CS594 – Advanced Machine Learning

Report – Project 1

Team 4: Somshubra Majumdar(smajum6@uic.edu), Amlaan Bhoi(abhoi3@uic.edu), Chandrasekhara Ganesh Jagadeesan(cjagad2@uic.edu)

Answer 1)

Answer 1a)

$$P(y^t|x^t) = \frac{1}{Z_{x^t}} \exp\left(\sum_{s=1}^{m} \langle w_{y_s^t}, x_s^t \rangle + \sum_{s=1}^{m-1} T_{y_s^t, y_{s+1}^t}\right)$$

$$\log P(y^t|x^t) = -\log Z_{x^t} + \sum_{s=1}^{m} \langle w_{y_s^t}, x_s^t \rangle + \sum_{s=1}^{m-1} T_{y_s^t, y_{s+1}^t} \quad - (1)$$

$$\nabla w_y = \sum_{s=1}^{m} [\![y_s^t = y]\!] . x_s^t - \frac{\partial \log Z_{x^t}}{\partial w_y}$$

$$\frac{\partial \log Z_{x^t}}{\partial w_y} = \frac{1}{Z_{x^t}} \sum_{y \in \mathcal{Y}} \exp\left(\sum_{s=1}^{m} \langle w_{y_s}, x_s^t \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}}\right) \sum_{s=1}^{m} x_s$$

$$= P(y_s = y | x^t) . \sum_{s=1}^{m} x_s^t$$

$$= \sum_{s=1}^{m} P(y_s = y | x^t) . x_s^t$$

$$\therefore \frac{\partial}{\partial w_y} \log P(y^t|x^t) = \sum_{s=1}^{m} \left([\![y_s^t - y]\!] - P(y_s = y | x^t) \right) x_s^t$$

w.r.t T_{ij}

$$\frac{\partial}{\partial T_{ij}} \log P(y|x) = [\![y_s^t = i, y_{s+1}^t = j | x^t]\!] - \frac{\partial}{\partial T_{ij}} \log Z_{x^t}$$

$$\frac{\partial \log Z_{x^t}}{\partial T_{ij}} = \sum_{s=1}^{m-1} P(y_s = i, y_{s+1} = j | x^t)$$

$$\therefore \frac{\partial}{\partial T_{ij}} \log P(y|x) = \sum_{s=1}^{m-1} \left[[\![y_s^t = i, y_{s+1}^t = j]\!] - P(y_s = i, y_{s+1} = j | x^t) \right]$$

Ans 1b)

$$\frac{\partial}{\partial w_y} \log Z_x^t = \sum_{s=1}^{m} \underbrace{p(y_s = y \mid X^t)}_{\text{Marginal Probability of } y \text{ at position } s \text{ in the word}} \cdot x_s^t$$

$\Rightarrow$ This equation is the expectation of features X_s^t in the margenal distribution.

For T_{ij}, we can see from $\frac{\partial}{\partial T_{ij}} \log Z_x^t$, the features are i and j

$\Rightarrow$ It is the expectation of features i and k ~~under~~ in the marginal distribution of these pairwise features.

c) The max objective value obtained = 200.185149689205

Answer 2)

a) $\frac{1}{n}\sum_{i=1}^{n} \log p(\boldsymbol{y}^i \mid \boldsymbol{X}^i) = -31.28843743965$

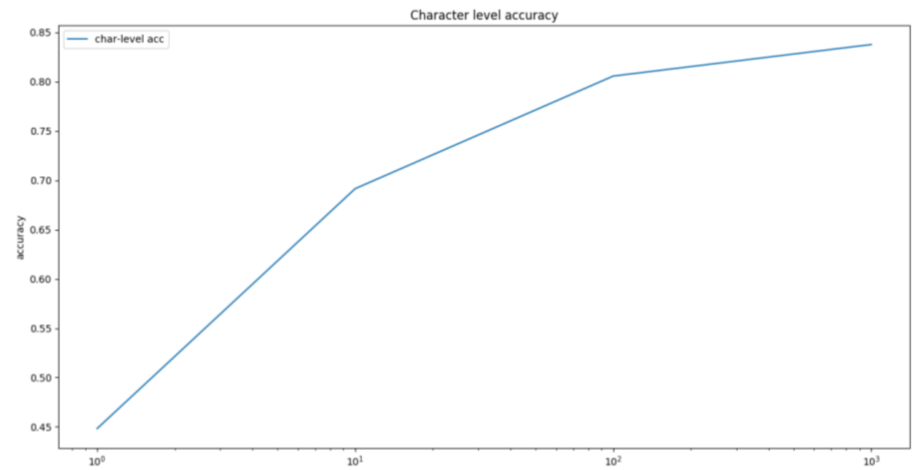
b) Optimal Objective value = 3701.1579986480165

