CS230

Generating Six-Word Stories

Gianna Chien
Department of Computer Science
Stanford University
ggchien@stanford.edu

Abstract

Text generation has the potential to test and expand the creative ability of AI. The six-word story is a unique format of flash-storytelling where stories must be exactly six words long. In this project, I scraped data from Reddit's r/sixwordstories subreddit and used an LSTM and fine-tuned GPT-2 model to generate six-word stories. Evaluating models on readability, cohesiveness, humanness, and overall quality, I found that the stories generated by GPT-2 scored higher across all categories than those generated by the LSTM, with more consistent punctuation, capitalization, and plot development or implication. However, the GPT-2 stories varied more in length, and many stories were direct copies of those in the training set. Although both models were able to output readable and understandable stories, neither managed to capture the depth of humor, cultural references, and wordplay present in the human-written stories.

1 Introduction

Text generation has the potential to test and expand the creative ability of AI. Although this natural language sequence generation task is not new, applications to different fields can test the limits of machine comprehension within constraints particular to each situation. The six-word story is a format of flash storytelling that rose to popularity through the famous tale allegedly written by Ernest Hemingway:

"For sale: baby shoes, never worn." [14]

In addition to the challenges that come with generating grammatically correct text that is coherent and human-understandable, an AI to generate six-word stories must also conform to the word constraint, thus creating an additional level of complexity as it must balance its ability to tell a story - in that it details or implies some event, emotional journey, etc. - with maintaining a word count of exactly six.

In this project, I used an LSTM as well as the transformer-based GPT-2 to create an AI to generate six-word stories. The AI took no input, and generated a story beginning with a start token and either ending with an end token or at a story length of 20 tokens. A well-formatted story had exactly six words.

2 Related Work

Although no work had specifically been done concerning the generation of six-word stories, previous papers explored language modeling to produce other forms of text with restrictions. Keppler and Chen used Jaccard similarity, LSTMs, and other Seq2Seq models to generate the second sentence of

CS230: Deep Learning, Fall 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

a two-sentence horror story when provided with an input sentence [1]. Poetry generation, which often imposes strict form requirements upon the text generated, has also been explored. Ghazvininejad et al., for example, generated sonnets when given an input topic by first choosing rhyme words that related to that topic using the word embedding space, then creating a finite state acceptor (FSA) with a path for every possible sequence of vocabulary words that obeyed the rhyme and rhythm constraints and generating the poem backwards using an RNN to select the most fluent path through the FSA [7].

Transformer models, which are based solely on attention mechanisms [4] have previously been used for language modeling tasks. OpenAI's GPT model uses generative pre-training followed by discriminative fine-tuning to create a model that can easily adapt to different tasks [3]. Its successor, GPT-2, generally follows the architecture of GPT, but also moves layer normalization to the input of each sub-block, increases the vocab size, and increases the context size, resulting in a model with more parameters [2].

The successor GPT-3 has shown even more improvements over GPT-2, with text generated using GPT-3 going viral for its realistic imitation of human language [6]. However, users have only recently been able to request access to a public API for GPT-3, so its use has not been widely established. Therefore, GPT-2 was used for this project.

3 Dataset

I acquired data by using PRAW (the Python Reddit API Wrapper) [9] to call the Pushshift API [5] to query data from the r/sixwordstories subreddit [10]. Within this subreddit, stories are provided as the title of each post. I retrieved 28500 examples for model training on October 29th. Although not every one of these examples was guaranteed to be a story, as subreddit moderators occasionally made rule update or community challenge posts, such non-story posts were infrequent enough that with the quantity of data they likely had minimal effect on the model results. An example story from this subreddit is given below:

Drill sergeant couldn't help but laugh. [13]

3.1 LSTM Data Processing

For the LSTM approach, each example was then tokenized using the `nltk.word_tokenize` [12] method. To encode a single example into a source and target for training, start and end tokens were added to the story. All tokens except the last were used as the input `X`, and all tokens except the first were used as the output `Y`. From the training examples, a vocabulary was created to map each token to a unique index. This was used to convert examples to `int` tensors, to be passed into the model for training.

3.2 GPT-2 Data Processing

Like with the LSTM, a start and end token was appended to each training example when used for GPT-2. Chunks were then randomly sampled from the data for training, then represented using Byte Pair Encoding (BPE), as described in the GPT-2 paper [2].

