



Springer



6TH INTERNATIONAL CONFERENCE INFORMATION,
COMMUNICATION & COMPUTING TECHNOLOGY
(ICICCT-2021)

EMPIRICAL LAWS OF NATURAL LANGUAGE PROCESSING FOR NEURAL LANGUAGE GENERATED TEXT

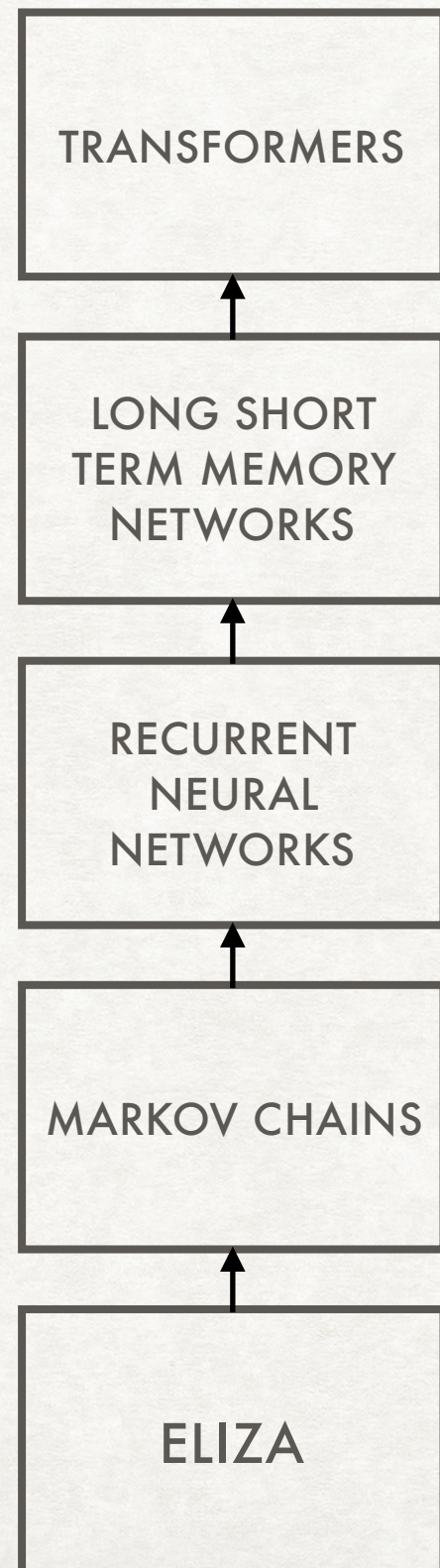
PAPER ID: ICICCT_2021_122

BY: SUMEDHA
PROF. RAJESH ROHILLA

INTRODUCTION

- In the domain of Natural Language Generation and Processing, a lot of work is being done for text generation. As the machines become able to understand the text and language, it leads to a significant reduction in human involvement.
- Standard Neural Networks are not used for text generation because the length of input and output is not fixed, it may vary with each sentence. So many sequential models are being used for generating the text.
- But the amount of research work done to check the extent up to which their results match the man-made texts are limited in number.
- In this paper, the text is generated using Long Short Term Memory networks (LSTMs) and Generative Pretrained Transformer-2 (GPT-2).
- After generating text, we have evaluated how close the neural language generated text is to human-generated text by checking if it follows Zipf's law and Heap's law.
- Along with this, we have seen the dependence of text generated on a hyper-parameter called Temperature and compared text generated by LSTMs to that generated using GPT-2.

RELATED WORK



The most recent and well-performing machine learning models for text generation are Transformers. These work on self-attention mechanism and are developed using encoders and decoders. All the best performing neural network architectures in the field of Natural Language Processing are found to be variants of transformers e.g.: BERT, GPT-2, etc.

The problem of vanishing gradients is solved by using LSTMs. These have an extra input known as cell state. The cell state is updated in each step such that if weights are too high it lowers them, if they are much less it increases them, hence avoiding the problem of vanishing or exploding gradients

In RNNs the current output is a function of present input as well as output of the previous neural network. As we go on with training RNNs, weights try to adjust themselves to minimize the error, in this process weights at the end will have more influence on the text generated as compared to weights at beginning of the network and the weights at the start slowly become zero. This process is known as vanishing gradients.

More commercial and intelligent text generation systems using Markov Chains first appeared in March 2017. Markov Chains predict the next word using the current word, as the output depends only on the current word, text loses semantics and context

The task of text being generated by a machine was first seen in mid the 1960s with the emergence of ELIZA, It breaks the input into sentences, parses it based on a simple pattern, and searches for keywords. Based on that keyword it generates a generalized response. Although it works well, the amount of intelligence involved is very little.

ZIPF'S LAW

All human generated texts in all the languages follow zipf's law and heap's law. If our machine generated text also follows these laws we say that this neural language generated text is close to natural language generated text

According to Zipf's law for all human generated texts the rank-frequency distribution is an inverse relation (frequency is proportional to inverse of rank).

