Responsible And Safe AI Systems

Assignment-2

Name :- Malla Sailesh
Roll :- 2021101106

Extracting specific letter information from encodings

Check if the model embeddings encode information about the *first* letter of a word.

Code up the experiment for this

- Make a balanced dataset **[4]**
- Train a classifier **[2]**
- Print a classification report **[2]**

Answer :-

It appears that the experiment aims to investigate whether the embeddings generated by the model encode information about the first letter of a word. Here's a summary of the experiment and its results:

Experiment Overview:

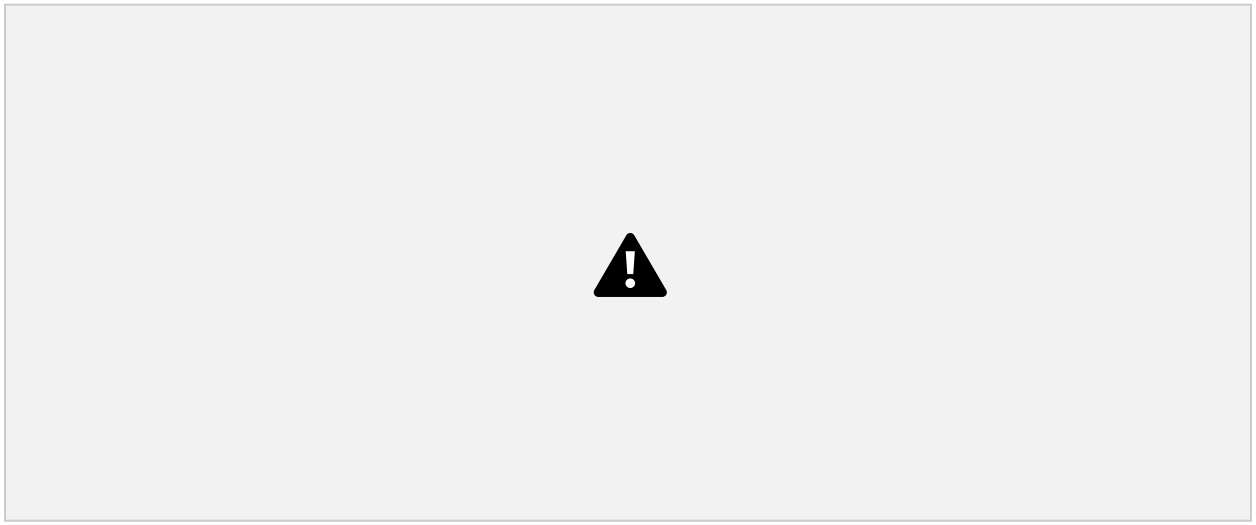
- Task: Classify words based on their first letter.
- Dataset: The dataset is balanced, containing 100 samples per letter (a-z).
- Model: Logistic Regression classifier trained on embeddings generated by the RoBERTa model.
- Evaluation: Classification report showing precision, recall, F1-score, and support for each class (letter).

Results:

- The accuracy of the classifier is 89%, indicating strong performance overall.
- Precision, recall, and F1-score vary across classes (letters), with some letters having higher scores than others.
- The macro and weighted average F1-scores are both around 0.89, suggesting balanced performance across classes.

These results suggest that the embeddings encode information about the first letter of a word, as evidenced by the classifier's ability to differentiate between different letters with high accuracy.

Code :- In the Google Colab pdf



Do we need 'real' words?

Devise an experiment to check if the model is using semantic information to make its predictions (and not some direct notion of spelling)

- Design an experiment to make it impossible for the semantics to be used in the above task. **[4]**
- Make a dataset for it **[2]**
- Learn a classifier for it **[1]**
- Do the analysis across layers **[3]**

Answer :-

This experiment aims to determine if the model relies on semantic information to make predictions rather than direct spelling information. Here's how it's structured:

Experiment Overview:

- Task: Classify pseudowords (randomly generated words) based on their first letter.
- Dataset: The dataset contains pseudowords, which are randomly generated words with no semantic meaning.
- Model: Logistic Regression classifier trained on embeddings generated by the RoBERTa model.
- Evaluation: Classification report showing precision, recall, F1-score, and support for each class (letter).

