Text Based Speaker Identification on Domain Specific Conversation Corpus

Shaswat Anand
University of Southern California
shaswata@usc.edu

Jeevitha Gowda Chandramouli
University of Southern California
gowdacha@usc.edu

Revanth Madamala
University of Southern California
madamala@usc.edu

Sahithi Ramaraju
University of Southern California
ramaraju@usc.edu

Aamulya Sehgal
University of Southern California
aamulyas@usc.edu

Abstract

Profiling users based on linguistic patterns of communication to identify the speaker in the age of big data, where we have access to a lot of online information, has a huge scope. This knowledge can be used to personalize actions for each user. This paper focuses on the main part of identifying characters from given scripts in the domain of TV series, which can be further extended to recognizing users through their dialogue corpus. In this paper, we intend to develop machine learning models using traditional and neural based approaches to achieve this task.

1 Introduction

The project[1] is to perform speaker identification using scripts and screenplays of TV series. Labeled dialogues of various speakers has been used to train models with the objective of correctly tagging unseen dialogues with an identified speaker.

The motivation is to be able to use large corpus of conversational dialogues present in scripts of a TV show to be able to unearth linguistic patterns to be able to identify the speaker and indirectly create a profile of the speaker.

The problem is a multi-class classification problem of mostly conversational dialogues from TV series which makes it difficult to identify interesting features to use for training and inference. Speaker identification can be used by new writers to see if the new dialogues that they attribute to certain characters are consistent with the existing dialogues written for that character by earlier writers.

2 Related Work

We observed that most speaker identification work involves auditory features. Few text-based speaker identification studies have employed diverse methods.

Film Dialogue Speaker Identification has been studied (Kundu. et. al., 2012). They organized cinematic conversation speakers by linguistic stylistic elements using a text corpus of film scripts from the Internet Movie Script Database (IMSDB) archive and the K-Nearest Neighbor (KNN) Algorithm, Naive Bayes Classifier, and Conditional Random Field (CRF). A KNN speaker identification model for Friends has been proposed. They differentiate characters on ”What the speakers say” rather than ”How they speak.” (Agarwal. et. al., 2018). Another offered a multi-model for Big Bang Theory and eighteen genre-specific movie scripts including visual, textual, and audio modalities in a unified optimization framework.(Azab. et. al., 2018)

We are proposing a speaker identification system extending these previous methods, leveraging scripts and screenplays of TV shows and use conventional models and transformers to classify characters.

3 Method

3.1 Data Collection

We decided to use two datasets which contain speaker and their dialogue. The first dataset selected for this was a ’Friends’[2] datasets. This

[1] https://github.com/madamalarevanth/Speaker-Identification-from-dialogue-scripts

[2] https://fangj.github.io/friends/

dataset contains all 10 seasons of dialogues. This dataset had to be web scraped.

The second dataset was the 'The Big Bang Theory (BBT)'[3] dataset. We found a raw Kaggle dataset that contained web scraped dialogues for 10 seasons of the show.

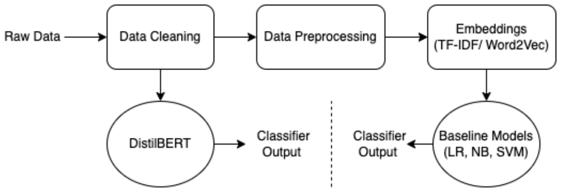

Figure 1: Methodology Flowchart

3.2 Data Cleaning and Pre-processing

Dataset cleaning was performed since they had incorrect formatting and incorrect components like dialogue description, who the dialogue was being addressed, etc. It also had misspelled speaker names which were corrected. In all, a significant amount of data wrangling was performed to improve the dataset quality. Feature engineering techniques like TF-IDF, CountVectorizer, and Word2Vec models built from the dataset were used for baseline model embeddings, while the pre-trained DistilBERT tokenizer and model from huggingface transformers library were used for uncontextualized sentence level embeddings for the BERT-based model.

3.3 Data Analysis

The processed datasets were analysed to find some interesting information about the data. Analysis like top speakers by lines was used to focus the dataset on major characters and remove dialogues from single appearance or low dialogue characters. We can also see that the Big Bang Theory dataset is much more imbalanced as compared to the Friends dataset.

