EGAN: Generatives Adversarial Networks for Text Generation with Sentiments

Andres Pautrat-Lertora, Renzo Perez-Lozano and Willy Ugarte[a]
Universidad Peruana de Ciencias Aplicadas (UPC), Lima, Peru

Keywords: GAN, Text Generation, NAO.

Abstract: In these last years, communication with computers has made enormous steps, like the robot Sophia that surprised many people with their human interactions, behind this kind of robot, there is a machine learning model for text generation to interact with others, but in terms of text generation with sentiments not many investigations have been done. A model like GAN has opportunities to become an excellent option to attack this new problem because of their discriminator and generator competing for search the optimal solution. In this paper, a GAN model is presented that can generate text with different emotions based on a dataset recompiled from tweets labeled with emotions and then deployed in an NAO robot to speak the text in short phrases using voice commands. The model is evaluated with different methods popular in text generation like BLLEU and additionally, a human experiment is done to prove the quality and sentiment accuracy.

1 INTRODUCTION

Text generation is a stringent computational task that last years have has many utilities like improvements in virtual assistants and Human-robot interaction(HRI) with more elaborate dialogues. However, the text generated is not totally accurate and presents not realistic phrases as (Huszar, 2015) says, this includes that if the text generated is more extensive the problems will be more frequent. Even though the objective is the human interaction, the sentiments are not usually considered which in a real conversation is usually a really important topic. Nowadays there are many good models with the main function of text generation, some of these can be transformers, that don't consider a sentiment but can be modified, or the best known model GPT3 that have different uses.

Many people have incorporated voice assistants as a tool in their daily lives to control appliances, play multimedia products, create notes or reminders, etc. The relevance can be noticed in (Newman, 2019) where indicates that on 2018 in the United States 14% of adults regularly use one of these devices, while in the United Kingdom it is 10%. Also, the addition of sentiments will bring a better experience to the user, can be more personal and have benefits as explained in[1] like recognizing the feeling and answering something depending on the user emotion.

The generation of coherent text is always a hard task, all languages have a structure, grammatic and correct order to be understood, furthermore, inside every sentence, many topics can be mentioned and it needs logic to don't jump from one to other without context. For that, a good text generation model needs to catch many attributes of the already existing sentence and then process them to generate more content. Adding an emotion to the generation makes this work even more problematic, there will be more attributes to process and each word has to consider the sentiment of the previous text to continue the task. Originally the GAN architecture just have 2 models inside the generator and the discriminator to evaluate how realistic the samples of the generation are, but in our model, there is another parameter, the sentiment, that needs to be evaluated, for this, we use a third model for the task and specialize that model on sentiment analysis and leave the first one for its original task.

There are different solutions made for the text generation that uses a model based on LSTM (Long short-term memory), like (Cai et al., 2021), which is a variation of an RNN (Recurrent neural network), this is used because this kind of network work with a se-

[a] https://orcid.org/0000-0002-7510-618X

[1]Emotion AI will personalize interactions - Gartner - https://www.gartner.com/smarterwithgartner/emotion-ai-will-personalize-interactions

Pautrat-Lertora, A., Perez-Lozano, R. and Ugarte, W.
EGAN: Generatives Adversarial Networks for Text Generation with Sentiments.
DOI: 10.5220/0011548100003335
In *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K 2022) - Volume 1: KDIR*, pages 249-256
ISBN: 978-989-758-614-9; ISSN: 2184-3228

quence of data instead of single data values, making this perfect to generate new content from a context like a sentence, some models that use this method are (Li et al., 2018). But the model mentioned before only can work if they have a big amount of data which is not always the scenario, so the GAN (generative adversarial networks) are used because can have a good performance with a smaller dataset which can be seen in models like (Rizzo and Van, 2020). On the other hand, some investigators have tried to go further away to make a generative model and add a sentiment label (Li et al., 2018), with the time many works are done to improve the text generation to make it indistinguishable from a real text from a human.

We develop a GAN based model, which is composed for two discriminators that are based in a convolutions network, one to recognize if the text evaluated is true or fake and one that determines the emotion of the text by extract each characteristic of the sentence. Finally, a generator model based on LSMT trains with the output of the other two models trying to make them fail with the text generated.