In simple terms it can be explained as, the word that occurs the greatest number of times occurs two times more than the word that occurs the second most number of times, three times more than the third most frequently occurring word, and so on

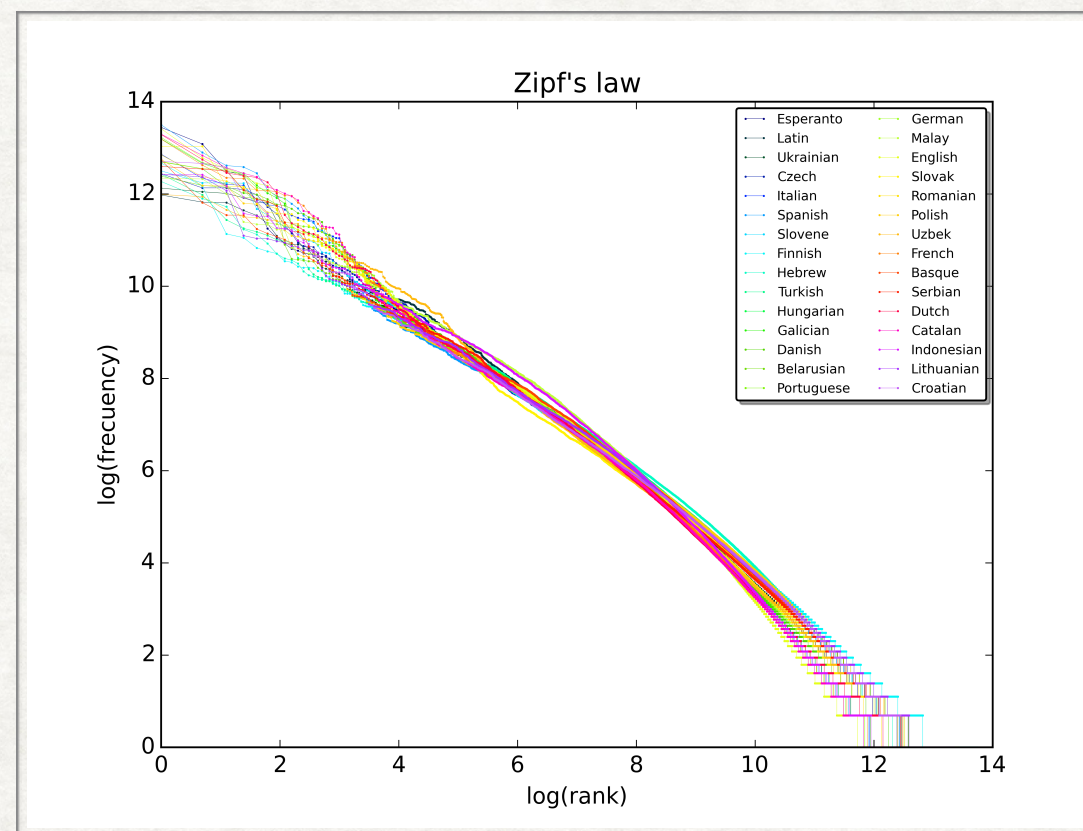


Fig: A plot of the rank versus frequency for the first 10 million words in 30 Wikipedias in a log-log scale

HEAP'S LAW

The law can be described like as the number of words in a document increases, the rate of the count of distinct words available in the document slows down.

e.g: Suppose in a document with 1000 words no. of unique words are 100, then for a document with 2000 words no. of unique words will be less than 200, for a document with 3000 words no. of unique words will be much less than 300 etc.

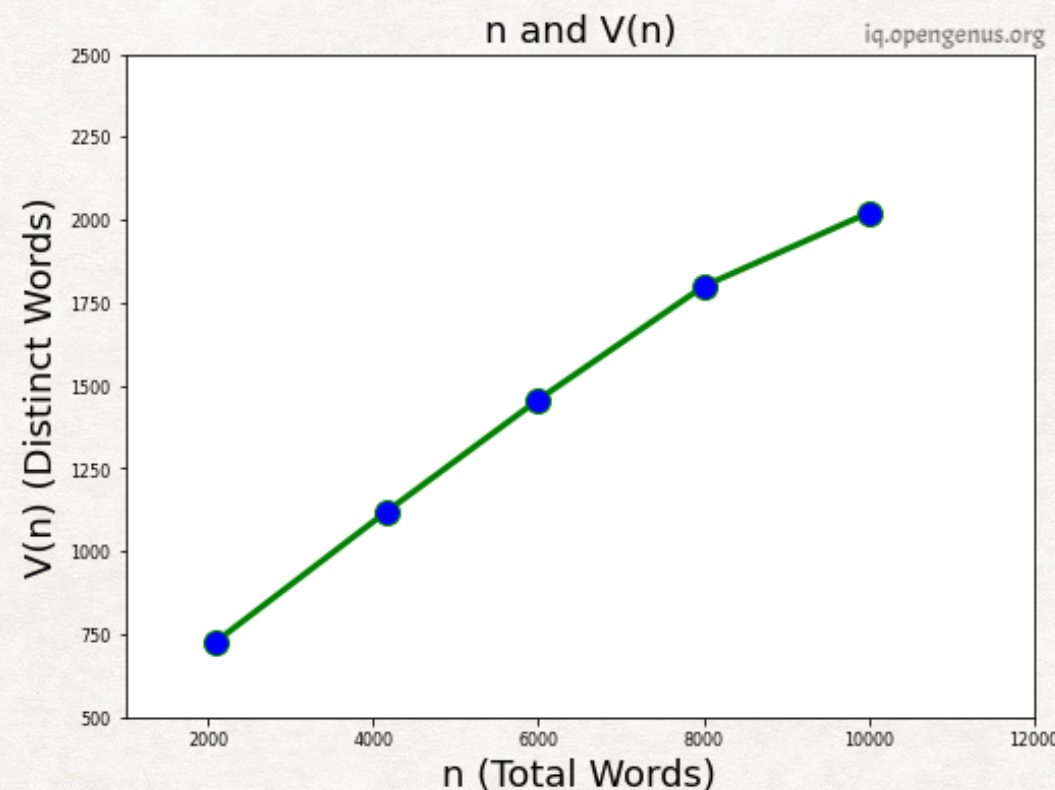


Fig: A typical Heaps-law plot. The x-axis represents the text size, and the y-axis represents the number of distinct vocabulary elements present in the text