4 Methods

4.1 LSTM

I implemented a baseline LSTM model that, given an input sequence as described in the Dataset section, outputted a probability distribution over all possible words in the vocab indicating the probability that each word should be chosen next in the sequence. The overall model architecture was as follows:

The model zero-padded the input sequences to the same length so that they would be batched appropriately. Sequences were then fed into an embedding layer, which added an additional embedding dimension of 256. This output was then fed into an LSTM with 64 hidden units, then into a dense layer that used softmax activation to create a probability distribution over all words in the vocab.

The model was trained using an Adam optimizer with a learning rate of 0.01, learning rate decay of 0.01, batch size of 32, gradient clipping at value 5.0, β_1 of 0.9, and β_2 of 0.999 for 100 epochs. Sparse categorical cross entropy loss between the outputted probability distribution and the `Y` labels was used.

During prediction, the vocab index of the start token `'<SOS>'` was provided as input to the trained model. The model then generated words until either the end token `'<EOS>'` was reached, or 20 tokens were generated. At each timestep, the next word was randomly selected from the vocabulary using the probability distribution outputted by the model. Once all tokens were generated, they were concatenated into a single sentence using the `detokenize` method of the nltk `TreebankWordTokenizer` [11].

4.2 GPT-2

As previously stated, GPT-2 is a transformer-based model that closely follows the structure of the original GPT created by OpenAI [2]. For a diagram showing this architecture, see Appendix B.

Using the Python package created by Max Woolf [8], I fine-tuned the small (124M hyperparameter) pretrained GPT-2 model to my training set. The model was trained using a batch size of 1, learning rate of 0.0001, and Adam optimizer. Training took place over a period of approximately 3.5 hours, for 15400 total training steps.

When producing output examples, the start token was provided to the model. The model then generated text either until the end token was encountered, or a maximum of 20 tokens were generated. Generation took place with temperature 1.25 in order to provide more varied output.

5 Results and Discussion

5.1 LSTM Output

The LSTM model was able to achieve a loss of 49.3697 by the last epoch. Preserving punctuation and capitalization, but removing start and end tokens, some hand-picked model outputs include:

Dad is stopped by his wind

I love them beyond my heart .

Killing happened: Farewell, Adopted without disappointment

Call him agreed . Ready for war .

These generated stories conform to the six word constraint, are grammatically human readable, and seem to stay on one specific topic. However, the spacing surrounding punctuation is inconsistent, and the capitalization scheme is not typical for human-written stories. This may indicate that a better tokenization scheme or better data preprocessing is necessary to ensure consistently realistic text throughout the training examples.

Additionally, some generated examples conform to the six word constraint, but are not comprehensible stories, whereas other stories had realistic punctuation and storytelling, but did not conform to the six word constraint. For example:

Shoot me?! God, too late

Though only five words, this last example seems to display the dark and cynical humor common on the r/sixwordstories subreddit.

5.2 GPT-2 Output

Some hand-picked outputs from the GPT-2 model include:

Mistakes: made. Clinic: paid. Ass: saved

My stepsister is also a potato??????????????????

College fund, spent on the funeral.

Cold feet, send warmth and grins.

Like those generated by the LSTM, these stories conform to the six word constraint, are grammatically human readable, and stay on one specific topic. The punctuation and capitalization is much more consistent than the LSTM outputs, likely due to the better data preprocessing employed by GPT-2. Nearly all examples are comprehensible stories.

Additionally, GPT-2 demonstrates better understanding of the larger world since it was pretrained on the more general WebText corpus [2], as shown in this example:

Police executed black man. Protests worldwide.

Despite this pretraining, GPT-2 still maintains the self-aware humor often found on the r/sixwordstories subreddit. For instance, the following outputted example seems to make direct reference to the length requirement placed upon the six-word story:

This really feels like too short

However, the degree of fine-tuning used to produce these realistic examples often resulted in generated examples that were direct replicas of those within the training set. From a random sample of 50, 13 were exactly the same as stories found in the training set (26%). In order to counteract this, I increased the temperature used for text generation; although this helped produce more varied outputs, it also resulted in a decline in the quality of the syntactical structure of the sentences.

5.3 Quantitative Evaluation

As a quantitative metric, I attempted to measure the percentage of generated outputs that conformed to the six-word constraint. To do this, I randomly sampled 50 stories from 1000 generated LSTM outputs, as well as 50 stories from 1000 generated GPT-2 outputs. I then manually counted how many words were in each of these 100 stories.