Topic modeling was performed on both datasets to understand character complexity and thought diversities using Latent semantic indexing model, affinity analysis to understand each character's relationship with one another and sentiment analysis using TextBlob library.

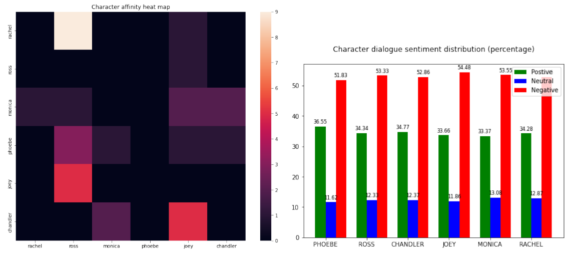

Figure 2: Affinity Heat Map and Sentiment Analysis - Friends

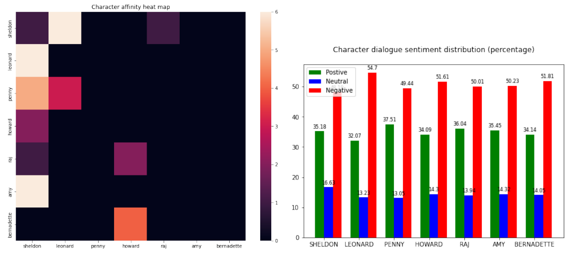

Figure 3: Affinity Heat Map and Sentiment Analysis - BBT

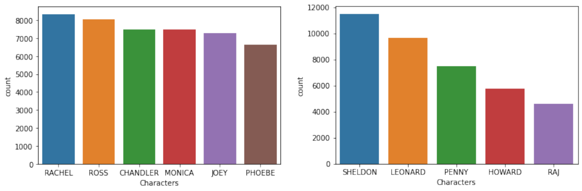

Figure 4: Dialogue Count Per Character

3.4 Models

Three baseline models (Logistic Regression (LR), Naive Bayes (NB) - Multinomial and Support Vector Machine (SVM) - Support Vector Classifier) and Transformer models using BERT Base Uncased and DistilBERT Base Uncased (Sanh. et. al., 19) are used to learn from data

3.4.1 Training

Baseline models are trained using feature engineering methods mentioned above. GridSearch with 5-fold cross validation is used for hyperparameter tuning of regularization parameter for LR, smoothing parameter (alpha) for NB, kernel coefficient (gamma) and regularization parameter (C) for 'rbf' kernelized SVM. The obtained results are shown in table 1.

Apart from that, we used BERT, and DistilBERT Base (Dogra. et. al., 21) Uncased model with and without pre-trained weights for obtaining text embeddings from the last hidden layer with a

[3] https://www.kaggle.com/datasets/mitramir5/the-big-bang-theory-series-transcript

batch size of 64. These embeddings are then converted to TensorFlow datasets (train, val, and test). With the pre-trained TensorFlow model from DistilBERT, we use Adam as the Optimizer with Categorical Cross Entropy as the Loss function for model training.

4 Experimental Setup

For training DistilBERT model, we obtain the sentence level embeddings from the last hidden layer of the pre-trained model and we employ the Adam Optimizer with a learning rate of 5e-5, Sparse Categorical Cross Entropy as the Loss function, Sparse Categorical Accuracy as metric, and we fit the model for 5 epochs. The DistilBERT model takes approximately 30 minutes per epoch in the train phase.

4.1 Datasets

We used our cleaned up datasets - Friends and The Big Bang Theory (BBT) - for the task of speaker identification. The datasets were split into train, validation and test datasets in a 7:1:2 ratio. The models were trained and tested separately on the two datasets.

In addition to the main classes of the datatsets (representing the main characters) which are to be predicted, we augmented our dataset by merging the dialogues of all other characters into a class 'Others' to perform negative sampling.

4.2 Baseline methods

We trained and tuned the baseline models - Logistic Regression, Naive Bayes and Support Vector Machine - with each of our feature engineering techniques - Count Vectorization, TF-IDF, Word2Vec with averaging and Word2Vec with TF-IDF averaging - so in total 12 baseline models.

