Generation of undirected responses;
Learning models on Reddit data

Bauke Brenninkmeijer
S4366298
email: bauke.brenninkmeijer@student.ru.nl

Ties Robroek
S4388615
email: ties.robroek@student.ru.nl

Abstract

Forums such as Reddit are hugely popular right now. Some of these contain strong self-contained communities. In this research we have explored multiple ways of modeling these communities. We have made generative systems that can produce comments to the likeness of what can be found on these subreddits. We explored the capabilities of Markov Chains, RNNs and GANs. The models have been trained on several subreddits of which we chose one for more thorough testing. We have conducted a survey and found that our RNN model is capable of producing convincing sentences.

1. Introduction

Conversational modeling is an important subject in natural language processing and machine learning. More specifically, modeling sensible answers to arbitrary question has proven to be a challenge for a long time now. This project is about generating comments for the website Reddit, one of the biggest online forums to date. We wanted to achieve this by directly using Reddit data. We chose three subReddits (specific topic-bound subforums) which we deemed fit for the task. The generation was done in via multiple models. We have made a Markov chain model to start things off, followed by an RNN (Recurrent Neural Network) and a GAN (Generative Adverserial Network).

Section 2 details our choice of dataset and our choice of models. In section 3 our approach and implementation is explained. The results are presented in section 4.1 and we discuss possible improvements in section 5

2. Background

Our approach resembles that of creating a chatbot with a personality such as in [1]. We have also found inspiration from work by Li et al. [2, 3]. They have used adversarial networks and reinforcement learning for building models that can generate dialogue. Tests akin to the Turing Test were performed which we have also included following their example. We will be discussing our choice of dataset and models below.

2.1. Reddit

We have built several models that outputs comments with the "personality" of the average user of that subreddit. Many websites in the modern world report the activity of "trolls" and "bots". Reddit is no exception[4]. Automatically generated comments can provide a variety of uses, both malicious and helpful. Bots trained on a specific subreddit may be deployed on the very same subreddit to strengthen it's believes and "reinforce it's bubble". Perhaps counterintuitively, bots trained on more moderate subreddits may be able to reduce hate speech and such on extremist sections. Finally, as with any trained neural network, a bot may be able to somewhat accurately represent it's source. This may make it easier to control and monitor message boards. For our research we have limited our scope to three subreddits. The forums '/r/the_donald' and '/r/politics' are both highly political subreddits. Both subreddits are highly opinionated and show a high degree of cohesion. We have chosen these subreddits as we suspect they can be represented well in a limited term space. We have also included the subreddit '/r/askreddit' as a control group. This is a forum whose comments specifically answer the topic post. We expect our models to perform poorly on this data as the variation between topics can be used.

2.2. Markov chains

A Markov chain is a model describing a sequence of states. A state is solely determined by the previous state. We can generate a sentence by simply "walking" through the model. It is regarded as a memory-less model, looking at nothing but the current state. Being a quite simple model, it is often used as a good baseline to compare more advanced methods with. A notable example of Markov Chains in action is the subreddit '/r/subredditsmulator'[5] in which only Markov Chain-based bots are allowed to post.

2.3. Recurrent Neural Networks

Recurrent Neural Networks are a class of neural networks that are specifically meant for recursive data. Unlike feed-forward networks, RNNs have an internal state which stores information. The capacity to take older information into account is essential for temporal related tasks such as speech. Classifying a new piece of data is not just done by taking the data as input but by taking both the data as input and the previous state.

There are two general classes of RNNs which share a general structure: Finite impulse networks and infinite impulse networks. A finite impulse recurrent neural network is a directed acyclic graph. The graph in these networks specifically can always be substituted by a strictly feedforward network. An infinite impulse is an directed cyclic graph that cannot be exchanged for an feedforward network.

2.4. Generative Adversarial Networks

Generative Adversarial Networks (GANs), proposed by Goodfellow et al. [6] in **(year?)**, are a class of artificial algorithms built from two neural networks. The system consists of a generator and a discriminator. The generator generates new data samples. The discriminator tries to distinguish between the generated "fake" data samples and samples from the dataset. This has lead to some very impressive results such as images that resemble real images quite closely.

