POS Tagger

Comparison between models:

Model	Accuracy
HMM	95.19%
SVM	92.69%
BiLSTM	96.38%

Table 1: Accuracies (5-fold CV) for different models are shown in the table

POS Tag	HMM	SVM	BiLSTM
.	99.98%	100%	100%
ADJ	89.29%	87.54%	88.15%
ADP	96.64%	93.65%	97.16%
ADV	88.98%	86.88%	90.10%
CONJ	99.44%	99.51%	99.58%
DET	98.68%	98.62%	98.86%
NOUN	92.18%	91.18%	96.66%
NUM	97.06%	94.12%	92.01%
PRON	98.40%	94.54%	97.89%
PRT	90.05%	76.23%	90.92%
VERB	94.60%	95.76%	96.02%
X	34.92%	0%	0.60%
^	100%	(NOT USED)	100%

Table 2: Per POS accuracy for different models is shown in the table

POS Tag	HMM	SVM	BiLSTM
.	99.23%	99.65%	99.99%
ADJ	90.06%	86.05%	91.26%
ADP	94.18%	84.04%	96.26%
ADV	90.45%	90.44%	92.44%
CONJ	99.25%	99.37%	99.18%
DET	94.41%	98.2%	98.76%
NOUN	95.64%	95.56%	94.15%
NUM	97.06%	95.73%	96.09%
PRON	91.18%	99.74%	98.68%
PRT	90.16%	70.7%	90.65%
VERB	96.05%	97.001%	96.87%
X	69.04%	0%	91.65%
^	100%	(NOT USED)	99.99%

Table 3: Per Part of Speech recall for different models is shown in the table

Analysis for SVM:

Confusion matrix for third fold:

	DET	NOUN	ADJ	VERB	ADP	.	ADV	CONJ	PRT	PRON	NUM	X
DET	23917	3	0	0	253	1	57	13	0	7	0	0
NOUN	4	41565	1200	785	1464	56	149	1	244	4	88	25
ADJ	0	683	1158 9	182	144	14	334	0	274	0	0	18
VERB	0	915	84	33322	266	5	182	0	22	0	0	0
ADP	33	26	12	27	22901	0	202	9	1228	15	0	0
.	0	0	0	0	0	27910	0	0	0	0	0	0
ADV	62	139	388	30	644	0	9413	18	140	0	0	1
CONJ	10	0	0	0	7	0	15	6524	0	0	0	0
PRT	0	54	106	1	1247	0	53	0	4692	0	0	2
PRON	327	8	2	0	246	0	0	0	0	10088	0	0
NUM	0	21	67	0	0	0	1	0	33	0	2084	8
X	2	81	19	5	76	22	2	0	3	0	5	0

Error analysis:

The lesser the data available about a particular word, the lesser the accuracy. Since the model is completely dependent on the feature vector being provided, unless the feature vector does a decent job in depicting the words, the model cannot provide accurate results. This is a major issue in any unseen words: generic vector compromises the results.

To improve accuracy of numbers, all digits have been converted to 1. Although, this is ineffective in cases when the number is written as word eg. twenty-three. This led to an increase in accuracy by .2% overall, and a 7% increase in the NUM accuracy. The addition of prefix and suffix improved the overall accuracy, with improvements seen in NOUN, ADJ and VERB accuracy, with the overall accuracy increase by 1%

Particular Parts of Speech Error Analysis:

For the ‘.’ part of speech, the feature vector generated is rich enough to generate correct results.

For ‘ADJ’ and ‘ADV’: High number of unique adjectives and adverbs, due to which a generic feature vector is being given out to the model to make predictions. The SUFFIX and PREFIX features helped in improving the results.
For the ‘ADP’ tag, the recall is low, which depicts the weight vector has a very high bias.
For the ‘NOUN’ tag, despite high unique elements, the weights for the noun class are strong enough to handle them. This must be thanks to the number of NOUN labels in the corpus.
For the ‘PRT’ tag, a bias is generated in the weights for the ‘PRT’ tag by the feature vector of ‘to’. Therefore the results are accurate only for ‘to’ and the rest are being incorrectly predicted. The term ‘to’ is used 14,700 times out of the 29,300 occurrences of the PRT tag in the entire dataset.