Our contributions are as follows:

- We develop a multilabel text generation with sentiments.
- We build a implementation of a GAN model for text generation with sentiments.
- We make a analysis and comparison of our model with other similar.

In Section 2, solutions with similar subjects will be compared to ours. In Section 3 we explain the different architectures and algorithms related to GAN models and text generation and then we present the structure of the algorithm developed and the contributions made. In Section 4, the experiments and results will be explained and detailed. Finally, the conclusion will be presented in section 5.

2 RELATED WORKS

Many projects have proposed different solutions using their own implementation of GAN with details that make unique models. We use some models as inspiration for our implementation some of these can be found next.

The model proposed on (Wu and Wang, 2020) named TG-SeqGAN is based on SeqGAN(Yu et al., 2017) with an addition of Truth-Guilded method to make it closer to the real data. This model includes an initial state where a transfer model and a cost function, defined as the distance between the generated text and the real text, are used to find the next value quicker with a CNN model for extract all characteristics and context of words. In our model, we use a CNN model as the discriminator, and an additional classificator model is added to evaluate the sentiment and generate sentences with a specific input.

In (Li et al., 2018) a novel framework is proposed, where the author train a GAN model for text generation with categories based on SeqGAN(Yu et al., 2017), with a LSTM as generator and a RNN model for the discriminator for veracity and clasificator for the category. Inspired by this work, we have implemented an LSTM model as our generators and separated the discriminator and classificator but for the last two models, we have used a CNN to obtain more accurate predictions.

The model CD-GAN for text generation proposed on (Yan et al., 2021) use an LSTM generator and a CNN as a discriminator-augmented, this means that their discriminator evaluates each word of the sentence individually and founding the incoherence in the sentences with the objective of avoiding the pretraining. This is a novel technique, the CNN used has a great performance on sequence classification but we consider that the individual word analysis is too complicated and doesn't have enough benefits so we use only one classification for the whole sentence.

EmoSen (Firdaus et al., 2020) is a framework end-to-end with the job of generating dialogs, and controlling the sentiment as happy, sad, angry, etc, and the feeling as positive, negative, and neutral, this is approached by analyzing reference text, audio, and visual helpers. This model can manage a lot of sentiments, feeling,s and contexts because of the used train dataset Sentiment and Emotion aware Multimodal dataset (SEMD), for this the dialogue system is robust. Compared with our proposed model we use the controlled training for sentiments on the text, but the feeling as positive, negative, and neutral is not considered like this project instead of these, we present six feelings, enjoyment, sadness, love, anger, surprise and fear..

Finally, on the work made on (Chen et al., 2020), CTGAN model is proposed based on SeqGAN, it has the addition of being able to generate text with variable length and a label of sentiment as positive, negative, and neutral. Also, an algorithm of word replacement is used to guarantee the quality of the generated text in a specific context. In this project, the discriminator is used to evaluate if the text is real or generated and if the sentiment is accurate, on our project these functions are separated into two models to make each one specialized on evaluate if the text is real or fake and if the sentiment is accurate, since that if use a one model that describe if is real or fake and add the sentiments will have many parameters that It will loss in the training.

3 METHOD

In this section we present the main concepts and architecture for our proposal.

3.1 Preliminary Concepts

The text generation with GAN models for human and robot interaction has many challenges in the development process. The adaptation and modification of GAN models to generate text classified with a sentiment label is the main problems, but this brings other problems like how to generate the text, how to identify the sentiment, and make the interaction with the user. In this section, the necessary background will be explained.

3.1.1 Text Generation

Text generation is a subarea of natural language processing, so it acquires knowledge of the computational area of language and artificial intelligence, the objective is to generate coherent and readable text. To make this task easier for the computer the words use to be tokenized giving each word on a dictionary a corresponding number and transforming the inputs sentences into a new format like one hot where an array of the size of the dictionary is replaced with ones and zeros depending on the words on the sentence. There are many models that can be used for this objective, one that performs well in many projects like (Cai et al., 2021), are the long short-term memory (LSTM) which can process many data at the same time, which means that can manage multiple words at the same time. One of the main characteristics of LSTM is the states it remembers over time and uses this information for the followings generations, this is done by implementing loops and gates within the model, as can be seen in Fig. 1.

