**CS 594: Advanced Machine Learning** **Due: 13:00, Mar 8, 2018**

# Project 1: Conditional Random Fields for Structured Output Prediction

*Student: Kristine H. Lee, Ashwini Naik, Juan Trelles* *Email: khlee2,anaik3,jtrell2@uic.edu*

1. Kristine H. Lee, khlee2@uic.edu, UIN:663618002
2. Ashwini Naik, anaik3@uic.edu, UIN: 670912216
3. Juan Trelles Trabucco jtrell2@uic.edu, UIN: 672755496

## 1 Conditional Random Fields

(1a) Show that $\nabla_{\mathbf{w}_y} \log p(\mathbf{y}|X)$—the gradient of $\log p(\mathbf{y}|X)$ with respect to $\mathbf{w}_y$—can be written as:

$$\nabla_{\mathbf{w}_y} \log p(\mathbf{y}^t|X^t) = \sum_{s=1}^{m} (\llbracket y_s^t = y \rrbracket - p(y_s = y|X^t))\mathbf{x}_s^t, \tag{1}$$

where $\llbracket \cdot \rrbracket = 1$ if $\cdot$ is true, and 0 otherwise. Show your derivation step by step.

Now derive the similar expression for $\nabla_{T_{ij}} \log p(\mathbf{y}|X)$.

$$p(\mathbf{y}|X) = \frac{1}{Z_X} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}}\right) \tag{2}$$

$$\text{where} \quad Z_X = \sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right). \tag{3}$$

$$\log p(\mathbf{y}|X) = \log\left(\frac{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}\right)$$

$$\log p(\mathbf{y}|X) = \sum_{s=1}^{m} \langle \mathbf{w}_{y_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{y_s, y_{s+1}} - \log\left(\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)\right)$$

$$\nabla_{\mathbf{w}_y} \log p(\mathbf{y}|X) =$$

$$\sum_{s=1}^{m} \llbracket y_s = y \rrbracket \cdot \mathbf{x}_s - \frac{\sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right) \cdot \left(\nabla_{\mathbf{w}_y} \left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, y_{\hat{z}+1}}\right)\right)\}}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}$$

$$\sum_{s=1}^{m} \llbracket y_s = y \rrbracket \cdot \mathbf{x}_s - \sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \left\{\left(\frac{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}\right) \cdot \sum_{s=1}^{m} \llbracket \hat{z}_s = y \rrbracket \cdot \mathbf{x}_s\right\}$$

$$\sum_{s=1}^{m} \llbracket y_s = y \rrbracket \cdot \mathbf{x}_s - \sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{p(\hat{\mathbf{z}}|X) \cdot \sum_{s=1}^{m} \llbracket \hat{z}_s = y \rrbracket \cdot \mathbf{x}_s\}$$

$$\sum_{s=1}^{m} \llbracket y_s = y \rrbracket \cdot \mathbf{x}_s - \sum_{s=1}^{m} \mathbf{x}_s \left(\sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} p(\hat{\mathbf{z}}|X) \cdot \llbracket \hat{z}_s = y \rrbracket\right)$$

$\sum_{s=1}^{m} [\![y_s = y]\!] \cdot \mathbf{x}_s - \sum_{s=1}^{m} \mathbf{x}_s \cdot p(\hat{\mathbf{z}}_s = y|X)$

$\sum_{s=1}^{m} ([\![y_s = y]\!] - p(\hat{\mathbf{z}}_s = y|X)) \cdot \mathbf{x}_s$

$\nabla_{T_{i,j}} \log p(\mathbf{y}|X) =$

$\sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!] - \frac{\sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right) \cdot \left(\nabla_{T_{i,j}} \left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, y_{\hat{z}+1}}\right)\right)\}}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}$

$\sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!] - \sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{ \left( \frac{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)} \right) \cdot \sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!] \}$

$\sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!] - \sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \{p(\hat{\mathbf{y}}|X) \cdot \sum_{s=1}^{m-1} [\![y_s = i, y_{s+1} = j]\!]\}$

$\sum_{s=1}^{m-1} ([\![y_s = i, y_{s+1} = j]\!] - p(y_s = i, y_{s+1} = j|X))$

(1b) Derivation that the gradient of $\log Z_X$ with respect to $\mathbf{w}_y$ and $T$ is exactly the expectation of the features with respect to $p(\mathbf{y}|X)$. The features are a column vector of indicator functions with respect to the possible values of $y_s$, $\phi_{\mathbf{w}_y}(x)$ in equation (4), and indicator functions with respect of $y_s$ and $y_{s+1}$, $\phi_T(x)$ in equation (5).

