Decoding the Force: Algorithms for Author Attribution in Star Wars

Aaron Jin aaronjin@stanford.edu
Collin Jung collinj2@stanford.edu
Akshath Mahajan amahaj@stanford.edu
Aryan Sahai aryan22@stanford.edu

Abstract

In this paper, we harness the power of DistilBERT, a machine learning model, to identify the scriptwriters behind dialogue lines in the Star Wars saga (Episodes IV to VI). This research, grounded in the broader context of plagiarism detection and authorship verification in digital texts, employs a novel application of natural language processing to cinematic scripts. The study meticulously compares the effectiveness of cross-entropy and hinge loss functions in the context of a multilabel classification problem, revealing the nuances of each approach's impact on the model's ability to accurately attribute authorship.

Through extensive experimentation, our findings demonstrate the superior performance of the hinge loss function in distinguishing between the writing styles of different characters. This insight, alongside a critical evaluation of the baseline model's limitations and the computational challenges encountered, underscores the potential of fine-tuned machine learning models in the field of stylometry. The implications of our work suggest promising directions for future research, including the exploration of larger and more varied datasets, and the application of these methods beyond the realm of fiction to historical document analysis.

1 Introduction

The Star Wars movies are known for their fascinating plots and diverse characters, serving as an ideal subject for exploring authorship attribution in screenwriting using advanced computational methods. In this study, "Decoding the Force: Algorithms for Author Attribution in Star Wars," we aim to identify the scriptwriters of specific dialogue lines from the Start War movies IV to VI using machine learning techniques.

Our motivation for this project is due to the larger issue of authorship attribution, particularly in the context of plagiarism. This represents a significant challenge as most text is available/produced digitally. In academia, plagiarism is a widespread issue. According to a study conducted by the International Center of Academic Integrity across 24 schools in the US over 12 years, a survey found that "95% among surveyed students cheated on tests or homework, or plagiarized at least once" [3]. In fact, according to a study by Fixgerald, 12.2% of people admitted to plagiarizing another author's work in their own paper or paraphrased ideas without reference [2]. *Figure 1* below depicts more such data.

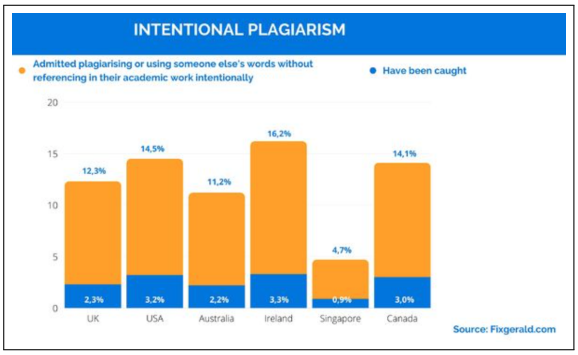

Figure 1: Intentional plagiarism statistics from Fixgerald.com

Given how pertinent this problem is, we were inspired to do a project in relation to this topic. Our work in this project employs principles learned in CS221, specifically taking them further, into the field of computational linguistics; we combine language and machine learning to solve complex problems. Our goal is to distinguish between different scriptwriters' styles in a dataset comprising dialogues from the Star Wars series. To tackle this, we employ DistilBERT, a streamlined variant of the BERT (Bidirectional Encoder Representations from Transformers) model. We chose DistilBERT as it is efficient given our computation power without compromising results.

We discuss the preprocessing of the Star Wars script data, which we found on Kaggle, which

involves tokenizing the dialogues and converting them into a format that DistilBERT can process. In CS221, we discussed Zero-one loss, Hinge loss, and Logistic loss. However, we researched Cross-Entropy Loss as well in order to extend our learnings in the class. During training, to enhance the model's accuracy, we employ two specific loss functions: cross-entropy loss and hinge loss; Our goal is to evaluate which of the loss function performs the task better.

Through this research, we aim to demonstrate the applications of machine learning in more unique fields. By applying DistilBERT to the Star Wars dialogues, we were able to showcase the intersection of artificial intelligence and creative arts. Ultimately, we were able to employ CS221 concepts to a real-life project and learned a lot in the process.

