CS 388 NATURAL LANGUAGE PROCESSING

HW 2 POS Tagging with HMMs and CRFs

VIvek Natarajan

Introduction:

In this exercise, the performance of Hidden Markov Models(HMMs) and Conditional Random Fields(CRFs) was analyzed on the POS Tagging task using the ATIS and WSJ Corpus of Penn tree bank. POS tagging is a more complex language model than N-grams wherein each token is assigned to a tag from a small set of tags. The set of experiments performed replicate that performed by Lafferty et al. 's original paper on CRFs although on a small scale.

HMMs are generative models which usually take a sequence of observed output symbols, like tokens of language, and infers the underlying states that generated each symbol, corresponding to the POS tags. It models the full joint distribution of labels and observations. This means it can also infer the most likely tokens (output), given a sequence of tags. The parameters to be estimated in a HMM model are the transition (from one state to another) and observation (of data given a state) probabilities. Learning and inference on a HMM model can be done quickly using dynamic programming (Viterbi algorithm).

CRFs are a discriminative approach to sequence labeling whereas HMMs are generative. Discriminative methods are usually more accurate since they are trained for a specific sequence labeling task. CRFs easily allow adding additional token features without making additional independence assumptions. Training time is increased since a complex optimization procedure (usually LBFGS) is needed to fit supervised training data.

Experiments:

The first task involved generating training and testing data in the format required by Mallet, the Java library with the CRF Tagger implementation. Then experiments were performed to evaluate the HMM Simple Tagger implementation on the ATIS and WSJ corpus varying the number of iterations and amount of training data. The performance criteria included training accuracy, testing accuracy, OOV accuracy and run-time. Subsequently, the CRF tagger was also evaluated on the ATIS and WSJ corpus varying the number of iterations. Then a number of orthographic features were added to the CRFs and the individual and grouped impact on the CRF performance were noted. These features included:

Prefix features - commonly used prefix like re, im, in, pre, mis, dis and un

Suffix features - commonly used suffix like s(plural), ing(Gerund), ly, able, ive, ic, ful, ed, ible, ious, ous, ant and ent

Morphological features - Caps(first alphabet capitalized), small(less than 5 letters), .(contains period), '(contains apostrophe), -(contains hyphen), 0(is alphanumeric).

The CRF was trained on two sections of the WSJ data instead of one and tested on two more sections to evaluate the impact of increase of training data. Out of curiosity, experiments were also performed on the ATIS dataset to see the impact of a Backward (sentences reversed) model on the HMM and CRF Taggers. However, due to shortage of time, results could not be reported on the ATIS dataset.

Results:

All experiments were on run on Stampede at TACC with number of threads set to 16. CRF+ indicates all three feature(pre, suf and morph) categories are added

Model	**Corpus**	**Training Accuracy**	**Test Accuracy**	**OOV Accuracy**	**Time(seconds)**
HMM	Atis	0.8896	0.8615	0.2916	2.5

CRF	Atis	0.9981	0.9287	0.3796	81
CRF + pref	Atis	0.9981	0.9334	0.3756	83
CRF + suf	Atis	0.9979	0.9392	0.4583	86
CRF + morp	Atis	0.9980	0.9294	0.3333	85
CRF + pr-suf	Atis	0.9979	0.9404	0.4583	86
CRF +	Atis	0.9982	0.9380	0.4583	87

Model	**Corpus**	**Training Accuracy**	**Test Accuracy**	**OOV Accuracy**	**Time(seconds)**
HMM	WSJ	0.8617	0.7849	0.3794	58
CRF	WSJ	0.8811	0.7341	0.4581	3415.5
CRF + pref	WSJ	0.8908	0.7444	0.4678	5289.7
CRF + suf	WSJ	0.9207	0.8117	0.6513	5518.3
CRF + morp	WSJ	0.9059	0.7805	0.5954	5347.8
CRF + pr-suf	WSJ	0.9865	0.8804	0.7812	6401.6
CRF +	WSJ	0.9888	0.8759	0.7907	6805.2

HMM - No of iterations and convergence results on WSJ Corpus

Iteration Number	**Training accuracy**	**Test Accuracy**	**OOV Accuracy**
1	0.862	0.7853	0.381
10	0.862	0.7850	0.3794
20	0.8618	0.7849	0.37963
50	0.8617	0.7849	0.3794
Final - (80) Convergence	0.8617	0.7849	0.3794

CRF - No of iterations and convergence results on WSJ Corpus

Iteration number	**Training accuracy**	**Test Accuracy**	**OOV Accuracy**
1	0.1160	0.0909	0.2294
5	0.343	0.319	0.2445
10	0.257	0.2306	0.2569
20	0.581	0.548	0.386
50	0.5989	0.55	0.3933
Final - (500) No convergence	0.8811	0.7341	0.4581

