CS 626 - A1

Mohammad Ali Rehan, Neel Aryan Gupta, Shreya Pathak

180050061, 180050067, 180050100

POS tagging

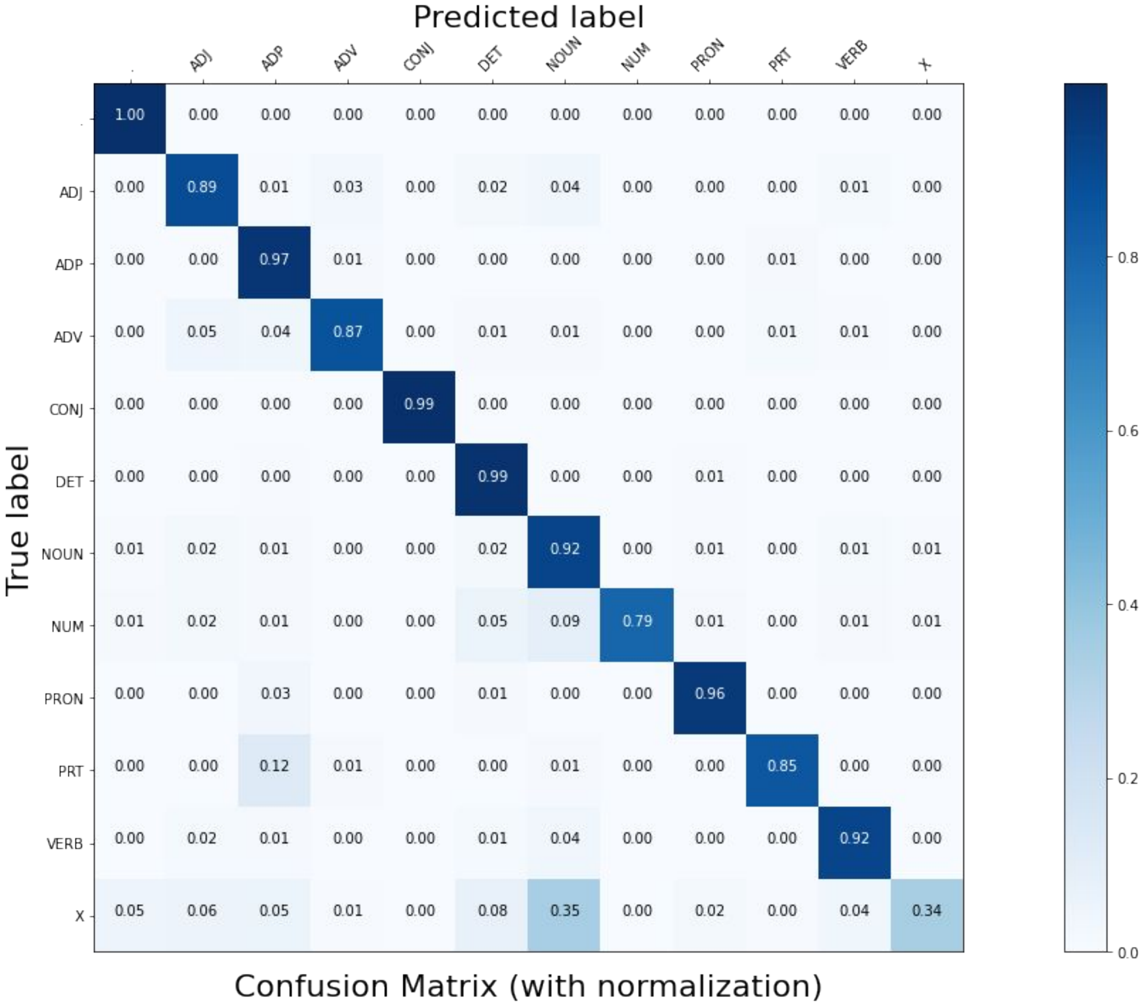
- Using HMM
- Using SVM
- Using LSTM

USING HMM

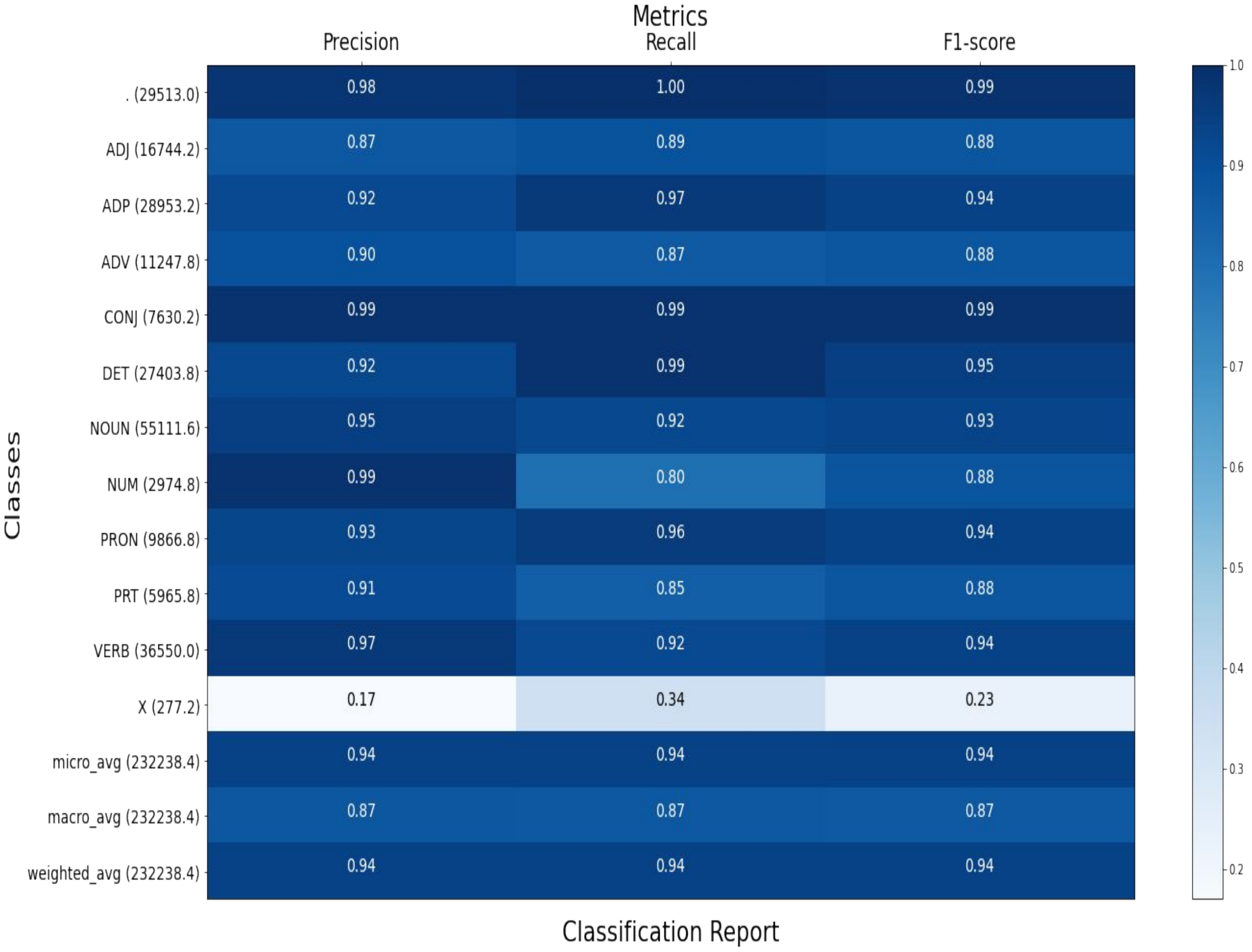
- Stochastic method.
- Uses training data to find empirical transition and emission probabilities.
- Viterbi algorithm is then applied to an HMM to predict tags for unseen statements.
- Add 1-Smoothing is used for OOV words

The model attains an average accuracy of around 93.8% across all folds. High per-POS accuracy (98-99%) is obtained for several tags. Some tags are misclassified due to possibility of multiple tags for a specific word and less probable tag sequences, eg, adjectives and adverbs are misclassified as each other. Lack of complex features and no long range context could be a possible reason for the errors.

