swedish names
π	1 in 5	Common in math
\$	4 in 5	Used in business
Ψ_1^2	1 in 40,000	Unexplained usage

equation:

$$\sum_{i=0}^{\infty} x + 1$$

and follow it with another numbered equation:

$$\sum_{i=0}^{\infty} x_i = \int_0^{\pi+2} f \tag{3}$$

just to demonstrate LaTeX's able handling of numbering.

6.3 Citations

Citations to articles [1, 3, 2, 4], conference proceedings [3] or books [7, 5] listed in the Bibliography section of your article will occur throughout the text of your article. You should use BibTeX to automatically produce this bibliography; you simply need to insert one of several citation commands with a key of the item cited in the proper location in the `.tex` file [5]. The key is a short reference you invent to uniquely identify each work; in this sample document, the key is the first author's surname and a word from the title. This identifying key is included with each item in the `.bib` file for your article.

The details of the construction of the `.bib` file are beyond the scope of this sample document, but more information can be found in the *Author's Guide*, and exhaustive details in the *LaTeX User's Guide*[5].

This article shows only the plainest form of the citation command, using `\cite`. This is what is stipulated in the SIGS style specifications. No other citation format is endorsed or supported.

6.4 Tables

Because tables cannot be split across pages, the best placement for them is typically the top of the page nearest their initial cite. To ensure this proper "floating" placement of tables, use the environment **table** to enclose the table's contents and the table caption. The contents of the table itself must go in the **tabular** environment, to be aligned properly in rows and columns, with the desired horizontal and vertical rules. Again, detailed instructions on **tabular** material is found in the *LaTeX User's Guide*.

Immediately following this sentence is the point at which Table 1 is included in the input file; compare the placement of the table here with the table in the printed dvi output of this document.

To set a wider table, which takes up the whole width of the page's live area, use the environment **table*** to enclose the table's contents and the table caption. As with a single-column table, this wide table will "float" to a location deemed more desirable. Immediately following this sentence is the point at which Table 2 is included in the input file; again, it is instructive to compare the placement of the table here with the table in the printed dvi output of this document.

Figure 1: A sample black and white graphic.

Figure 2: A sample black and white graphic that has been resized with the `includegraphics` command.

6.5 Figures

Like tables, figures cannot be split across pages; the best placement for them is typically the top or the bottom of the page nearest their initial cite. To ensure this proper "floating" placement of figures, use the environment **figure** to enclose the figure and its caption.

This sample document contains examples of **.eps** files to be displayable with LaTeX. If you work with pdfLaTeX, use files in the **.pdf** format. Note that most modern TeX system will convert **.eps** to **.pdf** for you on the fly. More details on each of these is found in the *Author's Guide*.

As was the case with tables, you may want a figure that spans two columns. To do this, and still to ensure proper "floating" placement of tables, use the environment **figure*** to enclose the figure and its caption. and don't forget to end the environment with figure*, not figure!

6.6 Theorem-like Constructs

Other common constructs that may occur in your article are the forms for logical constructs like theorems, axioms, corollaries and proofs. There are two forms, one produced by the command `\newtheorem` and the other by the command `\newdef`; perhaps the clearest and easiest way to distinguish them is to compare the two in the output of this sample document:

This uses the **theorem** environment, created by the `\newtheorem` command:

THEOREM 1. *Let f be continuous on $[a, b]$. If G is an antiderivative for f on $[a, b]$, then*

$$\int_a^b f(t)dt = G(b) - G(a).$$

The other uses the **definition** environment, created by

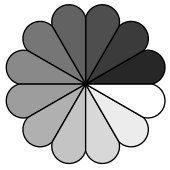

Figure 4: A sample black and white graphic that has been resized with the `includegraphics` command.

Table 2: Some Typical Commands

Command	A Number	Comments
`\alignauthor`	100	Author alignment
`\numberofauthors`	200	Author enumeration
`\table`	300	For tables
`\table*`	400	For wider tables

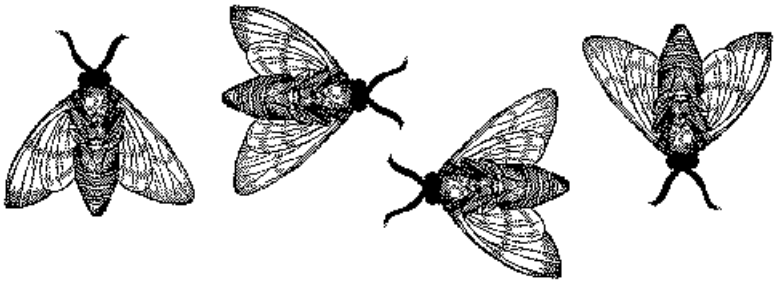

Figure 3: A sample black and white graphic that needs to span two columns of text.

the `\newdef` command:

Definition 1. If z is irrational, then by e^z we mean the unique number which has logarithm z:

$$\log e^z = z$$

Two lists of constructs that use one of these forms is given in the *Author's Guidelines*.

