Research Design and Data Cleansing Strategies

Research Strategy and Plan

The selected strategy for the research conducted on character-level language models is quantitative. This choice was made to enable a systematic comparison and in-depth analysis of various models to evaluate their performance.

The research plan involves building and examining several models, ranging from simple n-gram models such as bigram and trigram, Multilayer Perceptron (MLP), to more complex architectures like Recurrent Neural Network (RNN), and Transformers. The intention is to assess the strengths and weaknesses of each model and gain insights into their effectiveness for character-level language modeling.

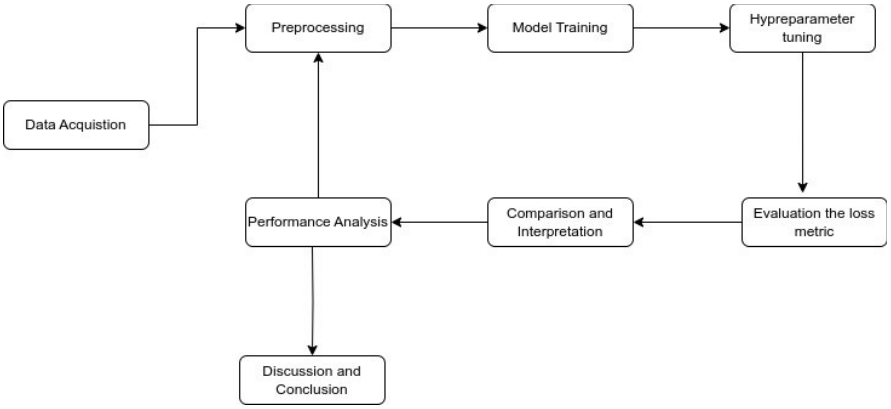

Figure 1: Flowchart of the proposed system

The plan includes following steps:

1. Literature Review: A comprehensive review of existing literature and research on character-level language models was conducted to gain insights into previous approaches, techniques, and their limitations. This step ensured a solid foundation for the research and provided valuable context for model selection, comparison as well as guidance for the hyperparameter tuning process.

2. Data Acquisition: The power of this system will be such that it will not require specific static dataset but rather any kind of name collection can be used to train and generate samples out of it. For the initial development test company names dataset is being used to train and evaluate the models but it can later be changed to test the system on different domains. It is visible from the Figure 1 that this steps goes into a loop from preprocessing and all the way up to the performance analysis. In the case of the company names dataset, quality and relevance were assessed to ensure it met the research objectives.

3. Model Development: Different models, including bigram, trigram, MLP, RNN, and Transformers, will be implemented and trained on the collected dataset. Each model will be carefully designed and configured to capture the nuances of character-level language modeling.

4. Hyperparameter Tuning: Conduct hyperparameter tuning to optimize the performance of the models. This process involves systematically varying the hyperparameters of the models, such as learning rate, batch size, regularization strength, and model-specific parameters, to identify the configuration that yields the best results. For instance, model specific hyperparameters for the MLP implementation that was referred from Bengio et.al 2003 are context length, size of the embedding space (feature vector), number of neurons in the hidden layer.

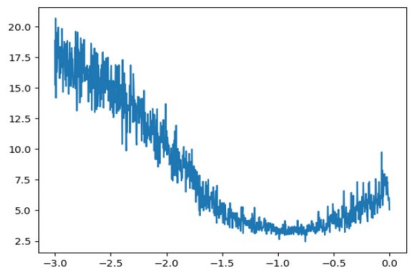

Figure 2: Tuning the learning rate and decay learning rate for MLP model

4. Evaluation Metrics: For each modeling stage, the dataset is randomly shuffled and divided into train, development (evaluation) and test set. The models are trained on the train set, tuned the parameters on the development set and at the latest stage evaluated on the test set as a final step. Cross entropy was chosen as the quantitative evaluation metric for assessing the performance of the character-level language models. Cross entropy measures the dissimilarity between the predicted probability distribution generated by the models and the actual distribution of the target data. It quantifies how well the models capture the patterns and structure within the dataset. The plot in the Figure 3 is very noisy as the model is trained on the mini-batches.

