Report

NLP Project

Aaradhya Gupta
Akhilesh A

Description

The project is Hindi - English Code-mixed text generator based on a simple LSTM architecture. We have trained the model over a large dataset taken from various social media sites.

Ideas/Experiments

We initially just generated text by passing the prompt into the trained model and then querying it to select a next word based on the output it gave by processing the current state. This was the normal implementation of an LSTM LM.

- **n-gram length**
  We had a context of 3 words, i.e. we used 4-grams. unigrams and bigrams were not considered, and trigrams did not perform as well as the 4-grams.

- **Number of epochs**
  We trained 2 models one with 10 epochs and the other with 15, we didn't see any considerable change between them with respect to perplexity, so we stuck with 10 to save time while training.

- **Optimization**
  Once we achieved a satisfactory level of training after changing the hyperparameters, we moved on to optimizing the way our model predicts and generates text.
  - We have implemented beam search. We kept the beam width as 5.
  - We used CodeMixIndex(CMI) as a parameter to limit our model vocabulary.

- **LSTM and Embedding Size**
  We also varied the embedding and LSTM hidden-layer sizes between 256 and 128

Results

S.no	Embedding size	Sequence length	Epochs trained	Perplexity	Standard deviation
1.	128	4	10	764.4676547992244	4807.196195890709
2.	128	5	10	945.598710314305	8272.959912688248
3.	128	6	10	638.4913159033806	634.2646330390024
4.	256	4	10	3072.8245940979064	99443.38454764397
5.	256	6	10	5963.116983721214	224365.15602054045

Although we observe an increase in the perplexity of the models as we make them more complex, the model with embedding size 256 and sequence length 6 and 4 actually gave the most grammatically coherent and meaningful sentences in all observations.
This is because average perplexity is a misleading data point as the average is greatly affected by very few values in the order 10e4, and it is better off to compare the models using standard deviation as that would better calculate how the values of the perplexity are changing over the validation corpus.
Standard Deviation can be seen increasing over the perplexity which supports our claim.

Proposed Improvements

- Cleaning of the corpus can always be improved as we still had a lot of instances of noise in the form of emojis(:p,:d,etc) and abbreviations(lol,xd,lmao,etc)
- A more linguistically competent set of rules for language restriction can be developed
    - For ensuring the generated sentence is Code-mixed, we restricted the language that the model can use,but it was done not considering any linguistic factors.
    - Triggering theory by Michael Clyne describes how certain words trigger code switching and code mixing. So a

GitHub Repo Link

https://github.com/aforakhilesh/Code-Mix-Generation

Data and trained models Link

https://iiitaphyd-my.sharepoint.com/:f:/g/personal/aaradhya_gupta_research_iiit_ac_in/El7K9ZGgLmBEoMHKuB26WgABbP_oInNebY3LcrNjnSwkWw?e=H8Mfsd

Literature/References

For understanding the theory behind LSTMs and their architecture:

1. https://colah.github.io/posts/2015-08-Understanding-LSTMs/
2. https://web.stanford.edu/~jurafsky/slp3/9.pdf
3. https://youtu.be/8HyCNIVRbSU
4. In-class lectures

For the Implementation:

1. https://lilianweng.github.io/lil-log/2021/01/02/controllable-neural-text-generation.html
2. https://github.com/irshadbhat/csnli

For codemixed text generation

1. https://www.researchgate.net/publication/228643287_Triggered_codeswitching_A_corpus-based_evaluation_of_the_original_triggering_hypothesis_and_a_new_alternative#:~:text=In%20a%20series,one%20language%20to%20the%20other.
2. https://amitavadas.com/Code-Mixing.html