Screenplay Quality Assessment: Can we predict who wins the Award?

Ming-Chang Chiu
mingchac@usc.edu

Abstract

Deciding which scripts to turn into movies is a costly and time-consuming process for filmmakers. Thus, script selection as the initial phase in movie production, building a tool to aid the process can be very beneficial. In this work, we present a method to evaluate the quality of a screenplay solely based on linguistic cues. We address this in a two-fold approach: (1) we define the task as predicting nominations of scripts at major film awards since we hypothesize that the peer-recognized scripts should have a greater chance to succeed. (2) based on industry opinions and narratology, we extract and integrate domain-specific features into common classification techniques. We face two problems (1) scripts are way longer than other document datasets (2) nominated scripts are limited and thus hard to collect. However, with narratology-inspired modeling and domain features, our approach sees clear improvements over strong baselines. Our work provides a new approach for future work in screenplay analysis.

1 Introduction

The motion picture industry is a multi-billion dollar business worldwide (Lash and Zhao, 2016). Decisions in selecting movies to be produced are critical to the profitability of a movie studio. However, the selection of the screenplay, happening at the initial phase, which has a large influence on the financial budget and quality of the final movie production, has a large subjective element.

A typical script review service costs a studio \$80 to \$150 to receive a report containing a short summary of the script and opinion as to its quality (Follows et al., 2019). Considering the amount of scripts a studio needs to filter through, it can be overwhelming.

Thus, Consider this scenario, if a tool can facilate the script review process and provide the chance of success, wouldn't this make an impact and cut down lots of budgeting decisions in the production process? An objective and reliable tool to help evaluate and narrow down the candidate scripts is of vital importance to aid the "green-lighting" (deciding which scripts to turn into movies) process. This work is not about measuring the art of rating how a good story is; it is about deciphering critical readers. We want to provide a tool that says "this is what picky reviewers may look for in a script".

The main idea of this work is to develop such a tool which gather custom analyses from various aspects, e.g., screenplay writing theory, character-focused linguistic behavior, to help assess the quality of the script.

In general, movie script writing can follow a well-defined *Three Act* structure (Field, 2007; McKee, 1997). Also, Weiland (Weiland, 2013, 2018) specifies a more fine-grained storytelling plan, starting from *hook, inciting event, 1st plot point, 1st pinch point, midpoint, 2nd pinch point, 3rd plot point, climax* to *resolution*, where we call Structural Points (SP). We believe knowledge like the above in strucuring a screenplay can bring benefits in selecting the most relevant textual properties for the prediction of script quality.

Aside from the event positioning, Follows et al. (2019) reported that how writers develop characters and events, i.e., *Characterization* and *Plot*, are two main foci of industry reviewers. We thus devise our domain specific features in these two aspects. We hope to offer an enhanced understanding of the essential elements in high-quality movie scripts.

To perform quality assessment, based on an assumption that the nominated scripts are recognized writings and thus should have had higher chance of passing green-lighting, we propose to perform an evaluation in a two-fold approach. First, we use award-nomination prediction as a proxy to the green-lighting process. Second, we examine our domain features and models by integrating them into existing document classification methods.

We admit the constraints of our metric in that the number of award venues has its limits, and not necessarily those without nomination would be any worse than the nominated. But due to the difficulty in collecting unproduced scripts with peer reviews, we adopt our current approach.

Our main contributions are:

- We defined a quality metric for screenplays and collected ground truths from peer-reviewed venues.
- Based on structural knowledge of screenplay narratology, we developed a simple narratology-inspired model for our task.
- Motivated by industry opinions and narratology, we devised domain-specific features to achieve our objective.
- We tested that for long document classification, a simple feature-based approach can work better than state-of-the-art models.

2 Related Works

Literary works-related research has gained interest in recent years. Bamman et al. (2013, 2014) has succeeded to learn latent character types in film and novels; Iyyer et al. (2016); Chaturvedi et al. (2016); Elson et al. (2010) try to model character relations in novels. Papalampidi et al. (2019) analyze narrative structure of movies by using turning points, and Chambers and Jurafsky (2008); Sims et al. (2019) seek to detect events in narratives.

