Genre prediction based on movie dialogues and metadata

Anvita Srinivas*
a5sriniv@ucsd.edu
University of California, San Diego
San Diego, California, USA

Pradyumna Sridhara*
prsridha@ucsd.edu
University of California, San Diego
San Diego, California, USA

Vishal Kundurli Sankaranarayanan*
vkundurl@ucsd.edu
University of California, San Diego
San Diego, California, USA

In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

1.1 Dataset

For this project, we use the Cornell Movie–dialogues Corpus from [6]. It contains 220,579 conversational exchanges between 10,292 pairs of movie characters, which were extracted from 617 movies. In addition to the dialogues, the corpus also contains metadata such as the genres of the movie, the release year, IMDb rating and the number of IMDb votes. Each training example of the dataset is one conversation, which can span from an exchange of two dialogues between one or more characters, to twelve dialogues (Figure 5).There are 83,097 conversations in this dataset. The movies chosen in the corpus are largely released around the 1990's to 2000's as can be seen in Figure 4. The creators of the dataset extracted around 100-200 conversations from each movie (Figure 1), and most movies have about 10-15 characters (Figure 3).

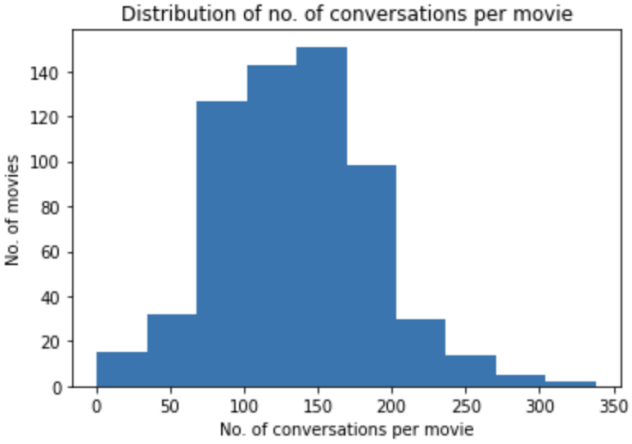

Figure 1: Distribution of conversations per movie

We analysed the popularity of movie genres from two standpoints – the number of movies created (released) that belong to a particular genre, and the number of viewers who actually watched (and rated) these movies. From Figure 2, we can see that drama, thriller, and action, are the top three most popular genres both in terms of the number of movies released and the number of votes received. Of course, the number of movies released and the number of votes received are highly correlated, but it is useful to compare the two when there is a large discrepancy, since it can signify that the audience's preference of a genre does not match with movie makers' perception of that preference. Two such categories are comedy and horror, where it appears that the perceived popularity of the genres might be higher than their actual popularity.

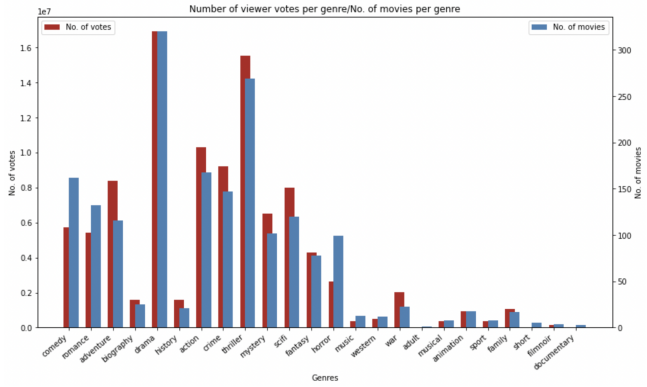

Figure 2: Distribution of votes and movies per genre

We also attempted to analyse whether we can get any indications about the genre of a movie from its ratings. To do this, we plotted the difference between the average movie rating per genre, with the average movie rating across the entire dataset. As seen in Figure 6, popular genres like action and thriller actually seem to have lower ratings than average, while relatively unpopular genres like war, film noir, biography, and history, have ratings which are higher by almost one point. Thus, based on the preliminary analysis of the data, we hypothesise that movie ratings might be a useful feature for the predictive task described in the next section.