$\log Z_X = \log \sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)$

$\nabla_{\mathbf{w}_y} log Z_X = \frac{\sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right) \cdot \left(\nabla_{\mathbf{w}_y} \left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, y_{\hat{z}+1}}\right)\right)\}}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}$

$= \sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{ \left( \frac{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)} \right) \cdot \sum_{s=1}^{m} [\![\hat{y}_s = y]\!] \cdot \mathbf{x}_s \}$

$= \sum_{s=1}^{m} \mathbf{x}_s \cdot p(y|X) \cdot [\![\hat{y}_s = y]\!]$

$= \mathbb{E}[\sum_{s=1}^{m} \mathbf{y}_s = y]$

$= \mathbb{E}(\phi_{\mathbf{w}_y}(x)), \quad$ where

$$\phi_{\mathbf{w}_y}(x) = \begin{bmatrix} [\![\mathbf{y}_2 = y]\!] \\ [\![\mathbf{y}_3 = y]\!] \\ \vdots \\ [\![\mathbf{y}_s = y]\!] \end{bmatrix} \tag{4}$$

$\nabla_{T_{i,j}} log Z_X = \frac{\sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right) \cdot \left(\nabla_{T_{i,j}} \left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, y_{\hat{z}+1}}\right)\right)\}}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)}$

$$= \sum_{\hat{\mathbf{z}} \in \mathcal{Y}^m} \left\{ \left( \frac{\exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{z}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{z}_s, \hat{z}_{s+1}}\right)}{\sum_{\hat{\mathbf{y}} \in \mathcal{Y}^m} \exp\left(\sum_{s=1}^{m} \langle \mathbf{w}_{\hat{y}_s}, \mathbf{x}_s \rangle + \sum_{s=1}^{m-1} T_{\hat{y}_s, \hat{y}_{s+1}}\right)} \right) \cdot \sum_{s=1}^{m-1} [\![ \mathbf{y}_s = i, \mathbf{y}_{s+1} = j ]\!] \right\}$$

$$= \sum_{s=1}^{m-1} p(y|X) \cdot [\![ \mathbf{y}_s = i, \mathbf{y}_{s+1} = j ]\!]$$

$$= \mathbb{E}[\textstyle\sum_{s=1}^{m-1} \mathbf{y}_s = i \wedge \mathbf{y}_{s+1} = j]$$

$$= \mathbb{E}(\phi_T(x)), \quad \text{where}$$

$$\phi_T(x) = \begin{bmatrix} [\![ \mathbf{y}_1 = i \wedge \mathbf{y}_2 = j ]\!] \\ [\![ \mathbf{y}_2 = i \wedge \mathbf{y}_3 = j ]\!] \\ \vdots \\ [\![ \mathbf{y}_{s-1} = i \wedge \mathbf{y}_s = j ]\!] \end{bmatrix} \tag{5}$$

(1c) We implemented a dynamic programming algorithm for the decoder based on the Viterbi algorithm ($O(m|Y|^2)$) and a brute force approach for validating our results. The results are in the `results/decode_output.text` file, and the maximum objective value on the provided set was 200.1852.