Of the 50 sampled LSTM stories, 37 had exactly six words (74%). In contrast, of the 50 sampled GPT-2 stories, only 33 had exactly six words (66%). This is most likely due to the high temperature I used for GPT-2 story generation in order to combat exact replication of the training set; more variation in sentence structure led to more variation in story length.

5.4 Qualitative Evaluation

Because six-word stories are inherently creative, much of my evaluation was qualitative. In order to perform this qualitative evaluation, I created a survey to gain insight from 15 human evaluators. This survey contained 15 six-word stories, of which five were human-written, five were generated by the LSTM, and five were generated by GPT-2. In a method similar to the one employed by Keppler and Chen [1], for each model, I chose the five stories by first choosing three stories completely randomly from the outputted examples (of which there were 1000 each for the LSTM and GPT-2, and 28500 in the human-written training set), then randomly selecting 20 stories and hand-picking my favorite two. When choosing stories, I ensured that they were all exactly six words so as to not make it obvious which were computer-generated, and also avoided choosing any stories generated by GPT-2 that exactly matched those in the training set. For an example page from the survey, see Appendix A. For the list of stories used in the survey, see Appendix C.

For each of the 15 stories in the survey, I asked responders to rate the stories on scales from 1 to 10 on:

- Readability: Grammatical correctness. 10 corresponds to easily readable.
- Cohesion: How well the story stays on one topic. 10 corresponds to very cohesive.
- Humanness: How likely it was that the story was written by a human. 10 corresponds to definitely human, and 1 corresponds to definitely computer.
- Overall quality: 10 is the best possible score.

I averaged the results for each category for each model:

Model	Avg Readability	Avg Cohesion	Avg Humanness	Avg Overall Quality
Human	8.92	8.6	7.71	7.88
LSTM	4.71	4.33	3.31	4.09
GPT-2	8.72	7.95	6.67	6.49

As shown above, in every category humans scored the highest, followed by GPT-2, then the LSTM. When compared to the LSTM, GPT-2 generated more consistently understandable output that concerned more concrete topics. Whereas it was possible to extrapolate metaphorical meaning from the stories created by the LSTM (ex. *Burned the nostalgia on her today*), the GPT-2 output more often implied a clear progression of action, such as in the story, *Spider strikes again. Found...killed it!*.

However, neither the LSTM nor GPT-2 seemed to be completely able to capture the depth of meaning that humans can assign to six words. Because six-word stories are so short, many include puns, wordplay, or references to popular culture or other literature that neither model was able to demonstrate. For example, the human-written story, *Vegan vampire killed with a steak.*, contains a pun on the words "stake" and "steak," and the story, *Orphan marries widow. Oedipus' bones howl.*, references the mythical Oedipus, who famously married his own mother by mistake.

6 Conclusion

Text generation is a complex task, and becomes even more so when a six-word constraint is added. By using an LSTM to generate six-word stories, it is possible to create stories that conform to the word constraint and are human-readable. However, the punctuation of these stories is often inconsistent, and though their meaning can be interpreted as vaguely metaphorical, they do not usually imply any events or actions in the way that a story typically would. In contrast, stories generated by GPT-2 are more consistently readable, cohesive, and human-like than those generated by the LSTM. However, GPT-2 has more inconsistent story length in its outputs, and many of its outputs are direct copies of the examples in the training dataset.

Overall, although neither the LSTM nor GPT-2 are able to capture the full depth of storytelling human authors can encompass in only six words, both have the ability to produce readable and coherent text; GPT-2 especially scored very well on all metrics when human-evaluated, producing stories with true plot progression or implication.

7 Future Work

In order to prevent GPT-2 from generating output stories identical to those in the training dataset, it may be beneficial to fine-tune the model for fewer iterations on the training data to prevent overfitting. Then, a lower temperature can be used during text generation, which may lead to more syntactically correct stories of exactly six words.

Another area for future work concerns the evaluation of generated stories. The development of an automated quantitative metric, for example one to automatically calculate the percentage of stories with exactly six words or to automatically assign a cohesiveness or readability score to a given story, could allow for a standardized, more objective way to evaluate model outputs.

References

[1] Adam Keppler, Jennie Chen. HorrifAI: Using AI to Generate Two-Sentence Horror. URL: https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/custom/15841116.pdf.

[2] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language Models are Unsupervised Multitask Learners. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[3] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. 2018.

