NLP Review 1

Team 4

Afzal Mukhtar	PES2201800675
Hritika Rahul Mehta	PES2201800024
Farheen Zehra	PES2201800651

Problem statement

Our problem statement is to “Analyse a comedian’s transcript and produce a similar script”.

Text Generation in our main goal for this project

Literature Survey
Paper 1

- **Published in** Elsevier 2018
- **Title -** Customizable text generation via conditional text generative adversarial network
- **Authors -** Jinyin Chen, Yangyang Wu, Chengyu Jia, Haibin Zheng, Guohan Huang
- **Description -** In this paper they propose a conditional text generative adversarial network(CTGAN) in which emotion label is taken as an input to specify the output text using variable length text generation. The CTGAN model has two modules - θ parameterised conditional generator and ϕ parameterised conditional discriminator. The initial texts generated by the CTGAN are modified to further match the real scene by an automatic word replacement strategy that extracts keywords like nouns from the training texts. To test the effectiveness of the model several datasets like the Yelp restaurant reviews and Amazon review dataset were used.

Literature Survey
Paper 1

- **Advantages -** There are many text generation methods which have been proposed like Markov chains and RNN but they are plagued with challenges to output a variable length text or a text different from the input sentence. The model mentioned in this paper overcomes the above hurdles.They have used two evaluation metrics which can broadly be classified into statistic based evaluation metrics and unstatistic based evaluation metrics. Based on these metrics the performance of the proposed CTGAN model is tested against other model like Markov Chain and Seq2Seq models and the given model performs better than all the other models considered.
- **Disadvantages -** When using a mixed evaluation metric to evaluate various text generators they came to a conclusion that text generated by the CTGAN is most difficult to be identified.

Literature Survey
Paper 2

- **Published in** ACM 2016
- **Title -** Smart Reply: Automated Response Suggestion for Email(Reference of paper 1)
- **Authors -** Anjuli Kannan, Tobias Kaufmann, Karol Kurach, Andrew Tomkins, László Lukács, Vivek Ramavajjala, Sujith Ravi, Balint Miklos, Marina Ganea, Greg Corrado, Peter Young
- **Description -** In this paper a new end-to-end approach for automatically replying to emails is described. The system proposed generates semantically diverse replies that are accessible by a single tap. This method is being used in Gmail as of now. A large scale deep learning network was used which contained LSTMs.The triggering module is the entry point into the Smart reply framework .It helps identify if an email can have short replies, a few exceptions include open ended emails and promotional emails. Data preprocessing methods include language detection(non english emails are discarded),tokenization,sentence segmentation,normalization and quotation removal.

Literature Survey
Paper 2

- **Advantages -** This paper introduced a new method for semantic clustering of user-generated content which requires a less amount of labelled data.Email,being one of the most popular modes of communication has led to an overload in the inbox of every individual. The given system helps suggest automated,short replies to users. Replies suggested by this model have an inherent diversity which leads to increased usage.
- **Disadvantages -** Every user sees only one response of an intent. An intent is a cluster of responses that convey a particular emotion. LSTMs which are an important component of this model have a strong tendency towards positive responses. In order to overcome this sometimes two passes of the LSTM must be done. Given that 10% of mobile responses utilise this system we can conclude that it is not widely used.

Literature Survey
Paper 3

- **Published in** ACM 2020
- **Title -** Diversity regularised auto encoders
- **Authors -** Hyeseon Ko, Junhyuk Lee, Jinhong Kim, Jongwuk Lee, Hyunjung Shim
- **Description -** The authors propose a new powerful text generation model in their paper, called Diversity Regularised Autoencoders. They talk about the other autoencoders that exist already and point out the issues they face. In Variation-Autoencoders has a challenge of collapsing posterior loss, which is also known as the Kullback-Leibler vanishing problem. The training is aimed at minimizing reconstruction loss. The authors have used the WGAN-GP approach to improve the training stability of the GAN and thereby improving the quality of the text generated. The noise injection strategy is applied to the encoder.The authors take four cases of noise injection – insertion, deletion, substitution, and masking.

Literature Survey
Paper 3

- **Advantages -** This paper modifies the existing model, which enhances the diverse nature of texts that are generated. The model maintains the readability and the grammar on noise injection, thereby it learns to correct grammar along with the training procedure too.
- **Disadvantages -** The model doesn't have any drawbacks per se, but in real life, the text is usually repeated a few times. The diversity in the model is great for learning the grammar and producing natural-looking texts, but too much diversity is not common in a normal text.

