Detective Novel Text Generation using RNN LSTM

Annysia Dupaya, Pranav Rajan, Safi Benyahia
KTH Royal Institute of Technology
dupaya, pranavr, safib@kth.se

Abstract

This project focuses on Natural Language Processing from simple recurrent-neural networks(RNN) to LSTMs and if time permitting, some of the modern approaches in the NLP toolkit including word embeddings (Word2Vec, Bert, Glove) and the HuggingFace Transformer Library.

0.1 Project Goal

We plan to implement default Project 3 + the extensions.

0.2 Overview

The goal of this project is to investigate and implement different neural network architectures including recurrent-neural networks(RNNs), LSTMS, GRU, Word Embeddings and Transformers for the text generation problem. Traditional Natural Language Processing has been based on statistical and computational linguistic approaches including NER, Logistic Regression that involve significant feature engineering to produce useful results. With the success of Deep Learning and the Transformer architecture, text generation and natural language processing has been transformed with the way the models are able to find and learn patterns in the data resulting in tools such as ChatGPT, highly accurate language translation tools, search tools and more.

In this project we will be extending Assignment 4 to experiment and understand the tradeoffs and implementations of different neural network approaches to text generation. Using the RNN architecture as a baseline, we will train neural network models on text chosen from Project Gutenburg to learn patterns and generate unique text with certain characteristics such as writing style, theme and language. We will compare the baseline RNN model with one and two-layer LSTM implementations both qualitatively and quantitatively using evaluation techniques such as prediction loss, n-gram frequency and quality of generated text.

0.3 Dataset Used

The group would like to propose the works of Conan Doyle (Sherlock Holmes) and Maurice Leblanc (Arsène Lupin) from the Project Gutenberg site. We will do an 80-10-10 split for training, testing, and validation. For example in our proposed base project using the Sherlock Holmes dataset with 12 books, to account for writing style changes that can happen over time, we will split each book 80-10-10.

For example, in one book, we use 80% of the first portion, then the remaining 20% with the latter portion of the book.

0.4 Technology Used

The group is considering using the following technologies:

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

- PyTorch
- PyTorch Lightning
- HuggingFace Transformers Library
- Spacy
- NLTK

0.5 Implementation

The group plans to implement the code from scratch using the technologies previously mentioned. The group will be extending Assignment 4, and following closely with the architecture used by Vaswani et al. [1] Some papers that we may reference include the original ChatGPT paper by Radford et al. [2] and the ChatGPT2 paper. All the members of this group have taken/are currently taking a class taught by Professor Johan Boye from the Speech, Hearing and Language Division and thus are familiar with the basics of how to set this project up.

0.6 Initial Set of Experiments

The group's first set of initial experiments will be to investigate how the different training parameters affect the model (ex. batch size and learning rate). Another set of experiments the group plans to do is checking the quality of the generated text of greater length. We plan on using strategies taught during the language engineering course such as using the BERTScore, entropy and the ideas described in [3].

0.7 Milestones Timeline

- E grade:
  - Complete the basic assignment as specified for Default Project 3
- D-C grade:
  - Train on two datasets: Conan Doyle + Maurice Leblanc
  - Check the performance when training the two datasets separately
  - Create a mixed dataset and compare the performance on the mixed and initial datasets
- B-A grade:
  - Use words as the basic entry in the network and use Glove for word embeddings and investigate Byte-Pair Encoding (BPE) tokenization.
  - Compare the quality of generated text when using word embeddings or BPE tokenizations compared to the base project.

0.8 Member Skills and Knowledge

- Annysia - has knowledge of RNNs in the course Scalable Machine Learning, mostly in their usage in an encoder-decoder configuration. She also has experience working with information systems and textual information. She hopes to gain deeper understanding of RNNs by implementing them with LSTM architecture.
- Pranav - first time taking a deep learning class but has completed DD2477 and currently taking DD2417 with Professor Johan Boye and is familiar with the language engineering techniques needed for the assignment. Also had exposure to machine learning/basic deep learning theory from DD1420 and DD2434. By the end of this project, Pranav hopes to understand how to implement the different architectures in the assignment, the tradeoffs and have a deeper understanding of neural network approaches to language engineering.
- Safi - Has completed the Machine Learning class and is currently taking a Language Engineering course with exposure to word embedding techniques like word2vec and GloVe. Safi is familiar with the basic concepts of neural networks and aims to gain practical understanding of implementing RNN architectures with LSTM layers, learning how they process sequential data for text generation, and developing skills in hyperparameter tuning to improve model performance.

  Also familiar with the language engineering techniques,

0.9 Target Project Grade

The group aims to get an A for this project.

References

[1] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[2] Alec Radford and Karthik Narasimhan. "Improving Language Understanding by Generative Pre-Training". In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[3] Ari Holtzman et al. "The curious case of neural text degeneration". In: *arXiv preprint arXiv:1904.09751* (2019).