LONG SHORT TERM MEMORY NETWORKS

- While training the network on large sequences RNNs begin to forget the starting part due to vanishing gradient issue which is solved utilizing LSTMs. Remembering details for a long time comes naturally to LSTMs because of their structure and design.

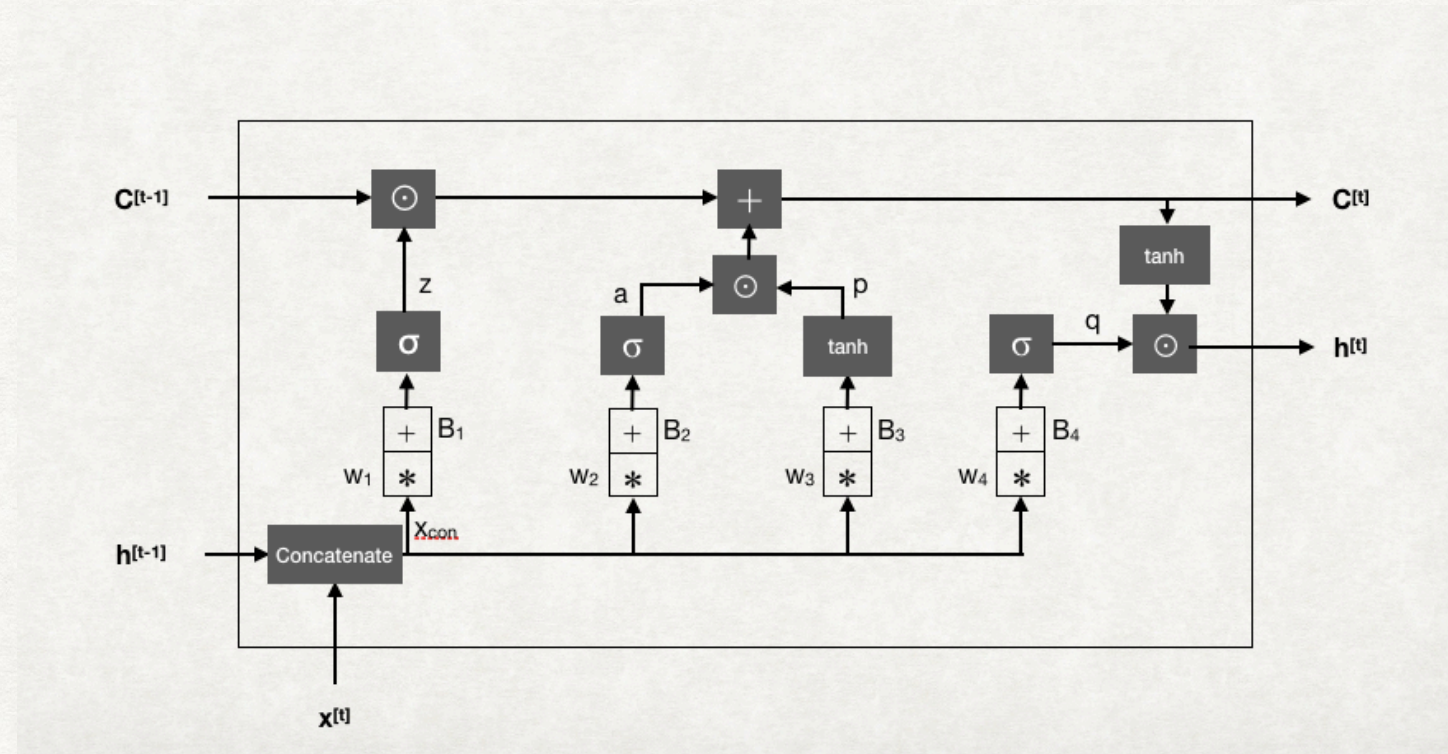


Fig: LSTM Cell

- In Fig. $C[t-1]$ represents provides the memory feature to LSTM cells, $h[t-1]$ is the output of the previous cell. The LSTM generates $h[t]$ and $c[t]$ as its outputs. First, a forget Layer makes a decision on information to be kept and thrown away. Secondly, we decide about the new information to be stored in cell state $c[t]$. In this step we update the old state cell by deciding the output for $c[t]$ and discarding information about old subject and adding new details. For the last step, we update the output $h(t)$.

TRANSFORMERS

- Transformers work on the self-attention model, this means they don't remember the whole sentence at once, but a perimeter α is assigned. $\alpha<1,1>$ decides how much value the first network holds while generating the first word, similarly $\alpha<2,1>$ decides how much value the first network holds while generating the second word and so on.