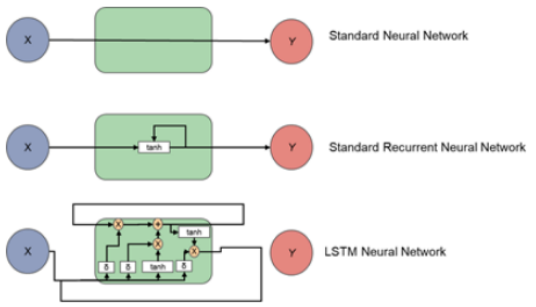

Figure 1: LSTM model (Zia and Zahid, 2019).

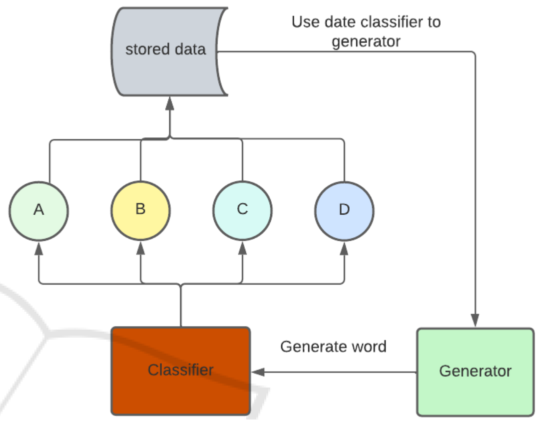

Figure 2: Sentiment analysis in training own elaboration.

3.1.2 Sentiment Analysis

According to EmoShape[2], the sentiment analysis refers to the fact that the machine can understand the feeling that the user wants to express, this can be by image recognition, speech recognition, or text analysis. When it comes to a text format, the task becomes a natural language processing job, whereas machine learning or deep learning techniques are needed. As mentioned when text has to be processed tokenization is really important and LSTM models make a good job like paper (Hochreiter and Schmidhuber, 1997), but other options like gated recurrent unit (GRU) and convolutional neuronal networks (CNN) are good too like (Liu et al., 2020) where a CNN model is used to identify the sentiment on a text other model that for use this criterion is SentiGAN (Wang and Wan, 2018) in where use multiple generators and one multi-class discriminator, to address the above problems. Since, yours multiple generators are trained simultaneously, aiming at generating texts of different sentiment labels without supervision.

The generator model passes the words in integers and the classifier identifies which of the sentiments they belong to. Then the words considering the labels re-enter the generator until they can enter the generator and the classifier consecutively Fig. 2.

3.1.3 Generative Adversarial Networks(GAN)

GAN models were proposed on (Goodfellow et al., 2014) and are based on having two models that compete with each other, a generator of content, and a discriminator that is responsible for checking if the content evaluated is generated by the first model or is

[2]EmoShape: Emotion Synthesis for Metaverse - https://emoshape.com/

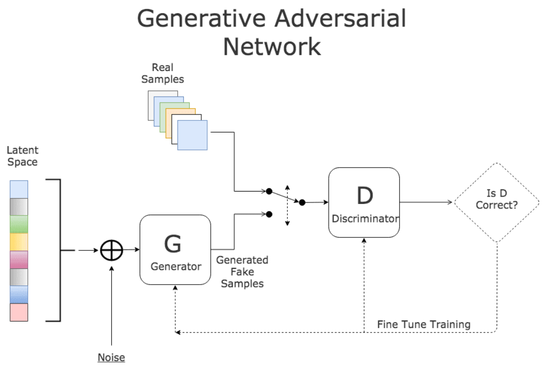

Figure 3: GAN arquitecture (Goodfellow et al., 2014).

real data from the dataset used, the structure of this first architecture is presented on the Fig. 3. Goodfellow develops the GANs to generate images, but because of their good results these models gain a lot of popularity and other researchers start using this model for the same purpose or adapting it to other investigation fields. One of the first investigations published that approached the GAN model for the text generation was (Firdaus et al., 2020) where an adaptation of the GAN model for images is used to generate sequences instead of images, making possible to generate text or music composition. Other paper have use gan and is DGSAN (Montahaei et al., 2021) in where is a model that upgrade of before model since that fix the gradient step problem and used 2 model in same network and subsequently generate text, obtaining results greater than 90 percent in BLEU 3.