[4] Ashish Vaswani et al. Attention Is All You Need. 2017. 31st Conference on Neural Information Processing Systems. URL: https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[5] Jason Baumgartner. Reddit statistics - pushshift.io. [Online; accessed 29-October-2020]. URL: https://pushshift.io/.

[6] Karen Hao. "A college kid's fake, AI-generated blog fooled tens of thousands. This is how he made it." MIT Technology Review. 14 Aug. 2020. URL: https://www.technologyreview.com/2020/08/14/1006780/ai-gpt-3-fake-blog-reached-top-of-hacker-news/.

[7] Marjan Ghazvininejad, Xing Shi, Yejin Choi, Kevin Knight. Generating Topical Poetry. 2016. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. URL: https://www.aclweb.org/anthology/D16-1126.pdf.

[8] Max Woolf. gpt-2-simple. GitHub. URL: https://github.com/minimaxir/gpt-2-simple.

[9] PRAW: The Python Reddit API Wrapper. [Online; accessed 29-October-2020]. URL: https://praw.readthedocs.io/en/latest/.

[10] Reddit. Six Word Stories. [Online; accessed 2-November-2020]. URL: https://www.reddit.com/r/sixwordstories/.

[11] "Source code for nltk.tokenize.treebank." NLTK 3.5 documentation. URL: https://www.nltk.org/_modules/nltk/tokenize/treebank.html

[12] "Source code for nltk.tokenize.punkt." NLTK 3.5 documentation. URL: https://www.nltk.org/_modules/nltk/tokenize/punkt.html#PunktLanguageVars.word_tokenize

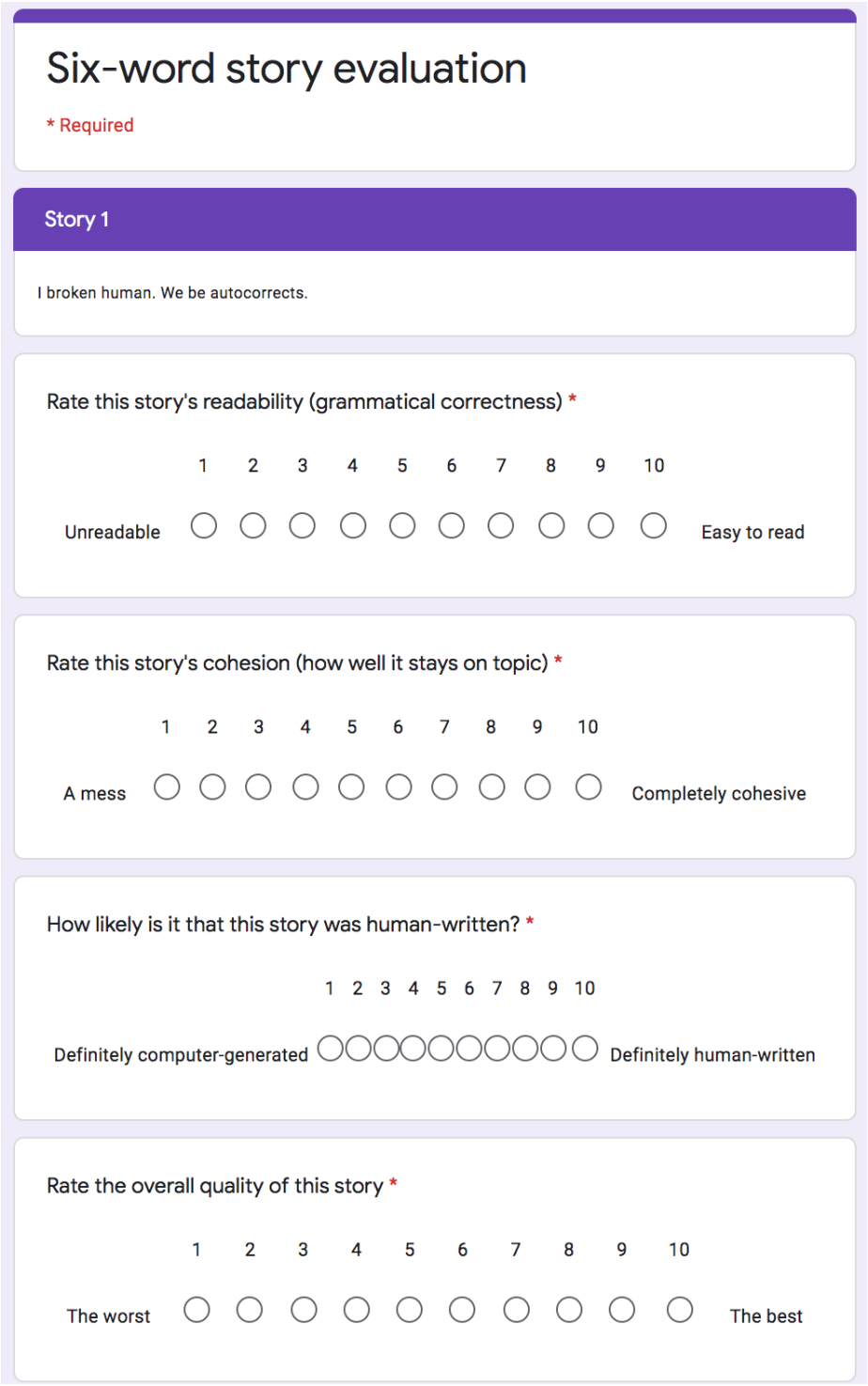
[13] u/DoubleOhOne. "Drill sergeant couldn't help but laugh." Six Word Stories. Reddit. URL: https://www.reddit.com/r/sixwordstories/comments/j7ti21/drill_sergeant_couldnt_help_but_laugh/