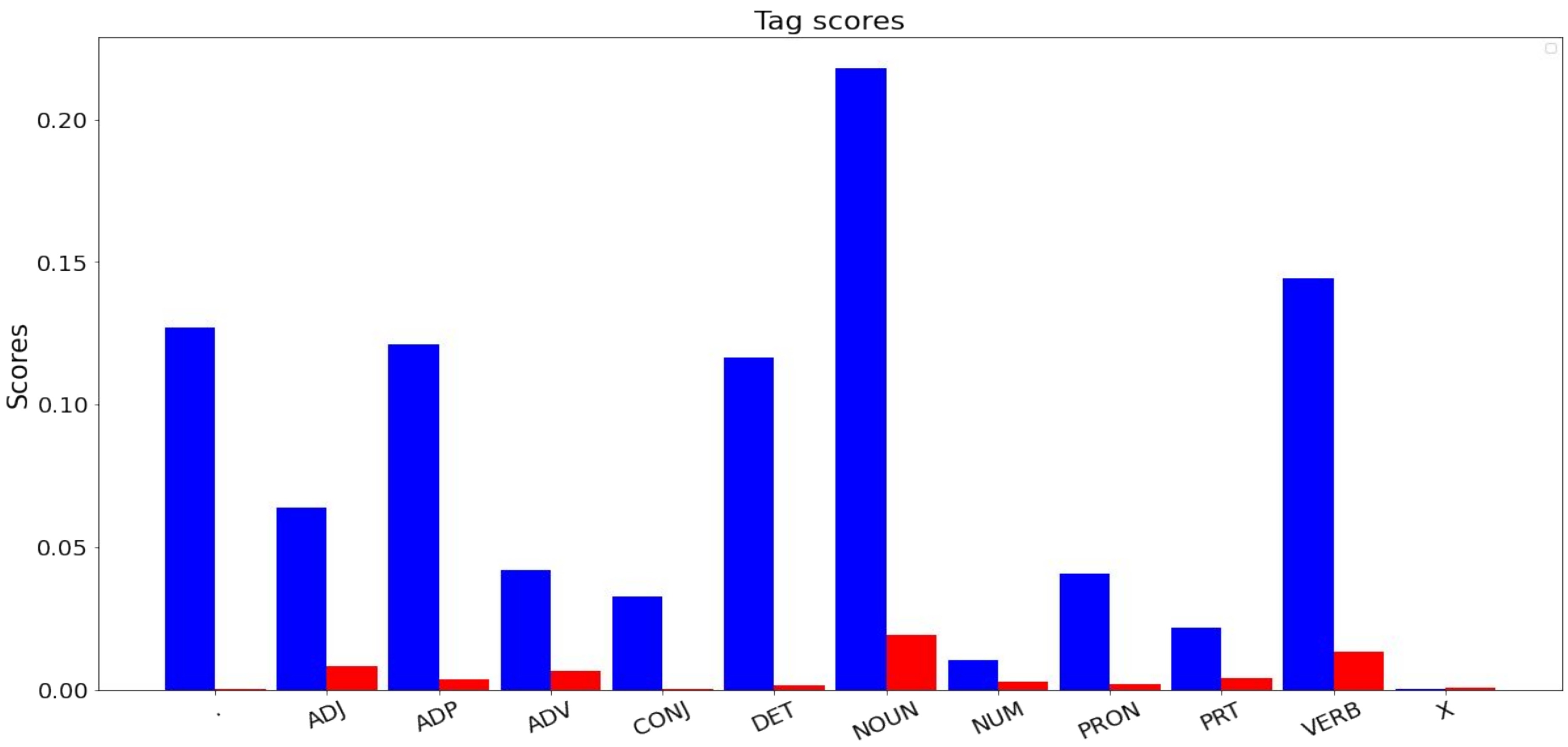
Confusion Matrix



Classification Report



Tag scores



Tag scores
Scores
0.20
0.15
0.10
0.05
0.00
.
ADJ
ADP
ADV
CONJ
DET
NOUN
NUM
PRON
PRT
VERB
X

Misclassifications (some examples)

The following phrases were misclassified by the model

- ('\^', '\^'), ('issue', 'VERB'), ('jury', 'NOUN'). Here 'issue' is classified as a noun by the model instead of a verb.
- ('go', 'VERB'), ('higher', 'ADV'), ('in', 'ADP'). Here 'higher' is classified as an adjective
- ('louis', 'NOUN'), ('crump', 'NOUN'), ('of', 'ADP'). Here 'crump' is misclassified as a verb.

These can be attributed to common tag sequences as well as multiple possible tags for the same word.

Misclassifications (continued)

The trigrams which are maximally misclassified by the model are shown below (Note that the middle word is the one misclassified in these cases):