2 Literature Review

There is some existing work related to our task that we have read and used for inspiration in our project. The most relevant and exciting paper we came across was "Authorship identification using ensemble learning" [1]. The approach Abbasi et. al. took in the study was to use ensemble learning and a multi-depth DistilBERT model. Their main goal is to "identify the author of an unknown text." They used a larger Kaggle dataset "All the News" [6] given that they had more computational power than we are able to access. A common aspect between the paper and our focus is the goal of using advanced machine learning techniques for authorship attribution. Moreover, the authors used DistilBERT in their approach which aligns with our application of the same model. However, there is a key difference between our focus and the focus of Abbasi et al.'s study. Abbasi's paper does not specify the use of particular loss functions like cross-entropy loss or hinge loss, which are central to our study. Instead, it uses ensemble learning which combines multiple machine learning classifiers to improve overall performance. Thus, the methodology of the two approaches differs while the end goal is the same.

Another insightful paper that we read for our project is "A Machine Learning Approach to Authorship Attribution of Literary Texts" [7]. This study focuses on the application of Artificial Neural Networks (ANNs) for authorship attribution in literary texts specifically. This study analyzes works by Polish authors Bolesław Prus and Henryk Sienkiewicz, with a focus on quantitatively analyzing their writing styles (stylometry). This definitely aligns with our objective of identifying scriptwriters to fil script. However, our approach diverges from that of the study. While Stańczyk and Cyran employ ANNs and prioritize lexical and syntactic features, our study uses DistilBERT. Moreover, in contrast to the study at hand, we focus on the application of specific loss functions like cross-entropy loss and hinge loss for enhanced model training. Additionally, their work is primarily centered on literary texts, whereas our research targets film scripts, representing a different aspect of authorship attribution. This paper provided us with valuable insights into the broader scope of machine learning applications in authorship attribution.

During our preliminary research, another paper we came across was "Authorship Attribution Using Stylometry and Machine Learning Techniques" [5], focusing on authorship verification. Specifically, their goal was to detect plagiarism in academic texts. This study utilized machine learning algorithms like k-NN and SMO and the results showed $> 90\%$ accuracy in attributing authorship to segments of ten PhD theses. While their focus on academic texts differs from our focus on film script analysis, there are still methodological similarities, particularly in feature selection and machine learning application. Their study on how document size and segmentation effect classification accuracy provides insights that could be useful in the context of script analysis. Although the algorithms differ, the core challenges and approaches in authorship attribution offer valuable insights for our project.

Finally, the paper "Misleading Authorship Attribution of Source Code using Adversarial Learning" [4] was very thought-provoking in allowing us to reflect on the limitations of our approach. While the paper still focuses on authorship attribution, it begins with a unique perspective on how machine-learning approaches in this field can be exploited. The authors in fact produce an "attack" on these systems that perform a "series of semantics-preserving code transformations that mislead learning-based attribution but appear plausible to a developer." The approach they use is a Monte-Carlo tree search where they find that their attack misleads attribution in 99% of cases. This is quite significant and something we have kept in mind when asserting our claims.

3 Dataset

The dataset we chose was "Star Wars Movie Scripts," a publically available dataset on Kaggle containing the entire script for 3 Star Wars movies (Star Wars IV through VI). The dataset contains over 3,000 lines of dialogue, providing a substantial amount of dialogue to train the model on.

Here is a snippet of what the dataset includes and how it is formatted:

Line	Character	Dialogue
237	BEN	"Obi-Wan Kenobi... Obi-Wan? Now that's a name I haven't heard in a long time... a long time."
238	LUKE	"I think my uncle knew him. He said he was dead."
239	BEN	"Oh, he's not dead, no... not yet."

To create our training and testing sets, we employed a 90/10 split. This division was chosen to ensure a robust training set that captures the complexities of the dialogue while still reserving a sufficient amount of data for testing the model's generalizability. The tokenization process of this dataset was facilitated by DistilBERT's tokenizer. This is discussed further in the *Data Preprocessing* section of the paper.

