CS626 A1

Mohammad Ali Rehan, Neel Aryan Gupta, Shreya Pathak

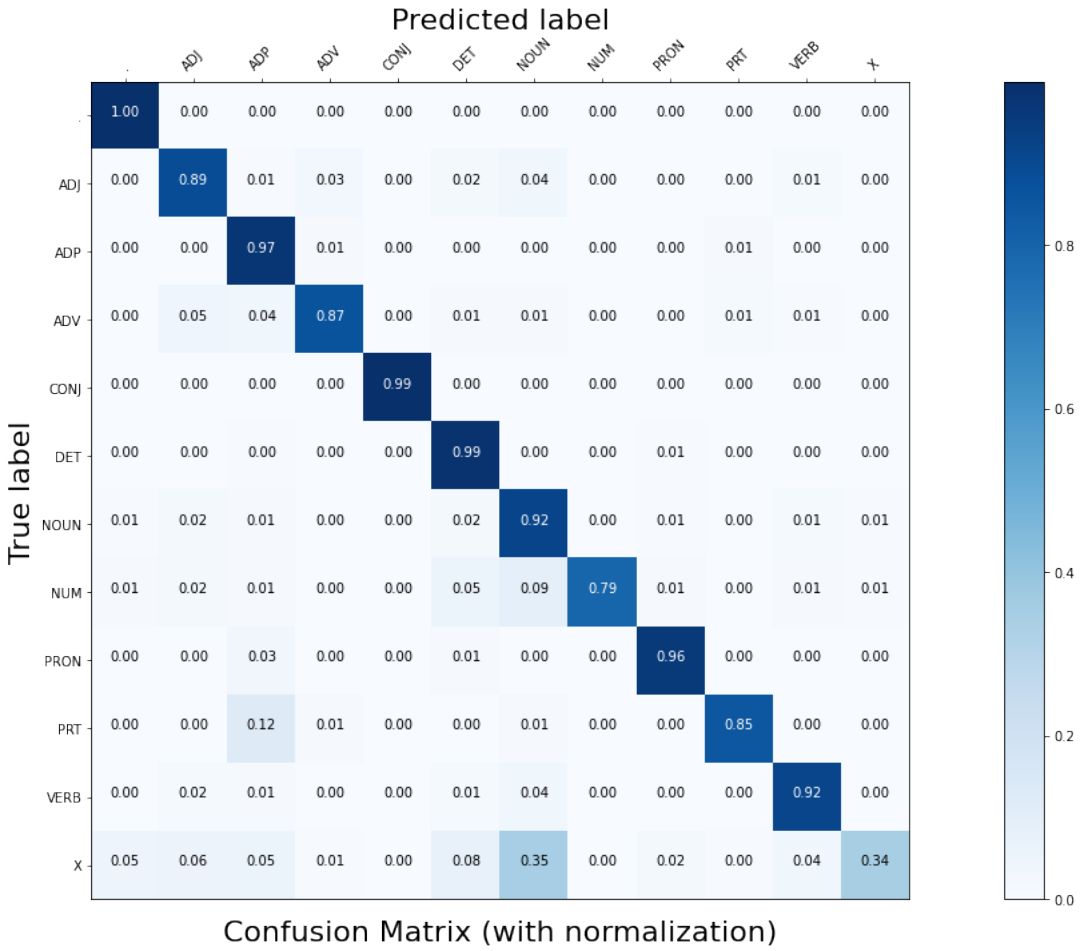
180050061, 180050067, 180050100

1 POS tagging using HMM

Using HMM, we are able to attain an accuracy of around 94% as shown below

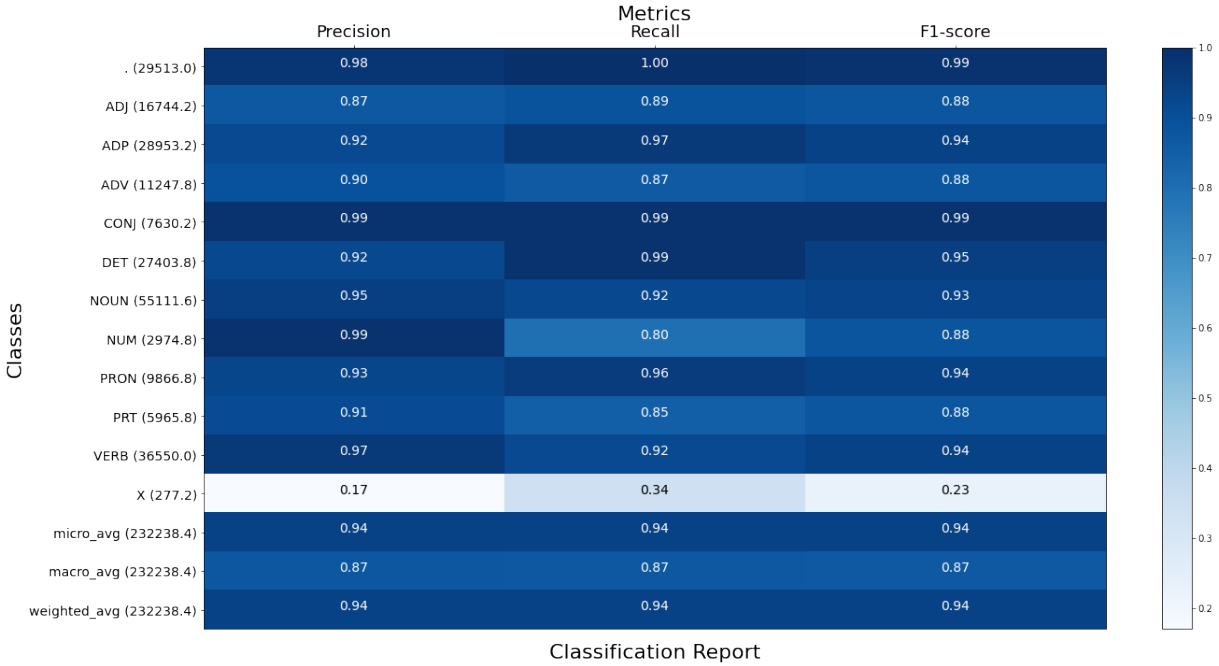
Fold number	Test Accuracy
1	0.9385
2	0.9385
3	0.9381
4	0.9391
5	0.9383

The confusion matrix is shown below



The HMM model's prediction are based on the emission and transition probability, i.e., the probability of a word having a tag (given the corpus) and the probability of a tag to follow the previous tag. The model achieves good

accuracies for conjunctions, pronouns, adpositions etc. Adjectives, Nouns are sometimes misclassified as determiners as they can occur in the positions often occupied by determiners thus causing the transition probability to be high. Verbs are sometimes misclassified as nouns as the same word can have both roles (-al rule) which will affect the emission probability. Adjective and adverb tags are also sometimes misclassified as the other. This can be attributed to the fact that same words can be used sometimes as adverbs and adjectives thereby affecting the emission probability.Also nouns and verbs are sometimes misclassified by our model so the transition probability would change accordingly. Also particles are confused with adpositions as they tend to occur in similar positions. The classification is shown below



The HMM model is a simple model in terms of implementation and complexity. There is no explicit training phase and the parameters are calculated empirically. Yet, it is able to achieve a fairly good accuracy on the brown tagged-sents corpus. Furthermore the time complexity is linear in the size of the data, thus, the algorithm is fairly fast. But, the time complexity is quadratic in the number of tags, so while the algorithm just takes around 1-2 minutes using the universal tagset containing just 12 tagsets, the time increases drastically for a richer tagset. Also, it is a short-sighted model, only looking at the previous tag and the current word to make a prediction. Thus, it makes mistakes while classifying words which can function as multiple POS tags and also if an uncommon pair of POS tags occurs consecutively. It is unable to incorporate more complex features of the text. Thus, while it does perform well for a simple task like POS tagging, it will not be useful for a more complex task.