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

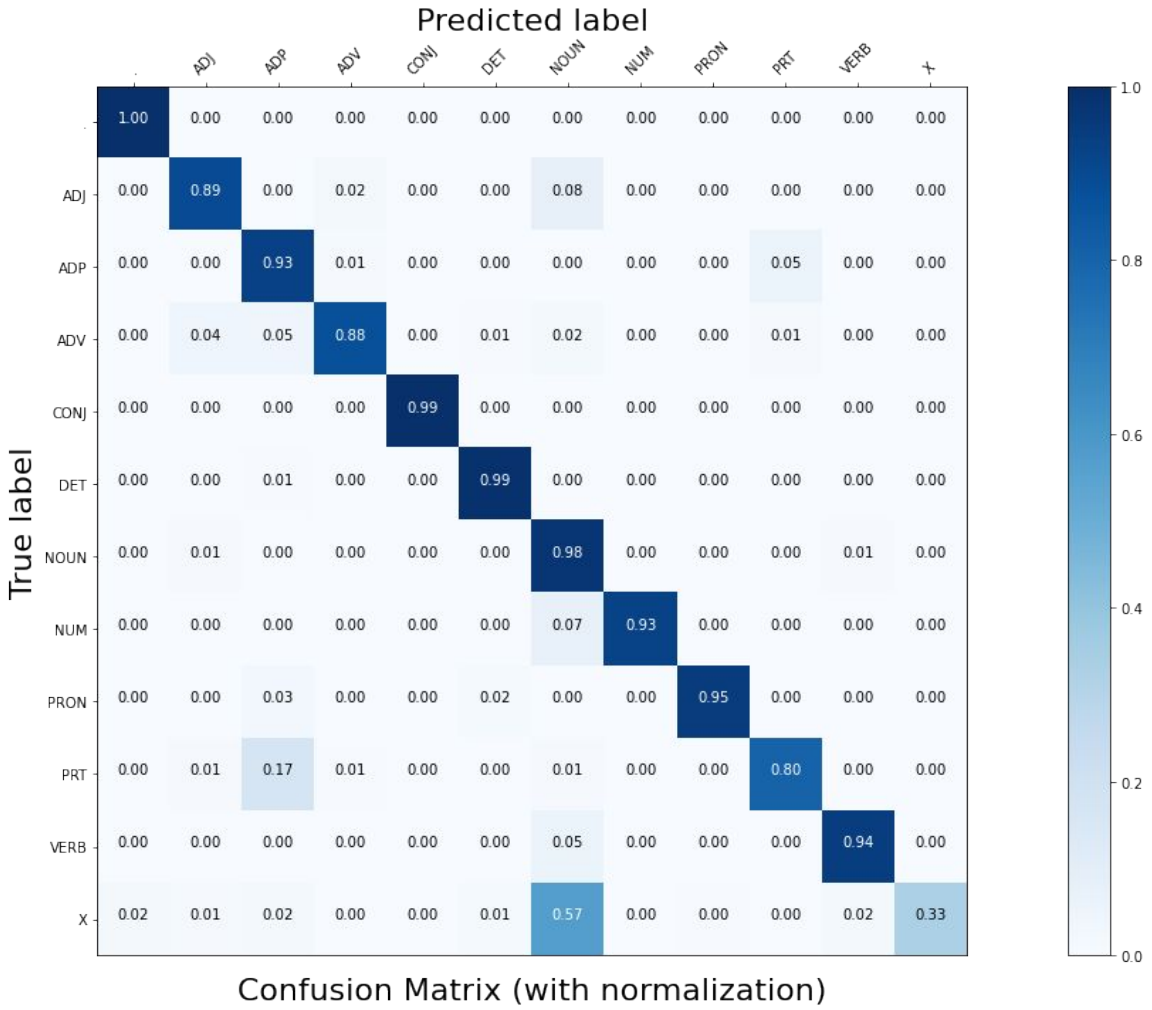
These errors can again be attributed to transition and emission probabilities

Using an SVM

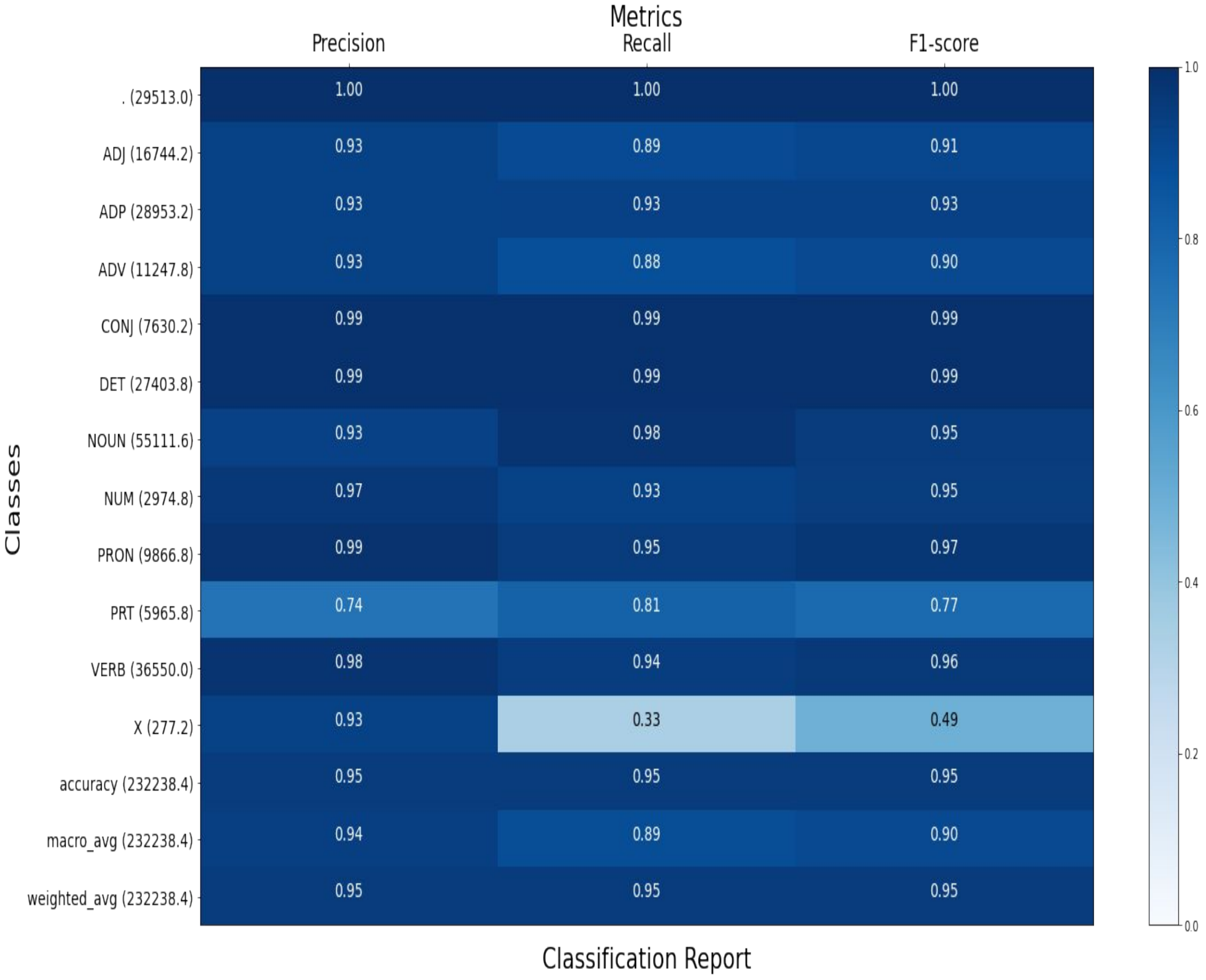
- Trained using Stochastic Gradient Descent (SGD) optimization
- Trigram-based count (with backoff) feature vectors
- Hinge loss function was used with L2 norm regularization