4 Baseline

4.1 microGPT

Our first approach to creating a baseline for solving this problem was through the creation of microGPT: a Generative Pre-training Transformer that we hoped to train to solve our problem. Critical components of the old microGPT's architecture included:

1. Layer Normalization (LayerNorm): Implemented with an optional bias, enhancing the model's stability during the training process.
2. Causal Self-Attention Mechanism (CausalSelfAttention): Ensures that predictions for each token are influenced only by preceding tokens, maintaining the chronological integrity of text generation.
3. Regularization Techniques: The model, as per our baseline configuration, excludes dropout regularization, set at 0.0 in the GPTConfig, to simplify initial training dynamics.

However, during the training process we realized that we simply did not have the computational power needed in order to properly train and deploy our microGPT to solve this issue, so we pivoted to a DistilBERT approach.

4.2 DistilBERT

Realizing that our original microGPT idea was not practical due to computer limitations, we switched gears and went with a simpler solution – using a DistilBERT base model as our starting point. In simple terms, we employed a pre-trained DistilBERT architecture known for turning text into numbers, making it handy for various language-related tasks. This change helped us overcome the computational hurdles, and the DistilBERT model, being efficient and versatile, improved the overall effectiveness of our approach. The intracacies of the DistilBERT model's design will be covered in the following sections.

5 Main Approach

5.1 An Overview of Loss Functions

In the realm of multi-label classification, three primary loss functions come into play: mean-squared error loss, cross-entropy loss, and hinge loss. However, our focus narrowed on the latter two, cross-entropy loss and hinge loss, as they hold distinctive roles within the machine learning landscape, particularly for classification tasks. In order to explain our approach, let us start by taking a deeper dive into these two loss functions.

Cross-entropy loss shines when the precise probabilities of predictions are crucial. This loss function operates by minimizing the divergence between the model's predicted probabilities and the actual distribution of labels, thereby providing a nuanced assessment of prediction confidence. It is particularly useful when insights into the certainty of predictions are essential, making it a preferred choice for tasks where probabilistic output interpretations are critical.

Conversely, hinge loss is tailored to the task of maximizing the margin between different classes. While cross-entropy emphasizes probability distributions, hinge loss shifts its focus towards fostering decisive classifications with clear boundaries between categories. This quality becomes especially valuable when the crisp distinction between

classes takes precedence over detailed probability estimates. Hinge loss excels in scenarios where creating unambiguous class separations is the primary objective, potentially leading to improved generalization in such cases.

5.2 DistilBERT as Our Model

Before delving into our technical process, let us provide a high-level overview of the approach we are taking. In this project, we aim to create a model for author attribution in Star Wars scripts, specifically focusing on dialogue lines from Movies IV to VI. Our chosen model architecture is DistilBERT, a distilled version of BERT, which is pre-trained on large-scale text data and fine-tuned for various natural language processing tasks. DistilBERT is well-suited for this task because it can capture contextual information from the dialogue lines and provide meaningful representations.

To prepare the data, we tokenize the text using the DistilBERT tokenizer, converting the input dialogue lines into numerical representations that the model can process. The DistilBERT model for sequence classification is then initialized, taking these tokenized inputs and producing logits, which are raw scores representing the likelihood of each character being the author of the dialogue line.

5.3 Algorithmic and Training Details

The crux of our algorithm lies in its ability to make informed decisions about the likely author of a given dialogue line. To facilitate this, we introduce two distinct loss functions within our training loop: cross-entropy loss and hinge loss. These loss functions serve as the guiding principles that shape the model's behavior. Cross-entropy loss provides nuanced probability estimates, while hinge loss promotes decisive and clear-cut classifications. Let us explore how these components come together in detail.

5.3.1 Data Preprocessing

Our algorithm begins with data preprocessing, where the text data is meticulously prepared for the model. This process involves tokenization, facilitated by DistilBERT's tokenizer. Through tokenization, we transform the textual inputs, representing dialogue lines in this context, into numerical representations that the model can effectively process. This conversion is crucial for the subsequent steps, as it enables the model to interpret and analyze the input text.

5.3.2 Model Initialization

The next crucial step is the initialization of the DistilBERT model for sequence classification. This model, pre-trained on vast text corpora, is a powerful tool for understanding the context and semantics of textual data. It takes the tokenized text as its input and produces logits, which are raw scores indicating the likelihood of each character serving as the author of the dialogue line. The model initialization is a pivotal moment, as it sets the stage for the subsequent training process.