The task of POS tagging using HMM is a stochastic approach which includes frequency, probability and statistics. It uses the Markov principle, i.e., the next state depends solely on the current state. It uses two types of probabilities, that of a word of having a particular POS tag and that of a particular tag 2-tuple, and calculates predictions on the basis of this. These probabilities are calculated empirically, obeying the principle of Maximum Likelihood Estimate. It uses the Viterbi algorithm applied using an HMM to find the outcome of a given sentence. The algorithm is a dynamic programming approach to reduce the complexity of the algorithm from exponential to polynomial. For out-of-vocabulary words we also implemented add 1-smoothing to get relevant results. Seeing the results we can conclude that the task of POS tagging is a fairly simple task and hence towards the bottom of the NLP stack as we are able to attain fairly high accuracy with a simple model. We also observe some pitfalls of the model which can be attributed to the fact that it does not incorporate rich features of the data and thus gives a way to achieve better results.

Some misclassified examples:

- (’$\hat{}$’, ’$\hat{}$’), (’issue’, ’VERB’), (’jury’, ’NOUN’). Here ’issue’ is classified as a noun by the model instead of a verb. This can be because nouns often tend to follow punctuation and ’issue’ in itself is used as a noun as well.
- (’go’, ’VERB’), (’higher’, ’ADV’), (’in’, ’ADP’). Here ’higher’ is classified as an adjective which can be explained as ’higher’ is often used as an adjective thus increasing the emission probability.

- ('louis', 'NOUN'), ('crump', 'NOUN'), ('of', 'ADP'). Here 'crump' is misclassified as a verb. Since, 'crump' is an uncommon word, due to the combined effects of smoothing and the fact that nouns are often followed by verbs thereby increasing the transition probability, we get that 'crump' has been wrongly tagged.

Some of the most misclassified trigrams are:

- ('ADP', 'NOUN', 'NOUN'), 2698 times
- ('DET', 'VERB', 'NOUN'), 2455 times
- ('DET', 'NOUN', 'NOUN'), 2399 times

2 POS tagging using SVM

Using a Multiclass SVM classifier, we are able to attain an accuracy of roughly 95% as shown below :

Fold number	Test Accuracy
1	0.9526
2	0.9528
3	0.9543
4	0.9525
5	0.9524

Support Vector Machines (SVM's) are supervised machine learning models typically used for binary classification, they are more robust versions of a standard perceptron. In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.
For this POS tagging task with the Universal Tagset of 12 basic tags, a linear multiclass SVM [1] was used, with the loss function being hinge loss. An anthropomorphic interpretation of the loss is that the SVM "wants" to have the scores for the correct class higher than the incorrect classes by atleast a fixed margin Δ (which behaves the same as the regularization parameter λ).

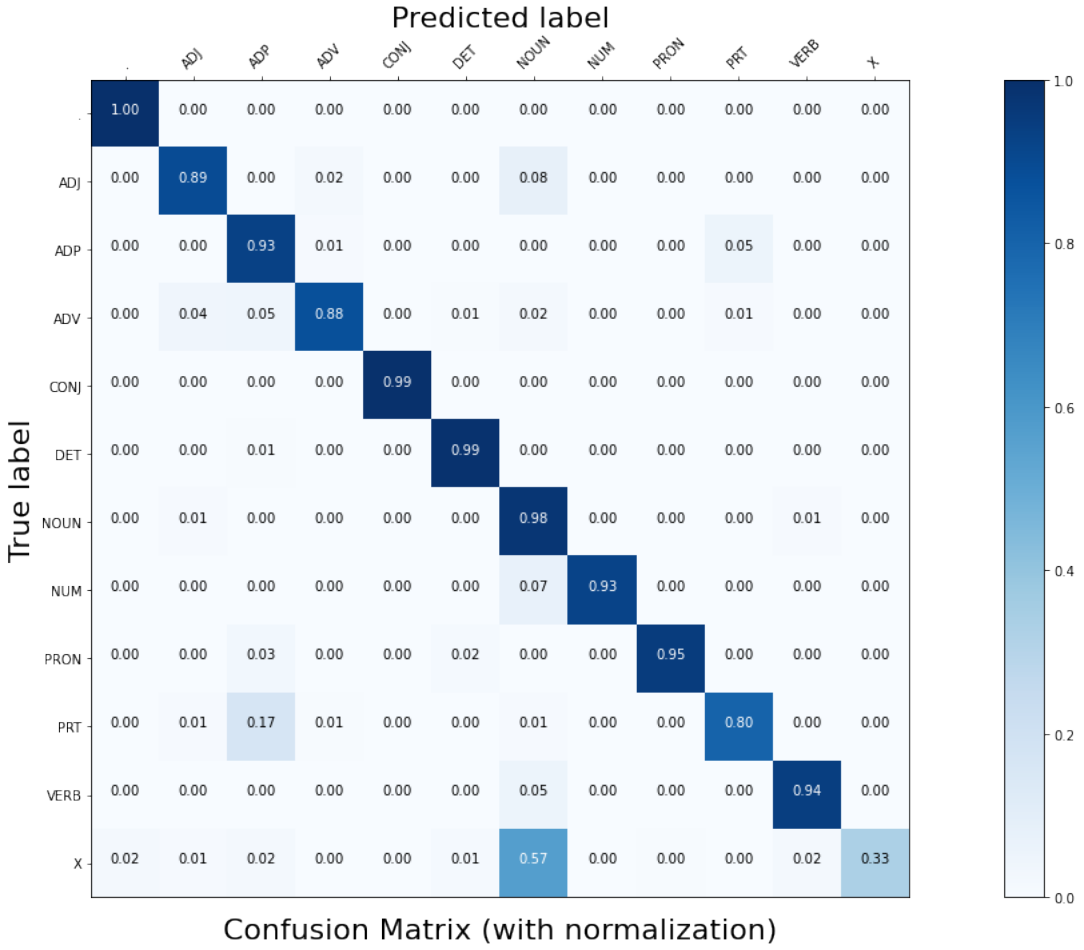
$$L_i = \sum_{j \neq y_i} \max(0, w_j^T x_i - w_{y_i}^T x_i + \Delta)$$