3.1.4 Human Robot Interaction (HRI)

This refers to the study of any interaction between humans and robots. In the last years, the interactions with robots have been improved, like the example of Sophia where the objective is o make a robot similar to the human appearance to interact with humans. One area of the HRI is verbal communication that can be translated into text, here artificial intelligence is needed to create coherent and understandable interactions. Interaction between machines and humans a better experience for the user can be reached by adding emotional analysis and proper response to his feeling (see Fig. 4).

3.2 Main Contribution

In this work of text generation based on sentiments, we proposed a GAN model with a discriminator model to evaluate the veracity of the text and classifier model to evaluate the sentiment of the sentence with the objective of training a text generator model

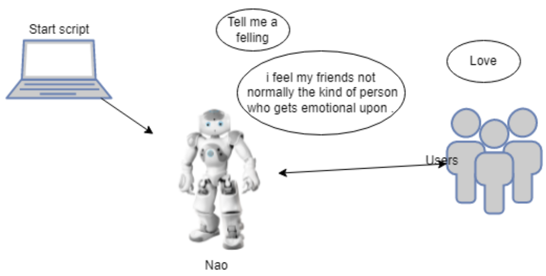

Figure 4: NAO robot interaction.

that can receive a sentiment and generate a coherent sentence corresponding to it.

3.2.1 GAN for Text Generation

In this section, we explain, the structure and function of the different models used inside the GAN which are the generator, discriminator and classification. This GAN model is based on different models as CatGAN (Liu et al., 2020) and SeqGAN (Firdaus et al., 2020), adding a second discriminator for sentiment analysis in the first and tuning the model parameters and structures to obtain better results in both cases. For other hand, this structure uses thresholds to train the discriminator with sentences with noise to make a better discriminator and classificator

Before working with a sentence, each one is transformed with a dictionary from words to integers and are codified with one hot method to be easier to manage them inside the models.

Discriminator. The objective of this model is to be able to differentiate between generated sentence and a original ones, then use the results a make a train step on the generator. Once this is done, the generator will be able to generate better sentences to make the discriminator wrong.

This model is based on a CNN, the model structure is presented in the Fig. 5. The first layer is an embedding layer as usually made on natural language processing(NLP), this help with the large input vector before the one-hot codification to be easier to manage for the model. Next, we use four 2D convolutional layers separately with one input channel and 300 output channels to extract features from the sentence and then make a max pooling to each. The result is concatenated and applied on a linear layer of input and output size of 1200 to evaluate the features extracted by the convolutional layers. Then an activation function is applied where x is the result of the previous linear layer and the function $f(x)$ can be expressed as following:

$$f(x) = \frac{1}{1+e^{-x}} \times max(0,x) * (1 - \frac{1}{1+e^{-x}})$$

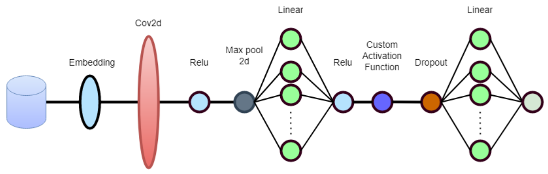

Figure 5: Discriminator Structure own elaboration.

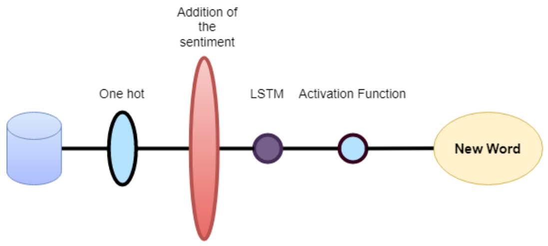

Figure 6: Generator Structure own elaboration.

Classifier. The purpose of this model is to evaluate sentences and classify the sentiment present in them, and this information is used to train the generator too. This model has a similar function to the discriminator but with more labels, although both tasks of the discriminator and classifier can be done by one model, separating these tasks into single specialized models ensures the efficiency of each one on its work and result in a better train for the generator model.

Because of the similar objective between the discriminator and the classification the structure used is the same, just variation the result depending on the number of sentiments that are being evaluated.

Generator. The generator is the main model to train, the discriminator and classificator work is done to help with the training on the generator. This model has the job of receiving a sentiment and part of the sentence if it already exists and generate the next word on the sentence or a dot to finish it. The objective of the training is that this model can generate coherent sentences that convey the feeling given.

