NLP Project Paper

Eric Risbakk, Thomas van den Broek

May 2019

1 Introduction

The goal of this paper is to explore and compare different methods to generate song lyrics. This has been done by comparing two different methods. The first one, which is considered our base line, is a N-gram model utilizing stupid back-off. Our second method, which is the more advanced one, utilizes a Long-Short-Term-Memory (LSTM) Recurrent Neural Network (RNN) using words on the character level to generate the text. We expect the LSTM method to perform better as it can utilize more information to generate the lyrics.

The dataset consists of more than 55'000 in over 15 million words. For pre-processing the data-set, all punctuation was removed, replaced by spaces, and same letters occurring for more 2 times were reduced to only be 2.

The data was obtained from kaggle. [Kuz]

2 Research

The research done was mostly to understand LSTM's better and to figure out how to score the lyrics once they were created. The information needed to perform method 1 came from the lecture slides.

3 Method 1 (N-Gram)

This method uses N-grams. It uses a bag of words for each size of the N-Gram given. The Lyrics are generated by picking the most commonly used words given the words prior to that specific position. it utilizes stupid back-off to guarantee that a word is selected. Generally speaking the lyrics generated will always be the same if the input is the same. To try and create some variation there is a 10 percent chance that the next chosen word is still a match but not the one that occurs most often. As mentioned in the introduction this is a baseline and is thus not too complicated.

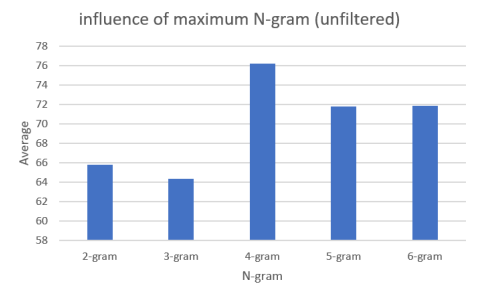
4 Method 2 (LSTM)

For LSTM we used Keras running on top of Tensorflow. Word embeddings were trained using gensim, on the song corpus. The word-embeddings were not made part of the model itself, rather, the word vectors were used as input instead. A single LSTM layer of 128 nodes then went on to a classification task of over 60'000 words. It was trained on the dataset in intervals of 10, to save memory, 10 times total, one epoch for each individual run, optimizer being RMSProp.

5 Testing

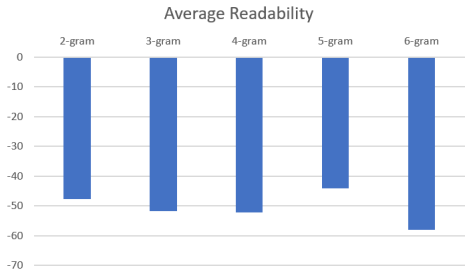
For testing we are using the Textatistic library [Hen]. This library can do multiple things such as counting words and sentences within the text. But most importantly, it generates a number representing the readability of the generated lyrics. Of course this is not perfect but it provides us with a way to measure the quality of the generated text.

For the first method, tests have been done to see how the number of n-grams change the result of the readability with the same lyrics length of 100. The results can be found in the table below.



	2-gram	3-gram	4-gram	5-gram	6-gram
1	78.04663621	80.84097254	77.25846273	67.48613805	70.97613971
2	71.99957692	58.57353896	77.26481577	78.49364189	70.44640264
3	62.32562178	46.89156757	75.04731892	75.28940821	79.41274247
4	67.23355638	59.00874545	72.36787879	72.36637143	67.19379611
5	49.32174208	76.4459375	79.00575553	65.49312314	71.33769559
	65.78543	64.35215	76.18885	71.82574	71.87336

We also tested what the influence was of cleaning the data. cleaning the data entailed getting rid of capital letter, punctuation and apostrophes. again the results can be found in the table below.



	2-gram	3-gram	4-gram	5-gram	6-gram
1	-48.9651485	-51.4758163	-57.3313402	-42.18	-59.6805263
2	-48.3227273	-41.979898	-47.1594898	-49.23	-51.6657895
3	-50.0318182	-62.85	-46.8653608	-41.29875	-73.0384211
4	-43.1954545	-50.0318182	-55.5870103	-44.82375	-46.3226316
5	-48.3227273	-52.3390816	-53.8426804	-43.376701	-59.6805263
	-47.7676	-51.7353	-52.1572	-44.1818	-58.0776

Song generation provides an interesting

6 Discussion

From the results that are obtained from the experiments of method 1 we learn a few interesting things. The most notable observation is that due to cleaning the input, the readability score becomes negative. This is caused by the fact that without capital letters and or punctuation there are no real sentences, only a string of words. The model was trained with either 5 or 10 words as input, with the next word in line as the expected output. Due to an oversight, it is probable that the NN was not trained on the entire section, but rather, a few number of songs - all models have an obsession with ABBA's song andante. As such, no experiments for this is reasonable. Sadly, by the time this was discovered, there was not enough time to repeat the experiments.

Furthermore, we can conclude that for unfiltered data, 4-gram seems to be the most readable according to the scoring function. And 5-gram is best for the filtered input. This seems to be because the larger n-grams have a more consistent result as can be seen in the table. Another reason is that it takes the context into account making more coherent sentences.

7 Conclusion

To conclude, it is hard to say what the performance of the LSTM could have been. Therefor it is hard to answer the question: how do the two compare? We can however conclude that the simple model is considered to be readable according to our scoring function. Unfortunately, when we look at the generates lyrics with human eyes, it leaves much to be desired. Given more time it would be interesting to see what the LSTM would have come up with and possibly how to improve it.

References

[Hen] Erin Hengel. *BIO/CVRESEARCHTEACHINGSOFTWAREDATA*. URL: `http://www.erinhengel.com/software/textatistic/?fbclid=IwAR2O5evMt-xbYwOuLgdhYFU9ItmhiQ3bQhrjSAOFOF8aY0p_Ytu1b4GjKqw`.

[Kuz] Sergey Kuznetsov. *55000+ Song Lyrics*. URL: `https://www.kaggle.com/mousehead/songlyrics`.