The generator maps the input to a particular data distribution while trying to increase the error rate of the discriminator. A rise in error rate of the discriminator intuitively describes that the generated data more closely resembles the real world data. A lot of caution should be taken with this assumption. Most research concerning GAN's is about generating new samples that humans cant distinguish. This is strictly not what a GAN does. After all, neural networks are not perfect, thus this also applies to the discriminator. In the end the generator has only learned how to fool it's discriminator. This effect results in experiments with GAN's being generally quite time consuming and precise, as a small change in parameters may spiral either discriminator or generator out of control. After training, empirical analysis is often still required to confirm the output does look like the real world data (ie. with images or text).

The generator is often initialized with a random input, sampled from a certain latent space. The discriminator on the other hand is fed real world data until a certain accuracy has been obtained. The other data is then used to compare the generated data with. This is very much like a train and test set for classifiers. In this model, the generator is typically a deconvolutional neural network, whereas the discriminator is a convolutional neural network.

3. Design

3.1. Dataset

We used publicly available reddit data [7]. Due to the huge amount of comments, we decided to take a specific time period, namely January, February and March of 2018. The data is not initially split on subreddit ($\sim$200 GB). After removing irrelevant subreddits and filtering the data we were left with 3GB for the_donald and politics and 6GB for askreddit in raw data. Initial tests on the dataset uncovered multiple problems. The variable sentence length posed a problem for in particular the RNN and the GAN. These networks can *technically* process sentences of varying length. In it's current form variable length sequences would drastically increase the memory usage and compute time (cudnn.benchmark) for both models. Additionally, variable length would require a much longer training process for both systems. Given the size of our research we have opted to only select sentences with a length of ten words. Our second issue with the dataset was that the term space was too large. Due to the massive use of internet slang and abbreviations (and the large amount of spelling errors) our encoder required a huge vocabulary size. We have used an English word set [8] to filter out any incorrect or otherwise unknown words. Our data sampling allows for such strict filtering as we can simply add more months worth of data if necessary.

3.2. Markov Chain

To give ourselves some baseline as to what a simple but most likely not very good sentence looks like from this dataset, we created a Markov chain.

3.3. Recurrent Neural Network

We used a RNN following the Pytorch name generation tutorial[9]. The network describes a simple recur-

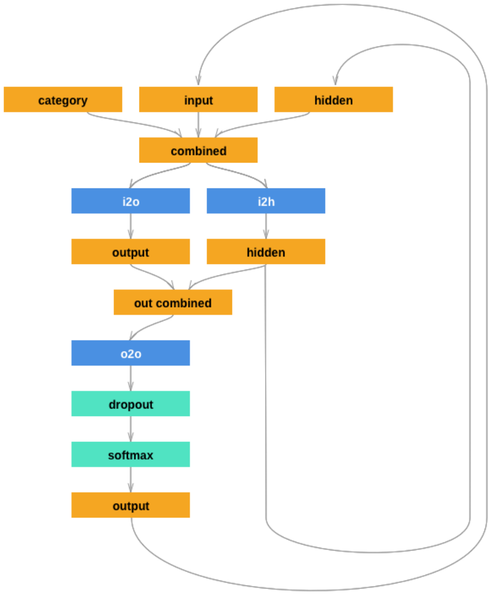

Figure 1: Overview of the recurrent network architecture.

rent system that tries to predict the next word given the previous word and the state of the system. The network we used is shown in figure 1

The model consists of 3 linear layers. One of the layers is for the hidden state. The second one is specifically for the output and the third one is used after the hidden state and output are combined again. This last layer also used dropout and softmax and results in a prediction. The result then goes back into the network, since the prediction of the next word is based on the previous word. The hidden state has the capacity to remember older information. The RNN is fed Pytorch tensors. These tensors consists of ten one-hot vectors. Every one-hot vector directly maps to a word. Running the network results in a tensor with ten one-hot vectors that can all be decoded back into words.