The model is based on an LSTM model because of its efficiency in generating text, architecture developed can be found on Fig. 6. Like the previous models, the first layer is an embedding model with the input size equal to the dictionary size and an output of 32. Then the label of sentiment is added and the information is passed to the LSTM layer with an input and output size of 32. Finally a linear layer an array with the size of the dictionary and a softmax function determinates the next word of the sentence.

Training. For the training, on each step of the training the sentences are transformed from words to integers with a dictionary of words as $X = [x_0, x_1, \ldots]$ and a one-hot encoding transforms it to an array of the size of the dictionary with ones and zeros depending on the words on the sentence. Before starting with the GAN training each model is pre-trained with only the real data, this gives the models an initial state to not start generating and evaluating sentences randomly. The final GAN train has 4 steps, first, a sentence is generated, second, the discriminator and classificator evaluate the sentence, third the fit of the 3 models is done and finally the discriminator and generator train with a real sentence. For the construction of the model, Pytorch is used, with a learning rate of 0.01 for the pre-training, GAN train of 0.0001, and a batch size of 8. The training was performed with 150 pretrain epochs and 2000 GAN train epochs, that was done on 3 separate trains of 4 hours each with a Tesla P100.

3.2.2 Connection Nao Robot

For the connection with the NAO robot we needed to use its specific IP on the local red to send it commands with the library Naoqi. This library contains all function that can be used with the NAO robot and is only available in Python 2, but our model was developed on Python 3, so several connected scrips were made. In the first step, a subscription to ALTextToSpeech was necessary for the robot to say the instructions, and ALSpeechRecognition to recognize the user voice of what sentiment is indicated. In the second step, the sentiment is sent to Python 3 to generate the sentence in English with the correct label, in this part, it is necessary to clarify that we have to present the sentences in Spanish due to academic issues of our institution, so we used a model to translate but for security, the red where the NAO robot is connected has no internet so the use of translation with APIs was not possible, for this reason, We use the Argos model for offline translate from English to Spanish, this model used help of OpenNMT toolkit, SentencePiece for tokenization, Stanza for sentence boundary detection[3]. In the final step, the translated text is sent to a script on Python 2 and uses ALTextToSpeech to interpret the phrase generate.

4 EXPERIMENTS

4.1 Experimental Protocol

To recreate the experimentation process we will mention the hardware, dataset, parameters, and the validation of the results of our project.

[3]Open Tech - https://www.argosopentech.com/

4.1.1 Development Environment

The model training environment used as the main platform was Google Colab with the Pro subscription, this tier of the platform was mainly needed for the extended runtimes compared to the Free version, this service offers us a Tesla T4 or a Tesla P100 GPU and 25GB of RAM.

4.1.2 Dataset

The used dataset was the emotions dataset for NLP[4], found on Kaggle, this one has a long recompilation of sentences labeled with one of 6 emotions and a total of 16000 sentences. The dataset was pre-processed, symbols were deleted, sentiments were separated into groups of 700 sentences to make the same number on each sentiment and the number of sentences was reduced due to the resource given by Google Colab Pro were not enough to make the training with all the data.

4.1.3 Models Training

The model was developed using Pytorch and the training was realized on Google Colab Pro GPU, this one has a maximum runtime of 24 hours and sometimes less, to approach this issue every 20 epochs the state of the model was saved, and if the runtime ends a new one was generated manually, load the last state and continue with the training, finally the train was done on around 48 hours. For the parameters of the model, we train 500 epochs with a batch size of 8, vocabulary size of 15213, and generator, discriminator, and classificator learning rate of 0.0001.

4.1.4 Testing Environment

We have developed two environments for testing and validation, the first is a user interface (UI), and the second is an interaction with de NAO robot. The UI is a simple environment where the user can select a sentiment and generate a sentence based on it, then a button can translate the English text to Spanish or vice versa. The NAO robot interaction is the main testing environment where it makes a presentation of the interaction and tells the user to say a sentiment, next the robot sends this sentiment to the model to generate 10 sentences, these sentences are classified with the text discriminator from the GAN which tell us which one is the more realistic one, finally, the selected sentence returns to the robot to say the phrase.

