Meme Generation

Walter Simoncini
Student / Maastricht University
w.simoncini@student.maastrichtuniversity.nl

Abstract

This paper details the construction of a text generator for memes. The meme generator is evaluated by human judges who categorize machine (or human) generated memes as real (human) or fake (machine). Two models are evaluated: a model based on convolutional layers and one based on Recurrent Neural Networks (RNN). The latter model is evaluated both when trained from scratch and when using pre-trained weights (specialized for meme generation).

1 Introduction

As stated by (Zhang and Sun, 2009): "Text generation is a subfield of natural language processing. It leverages knowledge in computational linguistics and artificial intelligence to automatically generate natural language texts, which can satisfy certain communicative requirements.". In the context of this paper I extend the work of (Wenzlau, 2019) in generating the text of memes belonging to a well-defined meme category. The model by (Wenzlau, 2019) is used as a baseline and compared with textgenrnn (Woolf, 2020) when the latter is trained from scratch or when the pre-trained weights are used. The evaluation of the models was done using a questionnaire to assess how well the models can deceive human judges in believing the memes were created by a human author.

2 Related work

The task of generating memes has already been tackled by a handful of researchers such as (Peirson et al., 2018), whose team developed a system to generate a meme's text from an image or a user defined label and (Wenzlau, 2019), who developed a system to generate text for memes belonging to a specific class. On a broader scale in the recent years various language models that can be used to generate text were developed, such as GPT-2 (Radford et al., 2019) and XLNet (Yang et al., 2019).

3 Data

The data used to train the models was obtained by scraping imgflip.com, a popular website used to share and create memes. Since memes do not follow a fixed format it was decided to use a single meme category, called "Drake Hotline Bling" (Figure 1)

Figure 1: The "Drake Hotline Bling" meme (imgflip.com, 2020)

The scraped data set contains 7852 memes, with an average length of 50 characters.

3.1 Data pre-processing

Before the scraper data set could be used for training the following pre-processing was applied:

- The text was made lowercase and newlines were replaced by spaces.
- Memes shorter than 15 characters (or longer than 80) were discarded.
- Non English memes were removed using the fasttext (Joulin et al., 2016b) (Joulin et al.,

2016a) lid.176.bin model (using an acceptance threshold of 0.35 for the English language).

- Memes containing non-ASCII characters were discarded to reduce the vocabulary size.

Finally, to be able to discern the top and bottom text boxes a semicolon after each text box was added, in a similar way to (Wenzlau, 2019). Thus a training sample would look like this: "text box 1; text box 2;". After pre-processing only 5315 memes (with an average length of 42.86 characters were left).

3.2 Data augmentation

Due to the relatively small data set size the memes were augmented using RoBERTa (Liu et al., 2019) via the nlpaug (Ma, 2019) library. Each meme (whose text was 20 characters or more) was augmented by substituting words with contextually similar ones. This resulted in 3061 additional unique memes (after pre-processing) bringing the total data set size to 8376 instances.

4 Models

Two neural network architectures were evaluated for the task: one based on textgenrnn (Woolf, 2020) and one based on convolutional layers (Wenzlau, 2019). Both architectures were set-up for character level prediction due to the short nature of a meme's text. Each model was trained for 10 epochs.

4.1 Textgenrnn

Textgenrnn (Woolf, 2020) is a neural network model based on LSTMs and attention. The configuration used in this paper used three RNN layers, each with 128 neurons and no bi-directionality. The Embedding layer had a dimension of 64 and would store up to 10000 words. Two versions of textgenrnn were trained: the first was trained from scratch on memes text only and the second used the textgenrnn default weights and was specialized for meme generation via transfer learning.

4.2 Convolutional network

The model based on convolutional layers is an extension of (Wenzlau, 2019), in which the tokenizer was replaced by the Keras tokenizer (Chollet, 2020) and the predictor script was enhanced by adding a spelling correction to predicted memes using pyspellchecker (Barrus, 2020). In a similar fashion to textgenrnn this model has three convolutional layers, each with 1024 filters with a kernel size of 5. Each convolutional layer is followed by batch normalization, a max-pooling layer with batch size 2 and a dropout layer with rate 0.25. The model also has an embedding layer with 64 dimensions and no limit to the number of words in the vocabulary.

5 Experiments

The architectures were evaluated on the amount of data required to generate realistic text and on the credibility of the generated memes in front of an human judge.

5.1 Required amount of data to generate memes

The convolutional and the trained from scratch textgenrnn models were trained using sampled subsets of 1000, 2500 and 5000 instances (using a 75-25 split between training and validation). Sample texts for the models are shown in tables 1 and 2. For smaller training set sizes the convolutional network outperforms textgenrnn but when 5000 samples or more are used textgenrnn is able to form more logical sentences. In general, we can say that 1000 and 2500 training instances are insufficient for both models to generate coherent text and recommend a training set size of 5000 or more.