Some performance stats: Gradient check in 8.6 sec

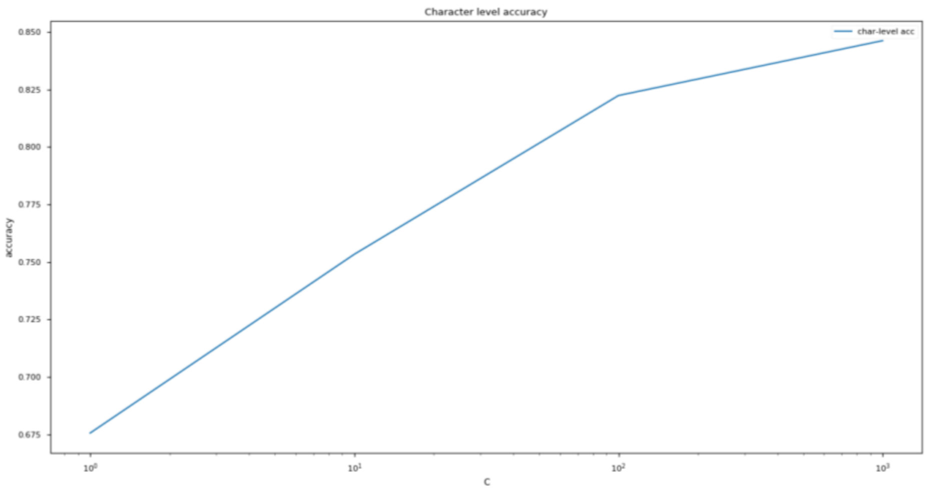
Test accuracies: Word wise = 47.25%, Char wise = 83.75%

Answer 3)