$$R(W) = \sum_k \sum_l W_{k,l}^2$$

$$L = \underbrace{\frac{1}{N} \sum_i L_i}_{\text{data loss}} + \underbrace{\lambda R(W)}_{\text{regularization loss}}$$

where w_j is the j^{th} row of W reshaped as a column and y_i is the true class label of the i^{th} data point (feature vector) x_i.
Note that the magnitude of the weights W has direct effect on the scores (and hence also their differences): As we shrink all values inside W the score differences will become lower, and as we scale up the weights the score differences will all become higher. Therefore, the exact value of Δ is meaningless and we can set it to 1.0, and control the regularization parameter λ instead.
The confusion matrix is shown below :

[1] adapted from CS 231n (Stanford University) - https://cs231n.github.io/linear-classify/#svm



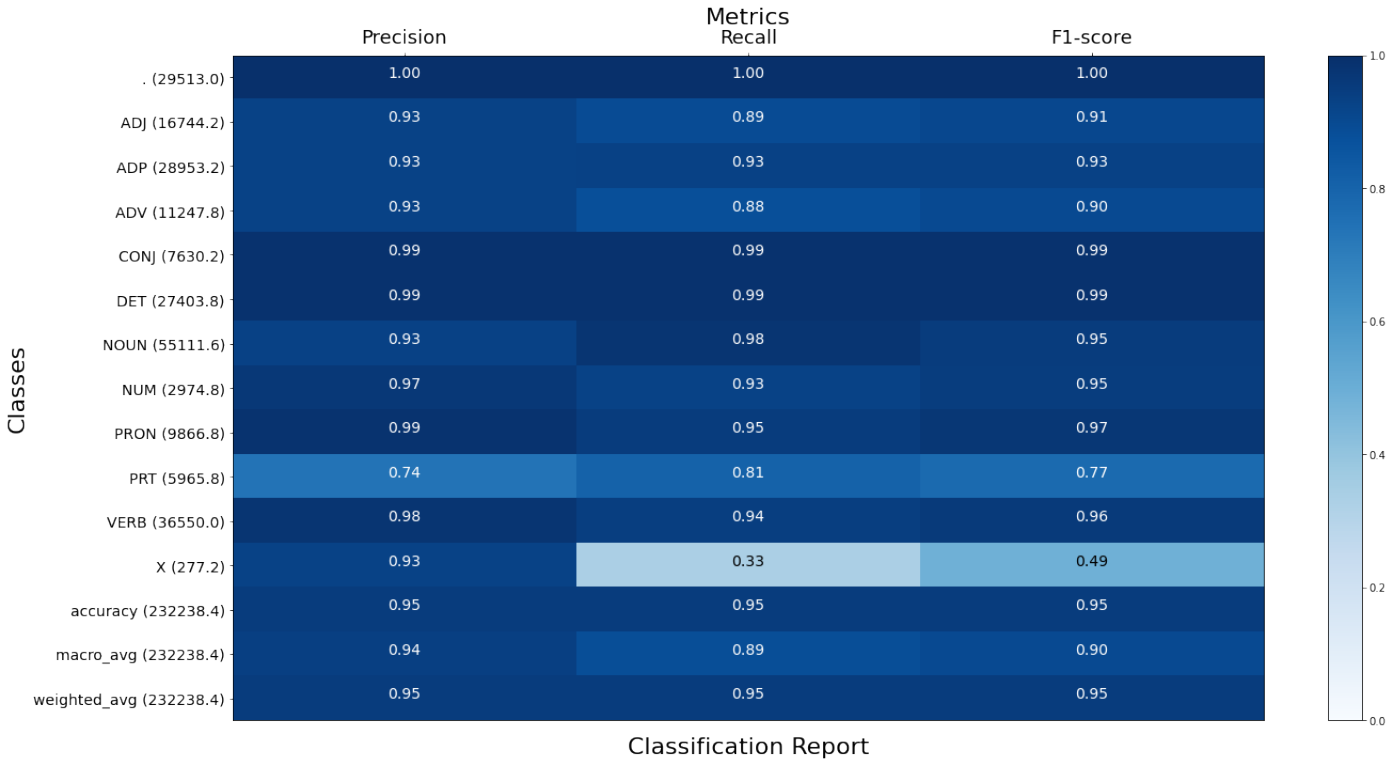
We see that recall scores for 'X' (Unknown) tag are very low, for which the major reason is the very small number of words in the training corpus which are tagged as 'X', and the predicted tag for such words is noun which happens due to the context backoff being set as noun. Verbs misclassified as nouns and particles misclassified as adpositions often occur together, that is the tuple (PRT, VERB) is often misclassified as (ADP, NOUN). Both these pairs of tags often occur together in the dataset, and the incorrect labelling of PRT causes the incorrect NOUN labelling due to the contextual dependence of word features on previously seen tags. Particles such as 'to', 'on' commonly occur as both as both particles as well as adpositions, which can possibly result in same feature vectors. This PRT misclassification majorly occurs because of the aforementioned reason only, as the feature vector for context of words such as 'to' is inadequate, so the majority tag for such features was selected as the true class label, as ideally, the training data should have a unique label per data point (feature vector). This can however be avoided with better and more complex word feature vectors. Another misclassification, although rare, is adverbs being mistaken for as adjectives. This probably happens because of their similar properties of describing the following word.

The most common misclassified words of the SVM model are shown below :

- 'to' (2011) : owing to the misclassification between the particle and adposition classes
- 'that' (659) : due to its occurrence before a NOUN and a VERB resulting in confusion between adposition and adverb tags respectively, also accounts to the error for Noun and Verb labels
- 'as' (261) : follows the same misclassifications as 'that' but with fewer instances
- 'her' (186) : due to the misclassification between the determiner and pronoun classes.

Note that most of the above misclassifications could have been avoided if our model accounted for the future context i.e. the words following the current word.

The classification report is as follows :