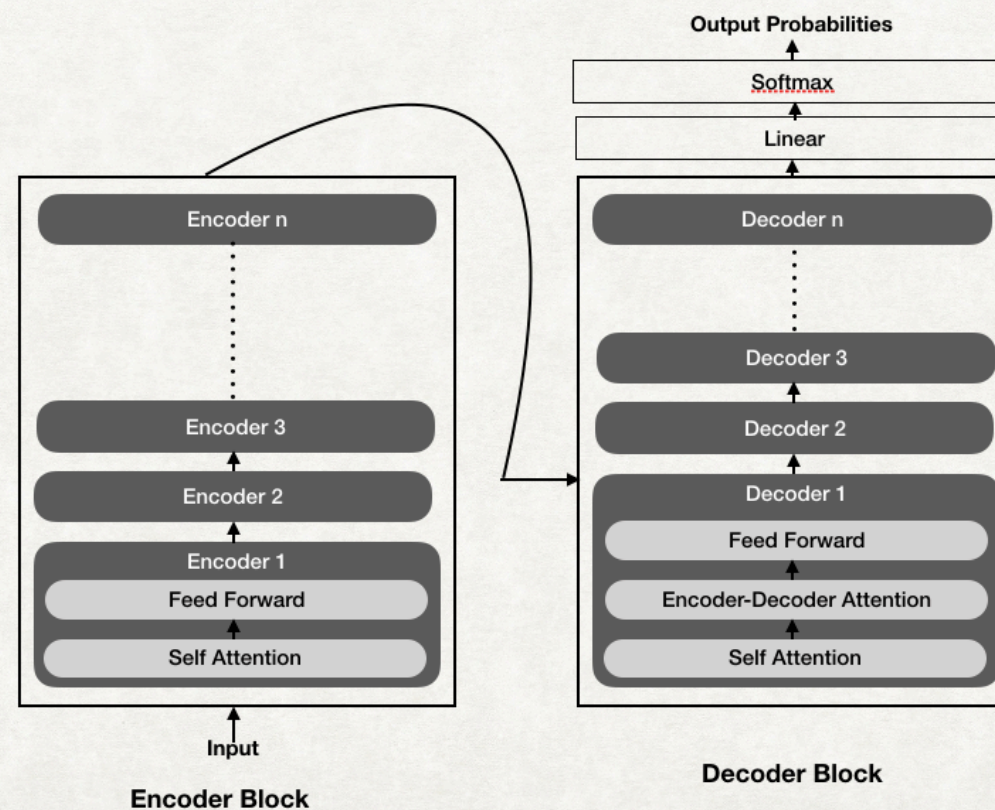
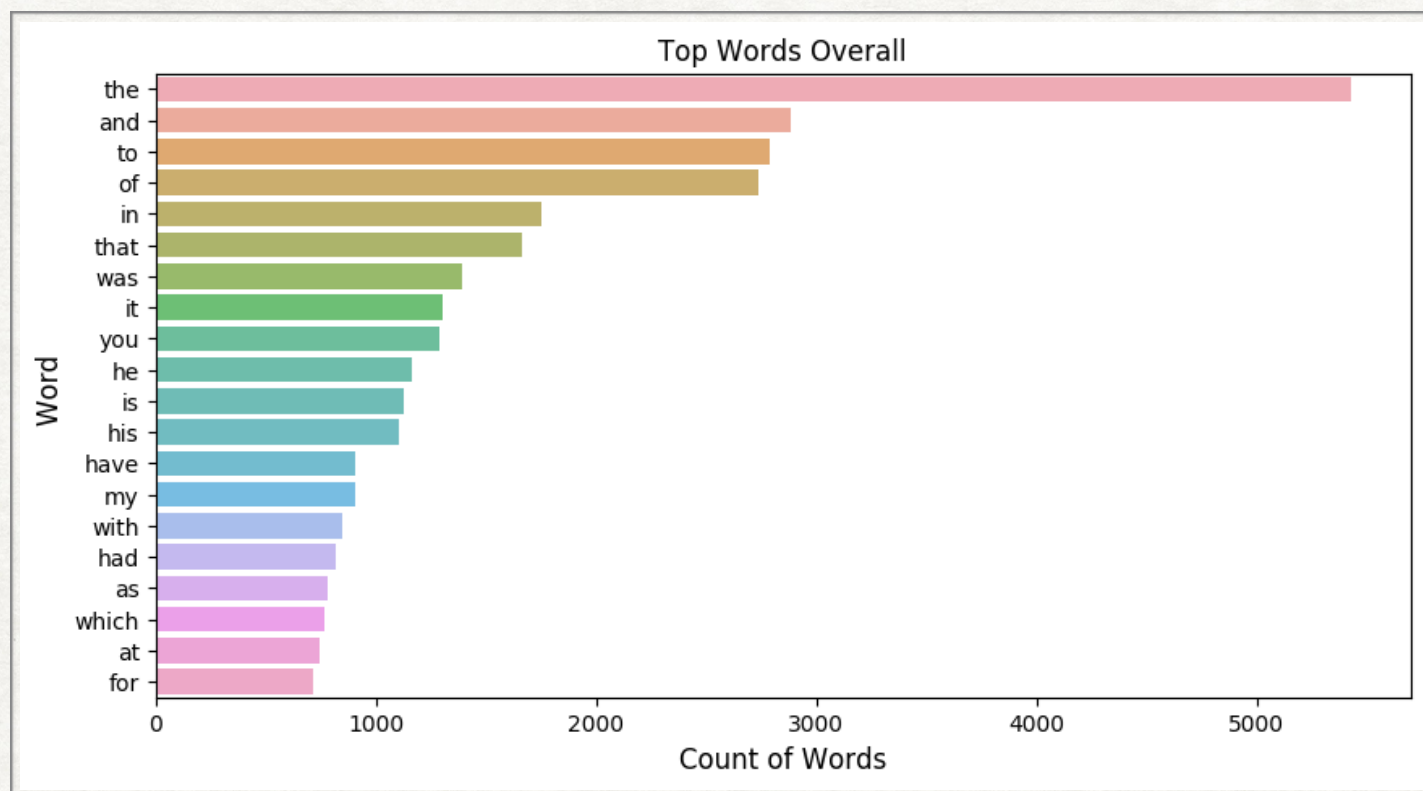