Literature Survey
Paper 4

- **Published in** Association for the Advancement of Artificial Intelligence (AAAI)
- **Title -** Improving Variational Encoder-Decoders in Dialogue Generation (Reference Paper)
- **Authors -** Xiaoyu Shen, Hui Su, Shuzi Niu, Vera Demberg
- **Description -** The authors have talked about improving an already present model for text generation, Variational Encoders-Decoders . They have written about the different types of models that have been used previously, and the issues they've faced. They also mentioned the KL-vanishing problem that the previous VAE models faced. In their paper, they have divided the training into two parts, the first part is to learn to encode discrete texts into continuous word embeddings, and the second part utilizes these word embeddings to learn the generalization of the latent representations. They split the model into a CVAE module and an AE module. The CVAE learns to generate the latent variables whereas the AE module builds the connection between them and the dialogue. Most of the models have a KL-vanishing problem because the RNN decoder is a universal function approximator and it tends to represent the distribution without the latent variables. The CVAE is less accurate than a GAN, theoretically as it needs to approximate the real posterior. The authors leverage a more powerful RNN encoder-decoder with this. The AE phase autoencoder the utterances to make the real posterior representable easily by the CVAE part. The experiment was performed in two datasets – the Daily dialogue and Switchboard.

Literature Survey
Paper 4

- **Advantages -** This model is good in maintaining the context and doesn't face issues with long sentence generation.
- **Disadvantages -** This model drops its performance if the BOW size increases. Even though it reduces the KL-vanishing problem, the performance to be in context reduces.

Literature Survey
Paper 5

- **Published in**- 2016 IEEE/WIC/ACM International Conference on Web Intelligence
- **Title -** Automatic Generation of Restaurant Reviews with LSTM-RNN
- **Authors -** Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, Dennis Morello, Fabiano Tarlao
- **Description -** The paper demonstrates the generation of machine generated reviews which are different from genuine reviews. The paper describes related work to generate text using ANN and RNN and their disadvantages like higher computational cost.The dataset collected is composed of 2,169,264 reviews distributed over 66,700 restaurants.

  They proposed a method to generate reviews given a restaurant category and rating. Their method involves 3 steps: (i) a generative phase based on a LSTM character-level recurrent neural network [They first train the network to predict the probability of the next token for a fixed-length sequence of tokens given as input—a token being a single character] (ii) a category classification phase [using Naives Bayes classifier] and (iii) a rating classification phase [pick similar sentences and assign a random R]. Two methods of evaluation used-extrinsic evaluation to assess the ability of fake reviews generated to influence the decision of a user. [Forms were set up-with name,categories,reviews(at least one fake and one genuine) of the restaurant]. Intrinsic evaluation to evaluate the ability of a human user to discriminate between genuine and generated reviews.[Forms with name of restaurant and 5 reviews were tested against 39 users]. The results show that about 30% of reviews generated using their method are considered useful by human users.

Literature Survey
Paper 5

- **Advantages -**

➔ The model used is LSTM which solves the problem of long term dependencies i.e. vanishing gradient.
➔ The authors claim that no other paper exists for automatic generation of product reviews.
➔ The authors try to involve human users for evaluation who could better assess their decision than a machine.
➔ A diverse dataset distributed over thousands of restaurants is used.

- **Disadvantages -**

➔ Naive Bayes classifier (for classification of type) indicates strong independent assumptions only- 1-grams, 2-grams and 3-grams words are considered. Instead a generative model can be used.
➔ The number of users considered for evaluation is less. Decisions may vary if a larger sample of users was considered.
➔ The authors focus only on generation of textual content of reviews.They could have considered generating reviews using other features like user activities.
➔ Ratings are assigned randomly to machine generated reviews.

Demonstration

We will demonstrate the following:

- Data Gathering
- Data Cleaning
- Exploratory Data Analysis

Contributions

Cleaning - Pass 1

Afzal Mukhtar
Farheen Zehra

Cleaning - Pass 2

Afzal Mukhtar
Hritika Rahul Mehta
Farheen Zehra

Final Dataset & Visualization

Afzal Mukhtar
Hritika Rahul Mehta

Thank you!

Standup Comedian Transcript Analysis and Generation

An NLP Project By Group 4

Group Members

Group 4

Afzal Mukhtar

PES2201800675

Hritika Rahul Mehta

PES2201800024

Farheen Zehra

PES2201800651

Problem Statement

What we want to solve

Generating a similar text for each standup artist, using Deep Learning Models

Approach

Using Deep Learning Models and testing different parameters to get the best generative model.