Results on extended WSJ Corpus for HMM and CRF

Model	**Corpus**	**Training accuracy**	**Test accuracy**	**OOV Accuracy**	**Time(seconds)**
HMM	WSJ extended	0.8870	0.8333	0.3957	126 (143 iterations)
CRF	WSJ extended	0.798	0.734	0.48	8 hours -(300 iterations completed)

Backward Model experiments on ATIS

Model	Corpus	Training Accuracy	Test Accuracy	OOV Accuracy	Time(seconds)
HMM	Atis	0.9019	0.8808	0.2916	5.4
CRF	Atis	0.9979	0.9182	0.4167	84
CRF+	Atis	0.9945	0.9400	0.4731	85

Observations and Discussion:

Test accuracy: The CRF (without tokens) almost always beats the HMM in both the very constrained airline transaction corpus (ATIS) and a more diverse corpus (WSJ). The results clearly highlight the key differences between the HMM and CRF. An anomaly is the fact that the WSJ HMM happens to beat the CRF in testing data accuracy, but it should be noted that the CRF performance varies significantly across different runs. A 10-fold evaluation of the CRF on the ATIS data gives test accuracy ranges between 0.914 and 0.942, which is significant. Probably a more robust CRF implementation would give less variable results.

OOV accuracy: The CRF being a discriminative model performs better in the OOV test accuracy compared to the HMM. The test accuracy among OOV words is however significantly worse than that among in-vocabulary tokens and leaves scope for improvement. The HMM poor performance can be attributed to the fact that it being a generative model, it tries to model the joint distribution of tokens and tags and it has never seen the OOV words before - hence it struggles here.

Training accuracy: The training accuracy of the CRF is significantly higher as compared to that HMM. The difference between training and test accuracy is also significantly higher for the CRF This can be attributed to that the fact that discriminative models tend to overfit the data compared to the generative models particularly small sized corpus. There is much less difference in the test accuracy between CRFs and HMMs and the HMM even trumps the CRF in test accuracy on the WSJ corpus. This illustrated the importance of validation on a standard unseen test set to understand the effects of overfitting.

Run time: The run time of CRF is about 10-100 times higher than that of the HMM. This is because the CRF runs a more complex LBFGS optimization procedure which requires significantly more number of iterations also optimizes over a larger number of parameters compared to the HMM. However, in most tasks, the increase in accuracy subsumes the run time increase of the CRF.

Orthographic features: Adding the orthographic features to the CRF significantly increased its accuracy in terms of training, test and OOV accuracy. The CRF+ model which had prefix, suffix and morphological features combined gave a significantly better performance in both cases of ATIS and WSJ corpus. The training times show a slight increase with the increase in the number of features. However, this needs to be validated on the basis of more experiments.

Best feature set: Among the three categories of features experimented with suffix features were found to be most useful followed by morphological and prefix features. This is especially validated by better OOV accuracy in case of suffix features. However this could also be attributed to the fact that there were more number of suffix features than the other two categories. It should be noted that each feature category is useful in its own right because the combination of all three features model (CRF+) performs significantly better on both corpus.

Number of iterations and convergence: The number of iterations required for convergence for the HMM is significantly lower compared to that of CRF. This is because the number of parameters to be learned in the HMM model is less and the optimization procedure is less complex. The CRF may often not converge thus requiring an upper bound on the number of iterations. This is illustrated in the results section. Reduced number of iterations for both CRF and HMM means less accuracy on OOV, training and test measures although the drop in accuracy is more pronounced for CRFs than HMMs. The OOV measures seem to converge faster as well. This may be because the percentage of OOV words is only around 20% in both corpus.

Increase in training data: On increasing the size of the training data, the number of iterations for HMM increase but the training, OOV or test accuracy do not seem to vary a lot (showed a slight increase). This was measured by training on sections 00 and 01 of the WSJ corpus and testing on sections 02 and 03. For CRFs on the other, even after **8 hours of run time**, only 300 iterations were completed with training and test accuracy scores of 0.798 and 0.734. The OOV measure was found to be 0.48. Thus the time required for an iteration seems to have significantly increased although it is hard to conclude if it is a polynomial function of the input size. Decreasing the training data definitely reduced the accuracy.

Backward Model experiments: The HMM and CRF tagger were also trained on the ATIS corpus with sentences reversed. It demonstrated a slight increase in performance although further experiments need to be done to validate this. For want of time, results on the WSJ corpus could not be reported.

Conclusion:

CRFs and HMMs illustrated the difference between discriminative and generative models for POS tagging. HMMs are the jack of all trades, have less training time, have less overfitting but are also less accurate. CRFs are the master of the task of sequence labeling with increased accuracy but have considerable increase in training time. The use of one over the other has this tradeoff and thus depends on the application.