- Model works great with about 95 % accuracy across all folds (5-fold cross validation)
- Simplistic features were used, trigram-based counts (previous 2 tags + current word) with backoff added as lower ngram-counts, and some morphological features pertaining to the current word
- High per-POS accuracy for most of the tags is seen
- Tag ‘X’ is poorly classified (high precision, low recall), owing to the insufficient data for the tag
- Common misclassifications : (PRT, VERB) --> (ADP, NOUN)
- Backoff for unknown words goes to NOUN tag as complex unknown words are likely to be nouns

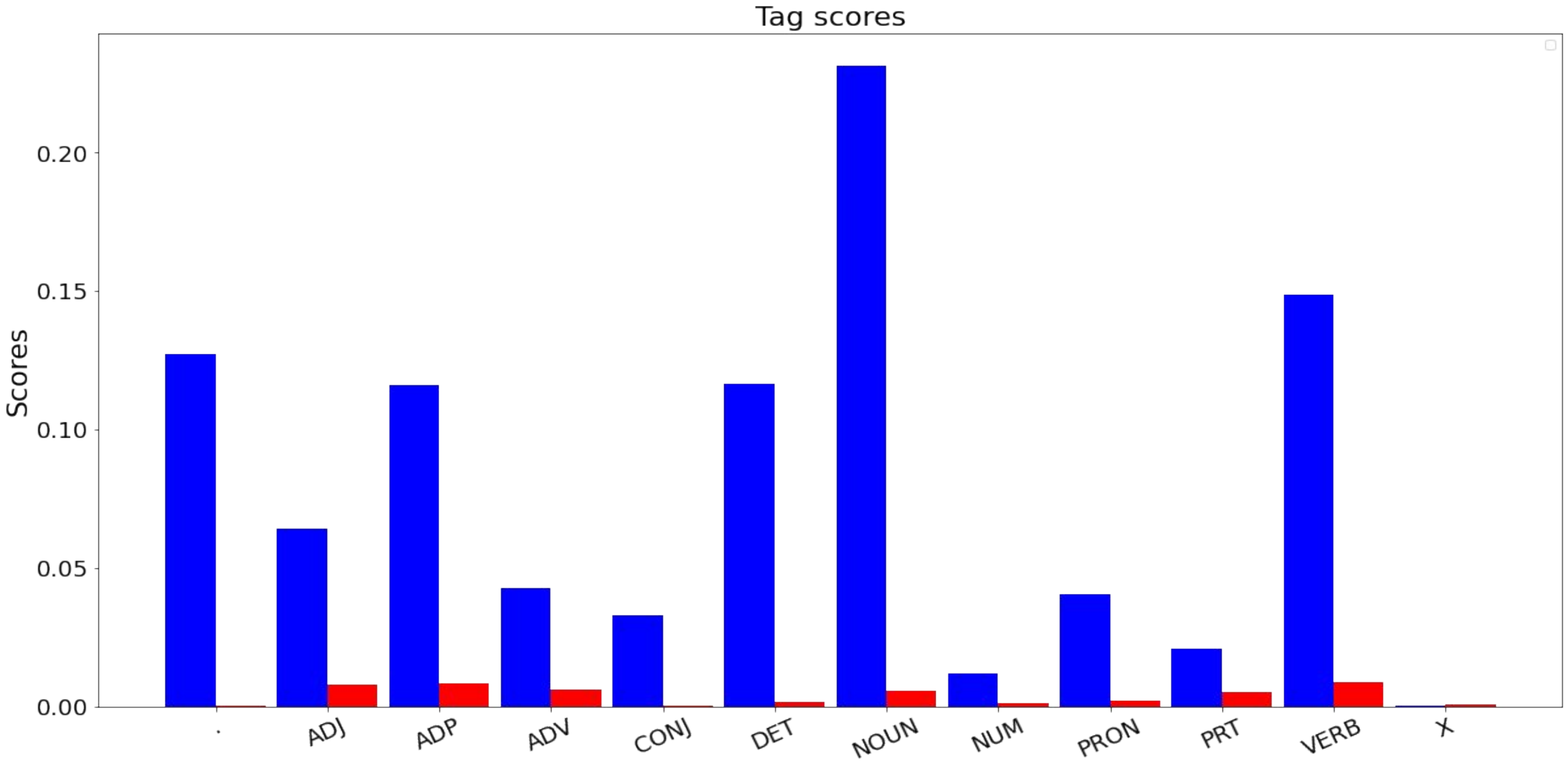
Confusion Matrix



Classification Report



Tag scores


Tag scores
Scores
0.00
0.05
0.10
0.15
0.20
.
ADJ
ADP
ADV