Overview of Research Methods

The research was done on various aspects of the generative models for Natural Language Processing.
Researching helped us understanding which model will be useful for our use-case.
The papers referred were:

- **Customizable text generation via conditional text generative adversarial network [Elsevier 2018]:** Jinyin Chen, Yangyang Wu, Chengyu Jia, Haibin Zheng, Guohan Huang
- **Smart Reply: Automated Response Suggestion for Email [ACM 2016]:** Anjuli Kannan, Tobias Kaufmann, Karol Kurach, Andrew Tomkins, László Lukács, Vivek Ramavajjala, Sujith Ravi, Balint Miklos, Marina Ganea, Greg Corrado, Peter Young
- **Diversity regularised auto encoders [ACM 2020]:** Hyeseon Ko, Junhyuk Lee, Jinhong Kim, Jongwuk Lee, Hyunjung Shim
- **Improving Variational Encoder-Decoders in Dialogue Generation (Reference Paper) [AAAI]:** Xiaoyu Shen, Hui Su, Shuzi Niu, Vera Demberg
- **Automatic Generation of Restaurant Reviews with LSTM-RNN [IEE/WIC/ACM 2016]:** Alberto Bartoli, Andrea De Lorenzo, Eric Medvet, Dennis Morello, Fabiano Tarlao

Data Collection Methods

Scraping and Cleaning

Scraping

The data for the transcripts for each artist was scraped from different websites using web scraping methods and APIs

Cleaning

The Data was cleaned to remove unnecessary words, emojis, emoticons, and expansion of contractions. They were further lemmatized ofr text normalization.

Data Preperation

The data was changed into a word to vector representation for word level text generation.

Project Synopsis

Model and Results

Model Used

An LSTM model was used with input sequence of 100 Words, where a one hot encoded Embedding was used. The softmax function was used as the output layer and a cross-entropy loss function was used for errors.

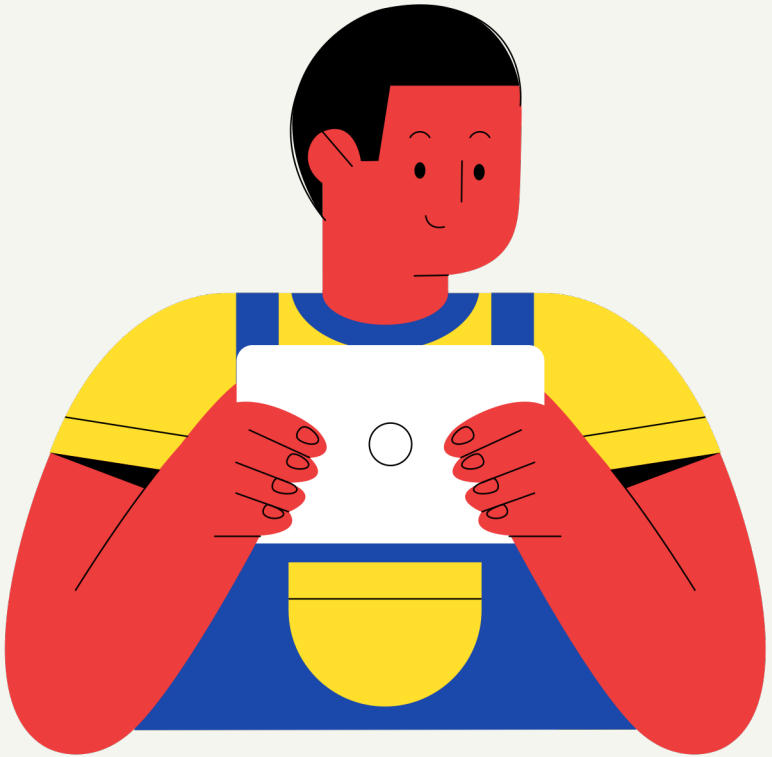
Training Data

The entire transcript of each artist was combined into one and passed to the model for training.

Results

Three different models were built. One with stop words, Second without stop words and Third with stop words and drop-out layers.
In all three the one with drop-out generalised better and had an overall lowest perplexity value.

Project Demo



LSTM Model

Model Architecture and Training Parameters

```
def build(self):
    model = Sequential()
    model.add(Embedding(self.vocab_size, 100, input_length=self.seq_length))
    model.add(LSTM(128, return_sequences=True))
    model.add(LSTM(128))
    model.add(Dense(128, activation='relu'))
    model.add(Dense(self.vocab_size, activation='softmax'))
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    self.model = model
    self.model.summary()

def fit(self, X, y, batch_size=256, epochs=100, validation_split=0):
    callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)
    self.history = self.model.fit(X, y, batch_size=batch_size, epochs=epochs,
                                  validation_split=validation_split, callbacks=[callback])
```

Model Results

Model without Stop Words

Model without Stop Words

Input To Model

and there is one guy who will come up with like i got 95 i do icse we will pay for your bill just icse who did icse oh my god you guy are sorted for life okay while we were learning addition and subtraction you were learning how to send mar rover i do not know what retarded curriculum they have of my god scary and obviously in kv they do a lot of torture technique they slowly break you down a a child the first thing is called morning assembly morning assembly is a phenomenon where you put kid