Also, there has been Gorinski and Lapata (2015) trying to summarize movie script with graph-based approach and automatically generate movie precis containing story, genre, and others (Gorinski and Lapata, 2018).

The only attempt in measuring quality of literary works we know of is made by Kao and Jurafsky (2012), who quantitatively analyze various indicators for discerning professional poems from amateurs'. However, in script writing, the cinematic success criteria lack evaluative consensus (Simonton, 2009) — previous works on evaluation of movies have largely focused on forecasting revenue or profit of movies using production, distribution, and advertising data (Ghiassi et al., 2015; Lash et al., 2015) or basic textual and human annotated features (Eliashberg et al., 2014).

The main differences between our work and previous works are: (1) our approach aims to process automatically without human annotated features. (2) our metric and method are geared towards evaluation that base soley on textual properties.

3 Data and Problem Setting

In this section, we introduce datasets we evaluate on, how we collect ground truths for out task, and the explain the difficulties we face.

3.1 Data collection

We evaluated our method using ScriptBase (Gorinski and Lapata, 2018) and Movie Screenplay Corpus (MSC) Ramakrishna et al. (2017) datasets. ScriptBase provides 917 scripts and MSC contains 945 Hollywood movies. We kept 897 and 868 suitable ones which have enough character utterances for our approach from each dataset respectively. Similar to Underwood (2019), which analyze high-prestige novels as works that have been reviewed by top journals, we collected the screenplays that have histories of nominations as quality "ground truth". The venues we collect from are well-known professional prizes, which include "Writers Guild of America Award", "Academy Awards", "Golden Globe Awards", and "British Academy of Film and Television Arts Awards". Since we focus on textual properties for success, we only gleaned nominations in the original screenplay and adapted screenplay categories. At the end, we obtained 212 (23.6%) movies out of ScriptBase and 113 (13.0%) from MSC as quality "ground truth" labels.

3.2 Problem Setup

Our work focuses on measuring quality as whether or not a movie would be nominated at a peer-reviewed venue. The basic assumption for using this approach as success metrics is simple — a screenplay that receives nominations by critical reviewers should have had higher chance of getting through green-lighting.

3.3 Challenges

In nature, a movie should be tough to be cleanly categorized, due to its length, complex storyline and turns, and the lack of evaluative criteria. Prior works in document classification (Yang et al., 2016; Liu et al., 2017; Adhikari et al., 2019; Johnson and Zhang, 2015) evaluated on datasets with small document size (Reuters, IMDB, Yelp, etc.). However,

Dataset	documents	average #w	%pos
Reuters	10,789	144.3	-
IMDB	135,669	393.8	-
Yelp 2014	1,125,457	148.8	-
ScriptBase	897	27,539.7	23.6
MSC	868	27,067.4	13.0

Table 1: **Dataset statistics and comparisons of datasets.** #w denotes the number of words and %pos denotes the percentage of positive class.

our document size on average is at least 65 times longer, which may be hard for NN-based models to train due to long sequences and the computational burden. Besides, the number of training data we have is at most 1000 times smaller than other datasets. With our datasets being **long**, **small** and **skewed**, state-of-the-art techniques may not work well. Summary of the comparisons is shown in Table 1.

4 Analysis of Domain Features

In this section, we introduce our domain features that are divised to achieve our goal and provide analysis based on our problem setup.

Characterization and *Plot* are major aspects of focus in the industry; inspired by which, we devised 6 novel features. For each, we provided intuitive motivations, and then detailed how we converted them computationally. We chose the top two most speaking characters of each movie to analyze for *characterization*.

According to Weiland (2018), a script can place 9 SPs roughly equally distributed, creating eight equal-lengthed development segments (DS) in between. We hypothesized that such structural hints should help to achieve our objective. Based on the statistics of both datasets, to leverage the SPs, we collected a context window of 1% (∼270 words) centered at SPs for all scripts.