Fig.: A Transformer Cell

- A transformer cell consists of encoder and decoders blocks. An encoder converts text to word embeddings which are fed to the decoder, which as output generates text.
- In this paper, text is generated using Generative Pretrained Transformer 2 (GPT-2), which is based on a transformer model and uses decoder blocks.

DATASET USED

- As dataset we utilized the famous book "The Adventures of Sherlock Holmes" by Sir Arthur Conan Doyle.
- The book is made available through Project Gutenberg
- Dataset Stats: Book contains a total of 594197 characters



```
the      5428
and      2888
to       2790
of       2737
in       1750
that     1664
was      1394
it       1303
you      1286
he       1168
is       1129
his      1103
have     909
my       907
with     850
had      822
as       780
which    770
at       742
for      716
Name: 0, dtype: int64
```

Fig: Most common words and their occurring frequency

First two lines of the dataset: I had seen little of Holmes lately. My marriage had drifted us away from each other.

FLOWCHART

EXPLORED THE DATASET (TOTAL NO. OF CHARACTERS, MOST COMMONLY USED WORDS)



PRE-PROCESSED THE DATASET (CLEANING, TOKENIZATION, REMOVING PUNCTUATIONS AND STOP WORDS)



PREPARED DATASET FOR TRAINING (APPLIED SLIDING WINDOW TECHNIQUE , CHARACTER TO INTEGER MAPPING)



TUNED HYPER-PARAMETERS (TRIED DIFFERENT COMBINATIONS OF BATCH SIZE, NUMBER OF EPOCHS, LAYERS AND CHOSE THE ONES GIVING MAX. ACCURACY & MIN. LOSS)



CREATED LSTM BASED MODEL, FITTED IT, AND GENERATED TEXT FOR DIFFERENT VALUES OF WORD LENGTH AND HYPER- PARAMETER TEMPERATURE IN RANGE 0.1 TO 3



GENERATED TEXT USING GPT-2 FOR DIFFERENT WORD LENGTHS



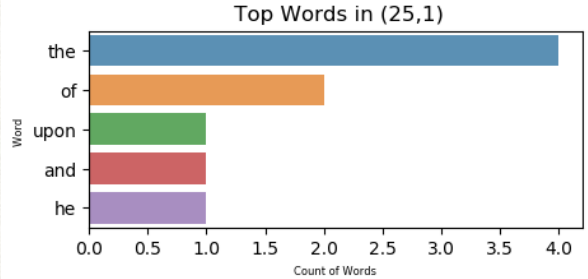
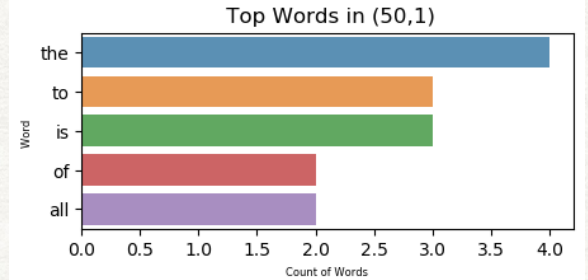
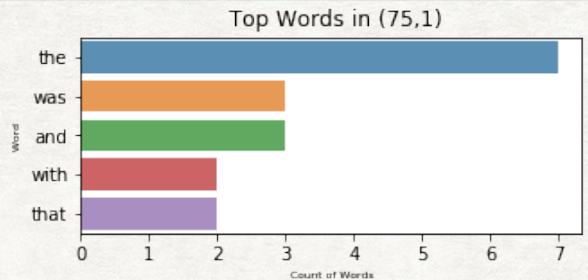
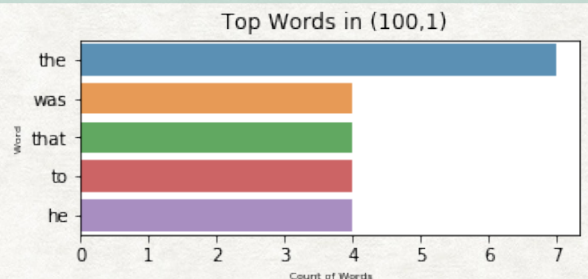
COMPARED HOW WELL LSTM AND GPT-2 GENERATED TEXT FOLLOWS ZIPF'S LAW AND HEAP'S LAW



COMPARED THE TEXTS GENERATED USING LSTMS AND GPT-2

RESULT

VERIFYING ZIPF'S LAWS FOR TEXT GENERATED USING LSTMS:

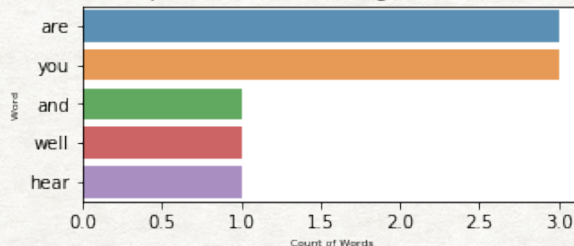
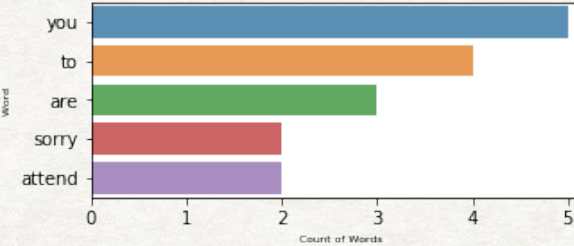
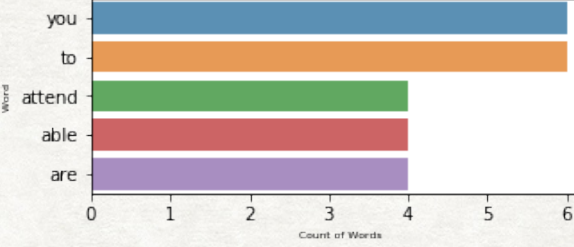
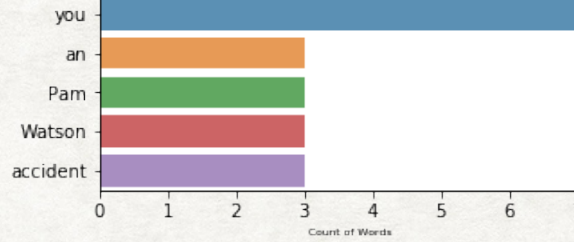
Document Length	Word Rank vs. Frequency	Unique Words
25		21
50		40
75		60
100		68

For verifying these two power laws for LSTMs and GPT-2 we generated text with various word lengths and calculated number of unique words and word rank vs its frequency distribution for each document length and got the results as displayed in the tables.

From Table, we can see that for all document lengths word rank and frequency are inversely proportional to each other.

RESULT

VERIFYING ZIPF'S LAWS FOR TEXT GENERATED USING GPT-2:

Document Length	Word Rank vs. Frequency	Unique Words												
25	<p>Top Words for word length = 25 - GPT2</p>  <table><tr><th>Word</th><th>Count of Words</th></tr><tr><td>are</td><td>2.8</td></tr><tr><td>you</td><td>2.8</td></tr><tr><td>and</td><td>1.0</td></tr><tr><td>well</td><td>1.0</td></tr><tr><td>hear</td><td>1.0</td></tr></table>	Word	Count of Words	are	2.8	you	2.8	and	1.0	well	1.0	hear	1.0	18
Word	Count of Words													
are	2.8													
you	2.8													
and	1.0													
well	1.0													
hear	1.0													
50	<p>Top Words for word length = 50 - GPT2</p>  <table><tr><th>Word</th><th>Count of Words</th></tr><tr><td>you</td><td>5.0</td></tr><tr><td>to</td><td>4.0</td></tr><tr><td>are</td><td>3.0</td></tr><tr><td>sorry</td><td>2.0</td></tr><tr><td>attend</td><td>2.0</td></tr></table>	Word	Count of Words	you	5.0	to	4.0	are	3.0	sorry	2.0	attend	2.0	30
Word	Count of Words													
you	5.0													
to	4.0													
are	3.0													
sorry	2.0													
attend	2.0													
75	<p>Top Words for word length = 75 - GPT2</p>  <table><tr><th>Word</th><th>Count of Words</th></tr><tr><td>you</td><td>6.0</td></tr><tr><td>to</td><td>6.0</td></tr><tr><td>attend</td><td>4.0</td></tr><tr><td>able</td><td>4.0</td></tr><tr><td>are</td><td>4.0</td></tr></table>	Word	Count of Words	you	6.0	to	6.0	attend	4.0	able	4.0	are	4.0	35
Word	Count of Words													
you	6.0													
to	6.0													
attend	4.0													
able	4.0													
are	4.0													
100	<p>Top Words for word length = 100 - GPT2</p>  <table><tr><th>Word</th><th>Count of Words</th></tr><tr><td>you</td><td>7.0</td></tr><tr><td>an</td><td>3.0</td></tr><tr><td>Pam</td><td>3.0</td></tr><tr><td>Watson</td><td>3.0</td></tr><tr><td>accident</td><td>3.0</td></tr></table>	Word	Count of Words	you	7.0	an	3.0	Pam	3.0	Watson	3.0	accident	3.0	51
Word	Count of Words													
you	7.0													
an	3.0													
Pam	3.0													
Watson	3.0													
accident	3.0													

From Table, we can see that for all document lengths word rank and frequency are inversely proportional to each other for GPT-2 generated text as well. Hence, we can say that Zipf's law is being followed by text generated by LSTMs and GPT-2.

RESULT

TEXT GENERATED USING LSTMS FOR DIFFERENT TEMPERATURE & WORD LENGTH:

- Temperature is a hyperparameter in neural networks which controls the randomness of text generated or any output
- In Table, we have measured the effect of variation of hyperparameter Temperature on the text generated
- We can see that as temperature (randomness) increases grammatical errors decrease and sentences start making comparatively more sense.