There is one other similar construct environment, which is already set up for you; i.e. you must *not* use a `\newdef` command to create it: the **proof** environment. Here is a example of its use:

Proof. Suppose on the contrary there exists a real number L such that

$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = L.$$

Then

$$l = \lim_{x\to c} f(x) = \lim_{x\to c} \left[gx \cdot \frac{f(x)}{g(x)} \right] = \lim_{x\to c} g(x) \cdot \lim_{x\to c} \frac{f(x)}{g(x)} = 0 \cdot L = 0,$$

which contradicts our assumption that $l \neq 0$. □

Complete rules about using these environments and using the two different creation commands are in the *Author's Guide*; please consult it for more detailed instructions. If you need to use another construct, not listed therein, which you want to have the same formatting as the Theorem or the Definition[7] shown above, use the `\newtheorem` or the `\newdef` command, respectively, to create it.

A *Caveat* for the TEX Expert

Because you have just been given permission to use the `\newdef` command to create a new form, you might think you can use TEX's `\def` to create a new command: *Please refrain from doing this!* Remember that your LATEX source code is primarily intended to create camera-ready copy, but may be converted to other forms – e.g. HTML. If you inadvertently omit some or all of the `\def`s recompilation will be, to say the least, problematic.

7. CONCLUSIONS

This paragraph will end the body of this sample document. Remember that you might still have Acknowledgments or Appendices; brief samples of these follow. There is still the Bibliography to deal with; and we will make a disclaimer about that here: with the exception of the reference to the LATEX book, the citations in this paper are to articles which have nothing to do with the present subject and are used as examples only.

8. ACKNOWLEDGMENTS

This section is optional; it is a location for you to acknowledge grants, funding, editing assistance and what have you. In the present case, for example, the authors would like to thank Gerald Murray of ACM for his help in codifying this *Author's Guide* and the **.cls** and **.tex** files that it describes.

9. REFERENCES

[1] M. Bowman, S. K. Debray, and L. L. Peterson. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.*, 15(5):795–825, November 1993.

[2] J. Braams. Babel, a multilingual style-option system for use with latex's standard document styles. *TUGboat*, 12(2):291–301, June 1991.

[3] M. Clark. Post congress tristesse. In *TeX90 Conference Proceedings*, pages 84–89. TeX Users Group, March 1991.

[4] M. Herlihy. A methodology for implementing highly concurrent data objects. *ACM Trans. Program. Lang. Syst.*, 15(5):745–770, November 1993.

[5] L. Lamport. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.

[6] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 115–124, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[7] S. Salas and E. Hille. *Calculus: One and Several Variable*. John Wiley and Sons, New York, 1978.
[8] C. Saxena. Enhancing productivity of recruitment process using data mining and text mining tools. 2011.

APPENDIX

A. HEADINGS IN APPENDICES

The rules about hierarchical headings discussed above for the body of the article are different in the appendices. In the **appendix** environment, the command **section** is used to indicate the start of each Appendix, with alphabetic order designation (i.e. the first is A, the second B, etc.) and a title (if you include one). So, if you need hierarchical structure *within* an Appendix, start with **subsection** as the highest level. Here is an outline of the body of this document in Appendix-appropriate form:

A.1 Introduction

A.2 The Body of the Paper

A.2.1 Type Changes and Special Characters

A.2.2 Math Equations

Inline (In-text) Equations.

Display Equations.

A.2.3 Citations

A.2.4 Tables

A.2.5 Figures

A.2.6 Theorem-like Constructs

A Caveat for the TEX Expert

A.3 Conclusions

A.4 Acknowledgments

A.5 Additional Authors

This section is inserted by LaTeX; you do not insert it. You just add the names and information in the `\addition-alauthors` command at the start of the document.

A.6 References

Generated by bibtex from your .bib file. Run latex, then bibtex, then latex twice (to resolve references) to create the .bbl file. Insert that .bbl file into the .tex source file and comment out the command `\thebibliography`.

B. MORE HELP FOR THE HARDY

The sig-alternate.cls file itself is chock-full of succinct and helpful comments. If you consider yourself a moderately experienced to expert user of LaTeX, you may find reading it useful but please remember not to change it.