[14] Wikipedia. "For sale: baby shoes, never worn." [Online; accessed 30-September-2020]. URL: https://en.wikipedia.org/wiki/For_sale:_baby_shoes,_never_worn.

Appendix

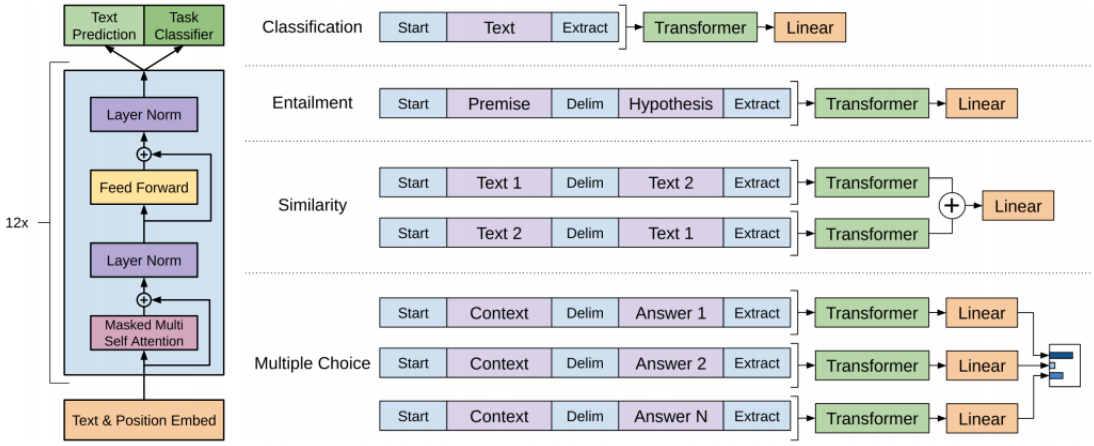
Appendix A

For qualitative evaluation, all respondents answered the same set of 4 questions for each of the 15 samples. Below is a sample page of the survey.



Appendix B

Architecture of the GPT model [3].



Appendix C

The following are the stories that were included in the survey sent to human evaluators:

Model	Story
LSTM	I broken human. We be autocorrects.
LSTM	Her winter entwined. I ain't them.
LSTM	Christmas: The last blowjob in out.
LSTM	Burned the nostalgia on her today
LSTM	I should love me this forever
GPT-2	But she did have a sister.
GPT-2	Spider strikes again. Found...killed it!
GPT-2	When Trump wins, Iens goes hostage
GPT-2	Well, what happened to the pilot?
GPT-2	He collected sourdough in his pocket.
Human	You go check. Leave the gun.
Human	YARD SALE: lawn mower not included.
Human	Your Love Was My Favorite Lie
Human	Orphan marries widow. Oedipus' bones howl.
Human	Vegan vampire killed with a steak.

Appendix D

Both the LSTM and GPT-2 generated a disproportionate number of stories about love. Some sample GPT-2 outputs include:

Infinite love is the only truth.

The patience of love is immeasurable

When acting with love, magic happens.

Perhaps this is an indication that many of the six-word stories written by redditors focus on themes of love.