5.3.3 Loss Function Selection

In our algorithm, the choice of the loss function is a critical decision that significantly influences the model's learning and output behavior. We implement two distinct loss functions during the training loop, each designed to address specific aspects of the author attribution task.

For one branch of our algorithm, we opt for cross-entropy loss as the loss function. This choice is motivated by the need for a fine-grained understanding of class probabilities. Cross-entropy loss operates by quantifying the dissimilarity between the predicted probabilities, obtained through a softmax transformation of the model's logits, and the actual label distribution. This measure provides a mechanism to fine-tune the model's output probabilities, enabling it to express a nuanced level of confidence in its predictions. The model aims to minimize the divergence between its predicted probabilities and the true label distribution.

Simultaneously, we employ hinge loss as an alternative in the other branch of our algorithm. Hinge loss is strategically chosen to emphasize decisive classification performance over nuanced probability estimates. Its primary objective is to maximize the margin between different text categories, thereby fostering clear and unambiguous decisions. The hinge loss calculation considers the raw scores for the correct character (the true author) and the scores for all other characters, introducing a margin term to encourage decisiveness. This choice is particularly advantageous when the task demands a crisp distinction between potential authors.

5.3.4 Training Loop

The training loop constitutes the heart of our algorithm, where the model learns and adapts to the data. Both the cross-entropy loss model and the hinge loss model follow similar training processes:

1. Input the tokenized dialogue lines into the DistilBERT model.
2. The model generates logits, representing raw scores for each character as a potential author of the dialogue line.
3. For the cross-entropy loss model, we calculate the cross-entropy loss between the predicted probabilities (obtained through a softmax function) and the actual label distribution.
4. For the hinge loss model, we compute the hinge loss, focusing on maximizing decision margins and clear class separations.
5. In both cases, the loss is backpropagated through the model, and the model's weights are updated using an optimizer, such as AdamW.
6. This iterative process continues until the model converges, ultimately fine-tuning its output behavior based on the selected loss function.

In essence, our algorithm encompasses data preprocessing, model initialization, and the strategic selection of loss functions. These components collectively drive the model's learning process, allowing it to provide nuanced probability estimates and decisive classifications for author attribution in Star Wars scripts. The algorithm's design is tailored to the unique demands of the task, ensuring that it can effectively capture authorship patterns within the dialogue lines.

6 Evaluation Metric

We mainly used a loss-based analysis to determine accuracy. To do so, we assessed both qualitative and quantitative metrics to gauge the model's performance effectively.

6.1 Loss-Based Analysis

Our primary quantitative metric for assessing model performance was loss, specifically training loss and validation loss. These loss values served as crucial indicators of how well the model was learning from the data and generalizing to unseen examples. Focusing on loss, the cross-entropy loss and hinge loss functions quantified the dissimilarity between the model's predictions and the true label distribution. We compared the training and validation losses across different models and loss functions. The model with the lowest validation loss was considered the most successful, as it demonstrated the best generalization to unseen data.

6.2 Accuracy

We also underscored the importance of accuracy to evaluate our model's success. Accuracy is a fundamental quantitative metric that measures the proportion of correctly classified dialogue lines. While loss is informative, accuracy provides a clear and intuitive measure of the model's performance. Accuracy is calculated as the ratio of correctly classified samples to the total number of samples in the validation set. We used accuracy to complement our loss-based analysis. A higher accuracy indicates that the model makes more correct authorship attributions.

7 Results & Analysis

7.1 Baseline Model Performance

First, we investigated authorship attribution using the DistilBERT model commenced with a baseline measurement. This initial model was trained on a dataset comprising dialogue lines from Star Wars movies IV through VI as mentioned earlier. Despite the model's advanced capabilities, the baseline performance was suboptimal, with a training accuracy of merely 2% and a validation accuracy of 1%. This rudimentary implementation served as a stark contrast to the subsequent refined models, highlighting the necessity for fine-tuning to enhance the model's predictive capabilities.

7.2 Cross-Entropy Loss Performance

The first of our fine-tuned models employed Cross-Entropy loss. Throughout the training epochs, we observed a marked decrease in loss, indicating learning and improvement. Specifically, the model exhibited a final training loss of 2.19 and a validation loss of 2.32, a reduction from initial losses of 3.4 and 2.8, respectively (see *Figure 2* below). Notably, the fine-tuning process on a Google Colab-provided A100 GPU spanned approximately 18 minutes, reflecting the computational intensity of the task.