For accuracy, we have provided the number for all baseline model configurations for both datasets. However, for weighted f1 scores we have just provided the numbers for the best performing configuration of the baseline models for both datasets.

4.3 Evaluation protocols

For evaluation the performance of our model, we used weighted F1 scores and accuracy numbers on the datasets. In addition, we also plotted the confusion matrix for DistilBERT predictions.

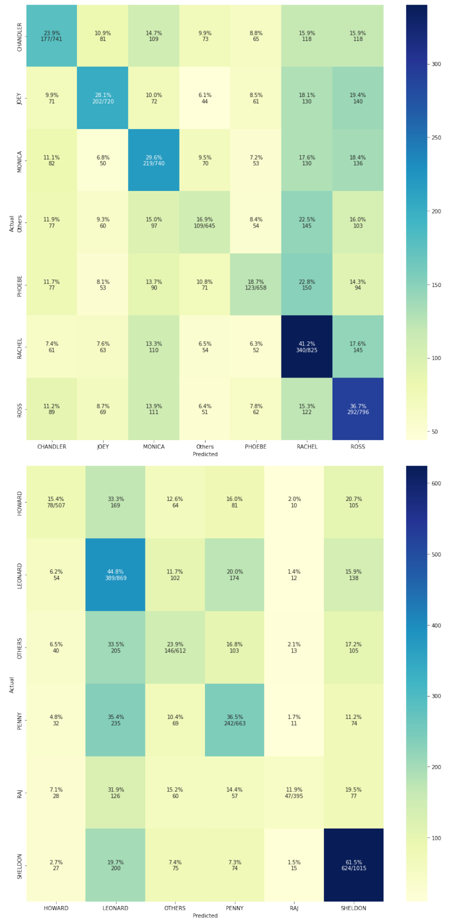

Figure 5: Confusion Matrices

5 Results and Discussion

Friends and Big Bang Theory test accuracies and weighted F1 scores for baselines and DistilBERT model are reported in tables 1 and 2.

In table 3, we have included the model performance on some distinguishable dialogues of these shows (which are usually spoken quite frequently and by one of the main characters) and we see that model performance is much higher than on them compared to the overall dataset.

We see that the performance of DistilBERT is better than the baseline models in both of our evaluation metrics. This seems to be due to the attention mechanism and contextualized embeddings generated by the model are better features especially for this corpora of conversations which doesn't have much diversity in the lines spoken by the characters.

In Table 4, we have included DistilBERT model's precision recall and F1 scores for all the characters in the two datasets. We can see that the model performance for the characters in Friends

Model/FE	**CV**	**TF-IDF**	**W2V A**	**W2V TF-IDF**
Dataset: Friends				
LR	24.8%	24.6%	21.1%	21.1%
NB	25.2%	24.7%	18.7%	18.1%
SVM	24.0%	24.5%	23.7%	23.6%
BERT	20.0%			
DistilBERT	**29.6%**			
Dataset: BBT				
LR	33.3%	37.3%	31.2%	30.9%
NB	34.0%	36.6%	28.9%	28.2%
SVM	33.0%	37.2%	33.1%	33.0%
BERT	28.0%			
DistilBERT	**38.5%**			

Table 1: Model Accuracy

Model/Metric	**Precision**	**Recall**	**F1**
Dataset: Friends			
LR	0.2465	0.2470	0.2453
NB	0.2539	0.2520	0.2403
SVM	0.2466	0.2452	0.2407
DistilBERT	**0.2988**	**0.2952**	**0.2910**
Dataset: BBT			
LR	0.3575	0.3733	0.3367
NB	0.3980	0.3661	0.3067
SVM	0.3746	0.3726	0.3218
DistilBERT	**0.3747**	**0.3852**	**0.3511**

Table 2: Weighted Precision Recall F1 Score

dataset is much closer to each other. Whereas, the model performance for ”Sheldon” is much higher than that for all other characters and ”Raj” is much lower for the Big Bang Theory dataset. This can also be attributed to higher imbalance in the lines in the dataset for Big Bang Theory as compared to Friends which can be seen in figure 4.