1.2 Predictive Task

The predictive task we aim to learn through models is to predict the genres of a movie based on the various features aforementioned in the dataset. This is a multi-output multi-class classification predictive task, and there could be multiple genres associated with each datapoint. The feature set comprises of a subset of conversation between one or more characters, the release year, rating of the

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/1122445.1122456

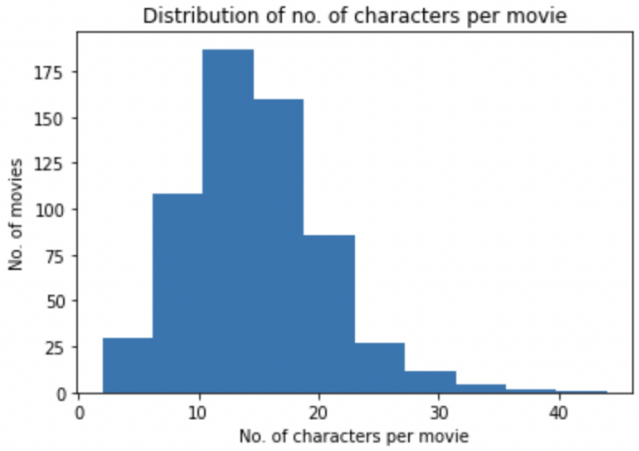

Figure 3: Distribution of characters per movie

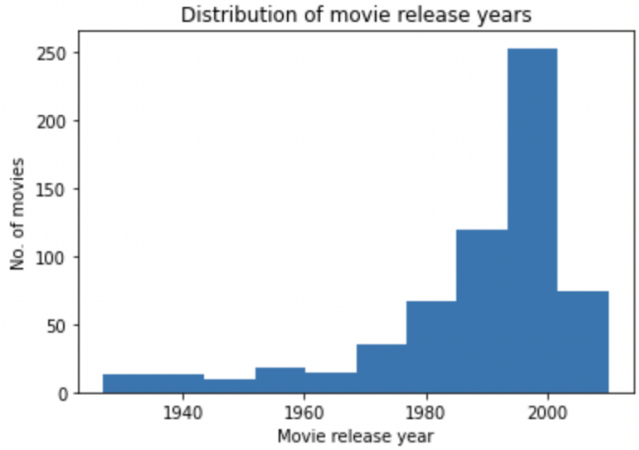

Figure 4: Distribution of movie release years

movie, number of votes, title of the movie, and the character names of the movie. For the text based features, we perform either TF-IDF or word2vec transformations to convert them into a vector that the model can learn. The output of the model is a vector of 0s and 1s where a 1 denotes that the genre is predicted positive, and 0 denotes the genre is predicted negative. The baseline we propose is a popularity baseline which predicts the most common genres to every datapoint in the test set. For instance, if our popularity threshold is 0.05, we provide as predictions all the genres that come in 5% of the data points. For our baseline, we utilized the threshold as 0.08% from our analysis of the dataset, which predicted the same three genres for all the data points. The detailed methodology in obtaining the feature set, and training the various models are explained in the methodology and experiments section respectively. In order to nor be misled in terms of a single metric, we did run the model predictions through various metrics such as Hamming loss, weighted precision, recall and the balanced error rate. [1] talks about the various metrics that were used for a similar predictive task, and they talk about how hamming loss and accuracy

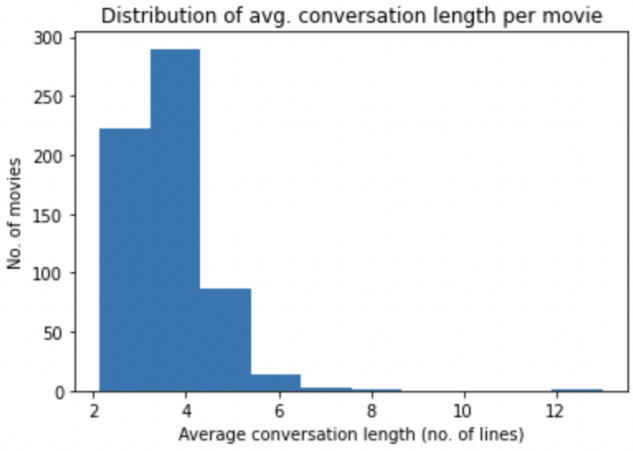

Figure 5: Average conversation length per movie

are overly generous assessment of the performance because of the class imbalance in the dataset as described in the previous section.