1. initialize memo table with the potential $\omega_i^\top \cdot x_1$
2. for $i = 2$ to remaining characters
3. for each entry in the memo
4. l = letter in memo that maximizes T(l,k) + memo(l), k in alphabet
5. update memo entry with T(l,i) + memo(l) + $\omega_i^\top \cdot x_i$
6. store the subword until the $i$th iteration appending l
7. end
8. end
9. Find the maximum value in memo table and return the sequence associated with it

## 2 Training Conditional Random Fields

(2a) We implemented the algorithms for computing $\log p(\mathbf{y}^i|X^i)$ and its its gradients with respect to $\omega_i$ and $T_{y_i, y_{i+1}}$. The implementation of $\log p(\mathbf{y}^i|X^i)$ uses the equations in (1) and (2). We obtained normalizer term $logZ$ from the memo table of the forwards algorithm where $logZ$ is specified in equation (Project 1: eq. 14). The resulting values for $\frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{w}_y} \log p(\mathbf{y}^i|X^i)$ and $\frac{1}{n}\sum_{i=1}^{n} \nabla_T \log p(\mathbf{y}^i|X^i)$ on the training data set are in the `results/gradient.txt` file as a column vector following the order specified in (Project 1: 9).
$\frac{1}{n}\sum_{i=1}^{n} \log p(\mathbf{y}^i|X^i) = -31.2884$

All our calculations used the memo trick ($M + \log \sum_i \exp(x_i - M)$) for numerical robustness. Also, we tested the values against gradest with sequences of different sizes and random values for $\omega$ and $T$.

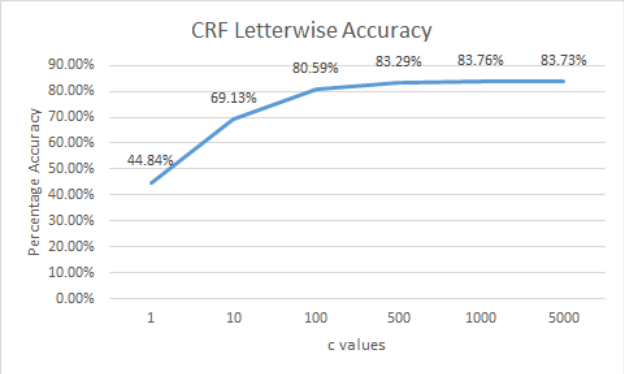


**Figure 1.** Letter-wise prediction accuracy per value of C of our CRF approach

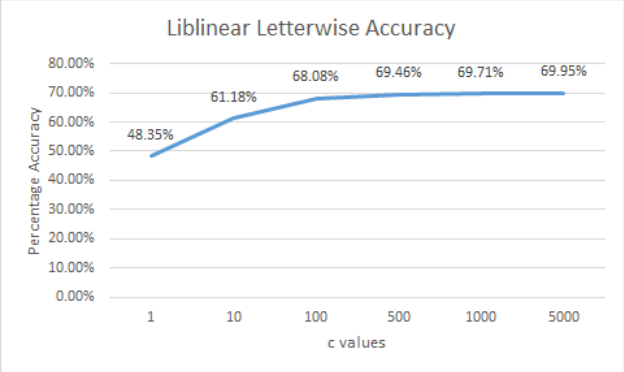


**Figure 2.** Letter-wise prediction accuracy per value of C using the SVM-MC approach (Liblinear wrapper for Matlab). The C values for liblinear were normalized by the number of letters in the training set.

(2b) We used the `fminunc` function from Matlab's Optimization Toolbox to train our model with a parameter $C = 1000$, $\omega$ and $T$ vectors initialized as zeros, and using the default parameters for the solver as specified in the reference file `ref_optimize.m`. We stored the optimal model in `results/solution.txt` and the predictions for the letters in the test data set in `results/prediction.txt`. After 100 iterations, we obtained an optimal objective value for (Project 1: eq. 8) of $f(x) = 3716.85$.

## 3 Benchmarking with Other Methods

(3a) We evaluated the letter-wise predictions of the CRF (Fig.1), SVM-MC (Fig.2) and SVM-Struct (Fig.3) approaches with different `c` values ($[1, 10, 100, 500, 1000, 5000]$). The results show that lower values of `c` obtain a lower prediction accuracy as the weight on the training data is smaller; hence, the generalization is poor. For the CRF and SVM-MC approaches, the prediction scores start to increase slowly for values of `c` greater than 100. The SVM-Struct starts to show the same behavior for values of `c` greater than 500. In general, the prediction score stabilizes from `c`=1000 and onwards.