Temprature, Word_Length	Text Generated
(0.1,25)	and reported has been complete shrugged through the same corridor room and there is no wonder that ha that is a line of a occur
(0.1,75)	and stone and such an hard that our foundation were eyes to replace the tattered grass and donations not and is a little little landau for included a considerable bad and may have been seeds about the subject as far as you could copy the unconscious think that there is where you earn to be all mccarthy cut they ' opposing in complete sent came to their shown " " yes " " i say
(1.0,25)	can asked gaiters a fee of calves tops of the bride the other lithe upon the rent rose while the front he was stated and
(1.0,75)	holmes there was the name of the private indexing i could see that the reason might bring this with lord still simon but soon before this about 's dress is provided out on winchester i do n't think that i am myself " he stepped once to the room and with the paper and brown the latter paragraph it was a quiet little committed his black white face was indeed ran up the seen and
(2.0,25)	sherlock holmes left fear as i had let me go upon the silence at last nothing them wooden and the files and a very instantly
(2.0,75)	attention there was a little brute absorbing large clouds gradually up and made a glass of brandy and an efforts of the shave it was even in a half black cigars made out of the stream of these bridge from examining where the blue carbuncle long the copying of the encyclopædia must be much whereabouts on which their should be eyes to discover much so as far as it were quite weary to any other

RESULT

VERIFYING HEAP'S LAWS FOR TEXT GENERATED USING LSTMS AND GPT-2:

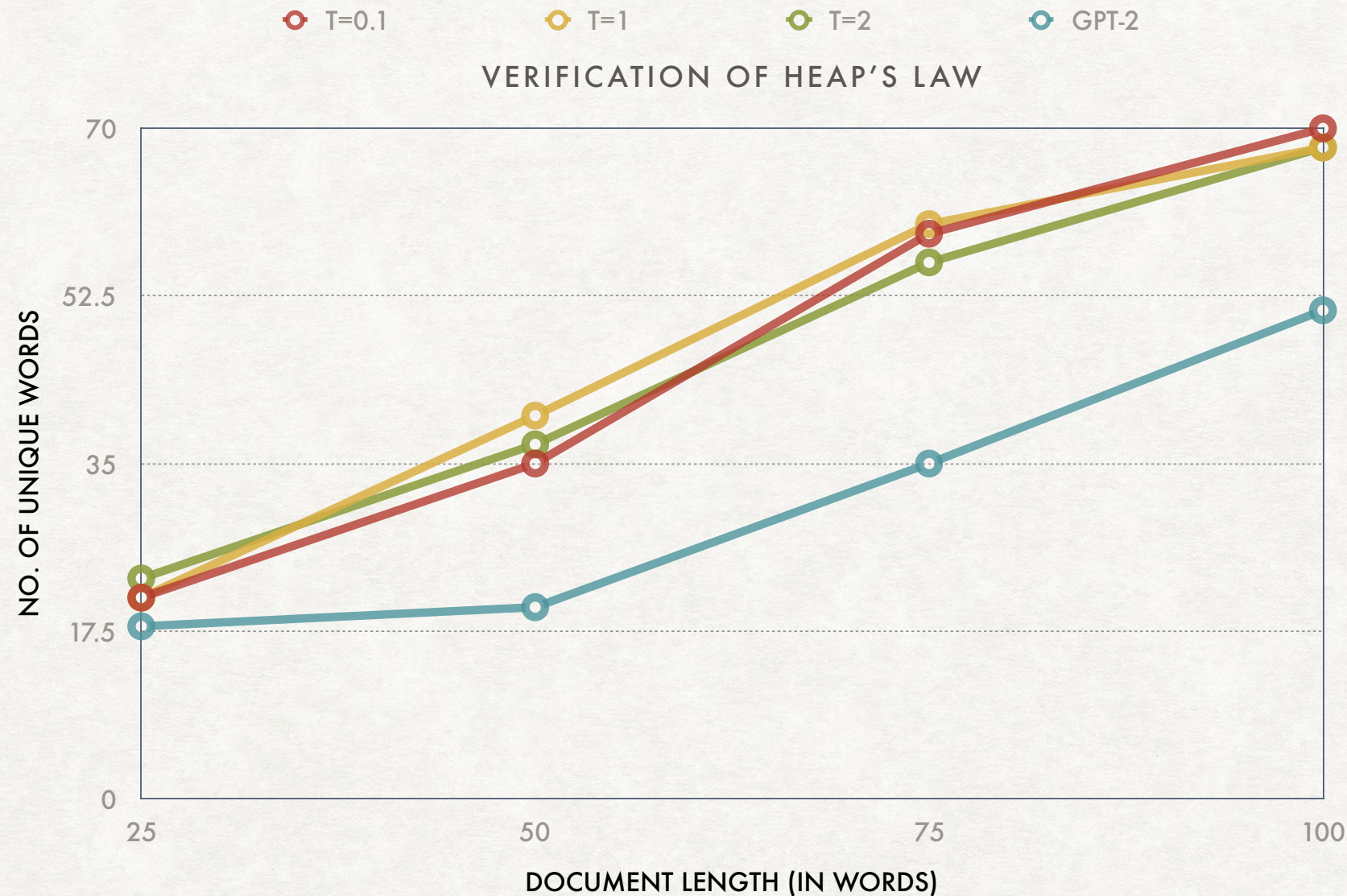


Fig.: Verification of Heap's Law

From Fig., we can see that as document size is increasing, the rate at which the number of distinct words is increasing has dropped. This tells us that Heap's law is also being followed for the text generated by the neural language models (both LSTM networks and GPT-2).