[4]Emotions dataset for NLP - https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp?select=train.txt

4.2 Results

We have used BLEU(Papineni et al., 2002) to validate the text quality of the sentences generated by our model, comparing 48 example sentences with the 16000 of the dataset we obtain the quality of the text generated. This metric use the number of words to compare sentences, the metric will be more demanding if this amount of words is higher. We evaluate our models with BLEU-1, BLEU-2, BLEU-3, and BLEU-4, where the number means the number of words used, the result of this metric can be found in the table 1.

Another metric that we have used is Jaccard which evaluates 2 groups of data to compare the similarity of these ones, we compare the 48 examples generated to the 16000 sentences of the dataset as on BLEU and obtain 0.0966 as shown in table 3, this means that the similarity is pretty low to the dataset. This metric is usually presented with BLEU because an overfitted model can generate the same sentences as the dataset, which means BLEU will be really high, so Jaccard help to discard that a good score is the result of overfitting.

Other metrics were considered to be used but many were discarded because they considers parameters that our model was not supposed to accomplish. One of these metrics was METEOR (Banerjee and Lavie, 2005), this is an improvement of BLEU focused on evaluating text translation, its bases on evaluating the matching unigrams considering the surface forms of the sentence. This makes the similarity of the generated and evaluation text important, but in our model, we don't want a big similarity with the dataset so our results are bad on this metric because is focused on novelty, not similarity. Another metric we consider was ROUGE (Lin, 2004) which is used to evaluate text summarization and translation and is based on counting the overlapped n-grams or sequences of words, this makes the similarity of the texts evaluated really important and this is not relevant for our model as explained before.

4.3 Discussion

As shown in table 2 our model presents better performance on BLEU metric than some similar implementations but worse in other cases, for this is important

Table 1: BLEU Metrics.

Metrics	BLEU-3	BLEU-4	BLEU-5
EGAN	.8127	.6138	.4574

too to compare the Jaccard score too in table 3, so we can notice that despite the higher score on BLEU the text of their model is more similar to the dataset used by them. Our model presents a great alternative to other project implementations despite can presents lower quality text these is more novelty.

On the found results we have the DGSAN model (Montahaei et al., 2021) that is one of the models with the best results on the benchmarking shown on table 2, it has better results than us on BLEU-3 and BLEU-5, and this quality difference can be noticed on the table 3

Next, on table 4 text generated by different models are presented. Comparing the text shown we can notice some of them are not too coherent. For example, in the case of the model WRGAN they present a good text coherence in the reading, our text has less coherence but it's longer than WRGAN examples.

Also, we have used the NAO robot to make testing and validation, as a proposition of making better the interaction for the user we make the robot say an introduction and then received a voice command telling a sentiment so the model can generate sentences with it and finally answer with that sentence. We make some surveys where with a video the respondents evaluate from 0 to 5 the quality of the interaction, from this the average score was 3.67 so we can conclude that the interaction with the NAO robot with the implementation of our model was good.

Also, continue with the survey we have the qualified results on the feelings of the generated from the model shown on table 5 where the sentences of Sadness and Surprises are the two feelings that users perceive. On the other hand, analyzing the coherence of

Table 2: BLEU Metrics.

Metrics	BLEU-3	BLEU-5
EGAN(ours)	**.813**	**.457**
SeqGAN	.807	.419
DGSAN	.945	.728
DoubAN-Full	.095	.056
WRGAN	.634	.303

Table 3: Jaccard Metrics.

Metrics	Jaccard
EGAN(ours)	**.097**
SeqGAN	.140
DGSAN	.254

Table 4: Text Comparation of differents models.

Model	Text Generate
EGAN(ours)	• I go through the time i had too much more and feel that she asked why you feel is • I dont feel apprehensive and apprehensive among my feelings that i m feeling reluctant to post • I dont feel extremely worthless feeling so apprehensive among a bit
TILGAN	• I was driving my van to work one day • She bought some new books • He saw some band members
DoubAN-Full	• What did the appletalk system say? • What is the immune system of? • Where did the grand canal occur?
WRGAN	• Could use a little more humanity and delight • So boring and meandering • A pleasant, but it's also extremely effective

Table 5: Sentiment accuracy by survey.