The evaluation metric we have considered to be the most apt for this predictive task is the weighted average F1 score, and we have leveraged the F1 score implementation from sklearn.metrics to achieve this.

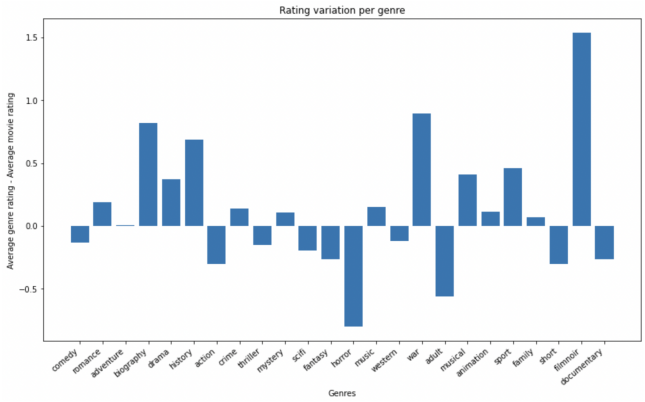

Figure 6: Rating variation per genre

2 LITERATIVE REVIEW

The Cornell Movie-dialogues Corpus was originally released in [6] to understand how conversational participants unconsciously adapt to each other's conversation styles. Subsequently, it has been used for many diverse applications, such as – language generation (for example, conversational response generation [14], stylistic language generation [15], dialogue generation[8]), and movie success prediction [2]. These are different from our predictive task which focuses on predicting the genres of a movie given its dialogues.

There are a multitude of publically available dialogue datasets. [3] contains around 130k movie dialogues and 6 million words, as well as context descriptions written on the script. However, the dataset does not guarantee that the conversation is only between

two people. To overcome this, the authors of [11] extract and filter approximately 1 million dialogues down to 86k dialogues that occur only between two individuals. Other dialogue datasets also use different sources such as trial proceedings [10] or TV scripts [12]. [13] also surveys various datasets that can be used for dialogue systems. A large number of these datasets have been used for language and dialog generation tasks. However, most of these datasets do not contain any metadata about the movies or dialogues (such as the genres, ratings, or release years) and hence cannot be used for our predictive task.

Common datasets for predicting movie genres involve features such as the movie plot and their posters. For example, [4] use the Full MovieLens Dataset from Kaggle [9] to learn to predict the genres of a movie using its posters. They leverage transfer learning on ResNet-34 to extract image features from the dataset. However, the dataset they use is considerably smaller than the Cornell Movie-Dialog Corpus, and contains only around 35k examples as compared to 83k samples. Another paper [7] studies the efficacy of methods such as K-binary transformation and probabilistic classification for the multi-label genre prediction task. They use a custom dataset containing plot summaries and genres of movies which was extracted based on the IMDb dataset. While our predictive task is the same as theirs, we are trying to solve a more challenging problem as it is harder to predict the genres of a movie from limited lines of dialogue than it is from the entire plot summary. [5] is the most similar to our work as the authors predict movie genres based on features extracted from movie scripts. They compare and analyse the performance of two classifiers – a Naive Bayes classifier and a Maximum entropy Markov Model classifier, and find that the latter performs better. A major difference between their work and ours is that they have access to all the dialogues of a movie to make a decision about its genre, while we only have access to one conversation excerpt at a time. However, we can harness features like the movie rating, character names, and release date, which can provide valuable signals about the genres of a movie.

3 METHODOLOGY

Our dataset contains a rich metadata of movie characteristics. We try to harness these optimally through feature engineering and feature selection. Based on the preliminary analysis of the data, we picked the following features to use for our multi-label classifier:

Movie rating: As seen in Figure 6, movie ratings can have some predictive power when it comes to identifying the genre of a movie. Often, less popular genres like war and history receive higher ratings, possibly because those movies have been selectively picked by the audience. We pass the movie rating as a feature to the model by subtracting the mean movie rating across the entire training dataset from it. This gives the model an indication about whether the rating of the movie is higher than average or lower than average, based on the training data.