4.1 Characterization

By the definition of *characterization*, we hypothesized that by measuring pattern change of characters, we may see how writers develop the characters' personality. We sought pattern change via two kinds of changes writers would make between SPs - linguistic (speaking pattern) change and emotional change. To do this, we proposed *Linguistic & Emotional Activity Curve*.

Linguistic & Emotional Activity Curve (*ling, emo*). For linguistic change, we extracted the dependency trees of characters; for emotional change we used normalized Empath (Fast et al., 2016) to get characters' emotion status. We extracted the linguistic distribution and Empath distribution of sentences in each development segment. We then applied *activity curve* (Dawadi et al., 2016), which uses a Permutation-based Change Detection in Activity Routine (PCAR) algorithm to calculate the change between two DSs of distributions.

Type-token ratio (*tt*). As Kao and Jurafsky (2012) show, in poetry, the *type-token ratio* related most positively to the quality of a poem. We believed this concept should work similarly on character analysis, and can show how much effort writers put in in characterization. We defined this feature as the number of unique words used by a character divided by the total number of words.

4.2 Plot

Moreover, we supposed a series of well-written dramatic events, i.e., *Plot*, should have emotional effect to readers and thus writers may use their lexica to achieve that. Therefore, we examined this hypothesis by leveraging two sentiment analysis lexicons to compute the emotional strength of SPs.

Valence-Arousal-Dominance (*VAD*). Mohammad (2018a) performed extensive study in getting an objective score for words in VAD dimensional space (Russell, 1980, 2003). We calculated average scores over the context window of each SP.

Emotion Intensity (*int*). Similar to *VAD*, we used the NRC Affect Intensity Lexicon (Mohammad, 2018b) over the SPs to score emotion intensity along four basic emotion classes (Plutchik, 1980).

Also, since events are usually adressed in units of scenes, we wanted to get a picture of how many different emotionally similar scenes across the dataset appear in a movie.

Empath Clustering (*clus*). We used Empath to extract lexical categories for each uttrance. We then clustered the lexical category distributions of all utterances with deep embedded clustering (Xie et al., 2016). We obtained the cluster distribution based on the lexical categories within a movie as a feature representation.

We visualized partial features in a "nomination v non-nomination" fashion to show the potential of our features. For some we can easily observe clear differences from one to the other, while some are more subtle. For instance, in *VAD*, *arousal* of MICA is ambiguous between the two, and yet we can easily discern nominated scripts along the same axis for ScirptBase.

5 Details of Activity Curve

In Sec. 4, we briefly introduce the concept of *Activity Curve*, but since later shown in Sec.8, this contributes consistent improvements to our task, we detail the procedure of getting this feature.

5.1 Definition

Linguistic interval represents a fixed portion of utterances. Particularly, we define a linguistic interval $\mathbf{W_{x,y}}$ where x and y are the start and end proportion in a character's utterance set. For instance, $\mathbf{W_{0,10}}$ indicates the first 10% utterances of a character. In this study, we segment a character's full dialog into equal-size windows, and each window is regarded as a time interval. Particularly, each of these segments has of 20% utterances of a character, and consecutive liguistic intervals are with an overlap of 10%.

Linguistic distribution is used to model linguistic information within a linguistic interval. As mentioned above, two linguistic distributions are made available in this analysis: utterance emotion disctribution and lexical disctribution. Distribution of seven basic emotions of each utterance is computed by using the Empath library (Fast et al., 2016). These seven basic emotions are *anger, sadness, joy, surprise, love, fear, and disgust*. The lexical distribution of each utterance is calculated from the lexical labels provided from (Gorinski and Lapata, 2018). We use the utterance emotion disctribution as the example in the following explanation. For each linguistic interval, we average the utterance emotion distribution from all utterances to generate the utterance emotion distribution for the linguistic interval. We define $\mathbf{D}_j = \{d_{j,1}, d_{j,2}, ..., d_{j,e}, ...d_{j,7}\}$ as the utterance emotion distribution for the j-th time interval for a character given the unique emotion set described above.