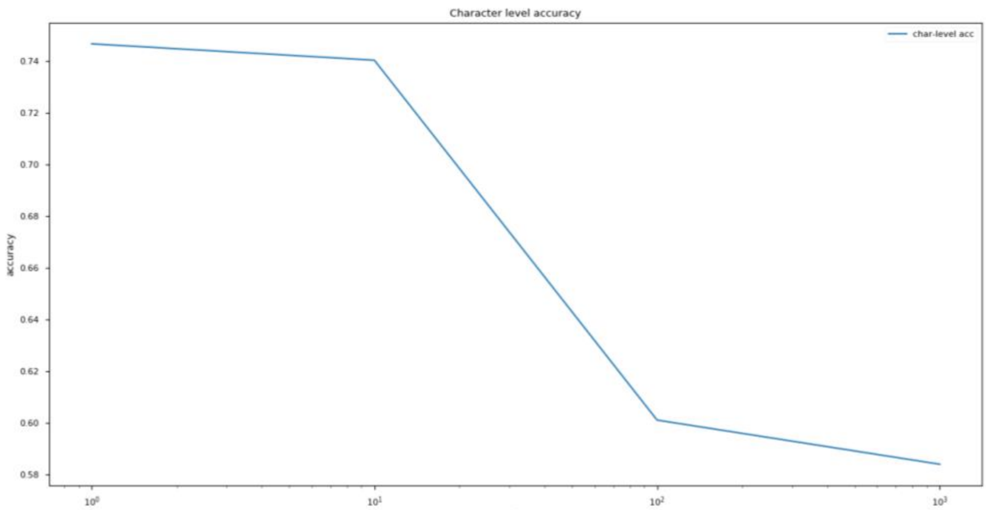
A1) CRF Letter-wise accuracy



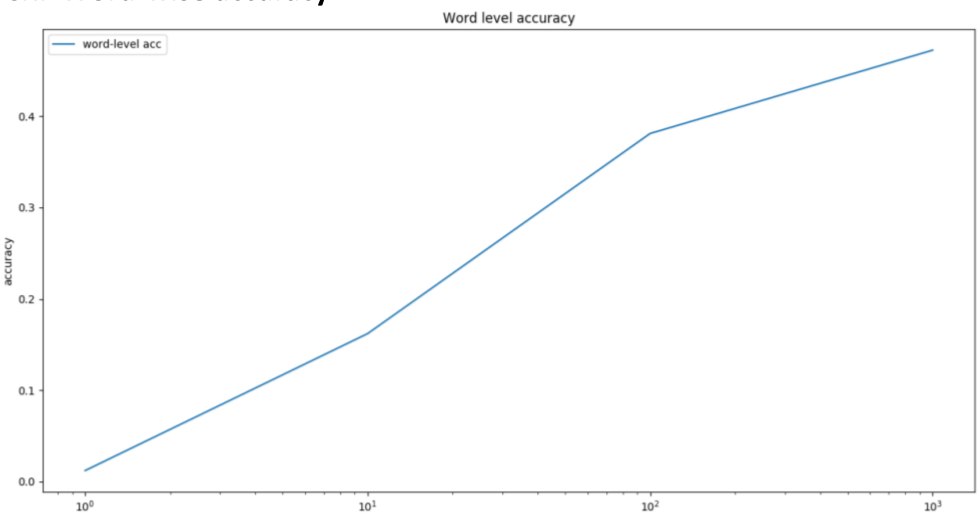
A2) SVM-HMM Letter-wise accuracy



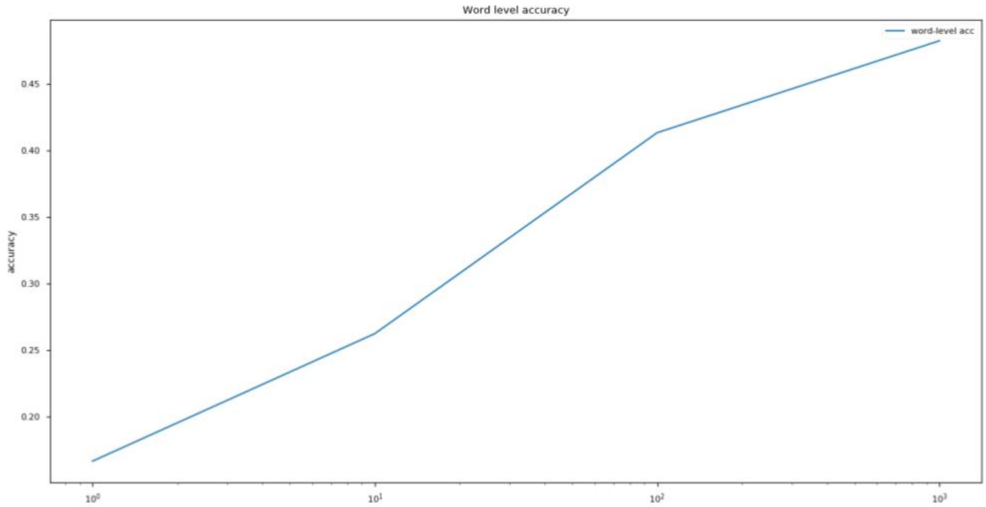
A3) SVM-MC (LibLinear) Letter-wise accuracy



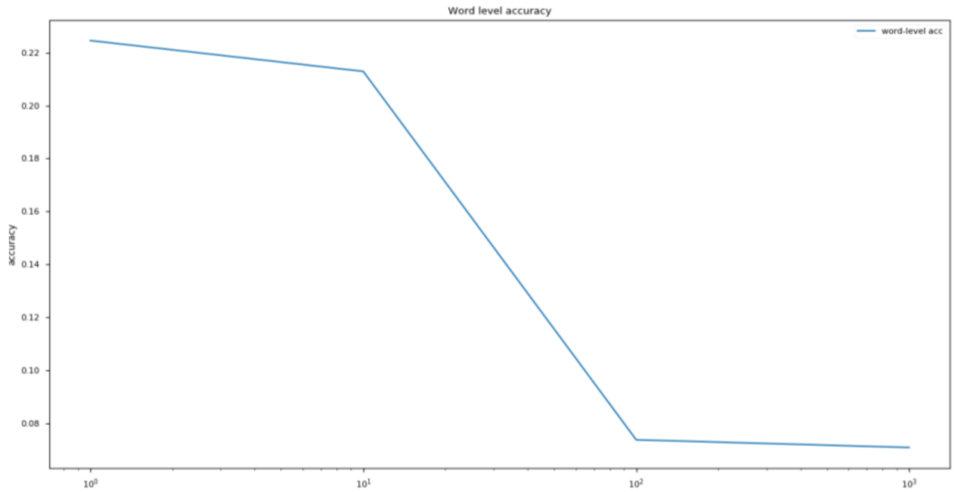
B1) CRF Word-wise accuracy



B2) SVM-HMM Word-wise accuracy



B3) SVM-MC (LibLinear) Word-wise accuracy



Observations:

As per the theoretical function of C, the penalty weight hyper-parameter determines whether the model is a “soft-margin” classifier (which largely ignores the training data overlap between classes and therefore obtains a poor validation and test accuracy) or a “hard-margin” classifier (which penalizes mistakes much more, therefore can cause the model to severely over-fit if a good value isn’t obtained for it via proper cross-validation).