Number of votes: Similar to the movie rating, the number of votes also indicates the preference of the audience towards a certain genre. However, having a distinction between the number of votes and the movie rating given a genre might prove to be helpful for the model, for the reasons outlined in the section describing the dataset. As the popularity of a movie is an important factor determining which movies (and hence which genres) the majority of the audience will watch, we hypothesise that the movie ratings and the number of votes received by a movie can be important features for the model. We process the number of votes for each movie in the same way as the movie rating – we subtract the mean number of votes across the entire training dataset from the number of votes of each movie.

Release year: We try to capture the temporal trends of the popularity of various movie genres by incorporating the release year of the movie as a feature. For example, a movie released in the 2000s might be more likely to be a fantasy movie than a movie released in the 1940s, due to the improvement in visual effects technology. We process the release year of each movie by subtracting the earliest release year from the training dataset from it. This way the model can weigh how important the recency of the movie release date is to predict which genre it belongs to.

Movie title: The movie title can be an important feature to identify the genre of the movie, as titles often give a fair idea about what a movie is going to be about. We encode each title as the TF-IDF vector that is computed on the set of all movie titles in the training set. We do not filter out any words from this dataset, and the resulting vocabulary size is 786 words.

Movie character names: Often, character names can also hint at the genres a movie belongs to. For example, character names with words like 'detective' and 'journalist' are more likely to belong to crime movies than romances. Hence, to encode this, we first collect a corpus of all the words in character names from the training data. Then, we filter out the words that occur in less than 3% of the training corpus. The resulting set contains 58 terms, including words such as 'doctor', 'secretary', 'cop', and 'nurse'. We then compute the TF-IDF vector for each character name using these words.

Movie dialogues: Movie dialogues are another source of information that can help the model predict the genre of a movie. We process conversations by concatenating each dialogue to form one paragraph. One conversation is analogous to a single document. We then remove stop words, and filter out words that occur in less than 10 conversations in the training set. We also limit the maximum number of words considered to 1000, based on the term frequency of the word. Lastly, we generate TF-IDF vectors for each conversation using the remaining words from the above step.

Our next step is to create models using a subset of the features described above. The various models we have attempted to train are described in the next section.

4 EXPERIMENTS

4.1 Models

We have experimented multiple models to accurately describe the data and map it to the relevant multi-label genres. For each model, the score returned was the average across labels weighted on each label's number of positive classes.

Naïve-Bayes. The Naïve-Bayes method is a well-known prediction algorithm that can work well for multi-class prediction. It is computationally more feasible, simple to tune and can work with less data (as opposed to other models such as deep learning). Multinomial Naïve Bayes is a popular model used for NLP and text processing. We wanted to see how we can extend this to multi-label

classification and more specifically to the task of predicting multiple genres for each movie data point. For this, we used one-vs-all classification using Multinomial Naïve Bayes. After tuning, this model gave an f1 score of 0.33. Since our data consisted of non-textual features such as release year, movie rating and number of votes as well, Naïve-Bayes would not be able to predict accurately over these features. Additionally, Naïve-Bayes assumes that the predictors are independent of each other, which is not the case for this dataset.

Logistic Regression. The next choice of model was Regression, namely, Logistic Regression. This too is easy to train and is a good starter model for textual data. Similar to the Naïve-Bayes approach, we used one-vs-all classification for Logistic Regression as well. This model given an f1 score of 0.32, very similar to Naïve-Bayes. Some of the drawbacks from the previous model exist here too. Additionally, Logistic Regression assumes that there is a there is a linear decision surface that separates the data. This might not be sufficient to model our dataset.

K-Nearest-Neighbors. We explored the K-Nearest-Neighbors approach, to find inherent patterns in the data, without explicitly having to label them. The features were grouped in an n-dimensional space and during prediction, the genres of datapoints in its vicinity were predicted. We tuned k,i.e the number of neighbours to consider to be 5. KNN was particularly suited to multi-class classification, which in our case could be easily extrapolated to multi-label classification. However, since our feature size was large, KNN failed to scale and ultimately could not learn over the data effectively. Additionally, our data was skewed towards 3 genres which added a bias on the model, which KNN was not particularly good at tolerating. We obtained a weighted f1 score of 0.23.

