CS 388 Natural Language Processing
Homework 2: Part-of-Speech Tagging with HMMS and CRFs

Pengxiang Cheng (EID: pc23445)
pxcheng@cs.utexas.edu

March 3, 2014

1 Introduction

The goal of this homework is to do Part-of-Speech tagging on real word data with the Penn Treebank tag set, using both Hidden Markov Model (HMM) and Conditional Random Field (CRF) to do the sequence labeling. We use the implementation of HMM and CRF from the Mallet package.

The implementation in Mallet provides a basic interface to train a sequence model iteratively until the log likelihood converges, and to calculate the training and testing accuracy after the convergence. Beyond that, we also want to calculate the out-of-vocabulary accuracy on testing dataset, which would indicate how well the model performs on labeling unseen words.

The last thing to do is running CRF model with additional orthographic features, including common suffixes, capitalizations, hyphens, etc. The CRF implementation from Mallet can deal with orthographic features, so we only need to handle the training and testing dataset to detect and include these features.

2 Implementation

First we need to convert the data file from Penn Treebank format to Mallet format, which can be achieved by separating all valid tokens and replace every "/" character with a space. This is implemented in "POSMalletFormat.java" file.

To calculate the OOV (out-of-vocabulary) accuracy, we simply modify the "TokenAccuracyEvaluator.java" file from the Mallet package by handling the training process and testing process separately according to the "description" parameter. In the training process, we use a static HashSet variable to store all seen tokens. In the testing process, we judge whether each testing token is OOV or not by referring to HashSet, and calculate the accuracy for both the whole set of tokens and OOV tokens.

To add extra orthographic features to the CRF model, we implement in the "POSAddFeatures.java" file to detect six most common features and add them in the raw data. The features we used here are listed in Table. 1.

Two corpora are used for the experiment. In the *atis* corpus, we use 80% of the data for training and 20%

Feature	Label	Judgment
plural	s	Ends with “s”
gerund	ing	Ends with “ing”
comparative	er	Ends with “er”
superlative	est	Ends with “est”
hyphen	hyph	Contains “-”
capitalized	caps	Starts with upper case characters

Table 1: Orthographic features used in CRF modeling.

of the data for testing, with a random split on the raw data. In the *wsj* corpus, we use the section 00 for training and section 01 for testing.

3 Results

The results for running HMM and CRF on atis and wsj corpora using only tokens are shown in Table. 2.

	HMM		CRF	
	atis	wsj	atis	wsj
Training accuracy (%)	89.05	86.19	99.83	98.57
Testing accuracy (%)	86.45	78.49	92.99	79.36
OOV accuracy (%)	29.17	37.95	33.33	47.61
OOV percentage (%)	2.80	15.33	2.80	15.33
Overall running time (s)	6.941	132.223	125.485	10069.513
Training iterations (#)	53	81	48	148
Average running time (s)	0.131	1.633	2.614	68.037

Table 2: Results for HMM and CRF using only tokens.

The results for add several different orthographic features to CRF sequence labeling are shown in Table. 3.

		only tokens	caps	s	hyph	3 features	6 features
atis	Testing accuracy (%)	92.99	92.99	93.57	92.99	93.57	93.81
	OOV accuracy (%)	33.33	37.50	41.67	33.33	41.67	50.00
	Overall running time (s)	125.485	114.802	121.873	120.987	114.557	104.212
	Training iterations (#)	48	43	46	48	45	43
	Average running time (s)	2.614	2.670	2.649	2.521	2.546	2.424
wsj	Testing accuracy (%)	79.36	82.34	82.73	78.98	85.54	86.29
	OOV accuracy (%)	47.61	56.19	58.24	47.76	68.53	71.70
	Overall running time (s)	10069.513	8439.253	8716.440	8748.273	7854.610	7309.522
	Training iterations (#)	148	135	121	123	121	123
	Average running time (s)	68.037	62.513	72.037	71.124	64.914	59.427

Table 3: Results for adding orthographic features to CRF.
(“3 features” indicates caps & s & hyph, and “6 features” indicates all features listed in Table 1)

4 Analysis

4.1 Comparison of HMM and CRF

1. For both corpora, the training accuracy, the overall testing accuracy, the OOV testing accuracy of CRF are all greater than those of HMM, which supports the theory that discriminative models (i.e. CRF) are better at sequence labeling than generative models (i.e. HMM).

2. For both corpora, the running time of CRF is much longer than that of HMM (approximately 20 times for atis, 80 times for wsj). This is partly due to the complex optimization procedure in every iteration of CRF training process. Another possible reason for this huge difference is that in CRF modeling, we inference the sequence tagging results of both training and testing dataset in each iteration, while in HMM we only do the inference after convergence. The time cost for inference is $O(TN^2)$, so the difference in running time grows even larger with a larger corpus (increases quadratically with increasing N), which explains the difference in the two corpora.

4.2 Impact of Adding Orthographic Features

1. For both corpora, adding orthographic features will, to some extent, increase both overall and OOV accuracy in testing dataset, and decrease running time. Adding 6 features to the *wsj* corpus, the OOV accuracy increases from 47.61% to 71.7%, and the running time decreases approximately 30%.

2. The increasing in accuracy is reasonable as orthographic features provide extra information for sequence tagging in addition to the context relationships. The decreasing in running time is mainly due to the decreasing in iteration counts, while maintaining almost the same average running time in each iteration, as additional information help the optimization to converge at a higher speed.

3. From the comparison of three distinct features (*capitalized*, *plural*, *hyphen*), the *plural* feature helped the most in increasing accuracy.