The distinguishable dialogues and F1 score for characters tables as well as the confusion matrices in figure 5 show that certain characters, like Joey, who is a lot more goofy compared to other characters, and Sheldon, who is a lot more anti-social compared to other characters, have much higher speaker identification performance in general as well as on their distinguishable dialogues. This is an interesting result which shows that the model is learning some unique features about these characters from their distinct dialogues.

Dialogue	**Accuracy**
Dataset: Friends	
”how you doin’?” (Joey) ”could you be any more?” (Chandler)	**41.66%** **35.71%**
Dataset: BBT	
”bazinga” (Sheldon) ”sweetie” (Penny)	**55.55%** **43.33%**

Table 3: Distinguishable Dialogue Fragments and DistilBERT Accuracy

Speaker/Metric	**Precision**	**Recall**	**F1**
Dataset: Friends			
JOEY	0.2660	0.4611	0.3373
ROSS	0.3223	0.2952	0.3081
RACHEL	0.3323	0.2787	0.3032
MONICA	0.2860	0.3270	0.3051
PHOEBE	0.3167	0.2811	0.2979
CHANDLER	0.2982	0.2294	0.2593
Dataset: BBT			
SHELDON	0.4792	0.7407	0.5819
LEONARD	0.3080	0.4293	0.3587
PENNY	0.3584	0.3589	0.3587
HOWARD	0.3069	0.2183	0.2551
RAJ	0.3045	0.1318	0.1840

Table 4: DistilBERT Precision Recall F1 Score For Speakers

One difference observed between the two datasets is that the speakers in Big Bang Theory don’t have a lot of character affinity amongst each other whereas speakers in Friends have more character affinity amongst each other. This makes distinct characters in the Big Bang Theory dataset more diverse compared to those in the Friends dataset. Hence, we see better results for Big Bang Theory dataset.

For the future, including more features like how the dialogues were said, such shyly, whispering, etc., and emotion while saying the dialogues, such angry, happy, sad, etc., and further fine-tuning the models could result in better performance. Maybe including the audio of the dialogues in addition to their transcription with ensemble methods could help as well.

6 Division of labor

We have identified the following areas into which the total work to be done can be divided. While all the authors would be collaborating in different areas while working on this project, we have listed the first names of the primary contributors in each domain.

- **Background Research**: Revanth
- **Dataset Collection and Cleaning**: Aamulya
- **Data Processing**: Shaswat & Jeevitha
- **Data Analysis**: Sahithi
- **Model Training and Evaluation**: Revanth & Sahithi
- **Final Metrics and Conclusion**: Aamulya & Revanth
- **Report** Shaswat & Jeevitha

References

Mohit Agarwal, Mayank Chaudhary, Purwa Maheshwari and Kaushal Kishor. 2018. *Text Based Speaker Identification.*

Mahmoud Azab, Mingzhe Wang, Max Smith, Noriyuki Kojima, Jia Deng, and Rada Mihalcea. 2018. *Speaker Naming in Movies. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1 (Long Papers), pages 2206–2216. New Orleans, Louisiana. Association for Computational Linguistics

J. P. Campbell Jr 1997. *Speaker recognition: A tutorial. Proceedings of the IEEE 85.9*, pp. 1437-1462.

V. Dogra, A. Singh, S. Verma, Kavita, N.Z. Jhanjhi and M.N. Talib 2021. *Analyzing DistilBERT for Sentiment Classification of Banking Financial News. Intelligent Computing and Innovation on Data Science. Lecture Notes in Networks and Systems,vol 248. Springer, Singapore.*

Amardeep Kumar and Vivek Anand. 2020. *Transformers on Sarcasm Detection with Context. In Proceedings of the Second Workshop on Figurative Language Processing*, pages 88–92, Online. Association for Computational Linguistics.

Amitava Kundu, Dipankar Das, and Sivaji Bandyopadhyay 2012. *Speaker identification from film dialogues. IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction.*

Kaixin Ma, Catherine Xiao and Jinho D. Choi 2017. *Text-based Speaker Identification on Multiparty Dialogues Using Multidocument Convolutional Neural Networks.*

Suneel Patel 2020. *NLP Pipeline: Building an NLP Pipeline, Step-by-Step. Medium*, Online.

Victor Sanh, Lysandre Debut, Julien Chaumond and Thomas Wolf 2019. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter arXiv preprint arXiv:1910.01108.*