Analysis for BiLSTM:

Confusion matrix:
NN -> NOUN, PR -> PRON, CJ -> CONJ, VB -> VERB

Tags	.	ADJ	ADP	ADV	CJ	DET	NN	NUM	PR	PRT	VB	X	^
.	30115	0	0	0	0	0	0	0	0	0	0	0	0
ADJ	0	16872	23	459	0	0	1163	5		5	409	0	0
ADP	0	11	30405	132	33	112	4	0	5	338	8	0	0
ADV	0	347	393	9904	33	31	89	0	0	76	96	0	0
CJ	0	2	1	16	7642	0	1	0	0	0	0	0	0
DET	1	2	126	1	14	29261	1	0	49	1	0	0	0
NN	0	1104	36	39	0	13	62819	75	8	3	2555	0	0
NUM	0	83	1	42	0	0	294	3368	0	0	138	0	0
PR	0		88	0	0	92	1	0	8021	0	2	0	0
PRT	0	39	449	59	0	0	18	0	0	5212	14	0	0
VB	0	118	35	21	0	0	904	0	0	3	36035	0	0
X	1	19	2	3	0	3	179	15	0	0	58	0	
^	0	0	0	0	0	0		0	0	0	0	0	11468

Strength: Words which have been seen frequently during training are predicted correctly. In addition, many unseen words get predicted correctly because of the bidirectionality. It is end to end and requires no pre trained embeddings or feature engineering. To improve the accuracy all numeric strings were replaced with “1”. This improved the overall accuracy from 95.8% to 96.38% and the per POS accuracy for ‘NUM’ from 84% to 92%.

Weakness: Since data is not balanced across tags, some unseen words which are outside the vocabulary (OOV) get tagged incorrectly by the more frequent tag. For tags having less number of instances like “X” accuracy is very poor. The model completely ignores the word features due to which some unseen words which could be tagged correctly based on these features do not get tagged correctly (e.g. words which have ‘NUM’ tag).

Error analysis:
Roughly 30% of incorrect predictions are due to unseen words. We observe the trend that unseen words get tagged generally with the more frequent tags.

Some of the commonly occurring wrong predictions are listed here:

- ‘as’ has two tags - ‘ADV’, ‘ADP’. Both are similar in number, so the model is not able to distinguish well between these.
- ‘that’ has three tags - ‘PRON’, ‘ADP’, ‘DET’. The model makes some errors in predicting the tag for this word.
- ‘to’ has two tags - ‘ADP’, ‘PRT’. Its usage as PRT and ADP, both are pretty common leading to errors in some cases.

Tag ‘X’ has very few occurrences due to which the model rarely predicts it correctly.

POS analysis:

- ‘.’ POS tag depends on sequence and the various punctuation symbols get encoded into vector representations due to which its accuracy is pretty high.
- For ‘ADV’, ‘ADJ’ there are a large number of distinct words used as adverbs and adjectives respectively. So vector representation of words is not enough to capture the information. In addition, the model is not based solely on transition of tags, so accuracy is not very high. Also note that the overall frequency of ‘ADV’ is small in the corpus.
- For ‘CONJ’, the accuracy is high because of the following reasons. Words used as conjunctions are mostly used as conjunctions only (e.g. and, but). A conjunction can easily be identified by the preceding tag sequence. In addition, there are not many distinct conjunctions in the vocabulary, so most conjunctions get seen during training.
- For ‘DET’, we obtain high accuracy because most determiners can be predicted by the following tag sequence. Since, our model is bidirectional we can capture this information. In addition, the vocabulary for ‘DET’ is not very large. So the chance of encountering an unseen ‘DET’ is quite small.
- For ‘NOUN’, the accuracy is high because our model stores preceding context (history) as well as following context. Many nouns can be predicted from the context around it. So many unseen nouns get predicted correctly in comparison

to other models. We observe that the recall for ‘NOUN’ is not very high. The reason is that many unseen words get wrongly predicted by more frequent tags. ‘NOUN’ is one of the more frequent tags.

- For ‘NUM’, numbers (like “twenty-three”) get predicted as other more frequent tags like ‘NOUN’, ‘VERB’. We observe that most incorrect predictions are on unseen words. On the other hand, SVM uses word features and so such cases do not get misclassified as ‘NOUN’ or ‘VERB’ because they do not have features for these tags.
- For ‘PRT’, low accuracy is observed due to some words being used more often as PRT in comparison to the rest. Words which are less frequently used as PRT get predicted incorrectly. (e.g. “to” is used 14,700 times out of 29,300 occurrences in the corpus, implying that other words are used quite less in comparison to the word “to”)

Nonetheless we observe the model is end-to-end and has a very nice performance overall.