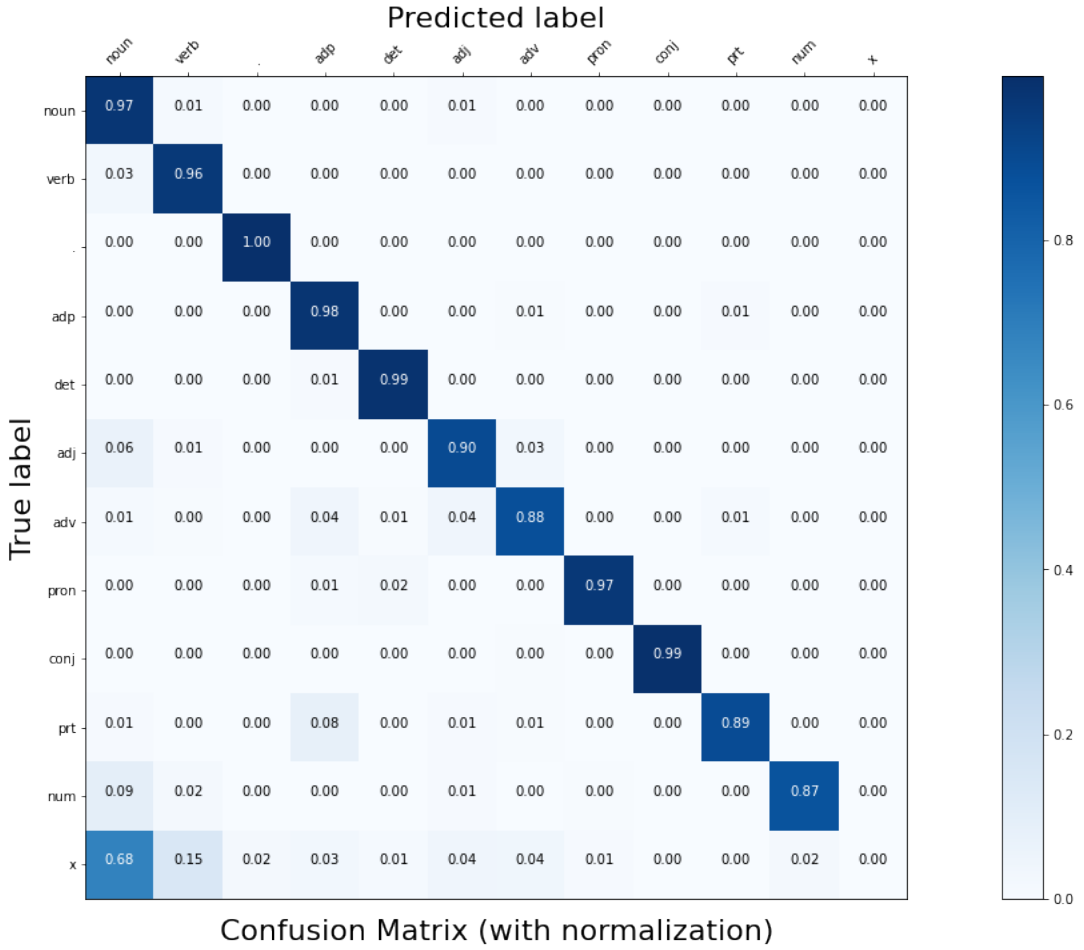


3 POS tagging using LSTM

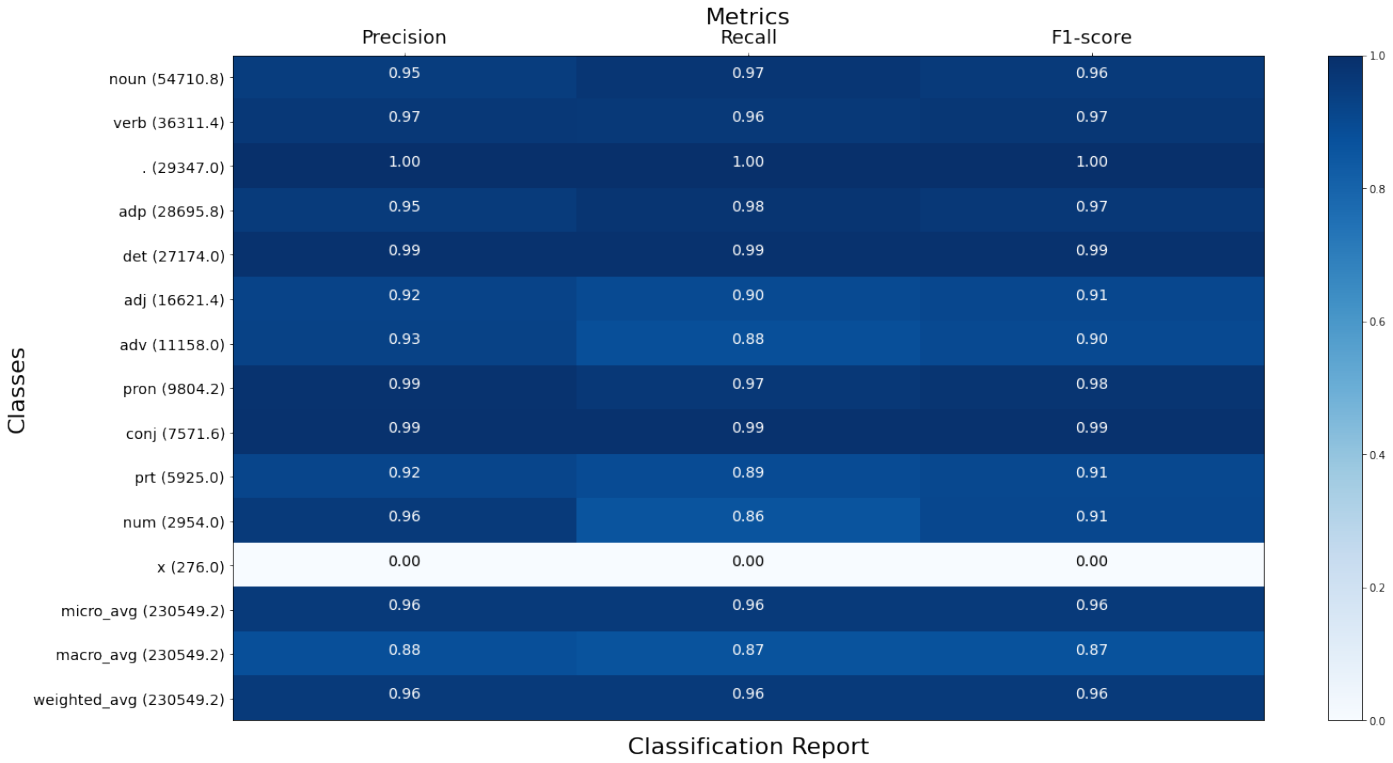
The LSTM model is able to carry out POS tagging quite well , as can be seen by the metrics below:

Fold number	Train Accuracy	Train Loss	Val Accuracy	Val Loss
1	0.9764	0.1310	0.9616	0.1411
2	0.9772	0.1303	0.9629	0.1420
3	0.9754	0.1328	0.9590	0.1435
4	0.9745	0.1327	0.9625	0.1423
5	0.9771	0.1329	0.9628	0.1443

The confusion matrix is shown below:



We can see that most classification are correct(weights on diagonal). Nouns are at times misclassified for verbs or adjectives perhaps owing to the 'al rule'. Verbs are mistaken for nouns although rarely. Misclassification is more for adverbs, as they role in describing something is quite similar to adjectives, Adpositions also occur close to words when they describe different objects relating to the action of the verb, That is why adverbs are confused for adpositions also.Compound words like "give up" etc. also complicate adposition and adverb demarcation. Because of sentence truncation and lack of occurrences of "x: we do not have enough data to classify "x" tag properly, That is why it is mostly classified as noun, which might be expected that a POS tagger tends to classify new or unworn words into proper nouns. Similar phenomenon occurs in NUM when the POS tagger might get a number it hasn't seen before, it classifies it as a noun or adjective in case the number is followed by a noun perhaps.



Some misclassification examples of POS LSTM are shown below

- 'to' in 'steps to remedy' was given the tag of adposition which is a common usage of 'to. Its rare occurrence as a particle in infitiv form wasn't detected
- 'top' in 'a top official' official was mistaken as a noun instead of adjective
- 'worth' was treated as an adjective instead of 'noun'
- 'consulting' in 'top consulting firm' was classified as verb.

Some of the most wrongly classified trigrams are

- ('det', 'adj', 'noun'): 2157 times
- ('det', 'verb', 'noun'): 601 times
- ('det', 'adv', 'adj'): 580 times

This is accordance with the cases highlighted above of adj being tagged as adv and adj being treated as verb if the words are same(eg soaring). Ambiguity in nouns also exists like 'raises' which can also be a present simple tense verb. This is also observed in the dataset.