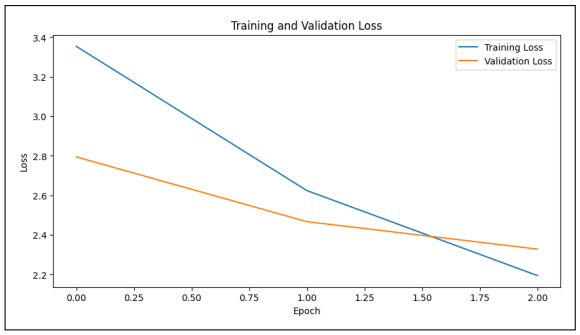

Figure 2: Cross-Entropy Loss: Epoch and Loss

7.3 Hinge Loss Performance

Contrastingly, the Hinge Loss model demonstrated a significant enhancement in performance, achieving a final training loss of 0.11 and a validation loss of 0.10. Despite a lengthier fine-tuning duration of approximately 24 minutes, the investment in computational resources yielded a substantially more competent model (see *Figure 3* below).

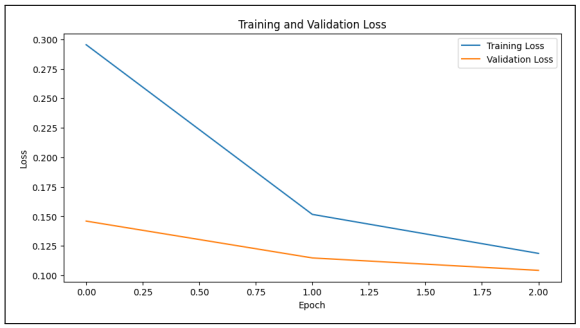

Figure 3: Hinge Loss: Epoch and Loss

7.4 Comparative Analysis of Loss Functions

Upon closer examination, the disparity in performance between the two loss functions becomes apparent. The Hinge Loss model's superior outcomes can be attributed to its intrinsic mechanism, which not only penalizes misclassifications but also enforces a margin between classes. This property is particularly beneficial for our multilabel classification task, where each dialogue line is explicitly associated with a character. The ability of Hinge Loss to create a buffer around the decision boundary ensures that the model is not merely accurate but also confident in its predictions.

Conversely, the Cross-Entropy Loss model, though efficient in its learning, demonstrated limitations in generalizing with the same level of precision. The primary mechanism of Cross-Entropy Loss, which focuses on the likelihood of class membership, does not inherently account for the margin of separation between classes. This distinction underscores the reason for the observed performance gap, with Hinge Loss emerging as the more robust choice for our dataset and task.

7.5 Implications for Further Research

The insights gathered from our project shed light on the critical role of loss functions in model training and their consequential impact on performance. The findings suggest that, for tasks similar to ours, where distinct class boundaries are paramount, loss functions that incorporate margin maximization principles, such as Hinge Loss, may offer substantial advantages. Moving forward, this helps to inform our future model fine-tuning strategies, as we continue to refine our approach to authorship attribution not just for movies placed in a galaxy far, far away but also for various real-life applications and literature.

8 Error Analysis

In the pursuit of refining our author attribution model, we conducted a series of experiments designed to bring about the strengths and weaknesses of our system. This iterative process involved manipulating the dataset and observing the resultant model performance, thus providing critical insights into the underlying mechanics of our approach.

8.1 Data Segmentation Experiments

Our initial experiment utilized a singular film script (Star Wars: Episode IV) as the training dataset. This narrowly focused dataset resulted in poor model performance, underscoring the necessity for a broader corpus to capture the nuanced variances in writing style. In response, we expanded our dataset to encompass scripts from three films (Episodes IV, V, and VI). This augmentation significantly improved model accuracy, affirming the hypothesis that a more extensive dataset leads to a more adept and discerning model.