3.4. Generative Adversarial Network

The GAN we used is based on the paper of Yu et al. [10]. They created a ”Sequential GAN” referenced to as SeqGan. They adapted the standard GAN model to be more suitable for discrete tokens such as text. The network is generally quite similar to normal GANs. A major reason for their adaptation lies in that the discrete outputs from the generative model make it difficult to pass the gradient update from the discriminative model to the generative model. Additionally, the discriminative model can only evaluate a complete sequence, while for a partially generated sequence, it is nontrivial to balance its current score and the future one once the entire sequence has been generated. Their adaptation to the gradient update policy allows the network to perform well on partial sequences. Even though our input and output tensors have a length of 10 words we still want our models to have a sense of sequence required for text generation. The network architecture can be seen in figure 2.

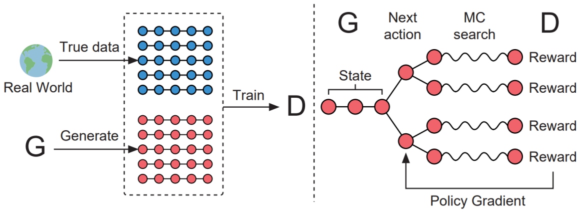

Figure 2: Overview of the SeqGan architecture.

The GAN does not use the same one-hot vector representation as the RNN. This is due to size constraints. The bigger scale of the GAN network makes the one-hot vectors incompatible with our hardware due to size constraints. We read and process the embeddings via the Pytorch embedding object. These embeddings provide the direct word representation as well as semantic information. The embedded vectors are much smaller in dimensionality than the RNN's one-hot vectors. The extra room we have due to this procedure is used to increase the size of the network layers, allowing for a 256-feature GRU (Gated Recurrent Unit) RNN for sequence generation. The discriminator uses a smaller GRU (64 features) as it is already very powerful out of the box. The dimensionality of our task makes it more difficult for the generator than for the discriminator.

3.5. Survey

Evaluating these models in a scientific way is quite hard, because how good a sentence is can not be analyzed by some statistic, but can only really be done empirically by humans. We have thus created a survey in order to evaluate the generated sentences. In this survey we sampled 20 sentences from both the RNN and the GAN, along with 40 sentences from the subreddit. We ommitted the markov model due to space constraints, opting for the two ”strongest” models instead. All punctuation was removed from the subreddit comments to resemble the sentences generated by the models. The sentences were split into 4 sections, with two sections for each model. The sentences were split 50/50 between human and model comments. Sentence order was randomized in every block, resulting in 4 blocks

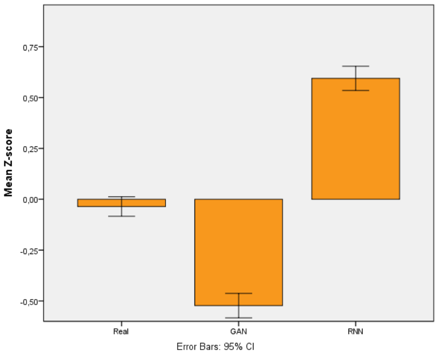

Figure 3: Survey Z-score results.

of 20 sentences in random order. The sentences were judged on ‘humanness’ on a 7-point Likert scale by 32 participants. The data from one participant were omitted as their answers did not show enough variance (rated almost every sentence equally).

4. Results

We will discuss the survey results first and reflect on our approach afterwards.

4.1. Survey

All ratings were converted to Z-scores to account for the variation in the use of the scale between participants. Then, the average Z-score for the real sentences, GAN sentences and RNN sentences was calculated. These averages were compared using three separate paired samples t-tests, with Z-score as the dependent variable. Real sentences were rated on average as more humanlike (M = -0.04, SD = 0.13) than sentences generated by the GAN model (M = -0.52, SD = 0.16). This difference, 0.49, 95% CI [0.40, 0.58], was significant ($t(30) = 10.96$, $p < 0.001$). Real sentences were rated on average as less humanlike (M = -0.04, SD = 0.13) than sentences generated by the RNN model (M = 0.59, SD = 0.16). This difference, 0.63, 95% CI [-0.72, -0.54], was significant ($t(30) = -14.29$, $p < 0.001$). Sentences generated by the GAN model were rated on average as less humanlike (M = -0.52, SD = 0.16) than sentences generated by the RNN model (M = 0.59, SD = 0.16). This difference, 1.12, 95% CI [-1.23, -1.01], was significant ($t(30) = -20.83$, $p < 0.001$).