5.2 Distance Measure Between Linguistic Distributions

To calculate the distance between two linguistic distributions, we used the Kullback-Leibler (KL) divergence measure. For instance, we have two text emotion distributions at the j-th linguistic interval and at the $(j+1)$-th linguistic interval: $\mathbf{D}_j = \{d_{1,j,1}, d_{j,2}, ..., d_{j,e}, ...d_{j,7}\}$ and $\mathbf{D}_{j+1} = \{d_{j+1,1}, d_{j+1,2}, ..., d_{j+1,e}, ..., d_{j+1,7}\}$. The KL divergence between these two distributions is defined as follows:

$\text{Dist}_j = \frac{1}{2} \times (D_{KL}(\mathbf{D_j}||\mathbf{D_{j+1}}) + D_{KL}(\mathbf{D_{j+1}}||\mathbf{D_j}))$
$= \frac{1}{2} \times \sum_{k=1}^{7}(d_{j,k}\frac{d_{j,k}}{d_{j+1,k}} + d_{j+1,k}\frac{d_{j+1,k}}{d_{j,k}})$

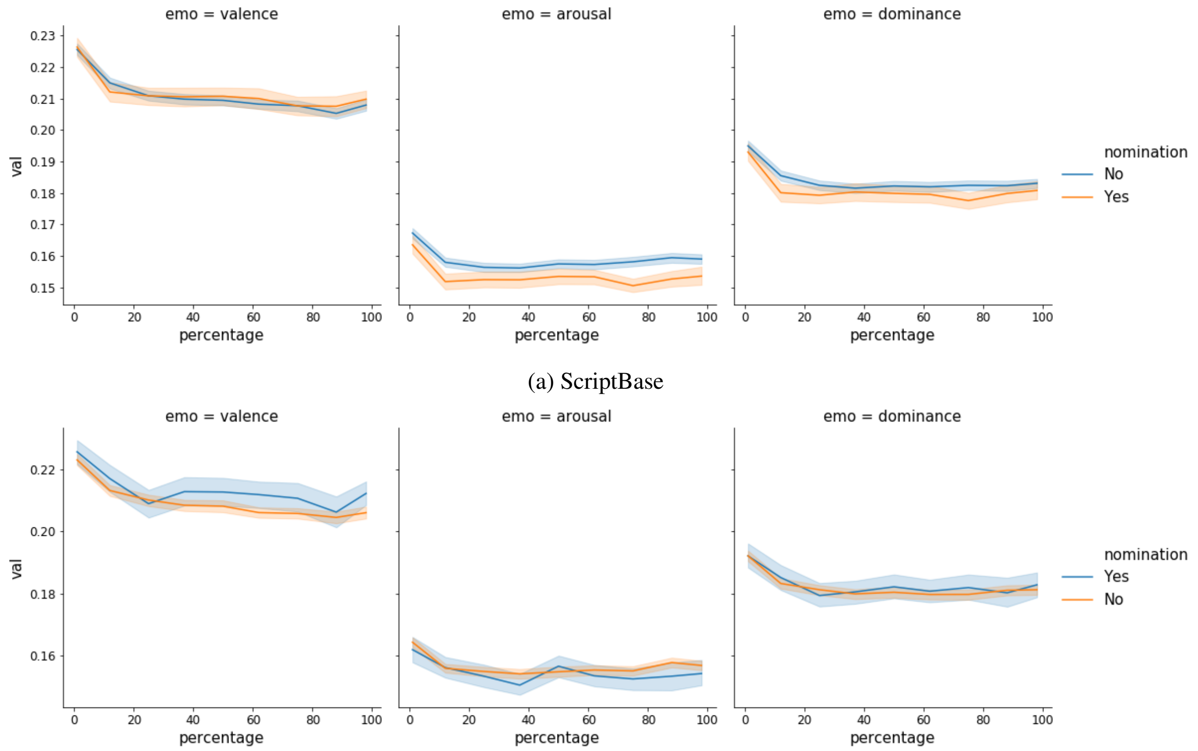
5.3 Consistency Model

We apply a Permutation-based Change Detection in Activity Routine (PCAR) algorithm proposed by (Dawadi et al., 2016) to calculate the change between two linguistic intervals. We apply PCAR to detect changes in linguistic based on a two-sample permutation test. The permutation-based technique first provides a data-driven approach to calculate an empirical distribution of a test statistic. The empirical distribution of a test statistic is then obtained by calculating the test statistic after randomly shuffling (rearranging) the data a specified number of times. For instance, to compare the change between $\mathbf{W_{0,20}}$ and $\mathbf{W_{10,30}}$, we have the following steps:

1. **Calculate the baseline statistic**: We calculate the baseline statistic as the KL divergence between two aggregated linguistic distributions in the original window $\mathbf{W_{0,20}}$ and $\mathbf{W_{10,30}}$ without permutation. We denote baseline statistic between j-th and $(j+1)$-th linguistic interval as $\overline{Dist_j}$.

2. **Calculate the empirical test statistic distribution**: First, we randomly permute recordings between $\mathbf{W_{0,20}}$ and $\mathbf{W_{10,30}}$. Then, we calculate the empirical distributions of the test statistic (KL divergence) by comparing the individual linguistic distribution within the two shuffled windows. We perform the shuffle for S times, and we define the test statistic generated from each shuffle between j-th and $(j+1)$-th linguistic interval as $Dist_{j,s}$.

3. **Significance testing**: To quantify the change of the linguistic information between two intervals, we quantify the significance in changing based on the number of times the test statistic from the permuted sample is equal to or greater than the baseline statistic. In other words, the change significance between j-th and $(j+1)$-th linguistic interval is computed as $sig_j = \sum_{s=1}^{S} 1(\overline{Dist_j} >= Dist_{j,s})/S$.



(a) ScriptBase

(b) MSC

Figure 1: **VAD.**

6 Predictive Modeling

In this section, we define our prediction task, and then propose our base model as and then move on to a paradigm which integrates domain features proposed in previous section. Also, with a similar background as our base model, we developed a simple end-to-end NN-based model.

6.1 Task Formulation

As a proxy to the original quality assessment task, we define a binary classification task as to predicting the nomination of a script.

6.2 Narratology-inspired Model.

Inspired by narratology and Manevitz and Yousef (2001), we propose *Tfidf-SVM*$_{narr}$ (Fig.2)— instead of using all texts in an entire document, we extract words in context window of SPs for each document, compute the tf-idf representations, and feed them into a SVM classifier. Due to the huge amount of unique tokens, we chose only the top 500 important features to represent a document. We test the results without choosing 500 features and our setting is better.

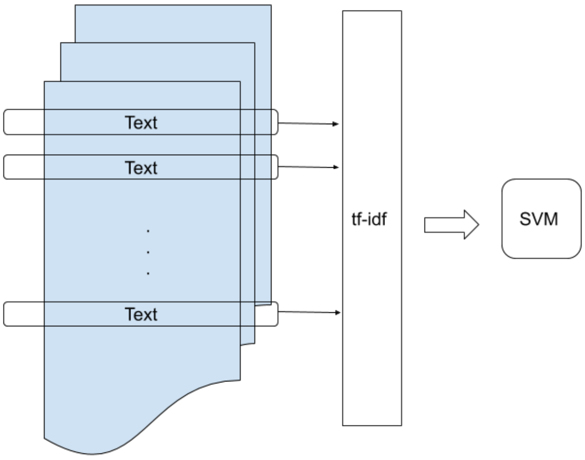


Figure 2: **Narratology-inspired Model.**

6.3 Deep Neural Prediction Model

Similar to *Tfidf-SVM*$_{narr}$, for each SP, we utilize a Bi-GRU over the corresponding context window, creating a hidden representation for that SP and concatenate all 9 hidden representations together. After that we use a fully connected layer for prediction. We call this *BiGRU*$_{narr}$. The idea is shown in Fig.3.