In comparison to LibLinear, SVM-HMM's accuracy is worse on lower c values, but increases more rapidly with higher c values. Also, our CRF method outperforms the other two approaches. For comparison purposes, we superimposed the results of the three methods for

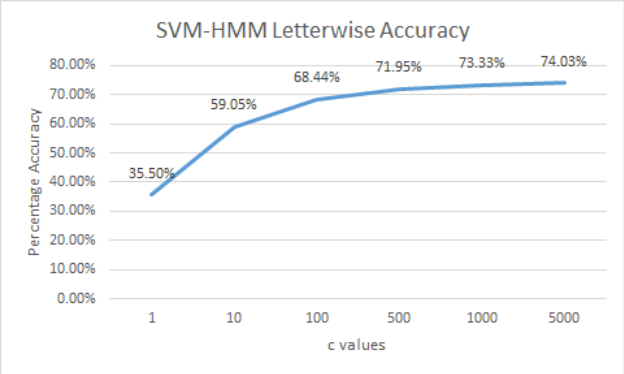


**Figure 3.** Letter-wise prediction accuracy per value of C using the SVM-Struct approach ($SVM^{hmm}$ binaries).

letters and images in Fig.4.

(3b) For the word-wise predictions, we observed that the predictions are not as accurate as for the letters case. This result is expected as it is less likely to match the whole letters in a word. We can also observe that our CRF approach is not longer outperforming the other two approaches for different values of `c` (Fig.8). Specifically, for a `c`=1000, the CRF had the highest prediction accuracy but for `c`=5000, it has the lowest among the three approaches. Moreover, while the SVM-Struct and CRF methods appear to be obtaining smaller accuracy values `c` greater than 1000, the SVM-MC shows an increasing slope.

## 4 Robustness to Tampering

(4a) We created the plots using the tampered datasets with no padding. We choose `c=500` for our CRF approach, and `c=5000` for SVM-MC. We noticed that the prediction degrades linearly when the number of randomly tampered images augment. In general, our approach works better than the SVM-MC. We can conclude that both methods are severely affected by the tampered images for letter predictions.

(4b) We used the same `c` values as in question 4(a) for the required plots (Fig.11 and Fig.12). Our approach performed better than SVM-MC only for 500 and 1000 tampered images. However, as the number of images increase, SVM-MC obtains higher accuracy in the predictions. We can say that both methods are severely affected by the rotations and translations applied to the test sample.

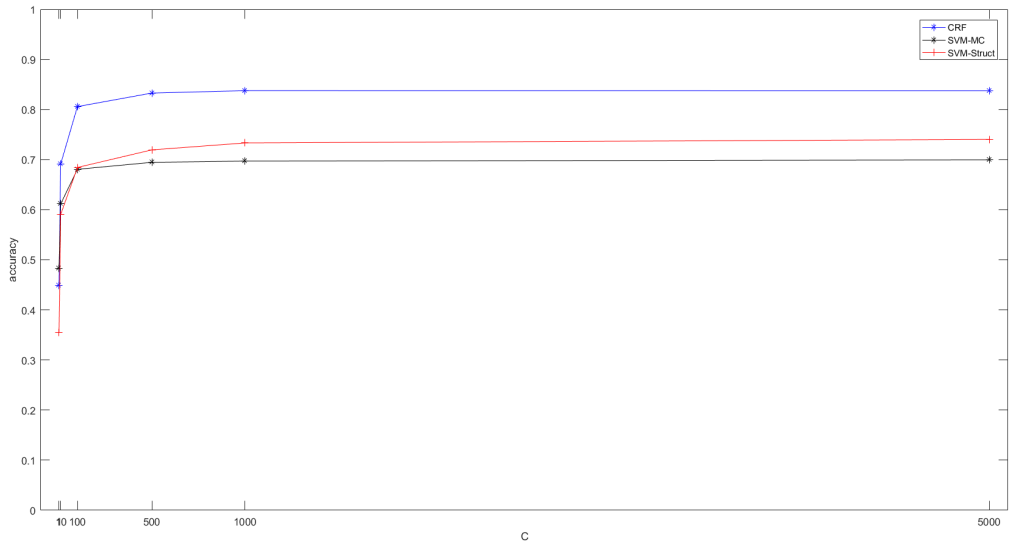


**Figure 4.** Letter-wise prediction comparison of the three approaches. The X-axis is not presented as categorical values to show the slope changes.