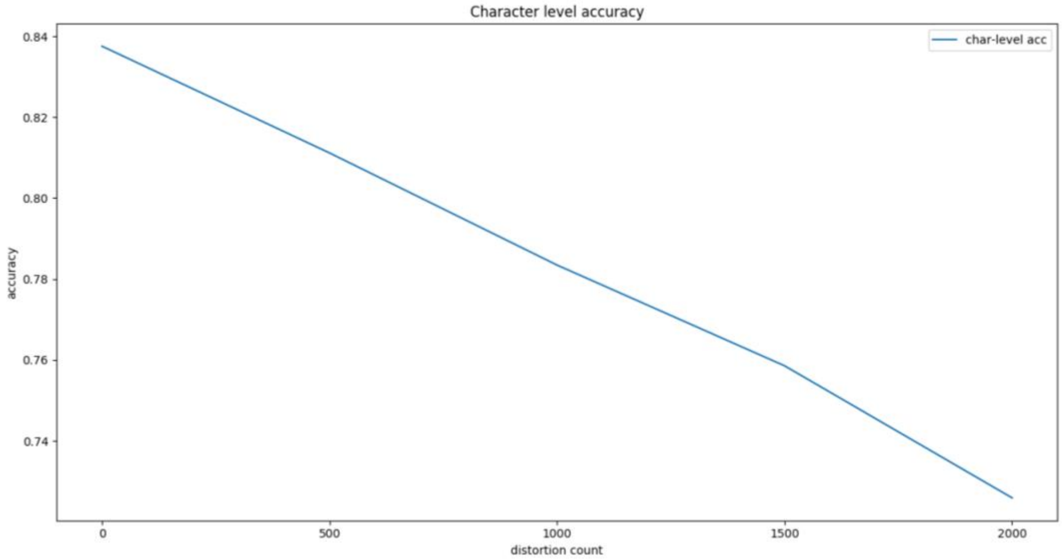
The above graphs for SVM-HMM and Conditional Random Fields correspond to the theoretical interpretation. As C increases, the model begins to fit the data better and obtains better test time word and character level accuracy.

However, counter to the other models, Linear SVMs from the scikit-learn package tend to perform poorly at test time with an increase in C. We feel that this is perhaps because the model is over-fitting, since we can measure the disparity between the train and test scores, where the train accuracy sharply rises, yet test accuracy drops as C increases.

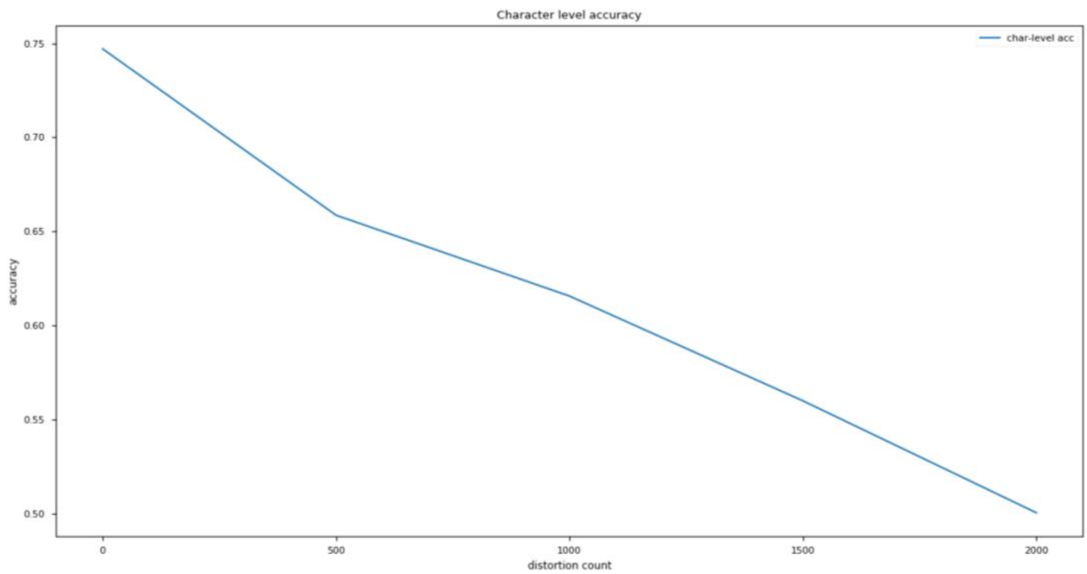
As an ablation test, we attempted to use smaller values of C from the range 1e-3 to 1.0, and found that at smaller Cs, both test and train accuracy are much lower than at C = 1. However, they both steadily rise till C = 1 point. This leads to a conclusion that perhaps the linear multi-class SVM is simply over-fitting at larger values of C, which corresponds to our theoretical understanding.