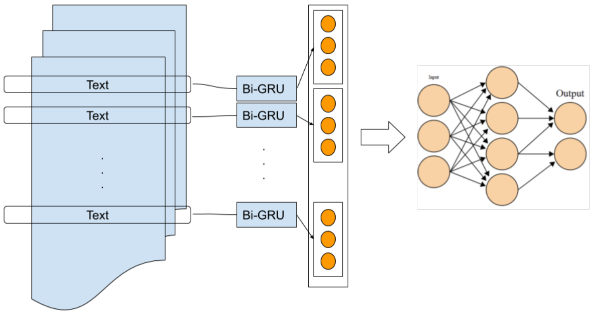


Figure 3: **Deep Neural Prediction Model.**

6.4 Feature-based Prediction

To examine the predictive power of proposed features, on top of *Tfidf-SVM*$_{narr}$, we add domain features along with tf-idf to SVM; for *BiGRU*$_{narr}$, we concatenate domain features with the 9 hidden representations before fully connected layer to see the efficacy of domain features.

7 Experimental Setups

7.1 Dataset usage

We performed random sampling on both datasets such that 80% is used for training, 10% for validation, and 10% for test.

7.2 Baselines

We adopted HAN (Yang et al., 2016), BERT$_{base}$, BERT$_{large}$ (Devlin et al., 2019) as our baselines. Since a script is subdivided into scenes, our HAN implementation, HAN$_{scence}$, uses scence as the second hierarchy instead of sentence.

7.3 Implementation details

We use Scikit-learn 0.21.3 to implement feature-based models, and PyTorch 1.3.1 for deep neural models. Also, we use gensim 3.8.1 for pre-trained Word2Vec embeddings for HAN, and Hugginface (Wolf et al., 2019) for BERT. Since the binary labels in both datasets are imbalanced, we weight the positive class by inverse frequency of class labels in the training set.

7.4 Hyper-parameters

To ensure a fair comparison, we tuned the hyper-parameters for all models. On feature-based models, we performed grid search. For NN models, we use embedding size 100 and Adam optimizer with 0.001 learning rate.

Method / Dataset	**ScriptBase**	**MSC**
HAN$_{scene}$	45.12	45.62
BERT$_{base}$	42.67	46.29
BERT$_{large}$	42.67	46.29
Tfidf-SVM	47.01	59.21
TFIDF-SVM$_{narr}$	**57.43**	**59.21**
+ emo + VAD	56.52	55.29
+ ling + emo + tt	**62.35**	62.73
+ int + ling + emo + clus	60.87	**64.79**
BiGRU$_{narr}$	52.92	42.40
+ ling + emo + clus + tt	57.47	54.46

Table 2: **F1 scores (%) of model predictions.** See Sec 4 for definitions of abbreviations.

8 Results and Discussion

We report the macro-averaged F1 scores of each model in Table 2.

8.1 NN v SVM

Interestingly, from Table 2 we see that NN-based document classification methods are no better than our proposed simple SVM narratology-based model. We suppose the length of document could be the main reason, RNNs may not handle "super long-term depdendencies" well for complex compositions like movie scripts.

8.2 Individual Feature

As to the effect of each individual feature, *Linguistic & Emotional Activity Curve* show improvements on both datasets, and yet the rest do not consistently help, especially on MSC, we think it may be because (1) the tfidf has 500 dimensions so individual feature may be overwhelmed, but, more features combined such as adding *Emotion Intensity, Linguistic Activity Curve* and *Type-token ratio* can generate consistent improvements, (2) the efficacy of feature can be dataset-dependent, e.g., we do not observe significant differences in *Arousal* of MSC as in its ScriptBase counterpart (Fig. 1), and so does the classifier.

8.3 Combination of Features

Adding different combination of features could add predictive power. But on different dataset, the efficacy varies. Also, features with negative correlations (Fig. 6) can damage the performance, e.g., adding *Emotional Activity Curve & VAD*.