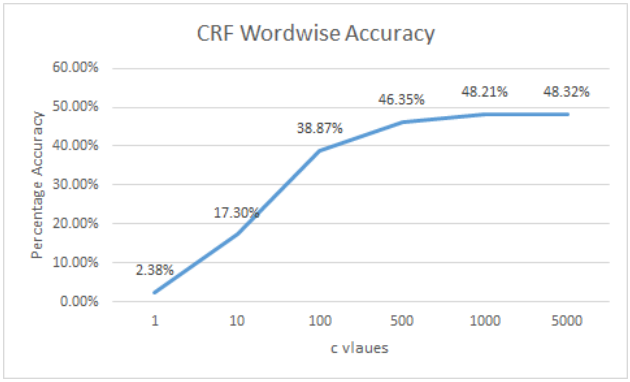


**Figure 5.** Word-wise prediction accuracy per value of C of our CRF approach

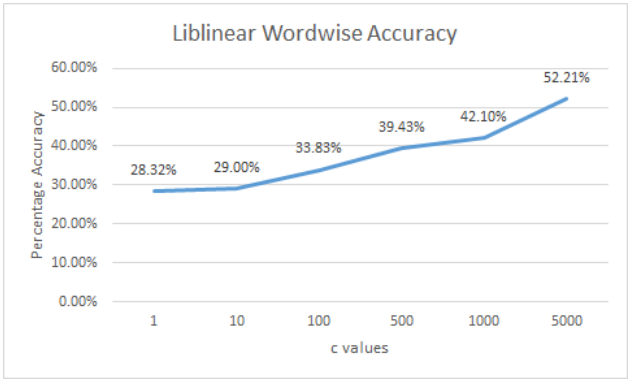


**Figure 6.** Word-wise prediction accuracy per value of C of the SVM-MC approach (Liblinear in Python)

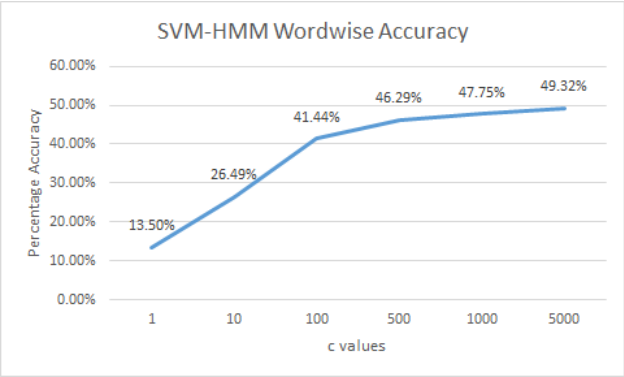


**Figure 7.** Word-wise prediction accuracy per value of C of the SVM-Struct approach

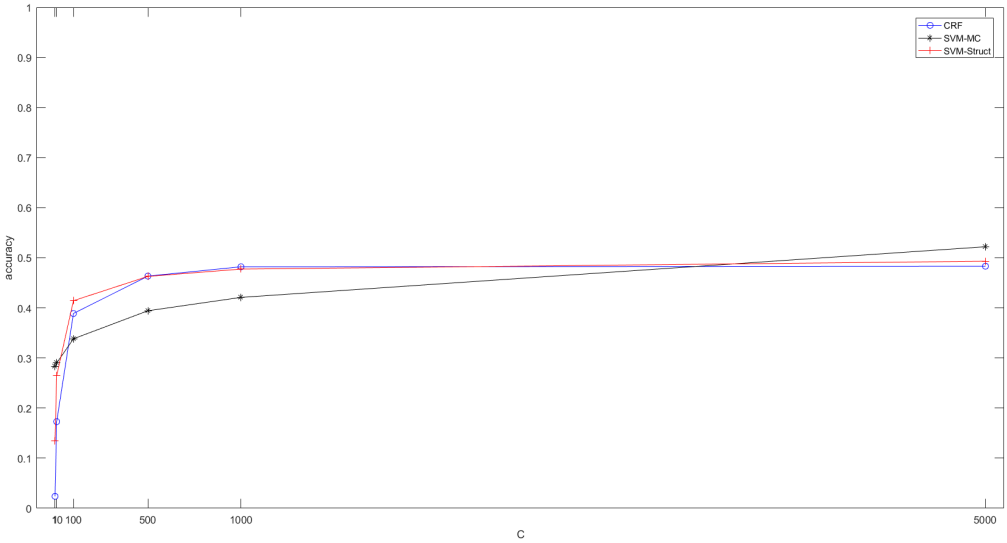


**Figure 8.** Word-wise prediction comparison of the three approaches. The X-axis is not presented as categorical values to show the slope changes.