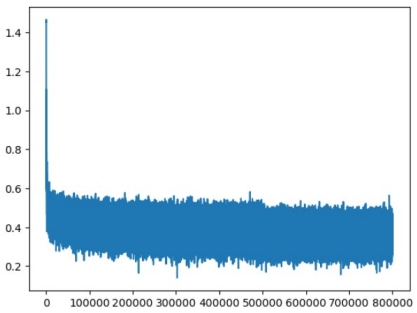

Figure 3: Negative log-likelihood loss function on the mini-batches for MLP model

5. Sampling Predictions: In order to visually see the words that are generated from the each model, the sampling is applied. This stage can be changed slightly for each of the developed model but the core idea is identical. As an example from the MLP model, the context length is 3 which means 3 character is taken to predict the next one. After the network is trained the embedding feature vector, we obtain logits (predictions by the model) and probabilities (which is obtained by softmax) to sample words from the model. Followings demonstrates the sampling from the MLP model. This model itself is intended to tune and make betters predictions using hyperparameters.

```
g = torch.Generator().manual_seed(10110608)

for _ in range(20):

    out = []
    context = [0] * block_size
    while True:
      emb = C[torch.tensor([context])] # (1,block_size,d)
      h = torch.tanh(emb.view(1, -1) @ W1 + b1)
      logits = h @ W2 + b2
      probs = F.softmax(logits, dim=1)
      ix = torch.multinomial(probs, num_samples=1, generator=g).item()
      context = context[1:] + [ix]
      out.append(ix)
      if ix == 0:
        break

    print(''.join(inttostr[i] for i in out))
```

```
rid.
forcend.
welluma.
gateway.
```

Figure 4: Sampling from the MLP model

5. Performance Analysis: The performance of each model will be analyzed in depth, examining their strengths, weaknesses, and trade-offs. Insights will be drawn from the evaluation metrics to identify patterns, trends, and areas of improvement for each model.

6. Comparison and Interpretation: The results obtained from the performance analysis will be compared across the different models. The similarities, differences, and relative advantages of each model will be highlighted. The implications of the findings will be interpreted to draw meaningful conclusions about the effectiveness of the various models.

7. Discussion and Conclusion: The research findings will be discussed in light of the research objectives and the existing literature. The implications, limitations, and potential future directions for character-level language modeling research will be explored. A comprehensive conclusion will be drawn, summarizing the key findings and their significance.

Data Cleansing

The code, analysis and findings for this stage can be found at the Github link which presents more comprehensive view of steps that are discussed in this report..

This stage is involved in the preprocessing stage which involves preparing the data for analysis or modeling. This includes various steps such as data acquisition, exploration, data cleansing. The raw data for mentioned company names

dataset. The raw companies dataset can be obtained via People Data Labs website. The dataset is about 13 million rows with following features:

- country - The country of company's current headquarters.
- founded - The foundation year of the company.
- industry - The self-reported industry.
- linkedin_url - The primary company LinkedIn URL.
- locality - The locality of company's current headquarters.
- name - The company's main common name.
- region - The region of company's current headquarters.
- size - A range representing the number of people working at the company.
- website - The primary company website.

Some of those features are useful to filter out portions of the dataset such as country, size. Also there is this idea to train the dataset only on some filters such as industry to only generate new company names that are specific to industries.

Data cleaning is an important step to ensure the quality and reliability of the dataset. It involves identifying and handling issues such as missing values, outliers, inconsistent formats, or invalid data points. Below are the steps for the data cleansing:

- Handling missing values
  - Any samples with missing name, country or size attribute is dropped since they are important in the filtering stage.
- Removing duplicates
  - This part was a bit tricky because even though duplicate deletion is performed before, they were generated after the data cleansing.
- Removing words with redundant characters
  - Only words with english alphabet are kept in the dataset since US is used to filter the dataset.
- Fixing words which has TLDs (top-level domains) in the name
  - Some words had TLDs such as ".com" in their names so they are removed.
- Filtering the dataset:
  - Using country feature
    - The country US is only used for now to reduce the size of the dataset. This is also done just because I am focusing on the English alphabet. Using English alphabet and names that are in Spanish would not make sense. But this is also not for sure, because there is probability that using different countries would generate better predictions, creative and unique words.
  - Using size of the names
    - For the sake of simplicity only names with the one word are kept in the dataset
  - Using the length of the words
    - Right now the filtering is such that minimum length is 3 and maximum is 16. There were name with more than 90 which were like random generated strings. Those can be because of the scraping by the source platform.
  - Using size of the company
    - This information is parsed from the size attribute which shows estimated min and max worker for the company as a string. This is used to filter out the companies which are less than 11.

Findings

The main section of the preprocessing of the dataset is the mentioned filtering options. Some of which are country, company size, industry, character length. In the case of the filtering using word length, the distribution of the dataset was analyzed to obtain normal distribution. But better approaches will be implemented and improved such as outliers detection can be useful. Also there is this case that some company names are abbreviated so that they are actually not names but abbreviation. This can be noisy in this particular dataset but it would not be issue in the case of person names.

Another comment is about the filtering the dataset using company size. Right now the companies below 11 is removed to reduce the size of the dataset but considering most of those companies can be small startups which has creative and nice names that can be helpful for models to learn better patterns. Best way to understand and filter prepare data, different filter options can be set and train models to see which works best and why is that.