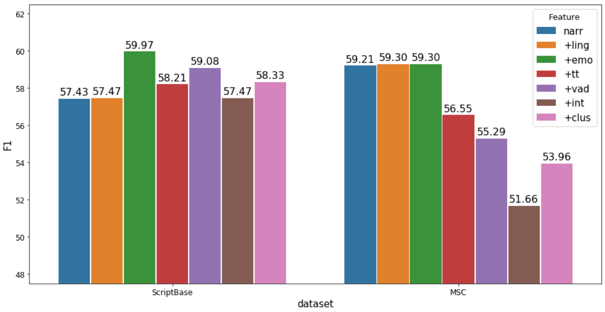


Figure 4: **Visualization of individual feature effect for *TFIDF-SVM*$_{narr}$** See Sec 4 for definitions of abbreviations.

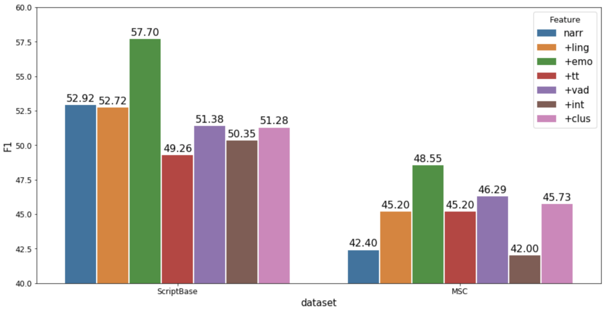


Figure 5: **Visualization of individual feature effect for *BiGRU*$_{narr}$.** See Sec 4 for definitions of abbreviations.

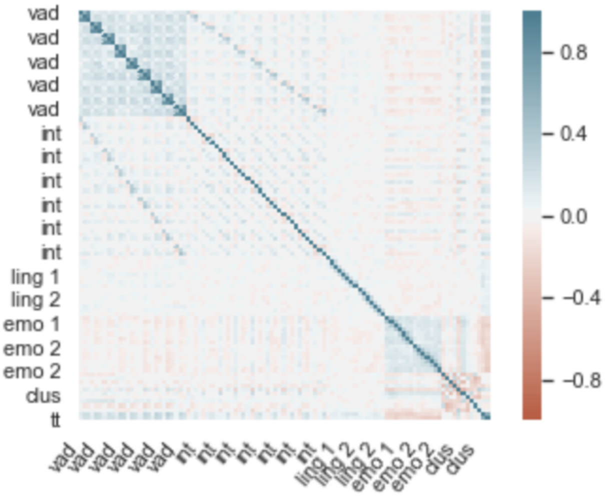


Figure 6: **Feature correlation of ScriptBase.** See Sec 4 for definitions of abbreviations.

9 Conclusion

We present a novel approach and features to analyze the quality of a screenplay in terms of its festival nomination-worthiness. This can serve as a preliminary tool to help filmmakers in their decision-making. Our results also show that simple lightweight approach can outperform state-of-the-art document classification methods. This also points out the current deficiency for long document classification research in the community.

For future work, it would be interesting to develop a more fine-grained approach by first decerning what structure the script use among commonly used ones (Miyamoto, 2018), and then go on to further modelling analysis based on our current approach.

References

Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy Lin. 2019. Rethinking complex neural network architectures for document classification. In *NAACL-HLT*.

David Bamman, Brendan O'Connor, and Noah A. Smith. 2013. Learning latent personas of film characters. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 352–361, Sofia, Bulgaria. Association for Computational Linguistics.

David Bamman, Ted Underwood, and Noah A. Smith. 2014. A Bayesian mixed effects model of literary character. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 370–379, Baltimore, Maryland. Association for Computational Linguistics.

Nathanael Chambers and Dan Jurafsky. 2008. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio. Association for Computational Linguistics.

Snigdha Chaturvedi, Shashank Srivastava, Hal Daumé III, and Chris Dyer. 2016. Modeling evolving relationships between characters in literary novels. In *AAAI*, pages 2704–2710.

Prafulla N Dawadi, Diane J Cook, and Maureen Schmitter-Edgecombe. 2016. Modeling patterns of activities using activity curves. *Pervasive and mobile computing*, 28:51–68.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*.

J. Eliashberg, S. K. Hui, and Z. John Zhang. 2014. Assessing box office performance using movie scripts: A kernel-based approach. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2639–2648.