Anger	Sadness	Love	Surprise
22.22	66.67	38.89	66.67

sentences with users indicates that 36 percent is bad, other 25 percent think that is normal and finally the 38.89 percent think it's excellent how to shown on table 6.

In conclusion, the diversity and quality of the text its really important to validate the results o text generation, metrics like BLEU and Jaccard can be really useful in these cases.

Table 6: Coherence accuracy by survey.

Bad	Normal	Excellent
36.11	25	38.89

5 CONCLUSIONS

Through the development of the project, the analysis of the metrics, and the results found in the other models, we concluded that our model has good text generation results, but it needs a high processing power to be trained. For this limitation, the model parameters were not the desired ones and that can be the cause for some incoherence in the generated text.

The CNN and LSTM models have provided a good performance on the GAN architecture for the text generation with sentiments. A benefit of using convolutional networks is that they are capable of feature extracting, this help to be more precise on the discriminator and classificator work. In the case of the LSTM generator, due to the information saved on each interaction on the generation, the text result has good coherence and quality.

A good upgrade to this work that can be done in the future, is the exchange of the internal models, similarly to GPT3 based models (de Rivero et al., 2021). Despite the good performance it presented, this can be improved, for example, by replacing the LSTM generator with a transformed-based generator or transfer learning from a CNN (Rodríguez et al., 2021).

REFERENCES

Banerjee, S. and Lavie, A. (2005). METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In *IEEvaluation@ACL*.

Cai, P., Chen, X., Jin, P., Wang, H., and Li, T. (2021). Distributional discrepancy: A metric for unconditional text generation. *Knowl. Based Syst.*, 217.

Chen, J., Wu, Y., Jia, C., Zheng, H., and Huang, G. (2020). Customizable text generation via conditional text generative adversarial network. *Neurocomputing*, 416.

de Rivero, M., Tirado, C., and Ugarte, W. (2021). Formalstyler: GPT based model for formal style transfer based on formality and meaning preservation. In *IC3K*.

Firdaus, M., Chauhan, H., Ekbal, A., and Bhattacharyya, P. (2020). Emosen: Generating sentiment and emotion controlled responses in a multimodal dialogue system. *IEEE Transactions on Affective Computing*.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. C., and Bengio, Y. (2014). Generative adversarial networks. *CoRR*, abs/1406.2661.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, 9(8).

Huszar, F. (2015). How (not) to train your generative model: Scheduled sampling, likelihood, adversary? *CoRR*, abs/1511.05101.

Li, Y., Pan, Q., Wang, S., Yang, T., and Cambria, E. (2018). A generative model for category text generation. *Inf. Sci.*, 450.

Lin, C.-Y. (2004). Rouge: a package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out of ACL*.

Liu, Z., Wang, J., and Liang, Z. (2020). Catgan: Category-aware generative adversarial networks with hierarchical evolutionary learning for category text generation. In *AAAI*.

Montahaei, E., Alihosseini, D., and Baghshah, M. S. (2021). DGSAN: discrete generative self-adversarial network. *Neurocomputing*, 448.

Newman, N. (2019). Journalism, media and technology trends and predictions 2018.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). Bleu: a method for automatic evaluation of machine translation. In *ACL*, pages 311–318.

Rizzo, G. and Van, T. H. M. (2020). Adversarial text generation with context adapted global knowledge and a self-attentive discriminator. *Inf. Process. Manag.*, 57(6).

Rodríguez, M., Pastor, F., and Ugarte, W. (2021). Classification of fruit ripeness grades using a convolutional neural network and data augmentation. In *IEEE FRUCT*.

Wang, K. and Wan, X. (2018). Sentigan: Generating sentimental texts via mixture adversarial networks. In *IJCAI*.

Wu, Y. and Wang, J. (2020). Text generation service model based on truth-guided seqgan. *IEEE Access*, 8:11880–11886.

Yan, Y., Shen, G., Zhang, S., Huang, T., Deng, Z., and Yun, U. (2021). Sequence generative adversarial nets with a conditional discriminator. *Neurocomputing*, 429.

Yu, L., Zhang, W., Wang, J., and Yu, Y. (2017). Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*.

Zia, T. and Zahid, U. (2019). Long short-term memory recurrent neural network architectures for urdu acoustic modeling. *Int. J. Speech Technol.*, 22(1).