CONJ
DET
NOUN
NUM
PRON
PRT
VERB
X

Common Misclassifications

- ’**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

Analytics

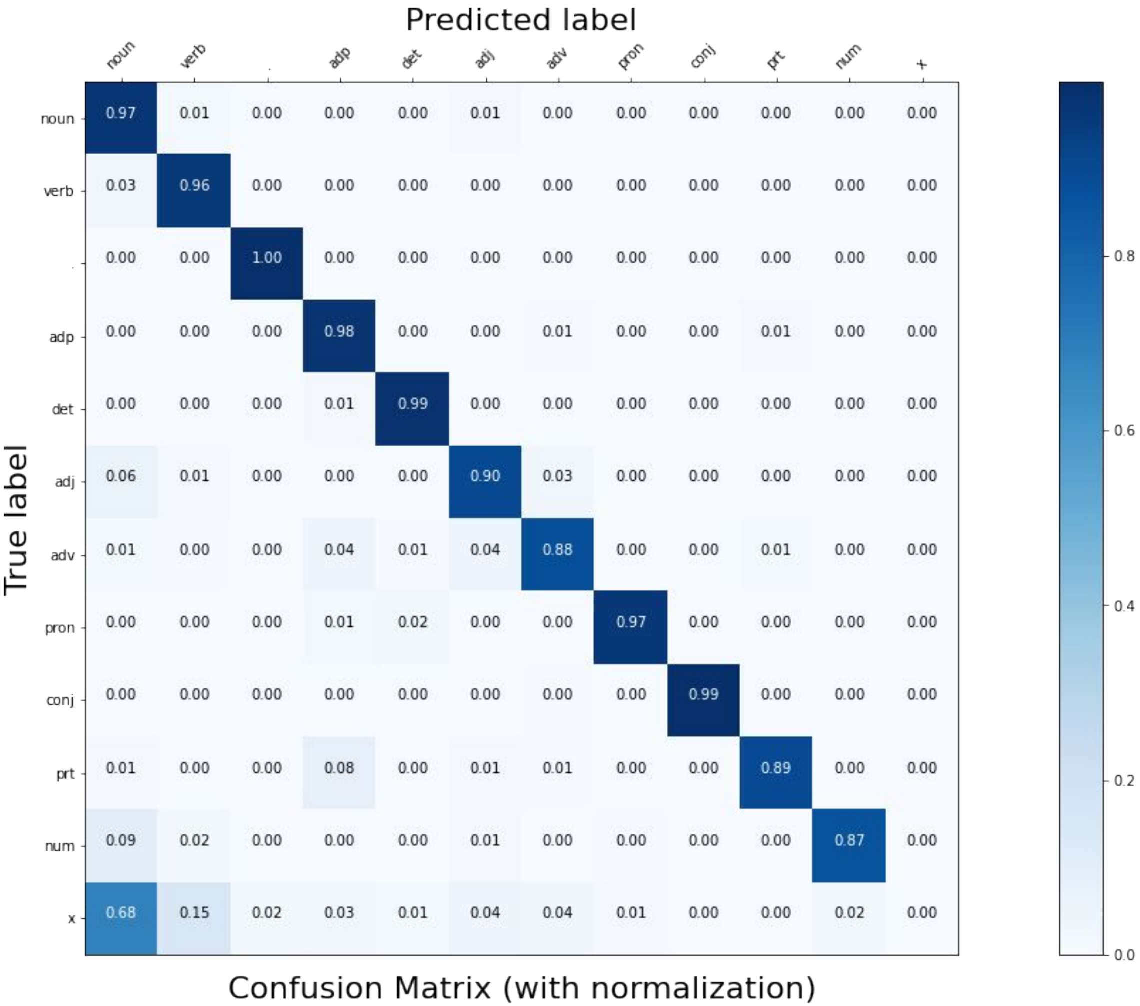
- Verbs misclassified as nouns and particles misclassified as adpositions often occur together, that is the tuple (PRT, VERB) is often misclassified as (ADP, NOUN).
- Articles such as ’to’, ’on’ commonly occur as both as both particles as well as adpositions, which can possibly result in same feature 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 use of simplistic features results in some feature vectors having multiple tags/labels across dataset, so choosing the most occuring label will result in the misclassification of classes with relatively low occurrences

