**MOSAIC PS-1**

***Detailed Description of Solution***

The Problem statement revolves around a masked language modelling task wherein a masked word in unseen data ought to be predicted. The primary logic used to accomplish this task is to ***mimic the human interpretation of filling in a blank word in a sentence by analyzing the positioning of different parts of speech in sentences individually so as to analyze which part of speech to use in what form most aptly to fill in the blank.*** Thereby the approach is to learn the missing word by learning individual parts of speech like nouns, pronouns, verbs, adverbs, prepositions, adjectives, adverbs etc., their usage in the correct form and their positioning in the sentence.

The procedure followed to accomplish this is as follows-

* First ***the pre-processing step*** is applied to the training data which includes lowercasing the whole sentence and removing punctuation marks. This is done to remove error from the tokenization process since if the data is not lowercased, and the same word has been used in two separate positions in the same sentence, supposing “The” and “the”, they might be treated differently while tokenizing. Punctuation marks again are removed to avoid unnecessary confusion while tokenization, since punctuation mark may be tokenized along with a word.
* The next step includes tokenization of every sentence in the training dataset to words. This has been achieved by using the ***spacy*** library so as to ***ensure best performance since spacy has much better and higher accuracy as compared to nltk, or use of simple regex or .split() method of python.***
* The third step includes converting these words to logical, semantic numerical representation or in other words vectorizing them so as to feed them to the neural network(Bidirectional LSTM in our case) since NNs are mathematical models with the capability of interpreting and manipulating mathematical data or numbers. In order to achieve this, the Google-News Word2Vec(300D) model has been used. It is a pre trained model trained on approximately **100 billion words** from Google News. The vocabulary contains around **3 million words and phrases**.

Hence, for better generalization, enhanced semantic and contextual understanding, it has been used to generate word embeddings.

(downloaded from Kaggle: <https://www.kaggle.com/datasets/leadbest/googlenewsvectorsnegative300>)

* The subsequent and most significant step is the masking of the different parts of speech in the training data set. For this purpose, the dataset is divided into 5 equal parts(since there are 50,000 sentences), wherein for every 10,000 sentences a main POS category has been defined. After creating a spacy doc object, which is a structured object created after undergoing the NLP pipeline also containing the different POS for a given text, is first created. The main parts of Speech in the solution are nouns ,verbs ,adverbs ,adjectives and determiners. Now these parts of speech have been arranged in the order ***adjective, verb, adverb, noun, determiners*** in the list of the POS categories. This order has been determined by hit and trial, that is tracking the number of skipped sentences by interchanging the positions of different POS and finally for this arrangement, ***1461 sentences are left unmasked, utilizing 97.078 % of the training data.***

Also apart from the main POS categories, a list of fallback POS categories also has been created, which include Prepositions, Auxillary verbs, Pronouns, Conjunctions(CCONJ and SCONJ as defined in spacy). Now for sentences in which a given POS is to be masked and is not present in that case the fallback POS are masked in the exact above given order( also determined by hit and trial- for masking as many sentences as possible).

**Refinements from previous methods and rationale behind choosing this given method of masking only:**

The first strategy applied was masking of random words, a single index is chosen randomly and the word at that given index is masked. Also customized vocabulary was generated on the training dataset using Word2Vec. This strategy when tested via the model’s evaluation led to the following issues-

* Repitition of a given set of words again and again
* Very poor retention of context and accuracy
* Imbalanced masked data since across a number of sentences the same word was masked, like ‘the’

The second strategy applied was masking 15% of random words in the sentence and using a pre-trained word embedding like that of Google-News for training. Issues with this approach-

* Imbalanced masking of data pertaining to the length of the sentences. Since the training data set is a collection of sentences of length approximately 11-12 words per sentence on an average with some very short sentences too of 4 words, short sentences were left unmasked and most sentences were subjected to masking of a single word, some of two words but not providing good semantics and context.

The third strategy applied was a frequency-based POS masking to achieve robustness and a balance across all POS categories in the dataset. Under this strategy first across the main POS categories the sentences were divided and masked and in the remaining unmasked sentences, fallback categories were masked. However this time using a python dictionary a frequency count of each masked POS is mapped and maintained. After both the masking procedures applied, the unmasked sentences were then masked taking into consideration all POS categories depending on which has the least frequency, that should be masked first.

Issues with this strategy-

* Overfitting and less generalization, since the cosine loss on cross validation dataset was much higher, also because balanced masking could not be achieved and more sentences being masked across diverse POS created confusion and increased perplexity.

Thus finally taking into account all these drawbacks, the strategy of main as well as fallback POS masking has been implemented.

**Training of BiLSTM Model:**

A bidirectional LSTM has been used for the purpose of masked word prediction. The choice of BiLSTM comes from the fact that for predicting a masked word, understanding the right and left context of the word is significant along with context retention. Since long term context retention is an issue with simple RNNs, hence using LSTM is a much better choice. However evaluation is not to happen in one direction only(for example in next word generation/recommendation LSTM may be good choice), since both right and left context are essential to predict the masked word correctly, a Bidirectional LSTM has been used.

The model utilized is Sequential, creating using tensorflow, which has three primary layers-

* The first layer is a BiLSTM layer with 256 units
* The second layer is a BiLSTM layer with 128 units and sequence = false, that is returning the last hidden state
* The third layer is a simple fully connected layer with linear activation, consisting of 300 units(since output is raw 300D embedding-hence more of a regression approach)
* After subsequent layers, 30% of neurons are randomly dropped to perevent overfitting

**Evaluation Metrics-**

(As described in the evaluation metric report)

**Hyperparameter Tuning-**

* Number of epochs=11
* For losses: alpha=0.7 and beta=0.3
* Learning rate=0.001(standard)
* Number of BiLSTM units in the first layer=256
* Number of BiLSTM units in the second layer=128
* Dropout=0.3

The above hyperparameters have been fine tuned by monitoring the cosine loss as well as the mse loss and bettering the performance of the model on test data.

**Final Prediction of words:**

After generating the predicted word embedding, using the most\_similar method of Word2Vec, the top 12 words are generated . These words are then subjected to a regex filter for filtering only the lowercase words and the first matching lowercase word is returned. In case no matching word could be found, “the” is returned as the fallback word. In case there is a KeyError(the embedding predicted is not in the vocabulary), a token “<UNK>”(unknown token) is returned.