David Elson, Nicholas Dames, and Kathleen McKeown. 2010. Extracting social networks from literary fiction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 138–147, Uppsala, Sweden. Association for Computational Linguistics.

Ethan Fast, Bin Bin Chen, and Michael S. Bernstein. 2016. Empath: Understanding topic signals in large-scale text. In *CHI*.

Syd Field. 2007. *Screenplay: The foundations of screenwriting*. Random House LLC.

Stephen Follows, Josh Cockcroft, and Liora Michlin. 2019. Judging screenplays by their coverage: An analysis of 12,000+ unproduced feature film screenplays and the scores they received, revealing what professional script readers think makes a good screenplay.

M. Ghiassi, David Lio, and Brian Moon. 2015. Pre-production forecasting of movie revenues with a dynamic artificial neural network. *Expert Systems with Applications*, 42(6):3176 – 3193.

Philip John Gorinski and Mirella Lapata. 2015. Movie script summarization as graph-based scene extraction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1066–1076, Denver, Colorado. Association for Computational Linguistics.

Philip John Gorinski and Mirella Lapata. 2018. What's this movie about? a joint neural network architecture for movie content analysis. In *NAACL-HLT*.

Mohit Iyyer, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III. 2016. Feuding families and former Friends: Unsupervised learning for dynamic fictional relationships. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1534–1544, San Diego, California. Association for Computational Linguistics.

Rie Johnson and Tong Zhang. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 103–112, Denver, Colorado. Association for Computational Linguistics.

Justine Kao and Dan Jurafsky. 2012. A computational analysis of style, affect, and imagery in contemporary poetry. In *Proceedings of the NAACL-HLT 2012 Workshop on Computational Linguistics for Literature*, pages 8–17, Montréal, Canada. Association for Computational Linguistics.

Michael T. Lash, Sunyang Fu, Shiyao Wang, and Kang Zhao. 2015. Early prediction of movie success - what, who, and when. In *SBP*.

Michael T. Lash and Kang Zhao. 2016. Early predictions of movie success: The who, what, and when of profitability. *Journal of Management Information Systems*, 33(3):874–903.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *SIGIR*.

Larry Manevitz and Malik Yousef. 2001. One-class svms for document classification. *Journal of Machine Learning Research*, 2:139–154.

Robert McKee. 1997. *Substance, Structure, Style, and the Principles of Screenwriting*. New York: HarperCollins.

Ken Miyamoto. 2018. The best screenwriting structures you can apply to your script.

Saif M. Mohammad. 2018a. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Saif M. Mohammad. 2018b. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

Pinelopi Papalampidi, Frank Keller, and Mirella Lapata. 2019. Movie plot analysis via turning point identification. *ArXiv*, abs/1908.10328.

Robert Plutchik. 1980. Chapter 1 - a general psychoevolutionary theory of emotion. In Robert Plutchik and Henry Kellerman, editors, *Theories of Emotion*, pages 3 – 33. Academic Press.

Anil Ramakrishna, Victor R. Martinez, Nikos Malandrakis, Karan Singla, and Shrikanth Narayanan. 2017. Linguistic analysis of differences in portrayal of movie characters. In *ACL*.

James A. Russell. 1980. A circumplex model of affect.

James A Russell. 2003. Core affect and the psychological construction of emotion. *Psychological review*, 110 1:145–72.

Dean Keith Simonton. 2009. Cinematic success, aesthetics, and economics: An exploratory recursive model.

Matthew Sims, Jong Ho Park, and David Bamman. 2019. Literary event detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3623–3634, Florence, Italy. Association for Computational Linguistics.

T. Underwood. 2019. *Distant Horizons: Digital Evidence and Literary Change*. University of Chicago Press.

K.M. Weiland. 2013. *Structuring Your Novel: Essential Keys for Writing an Outstanding Story*. PenForASword.

K.M. Weiland. 2018. Story structure qa: 6 outstanding questions about structure.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Junyuan Xie, Ross Girshick, and Ali Farhadi. 2016. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *HLT-NAACL*.