Using a BiLSTM

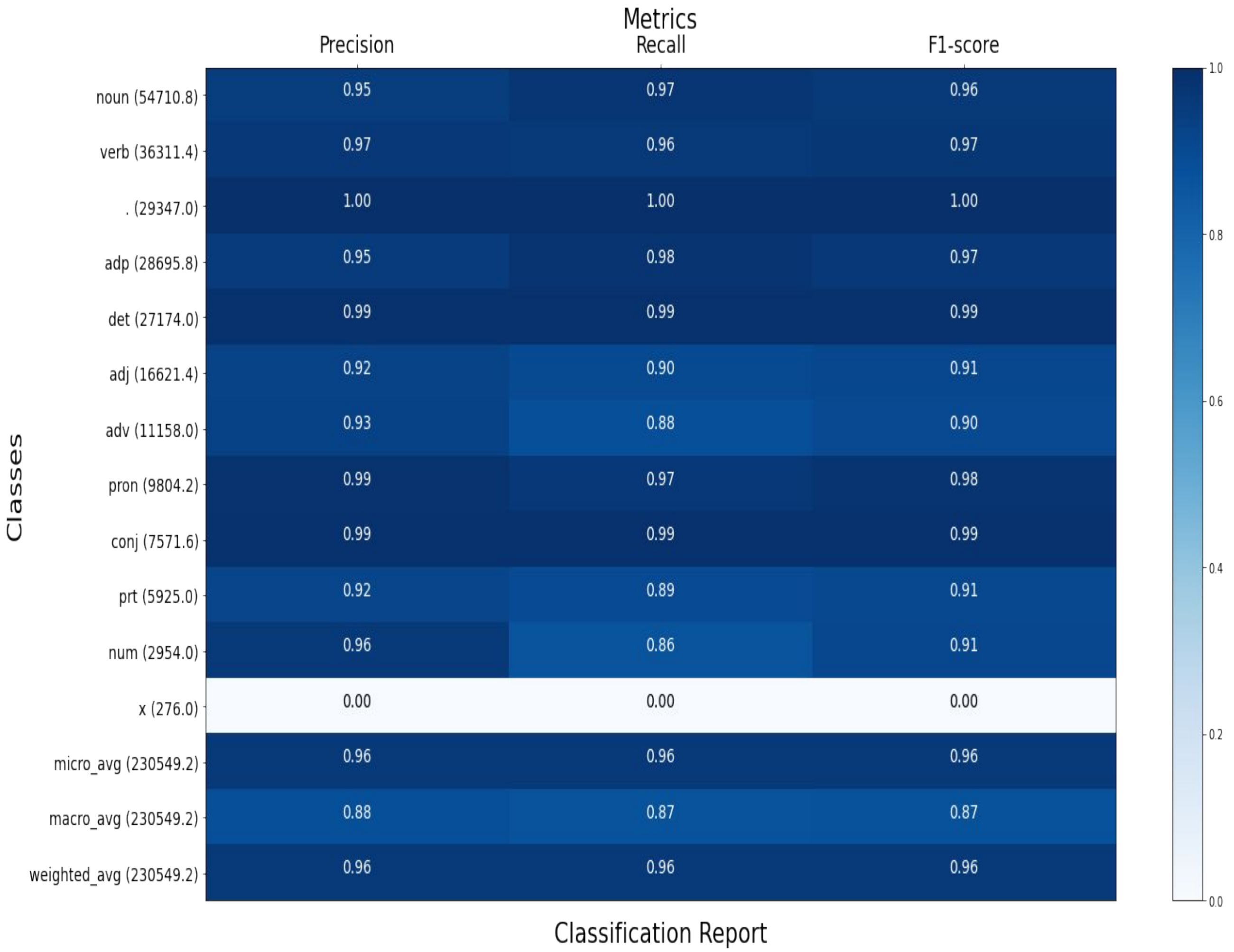
- Trained using ADAM
- Used word Embeddings of size 64 that were learnt during training
- Hidden layer size is also 64 in BiLSTM
- Useful in modelling long range word relations

The model performs fairly well and is able to predict with 96% plus accuracy, however it needs to see enough words per tag, eg: ‘x’ is poorly classified. Adjectives are mistaken for adverbs, perhaps where a verb can also be a noun. Nums are mistaken for nouns since they can occupy grammatical position in a sentence.