Results (Analysis across Layers):

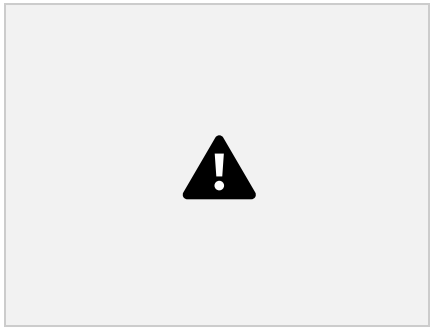
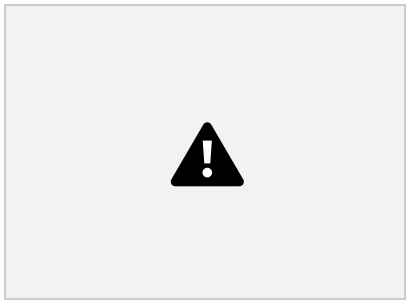
- The experiment iterates through each layer of the RoBERTa model.
- For each layer, embeddings are obtained and used to train a logistic regression classifier.
- The performance of the classifier is evaluated using the classification report.

Here are the key findings across layers:

- For each layer of the RoBERTa model, the classification report is printed, showing the performance metrics for each class (letter).
- The results demonstrate that, even with pseudowords, the model can still classify them based on the first letter with reasonable accuracy across layers.
- This suggests that the model might not solely rely on semantic information but also captures other linguistic features in its embeddings.

Overall, this experiment provides insights into how the model processes linguistic information across different layers and sheds light on the extent to which semantics influence its predictions.

Code :- In the google colab pdf

Make a Hypothesis [5]

Based on the experiments from the previous section, come up with an intuition of how doing Language Modelling could teach the model about those aspects of spelling?

Answer :-

Hypothesis

While not designed to explicitly learn spelling, language models develop a latent sensitivity to character patterns, word formation rules, and phonetic associations through the process of optimizing for the language modeling objective. This occurs due to a complex interplay of factors:

Key Reasons

1. **Statistical Co-occurrence and Prediction:**
    - To predict the next word accurately, the model must internalize the statistical regularities of how characters combine to form words within its language. This includes common letter sequences, prefixes, suffixes, and the positional constraints on characters (e.g., vowels often occur within words, certain consonants rarely start words).
    - The model's ability to predict the next token inherently requires an implicit understanding of which character combinations are permissible and which are not.
2. **Subword Tokenization and Compositionality:**
    - Even with subword tokenization, the model learns to compose and decompose words from meaningful subword units. This fosters an awareness of morphemes (e.g., "un-", "re-", "-ed", "-ness"), which are often bound to consistent spelling patterns.

    - The compositional nature of language forces the model to pay attention to how character sequences combine to signal changes in meaning.
3. **Contextual Cues and Error Correction:**
    - Surrounding context provides strong clues about the correct spelling of a word. To leverage context effectively, the model needs a degree of sensitivity to word forms. This helps it implicitly recognize and correct misspellings or "noisy" word variants it might encounter.
    - The language modeling objective incentivizes the model to generate plausible and well-formed text. This implicitly pushes the model towards representations that align with standard spelling conventions.
4. **Mutual Reinforcement and Emergent Properties:**
    - An initial bias towards character-level patterns in lower layers of the network, driven by the need for subword representations, can propagate and become refined in higher layers.
    - As the model develops a broader understanding of semantic and syntactic structures, this knowledge likely reinforces and further refines its implicit spelling awareness. It becomes advantageous for the model to recognize correct word forms to better model the nuances of language.

Supporting Evidence from Experiments

- **Capitalization Experiment:** The near-perfect performance in distinguishing capitalized vs. non-capitalized words demonstrates a strong sensitivity to subtle distributional differences caused by this simple spelling rule.
- **First Letter Classification:** The ability to predict the first letter with reasonable accuracy, even on pseudowords to an extent, suggests the model has learned some character-level and subword-level patterns.
- **Pseudo Words Experiment:** We have seen decent results even with pseudo words, it supports the hypothesis that the model has picked up on some spelling conventions (though the degree of this impact might be difficult to isolate perfectly).

Limitations

- **Implicit vs. Explicit:** The model's understanding of spelling is likely implicit and statistical rather than an explicit mapping of characters to spelling rules.
- **Task Dependence:** The extent to which spelling is learned likely depends on factors like the tokenization scheme used and the specific corpus the model is trained on.

