**Report on LSTM Model for POS Tagging**

**Introduction:** This report presents the development, training, and evaluation of an LSTM model for pos tagging. The task involves converting sentences into their vector representations and predicting part-of-speech tags for each word in the sentence. Various preprocessing steps, model architecture, training details, evaluation metrics, and potential improvements are discussed.

**Preprocessing:**

* Sentences are segmented into lists of words, with each sentence limited to a maximum of 15 words. Longer sentences are split to adhere to this constraint.
* A vocabulary dictionary is created, mapping each word to its GloVe embedding vector of 300 dimensions. Words which are not present GloVe vectors are assigned zero vectors.
* Part-of-speech tags are encoded into numerical labels using label encoding techniques.
* Labels are further converted into one-hot vectors to represent each tag.
* Sequences of words and corresponding labels are padded to ensure uniform length, using zero-padding for both the word vectors and label sequences.

**Model Architecture:**

* Input Shape: (8081, 15, 300) (8081 sentences with each containing 15 words, each word represented by a 300-dimensional vector)
* Output Shape: (8081, 15, 47) (8081 sentences with 15 labels for corresponding words, each label represented by a one-hot vector of 47 dimensions)
* LSTM Units: 100
* Dropout Rate: 0.2
  + LSTM Layer to capture temporal dependencies.
  + Dropout Layer for regularization.
  + Dense Layer for classification using softmax activation.

**Model Training:**

* Optimizer: Adam
* Loss Function: Categorical Crossentropy
* Metrics: Accuracy
* Early Stopping: Patience=3
* Epochs: 23
* Batch Size: 32
* Validation Split: 12%
* Test Split: 20%

**Evaluation:**

* Train Accuracy: 76%
* Validation Accuracy: 72%
* Test Accuracy: 71.21%
* Performance Metrics (Test Data):
  + Average Precision: 88% (weighted precision for all classes weights being supports)
  + Average Recall: 71% (weighted Recall for all classes weights being supports)
  + Average F1 Score: 67% (weighted F1 score for all classes weights being supports)

**Improvements:**

1. **Contextualized Word Vectors:** Utilize deep contextualized word vectors for better contextual understanding.
2. **Variable-Length Input:** Allow variable-length input sequences to handle longer sentences.
3. **Metrics Selection:** Use precision or recall for specific use cases instead of relying solely on accuracy.
4. **Transformer Models:** Explore transformer-based architectures like BERT for improved performance using attention mechanisms.

**Conclusion:** The LSTM model demonstrates promising performance in pos tagging, achieving competitive accuracy on the test data. Further improvements can be made by adopting advanced word representations and model architectures tailored to the task's requirements.