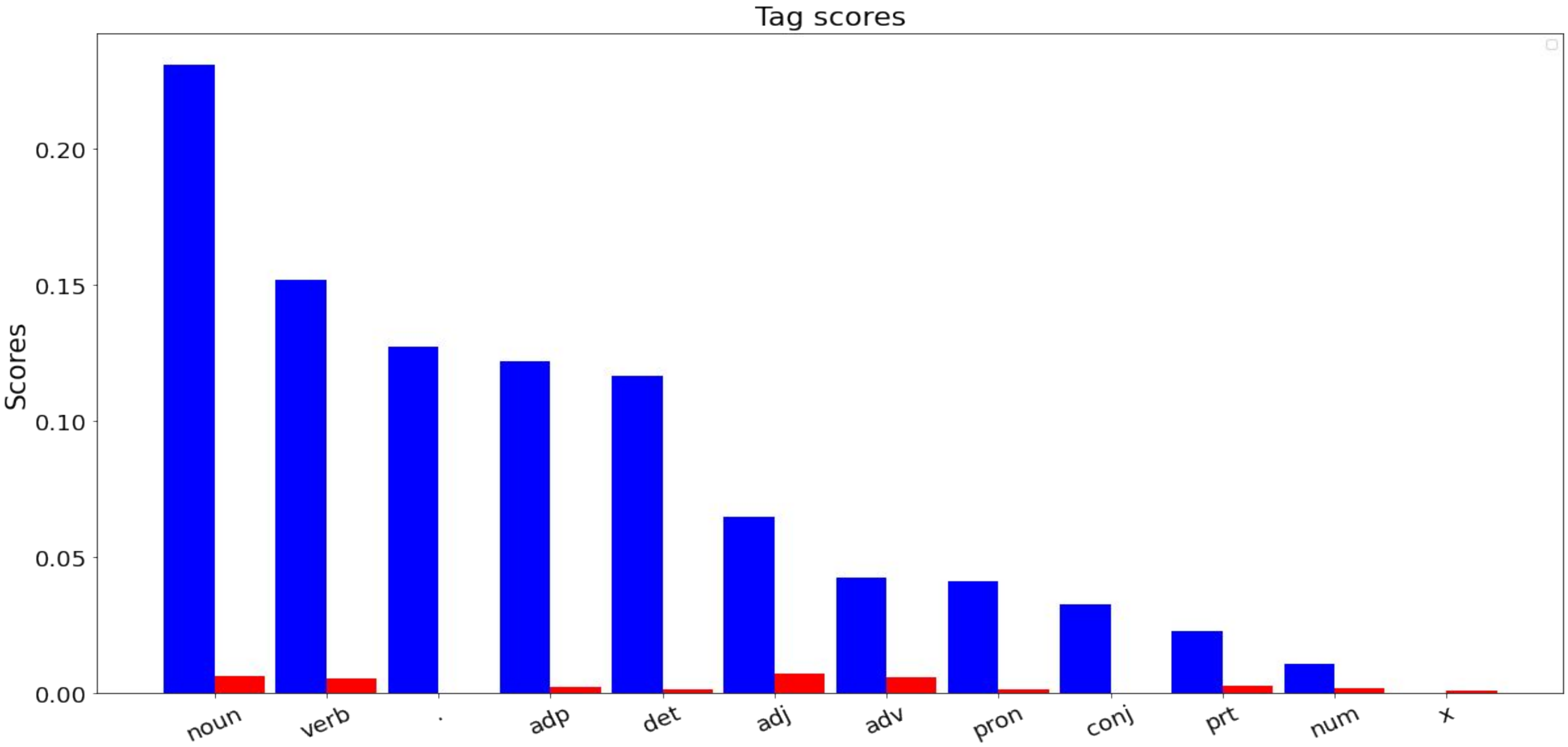
Confusion Matrix



Classification Report



Tag scores



Tag scores
Scores
0.20
0.15
0.10
0.05
0.00
noun
verb
.
adp
det
adj
adv
pron
conj
prt
num
x

Phrases that are wrongly classified

1. 'to' in 'steps to remedy' was given the tag of adposition which is a common usage of 'to. It's rare occurrence as a particle in infinitive form wasn't detected

2. 'top' in 'a top official' official was mistaken as a noun instead of adjective

3. 'worth' was treated as an adjective instead of 'noun'

4. 'consulting' in 'top consulting firm' was classified as verb

So, adj are tagged as adv and adj are also 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.

Top 3 POS trigrams where middle TAG is misclassified

1. ('det', 'adj', 'noun'): 2157 times
2. ('det', 'verb', 'noun'): 601 times
3. ('det', 'adv', 'adj'): 580 times

   As highlighted before, adj maybe mistake for adv or noun or verb depending on its other use. SImilarly adv is mistaken for adj too as highlighted by confusion matrix. Adv is also mistaken for adposition owing to similar positions in the sentence around the verb perhaps.

Conclusion

LSTM performs very well as it can capture long distance relationships between words which can help in POS tagging when sentences have long subordinate clauses

Linear SVM achieves great accuracy with very simple features, and the model can be improved further by using word embedding and exploiting the future word context. SMO algorithm (using the kernel trick) could also provide great results for even larger POS tag classes.

HMM performs well, though not as good as the more complex models, showing that POS is a shallow parsing task and does not require very rich features for good accuracy