Samples	Generated texts
1000	Astidat and dates; coastitaly condemnation homework;
2500	Asking the sees and might memes; making memes thats have mathrummer friends;
5000	Asking for votes; being a cough on a person;

Table 1: Texts generated by the convolutional network for different sample sizes (the semicolon splits the top and bottom texts)

Samples	Generated texts
1000	E mmrn etey t rit_ u; A_ g e;
2500	Then your here; Hark a hot to meses;
5000	School work; Having a good meme;

Table 2: Texts generated by textgenrnn for different sample sizes (the semicolon splits top and bottom texts)

5.2 Credibility of the generated memes

The models were evaluated by human judges via a questionnaire. The questionnaire contained 12 memes, 3 of which were created by humans and 9 were generated by the models (and cherry picked): three were generated by the convolutional network, three by the transfer learning textgenrnn model and three by the textgenrnn model trained from scratch. The judges had to assess whether a given meme was generated by a model (fake) or by an human (real). In total, the questionnaire collected 41 answers. The results are highlighted in the confusion matrix in figure 2

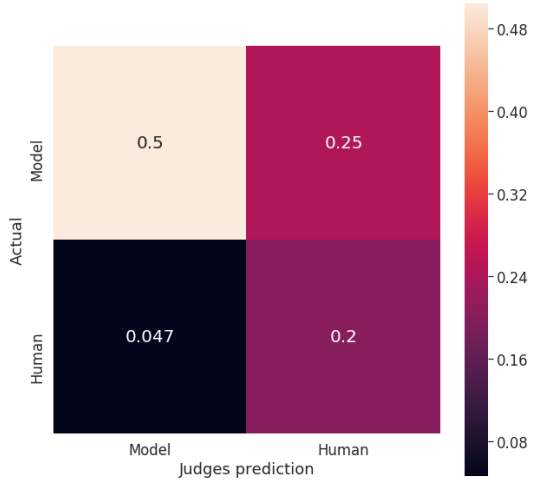

Figure 2: The confusion matrix for the human judges

From the confusion matrix in figure 2 we can see that the judges can tell, in around 70% of the cases, a machine generated meme from an human created meme, thus, in general, the models cannot deceive the human judges.

5.3 Best architecture for meme generation

Using the results from the questionnaire we can compare the three final models based on how well the can deceive the human judges. The confusion matrices in figure 3 highlight the percentage of generated images that were classified as human or machine created. By looking at the figure we can see that both the convolutional and the trained from scratch textgenrnn models perform poorly and are easily detected by humans meanwhile the transfer learning model manages to deceive humans half of the time.

6 Conclusion

In this paper we analyzed three models, and out of these three the most promising one is the pre-

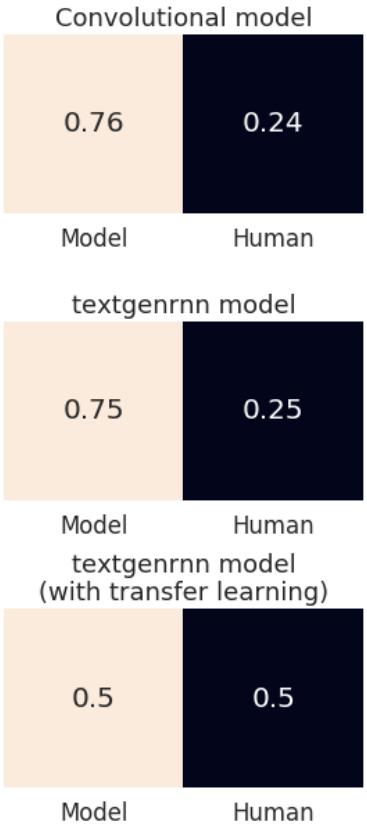

Figure 3: The confusion matrix for the three models

trained textgenrnn model, specialized for meme generation. Although these models are not able to deceive human judges in believing the generated memes are created by real humans the textgenrnn model shows some promising results and a larger data set should improve its deceiving ability. Moreover, we assessed that in order to generate coherent memes (with both architectures) a dataset of at least 5000 instances is required.

7 Future work

The models analyzed in this paper were trained on a relatively small training set, it might be worthwhile to investigate the effect of larger training sets on the model's performance. Moreover, given the improvement in score obtained by the pre-trained textgenrnn model it might be interesting to test different data sets for pre-training in case a larger training set is not available.

References

Tyler Barrus. 2020. pyspellchecker.

François Chollet. 2020. Keras.

imgflip.com. 2020. Drake Hotline Bling Meme.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Edward Ma. 2019. NLP Augmentation. https://github.com/makcedward/nlpaug.

V Peirson, L Abel, and E Meltem Tolunay. 2018. Dank learning: Generating memes using deep neural networks. *arXiv preprint arXiv:1806.04510*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Dylan Wenzlau. 2019. Meme Text Gen Convnet.

Max Woolf. 2020. textgenrnn.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Li Zhang and Jian-Tao Sun. 2009. *Text Generation*, pages 3048–3051. Springer US, Boston, MA.