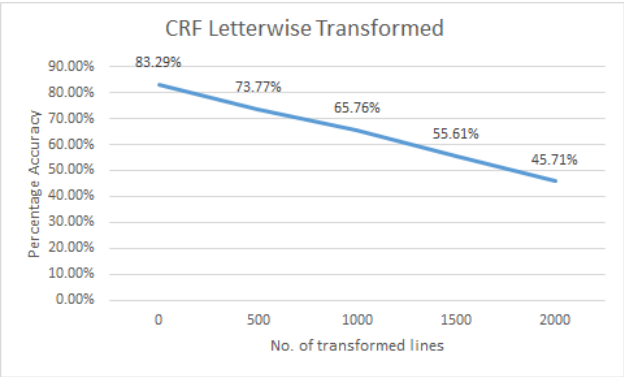


**Figure 9.** CRF predictions on a test set with different number of tampered images.

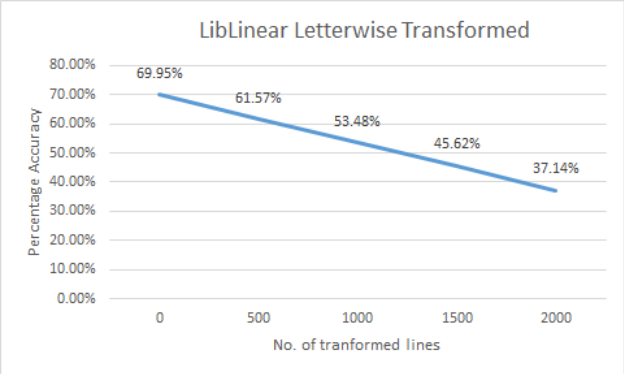


**Figure 10.** SVM-MC predictions on a test set with different number of tampered images.

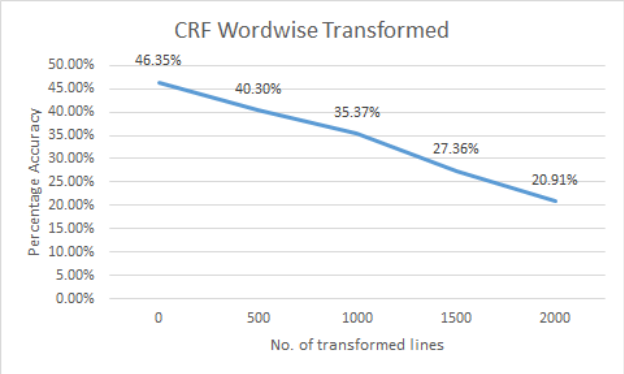


**Figure 11.** CRF word predictions on a test set with different number of tampered images.

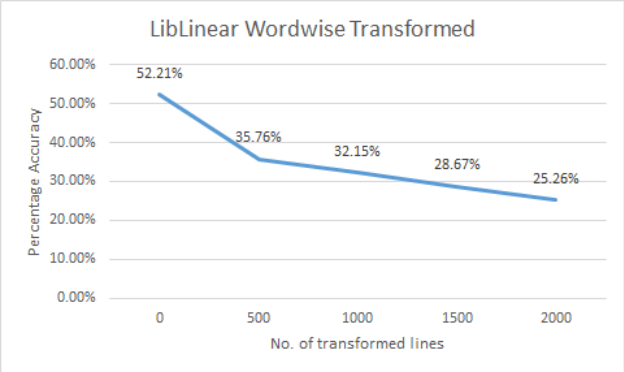


**Figure 12.** SVM-MC word predictions on a test set with different number of tampered images.