Decision Tree. Another class of models we implemented was the Decision Tree. We tried this approach as it was fairly intuitive for the text classification task at hand. A human can easily classify a movie based on its genre, using movie dialogues by following a rule-based method. Decision Trees try to mimic the same by creating decision branches and associating a match probability at each leaf. We tried an additional feature engineering step of PCA, to prune our features and retain only the important ones. This model took slightly more amount of time to train. We obtained an f1 score of 0.311 without PCA and a slight improvement to 0.377 with PCA.

XGBoost. A logical next step in model selection was gradient boosted trees. For this, we used to Sklearn library's XGBoost implementation. XGBoost after recently gaining widespread popularity seemed perfect for the task. It's approach of ensemble learning was promising as any one model would probably over-fit but a combination of different models would mostly likely perform better. However, in our case the results from XGBoost were not as positive as expected. After tuning, we obtained an f1 score of 0.42.

Sentence2Vec. Another variant of the tf-idf approach used in feature engineering was employing Word2Vec as part of the mix. The idea was the incorporate inherent semantics of the movie dialog sentences as a "Sentence2Vec". This sentence vector was constructed by weighting each word's Word2Vec vector by its tf-idf score. Then a single vector was obtained for each dialogue by taking the mean of the tf-idf weighted Word2Vec vectors. This was a simple aggregation strategy that ensured each word contributed to the sentence and each sentence had the same feature dimensionality. However, this approach encountered other implementation hurdles as it took significant amount of time and computational resources to train, which was beyond what we could afford, but we hope to implement this in the future.

SVM. Our best performing model was a Support Vector Machine (SVM) variant. SVMs are a popular model for text classification, with one of the main reasons being that they are independent of the dimensionality of the feature space. This worked very well for our use case as our task had a large feature size. Also, SVMs work well with sparse vectors, which also matched our scenario. We used Sklearns SVC implementation for this model. It showed promising results compared to all other approaches. For this reason, we invested more resources in further tuning our hyper-parameters. A small subset of the hyper-parameter space was explored through GridSearch. The regularization parameter was 1.5 and the radial basis function kernel was found to work best. This model led to an f1 score of 0.51.

Baseline Model. Analytics on the data revealed that the labels are skewed towards 3 specific genres. We should be able to get significantly accurate prediction rates by considering these 3 most popular genres as the prediction for all examples. We considered this as our simple baseline model. The f1 score for this model was 0.27.

Ablation Tests. The best performing model was SVM, and we intended to perform ablation tests on this model to learn the most contributing features and remove the ones which do not contribute as much or which tend to overfit the model. The ablation tests on SVM ended up taking a lot of processing power, and needed more which didn't seem feasible for this experiment, so we performed the ablation test on the next best model which was the XGBoost. We compared the F1 scores of all the models omitting each of the features, and is tabulated in Table 2. The score on each column implies the F1 score of the model omitting the feature in the same column. For instance, the first column release year, and the value 0.232 implies that the F1 score on the model omitting the release year is 0.232. On performing this experiment, one conclusive result we obtained was all the features were important to the predictive task, and the rating and number of votes were the most important.

5 DISCUSSIONS AND CONCLUSION

Our aim was to achieve a predictive model that predicts the genres given movie dialogues as well as other metadata. We have utilized various state of the art models and concepts that were proven empirically to be efficient in predicting any multi-output classification task. The baseline we developed was a rather simple one considering the dataset we created, and the patterns we noticed while analyzing the dataset. Our baseline was a naive predictor which predicts as genres for all movies the most common genres in the dataset based on a specific threshold. All the models we had developed and trained beat the simple baseline by a large amount in terms of the F1 score. The most efficient model that was proposed in this work was SVM, which gave a F1 score of more than 0.5, as compared to the baseline which was only 0.27. The best performing SVM was the one with regularization parameter of 1.5 and radial basis kernel function. A small set of hyper parameters were fed to