RESULT

TEXT GENERATED USING GPT-2 FOR DIFFERENT WORD LENGTHS

0: Watson you are in an accident?

Pam: Oh yes, and I would tell you, but I'm

1: Watson you are the greatest football fan ever, what can you say in your defence?

"I could probably give you one reason why I love being a fan of rugby union so much.

"I'm not going to pretend to

2: Watson you are no longer allowed to leave your job but your job does not matter. He tells me that his only aim is to get people to know and respect his intelligence; this is important for a successful company, otherwise he doesn't understand his customers. You are a smart man and I applaud your intelligence, but the rest is a complete waste of the time, energy

3: Watson you are very well. Now you are saying that you would not get away with not getting away with the stuff. Are you really implying that you were in a position where you knew that you were not making it a crime to say "fuck you" in public to the police officer? Is that what you are implying?

Hmmm. Is that what you are implying? Is that what you are implying? It certainly strikes me as a strong indication of some sort of crime of verbal

- We implemented a pre-trained transformer model and generated a text of lengths = 25, 50, 75 and 100.
- From the obtained text we can see that the text generated by GPT-2 is understandable to great extent, but sudden change in topic can be observed for each word length.
- Comparing this to LSTMs generated text we can say that text by GPT-2 makes more sense and is grammatically better.
- Also, the process of text generation by GPT-2 is more computationally expensive than the process involving LSTMs.

CONCLUSION AND FUTURE SCOPE

- We tried out various experiments and obtained results, which proved that LSTM and GPT-2 generated texts follow both statistical laws i.e., Zipf's and Heap's law.
- It is also observed that as Temperature (T) increases the quality of text also improves for LSTMs.
- Although the pseudo text given by LSTMs improves with T , it consists of some grammatical errors and stops making sense after some length, on the other hand, the text generated by GPT-2 is better than the text given by LSTMs in terms of grammar and quality, but lags in terms of computational cost.
- The Generative Pretrained Transformer-2 based model performs better than all the models previously used for text generation and generated samples are close to human-generated texts but, it also has some limitations such as because of its huge size it is more computationally expensive compared to previous models, this model gives good performance on generalized topics but performs poorly on scientific or technical data and abrupt changes in the topics can also be noticed.
- Future researches in the natural language processing community aim to reduce the computational cost of Transformers and LSTMs and to remove the limitations discussed to as much extent as possible.

REFERENCES

- M. Lippi, F., M. A. Montemurro, S., M. Degli Esposti, T., G. Cristadoro, Fo.: Natural language statistical features of LSTM-generated texts. In: IEEE Transactions on Neural Network and Learning Systems, volume 30, issue 11, pp: 3326 – 3337. IEEE (2019).
- Y. Qu, F., P. Liu, S., W. Song, T., L. Liu, Fo, M. Cheng, Fi.: A text generation and prediction system: Pretraining on new corpora using BERT and GPT-2. In: 10th International Conference on Electronics Information and Emergency Communication (ICEIEC), pp. 323-326. IEEE, Beijing, China (2020).
- M. C. Santillan, F., A. P. Azcarraga, S.: Poem generation using transformers and Doc2Vec embeddings. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, Glasgow, UK (2020).
- D. Wang, F., H. Cheng, S., P. Wang, T., X. Huang, Fo., G. Jian, Fi.: Zipf's law in passwords. In: IEEE Transactions on Information Forensics and Security, volume 12, issue 11, pp. 2776-2791, IEEE (2017).
- C. Li, F., Y. Su, S., W. LiuT.: Text-to-text generative adversarial networks. In: International Joint Conference on Neural Networks (IJCNN), pp. 1-7. IEEE, Rio de Janeiro (2018).
- J. Weizenbaum, F.: ELIZA – A computer program for the study of natural language communication between man and machine. In: Communications of the ACM, volume 9, pp. 36-45. ACM, United States (1966).
- A. Gatt, F., E. Krahmer, s.: Survey of the state of the art in natural language generation: core tasks applications and evaluation. In: JAIR, volume 61, pp. 65-170. AAAI Press (2018).
- B. Godor, F., World-wide user identification in seven characters with unique number mapping. In: 12th International Telecommunications Network Strategy and Planning Symposium, pp. 1-5. IEEE, New Delhi, India (2006).
- A. Vaswani, F., N. Shazeer, S., N. Parmar, T., J. Uszkoreit, Fo., L. Jones, Fi., A. N. Gomez, Si.: Attention is all you need, <https://arxiv.org/abs/1706.03762>, last accessed 2021/02/14. NIPS (2017).
- B. Raghav, F.: Text Generation in NLP - Springboard India, <https://in.springboard.com/blog/text-generation-using-recurrent-neural-networks/>, last accessed 2020/12/22.
- L. Lü, F., Z. K. Zhang, S., T. Zhou, T: Zipf's law leads to Heaps' law: analyzing their relation in finite-size systems. In: Journal.pone.0014139. PLoS One (2010).
- W. Li, F: Random texts exhibit Zipf's-law-like word frequency distribution. In: IEEE Transactions on Information Theory, vol. 38, no. 6, pp. 1842-1845. IEEE (1992)
- D. W. Otter, F., J. R. Medina, S., J. K. Kalita, T.: A survey of the usages of deep learning for natural language processing. In: IEEE Transactions on Neural Networks and Learning Systems, volume 32, issue 2, pp. 604-624. IEEE (2020).