4.2. Reflection

Following our results we can note that our RNN performs better than our GAN. The sentences produced by the RNN were on average rated as more humanlike than the sentences generated by the GAN. Sentences generated by the GAN were accepted less often as human sentence as the true sentences. Our current model does not satisfy the requirement of sufficiently mimicking the average /r/the_donald user. The sentences written by the RNN were accepted more often than the true sequences. At first glance this may be quite surprising, but the scope of the test should be considered. Not all respondees were familiar with the subreddit. All sentences were stripped of any interpunction and other stylistic formatting. Finally, the tested subreddit (/r/the_donald) contains a lot of figurative speech and community specific idioms. These may have negatively impacted the true sentence ratings in the survey.

5. Summary and Conclusions

In this project we have shown the feasibility of generating reddit comments using a GAN and an RNN. Our results have shown it possible for a generative model to generate convincing comments. The GAN did disappoint a bit in terms of performance, though through our error. We figure that by scaling the problem up and by improving the neural networks inside of the GAN it should be able to perform much better. We have not run exhaustive tests on our Markov chain system due to time constraints. A bigger research project may include large-scale tests that can also assess the quality of these systems. In this research, however, we primarily used them in an exploratory fashion. This would also include a survey on a larger set of people, perhaps people that are knowledgeable of the specific subreddits. Finally we would like to run the larger tests on more than just one subreddit. Perhaps it may be possible to base a GAN on our RNN creating a ’best of both worlds’ solution. We found the results promising and think further research could make these sort of systems very powerful in the future. We have learned a great deal about RNN’s, GAN’s, text generation and PyTorch, and would like to revise the topic in the future.

5.1. Author Contributions

Both of our group members have contributed equally to the research. Ties was in charge of the code and Bauke was in charge of the report in general terms. The introduction and background sections were initially written by Bauke. The design and result sections were initially written by Ties. We have both checked and rewritten parts all over the paper, making contributing specific sections difficult. Code: `https://github.com/Sipondo/Generation-of-undirected-responses`

References

[1] D. Morales, H. Nguyen, and T. Chin, "A Neural Chatbot with Personality," *Computer Science Department, Stanford University*, 2017. [Online]. Available: https://web.stanford.edu/class/cs224n/reports/2761115.pdf

[2] J. Li, W. Monroe, T. Shi, A. Ritter, and D. Jurafsky, "Adversarial learning for neural dialogue generation," *CoRR*, vol. abs/1701.06547, 2017. [Online]. Available: http://arxiv.org/abs/1701.06547

[3] J. Li, W. Monroe, A. Ritter, M. Galley, J. Gao, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," *CoRR*, vol. abs/1606.01541, 2016. [Online]. Available: http://arxiv.org/abs/1606.01541

[4] "Yep, russian trolls hit reddit too, on /r/funny and elsewhere," https://arstechnica.com/tech-policy/2018/04/reddit-identifies-nearly-1000-suspicious-russia -connected-accounts/, accessed: 24-06-2018.

[5] "Subredditsimulator," https://www.reddit.com/r/SubredditSimulator/, accessed: 24-06-2018.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," 2014. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[7] "Reddit data dump," http://files.pushshift.io/reddit/comments/, accessed: 20-05-2018.

[8] "english-words," https://github.com/dwyl/english-words, accessed: 05-06-2018.

[9] "Generating names with a character-level rnn," https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html, accessed: 05-06-2018.

[10] L. Yu, W. Zhang, J. Wang, and Y. Y. Shanghai, "SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient." [Online]. Available: https://arxiv.org/pdf/1609.05473.pdf

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks," 2016. [Online]. Available: https://arxiv.org/pdf/1511.06434.pdf

[12] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," pp. 1–16, 2016. [Online]. Available: http://arxiv.org/abs/1611.04558

[13] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, Ł. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," 2016. [Online]. Available: https://arxiv.org/pdf/1609.08144.pdf

[14] "Reddit just banned one of its most toxic forums. but it won't touch the_donald." https://www.vox.com/culture/2017/11/13/16624688/reddit-bans-incels-the-donald-controversy, accessed: 24-06-2018.