Generated Sentence

purse bluff did yes he supporting no mundu since car then two anything check funny horrible check doing madam want 2 it will grey these keep would cook good expensive raw but napkin expensive pain do round seen ill she ironing little whatever girl somewhere is element silent overconfident cameraman

Model Results

Model with Stop Words

Model with Stop Words

Input to the Model

and there is one guy who will come up with like i got 95 i do icse we will pay for your bill just icse who did icse oh my god you guy are sorted for life okay while we were learning addition and subtraction you were learning how to send mar rover i do not know what retarded curriculum they have of my god scary and obviously in kv they do a lot of torture technique they slowly break you down a a child the first thing is called morning assembly morning assembly is a phenomenon where you put kid

Generated Sentence

in the sun roast them to light medium brown make sure they turn brown and they make the guy stand in ascending order of insecurity shortest least self confident guy go in the first genetically gifted tall guy who is good looking will do well in life in the back

Model Results

Model with Stop Words and Dropout

Model With Stop Words and Dropout

Input To Model

and there is one guy who will come up with like i got 95 i do icse we will pay for your bill just icse who did icse oh my god you guy are sorted for life okay while we were learning addition and subtraction you were learning how to send mar rover i do not know what retarded curriculum they have of my god scary and obviously in kv they do a lot of torture technique they slowly break you down a a child the first thing is called morning assembly morning assembly is a phenomenon where you put kid

Generated Sentence

in the sun roast them to light medium brown make sure they turn brown and they make the guy stand in ascending order of insecurity shortest least self confident guy go in the first genetically gifted tall guy who is good looking will do well in life in the back

Model Evaluation

Model without Stop Words

Unigram Perplexity

```
PP(  animal  )    :    151.0000
PP(   sort   )    :    151.0000
PP(    a     )    :    50.3333
PP(admitting )    :    151.0000
PP(    in    )    :    75.5000
PP(  where   )    :    151.0000
PP(   have   )    :    75.5000
PP(  insane  )    :    151.0000
PP(  there   )    :    151.0000
PP(   can    )    :    30.2000
```

Bigram Perplexity

```
PP(  animal  )    :    12.2882
PP(   sort   )    :    12.2882
PP(    a     )    :    7.0946
PP(admitting )    :    12.2882
PP(    in    )    :    8.6891
PP(  where   )    :    12.2882
PP(   have   )    :    8.6891
PP(  insane  )    :    12.2882
PP(  there   )    :    12.2882
PP(   can    )    :    5.4955
```

Model with Stop Words

Unigram Perplexity

```
PP(   say    )    :    75.5000
PP(   that   )    :    50.3333
PP(    my    )    :    50.3333
PP(  friend  )    :    75.5000
PP(   like   )    :    18.8750
PP(    me    )    :    37.7500
PP(   and    )    :    151.0000
PP(  gopal   )    :    151.0000
PP(   have   )    :    75.5000
PP(   been   )    :    151.0000
```

Bigram Perplexity

```
PP(   say    )    :    8.6891
PP(   that   )    :    7.0946
PP(    my    )    :    7.0946
PP(  friend  )    :    8.6891
PP(   like   )    :    4.3445
PP(    me    )    :    6.1441
PP(   and    )    :    12.2882
PP(  gopal   )    :    12.2882
PP(   have   )    :    8.6891
PP(   been   )    :    12.2882
```

Model with Stop Words and Dropout

Unigram Perplexity

```
PP(   say    )    :    75.5000
PP(   that   )    :    50.3333
PP(    my    )    :    50.3333
PP(  friend  )    :    151.0000
PP(   is     )    :    37.7500
PP(    me    )    :    75.5000
PP(   and    )    :    151.0000
PP(   then   )    :    151.0000
PP(vegetarian)    :    151.0000
PP( version  )    :    151.0000
```

Bigram Perplexity

```
PP(   say    )    :    8.6891
PP(   that   )    :    7.0946
PP(    my    )    :    7.0946
PP(  friend  )    :    12.2882
PP(   is     )    :    6.1441
PP(    me    )    :    8.6891
PP(   and    )    :    12.2882
PP(   then   )    :    12.2882
PP(vegetarian)    :    12.2882
PP( version  )    :    12.2882
```

Model Evaluation

Rouge-2 Metrics

```
sentence                    : 0.3684
sentence_with_stopwords     : 0.3543
sentence_with_dropout       : 0.2294
```

Cosine Similarity

```
Sentence similarity:                        : 0.7111
Sentence similarity with stopwords:         : 0.6633
Sentence similarity with dropout:           : 0.5575
```

Thankyou

And Future works

Trying different Embeddings

Trying different models

Training on a Larger Dataset