Movie	# Lines	# Unique labels
Episode IV	1010	60
Episode V	839	49
Episode VI	674	53

8.2 Limitations of Baseline Model

An unforeseen revelation was the underwhelming performance of the baseline DistilBERT model, which proved to be a critical moment of learning for our team. The baseline's ineffectiveness

posed questions about our training methodology. A deep dive into the training procedure revealed that the baseline model's inability to predict the labels accurately could be attributed to potential mismatches between the model's pre-training and our task-specific requirements. Unlike the fine-tuned models, where the dataset was meticulously parsed and the model retrained, the baseline model likely needed adjustments in the dataset preprocessing or the prediction methodology to align with our classification labels.

8.3 Computational Constraints

Our computational resources were constrained by the limitations of the Google Colab high-GPU runtimes. Despite the high performance of these GPUs, our dataset's scale and the required computational time exceeded what was available. This limitation prevented us from expanding the dataset further, which could have included additional scripts or extended texts, potentially yielding improved results. The computational ceiling we encountered serves as a reminder of the importance of adequate resources in machine learning endeavors.

9 Future Work

While our current model results provided initial valuable insights into author attribution in Star Wars scripts, there are several avenues for future work and improvements that we aim to explore. In particular, let us explore some ways we could have improved our model if we had more time.

9.1 Expanded Dataset

One of the primary challenges we encountered was the limited size of our dataset, which may have hindered the model's performance. In the future, we intend to expand our dataset by incorporating dialogue lines from additional Star Wars movies (e.g. Star Wars I through III, and potentially the newer movies) as well as related content (e.g. The Clone Wars, The Mandalorian, etc.). A more extensive and diverse dataset would allow the model to generalize better and enhance its accuracy. We would also love to eventually expand our model to real-life use cases. One pertinent issue this could be applied to would be classifying writers of historical documents. It's often argued who the authors of certain Federalist papers are and an interesting application to this model would be to train it on the writings of the Founding Fathers and identify who wrote which contested Federalist paper.

9.2 Fine-Tuning Strategies

Experimenting with different fine-tuning strategies could be beneficial. We plan to explore variations in hyperparameters, such as learning rates and batch sizes, to optimize the model's training process further. More specifically, if we had more time, we were attempting to explore techniques like curriculum learning, where the model is exposed to easier examples before challenging ones, which could potentially enhance performance. Additionally, with more time we could have found other loss functions online that could have improved our accuracy and or given us better loss data.

9.3 Sharper Error Analysis and Mitigations

Understanding the specific cases where our model struggles is crucial. Conducting thorough error analysis can help identify patterns or challenges that the model faces. If we were permitted more time, we would have explored techniques like confusion matrix analysis and precision-recall curves to gain insights into the model's strengths and weaknesses. This could help identify patterns of misclassification and ambiguity. Additionally, this may help inform the development of mitigation strategies, including context augmentation and reinforcement learning techniques, to improve the model's performance on challenging dialogue lines.

10 Code

The code can be found as a Jupyter Notebook file. It has been hyperlinked.

References

[1] Ahmed Abbasi, Abdul Rehman Javed, Farkhund Iqbal, Zunera Jalil, Thippa Reddy Gadekallu, and Natalia Kryvinska. 2022. Authorship identification using ensemble learning. https://www.nature.com/articles/s41598-022-13690-4.

[2] Fixgerald. 2021. Cheating and plagiarism statistics among college students in 2021. https://fixgerald.com/blog/cheating-and-plagiarism-statistic.

[3] ICAI. 2020. Icai facts and statistics. https://academicintegrity.org/resources/facts-and-statistics#:~:text=This%20work%20demonstrated%20that%2064,test%2C%20plagiarism%20or%20copying%20homework.

[4] Erwin Quiring, Alwin Maier, and Konrad Rieck. 2019. Misleading authorship attribution of source code using adversarial learning. https://arxiv.org/pdf/1905.12386.pdf.

[5] Hoshiladevi Ramnial, Shireen Panchoo, and Sameerchand Pudaruth. 2016. Authorship attribution using stylometry and machine learning techniques. https://link.springer.com/content/pdf/10.1007/978-3-319-23036-8_10.pdf.

[6] Star Wars Movie Scripts. https://www.kaggle.com/datasets/xvivancos/star-wars-movie-scripts.

[7] Urszula Stańczyk and Krzysztof A. Cyran. 2007. Machine learning approach to authorship attribution of literary texts. http://www.wseas.us/journals/ami/ami-22.pdf.