Answer 4)

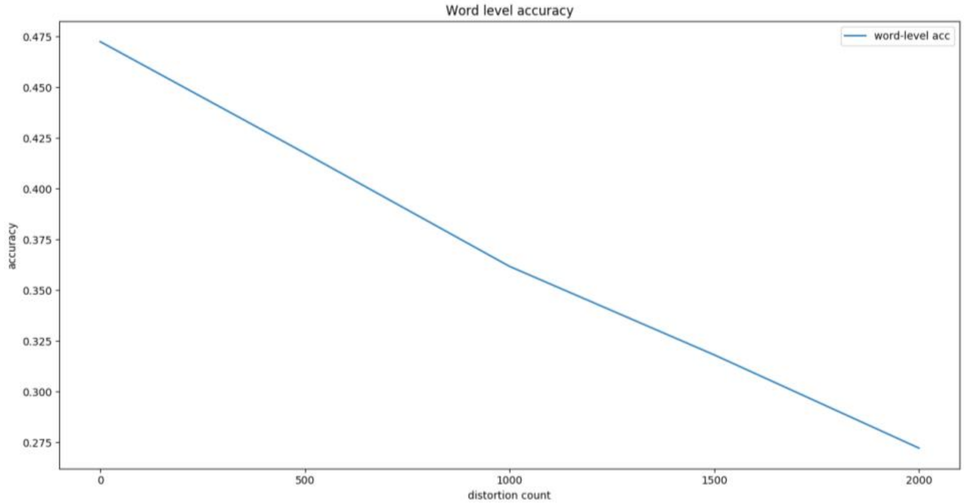
A1) Letter-wise prediction of Test using CRF at C = 1000



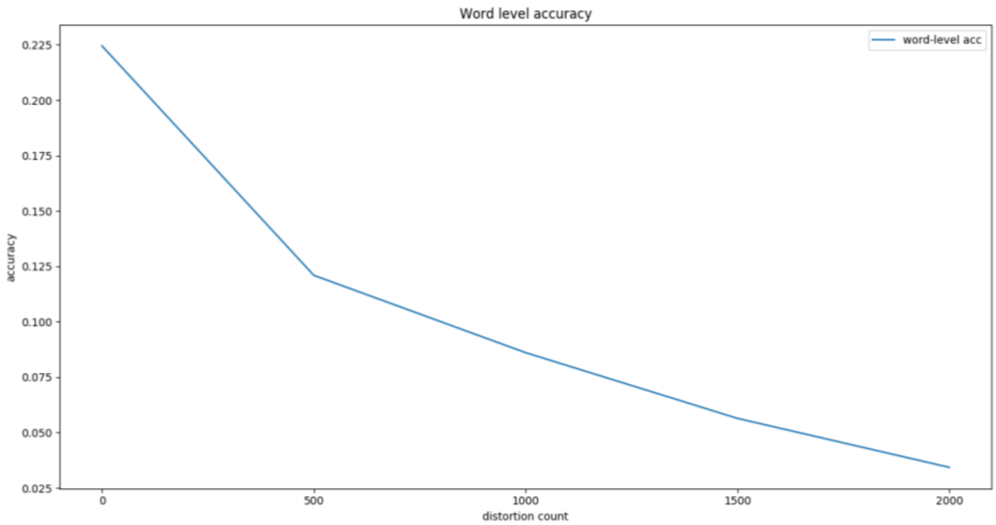
A2) Letter-wise prediction of Test using SVM-MC at C = 1000



B1) Word-wise prediction of Test using CRF at C = 1000



B2) Word-wise prediction of Test using SVM-MC at C = 1000



Observations:

As can be seen from the above graphs, distortions in the training data causes the overall performance of the model to degrade at test time (which are not affected by such distortions).

Somewhat surprisingly, we find that while both models tend to perform worse as more distortions occur, the drop in performance of the Conditional Random Field models is much smoother – dropping from **83.75** % to **71** % (character level accuracy) at 2000 distorted words. Word level accuracy also drops smoothly – from **47.5** % to **27.5** %.

This is in stark contrast to the Linear Multi-Class SVM, whose performance drops sharply – dropping from **75** % to **50** % (character level accuracy) and **22.5** % to **2.5** %. This sharp decrease in performance leads us to believe that the Conditional Random Field model is far more robust than the Linear Multi-Class Support Vector Machine.