Table 1: Experiment metrics

Model	F1 Score	Hamming loss	Weighted precision	Weighted recall	Balanced Error Rate
KNN	0.233	0.185	0.243	0.227	0.493
Decision Tree	0.377	0.145	0.405	0.39	0.472
SVM	0.514	0.125	0.609	0.495	0.499
XGBoost	0.423	0.130	0.490	0.462	0.464
Multinomial Naive Bayes	0.331	0.126	0.433	0.308	0.466
Logistic Regression	0.323	0.393	0.270	0.498	0.467
Popularity	0.270	0.170	0.191	0.483	0

Table 2: F1 scores on the Ablation experiment on XG Boost

Release year	Rating	Number of Votes	Title	Dialogues	Characters
0.232	0.197	0.187	0.231	0.233	0.221

the SVM to model the best possible set through a grid search, but a higher intensive CPU might aid in performing a grid search on a larger set of hyper parameters. Ablation experiments also implied that number of votes and rating were more crucial to this predictive task.

We believe there is potentially more work to be done with this dataset and the predictive task. Although we had done ablation task to try to filter out features that were not contributing greatly to the predictive task, we were unable to run grid search on most of our models to find the most optimal hyperparameters. Word2vec is another approach that could be utilized along with an n-gram model to provide context that could potentially improve the efficiency of said predictive task. We intend to do these experiments with a machine that could take a larger CPU load, as these experiments were not feasible on the modern day laptops.

REFERENCES

[1] [n. d.]. Predicting Genres from Movie Dialogue. https://towardsai.net/p/l/predicting-genres-from-movie-dialogue

[2] Vikas Ganjigunte Ashok, Song Feng, and Yejin Choi. 2013. Success with Style: Using Writing Style to Predict the Success of Novels.

[3] Rafael E. Banchs. 2012. Movie-DiC: a movie dialogue corpus for research and development. In *In Proc. of the 50 th Annual Meeting of the ACL*.

[4] Gabriel Barney and Kris Kaya. 2019. Predicting Genre from Movie Posters.

[5] A M Blackstock and Matt Spitz. 2008. Classifying Movie Scripts by Genre with a MEMM Using NLP-Based Features.

[6] Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs.. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

[7] Quan Hoang. 2018. Predicting Movie Genres Based on Plot Summaries. *CoRR* abs/1801.04813 (2018). arXiv:1801.04813 http://arxiv.org/abs/1801.04813

[8] Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. 2019. Way Off-Policy Batch Deep Reinforcement Learning of Implicit Human Preferences in Dialog. arXiv:1907.00456 [cs.LG]

[9] Kaggle. 2017. The Movies Dataset. https://www.kaggle.com/rounakbanik/the-movies-dataset/kernels

[10] Merja Kytö and Terry Walker. 2006. *Guide to A Corpus of English Dialogues 1560-1760*. Acta Universitatis Upsaliensis. http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-19360

[11] Lasguido Nio, Sakriani Sakti, Graham Neubig, Tomoki Toda, and Satoshi Nakamura. 2014. Conversation dialog corpora from television and movie scripts. In *2014 17th Oriental Chapter of the International Committee for the Co-ordination and Standardization of Speech Databases and Assessment Techniques (COCOSDA)*. 1–4. https://doi.org/10.1109/ICSDA.2014.7051436

[12] Anindya Roy, Camille Guinaudeau, Hervé Bredin, and Claude Barras. 2014. TVD: A Reproducible and Multiply Aligned TV Series Dataset. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. European Language Resources Association (ELRA), Reykjavik, Iceland, 418–425. http://www.lrec-conf.org/proceedings/lrec2014/pdf/751_Paper.pdf

[13] Iulian Vlad Serban, Ryan Lowe, Peter Henderson, Laurent Charlin, and Joelle Pineau. 2017. A Survey of Available Corpora for Building Data-Driven Dialogue Systems. arXiv:1512.05742 [cs.CL]

[14] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: Masked Sequence to Sequence Pre-training for Language Generation. arXiv:1905.02450 [cs.CL]

[15] Di Wang, Nebojsa Jojic, Chris Brockett, and Eric Nyberg. 2017. Steering Output Style and Topic in Neural Response Generation. *CoRR* abs/1709.03010 (2017). arXiv:1709.03010 http://arxiv.org/abs/1709.03010