Analysis for HMM:

Confusion matrix:
NN -> NOUN, PR -> PRON, CJ -> CONJ, VB -> VERB

Tags	.	ADJ	ADP	ADV	CJ	DET	NN	NUM	PR	PRT	VB	X	^
.	30697	38	20	5	0	0	155	1	0	0	46	9	0
ADJ	0	19522	11	499	0	1	1132	3	1	4	177	18	0
ADP	1	84	37331	392	1	103	400	0	112	373	212	25	0
ADV	0	520	497	10813	32	9	14	0	0	33	18	5	0
CJ	0	0	40	14	9264	4	0	0	0	0	0	4	0
DET	1	679	92	73	19	32611	944	11	29	1	155	41	0
NN	2	899	15	131	2	1	63892	68	19	18	1391	116	0
NUM	0	1	0	0	0	0	115	4408	0	0	0	0	0
PR	0	21	118	42	0	167	614	3	7925	4	18	11	0
PRT	0	63	380	77	0	0	12	0	0	5249	1	2	0
VB	0	174	51	59	0	3	1168	1	3	3	38516	34	0
X	19	2	1	0	0	0	24	3	0	0	1	180	0
^	0	0	0	0	0	0	0	0	0	0	0	0	11468

Implementation Details: All consecutive sequences of numbers are replaced by 0. For example, the string “123AB123” is transformed to “0AB0”. This helps in better performance on “NUM” tag from 88.90% to 97.06% and increase in overall performance of 0.2%.

Strengths: HMM has better explainability as compared to other neural based models. It achieves performance comparable to the modern approaches. It is reasonably fast as compared to other methods. Performs well on words frequently occurring with a particular tag.
Weakness: HMM performs poorly on unseen words(~55% accuracy) and on tag “X”.

Error Analysis:

- Nearly 45% of incorrect predicted tags are due to unseen words. The model has a tendency to predict unseen words with the most frequently occuring tags. This can be explained by the fact that tags for unseen words are predicted solely on the basis of transition probabilities.
- The model has poor performance on unseen words(~55% accuracy)
- ‘to’ has tags ‘ADP’ and ‘PRT’ with both of them quite common.This leads to errors in some cases. However the ratio of correctly predicted tags is still high(“to” is one of the most frequently occuring tags with nearly 25000 occurences).
- Tag ‘X’ has very few occurrences due to which the model rarely predicts it correctly since transition probabilities are low and words are unseen

POS Analysis

- **“.”:** It wrongly predicts “.” only when seeing unseen words since it is based solely on transition probabilities. Hence it has very high accuracy.
- **“ADJ”** & **“ADV”:** Are affected by lower emission probabilities since they have high unique occurrences and hence have lower emission probabilities. Another reason for lower accuracy for “ADV” is that words “to”, “that” and “as” have multiple commonly occuring tags and for each of them “ADV” is one of those.
- **“CONJ”:** Conjunctions are mostly used as conjunctions and there are fewer conjunctions. Hence they have high emission probability(P(w|CONJ) and for such words other emission probabilities are low.
- **“DET”:** Mostly for Determiners the case is the same as with Conjunctions. However the word “that” is sometimes incorrectly classified as ‘that’ and “That” both occur in the corpora reducing emission probabilities and it also features “PRON” and “ADP” among other tags.
- **“NOUN”:** Most unseen words are nouns(~43% of unseen corpora) hence has lower accuracy
- **“PRT”:** “To” is nearly 50% of total “PRT”. Hence there is a natural bias towards “To”. This affects emission probabilities of other PRTs. “to” also doubles up as an “ADV”, so some error is observed even in prediction of it. The same is with “in” and “on”.
- **“X”:** The frequency of tag “X” is significantly lower in the corpus. This severely affects the transition probabilities to tag “X”. Secondly, the emission probabilities for unseen words are taken uniform across all tags. This results in a lesser prediction of tag “X”.

Learnings

1. Feature Engineering: We learned a lot about how important feature engineering is, and therefore we got to explore a lot in this domain.
2. Implementation: We got a good practice of implementation in all the three tasks.
3. Error Analysis: The detailed error analysis helped us figure out where our model lags behind and therefore where we can improve upon if possible.
4. The grammar behind Parts of Speech: Given this assignment, we also got to explore the grammar behind the parts of speech.