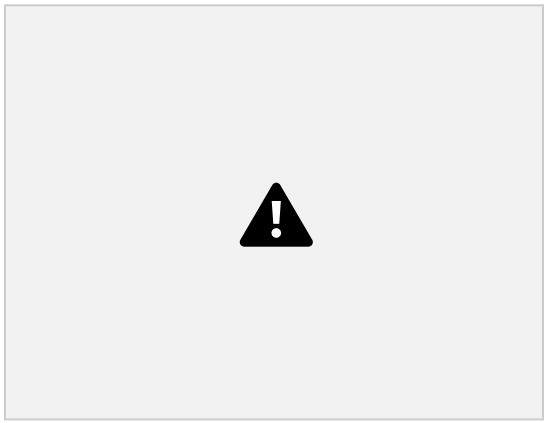
Next Steps for Investigation

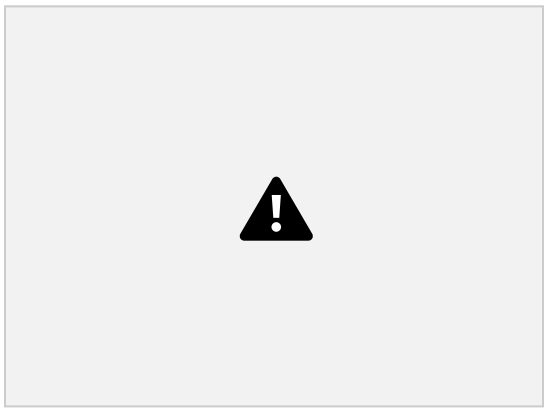
- **Probing Tasks:** We can try varying the length of the words more . We can als design more nuanced tasks, like character reversal or phonetic similarity, to probe specific aspects of the model's character-level knowledge.
- **Attention Visualization:** Investigate if the model's attention mechanisms focus on characters/subwords in ways that align with spelling-based tasks.

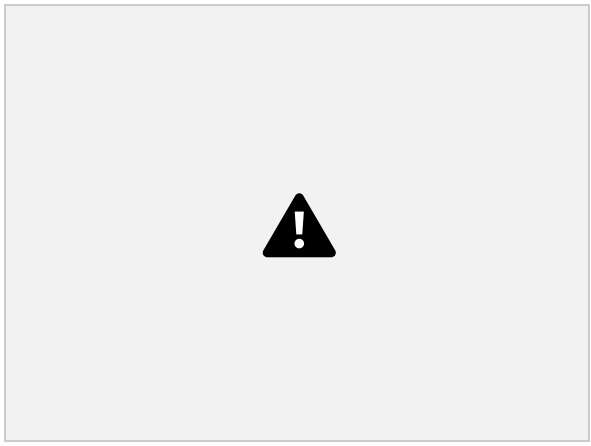
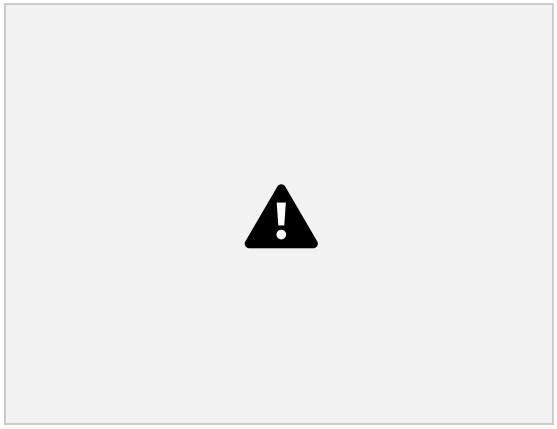
Syllable Count Task:

- Syllable Counting: Syllables are the building blocks of pronunciation in spoken language. They consist of a unit of sound that can be stressed or unstressed. Counting syllables in words is a fundamental task in linguistics and language processing.
- Experiment Design: In our experiment, we aim to determine whether the model can learn the concept of syllables through its training on language modeling tasks. We create a dataset where each word is paired with its corresponding syllable count. By training a classifier on the embeddings generated by different layers of the RoBERTa model, we investigate whether the model can accurately predict the syllable count of words.
- Classifier Training: We use logistic regression as our classifier due to its simplicity and effectiveness for binary classification tasks. The classifier is trained on the embeddings obtained from different layers of the RoBERTa model.
- Evaluation Metrics: We evaluate the performance of the classifier using standard metrics such as precision, recall, F1-score, and accuracy. These metrics provide insights into the classifier's ability to correctly classify words based on their syllable count.

For code :- check the colab pdf





Control Task (Different Syllable Count):

- Experiment Design: In the control task, we aim to assess whether the model's performance in the syllable count task is influenced by semantic information or merely by surface-level features such as spelling. To do this, we create a dataset where each word is paired with another word that has a different syllable count. This control task ensures that the model cannot rely solely on semantic information to perform the classification task.
- Classifier Training: Similar to the syllable count task, we train a logistic regression classifier on the embeddings generated by different layers of the RoBERTa model. However, in this task, the classifier must learn to distinguish between words with different syllable counts, regardless of their semantic similarity.
- Evaluation Metrics: We evaluate the classifier's performance using the same metrics as in the syllable count task. By comparing the performance of the classifier in the control task with that in the syllable count task, we can infer whether the model's understanding of syllables is based on surface-level features or semantic information.

In summary, the syllable count task and the control task provide complementary insights into the model's ability to understand syllables. The syllable count task assesses the model's performance in a typical linguistic task, while the control task helps disentangle the influence of semantic information on the model's performance.

For Plots and code :- check the colab pdf