CS 224N Assignment #1

1 Softmax

(a) prove:

$$softmax(x)_i = \frac{e^{x_i}}{\Sigma_j e^{x_j}}$$

$$softmax(x+c)_i = \frac{e^{x_i+c}}{\Sigma_j e^{x_j+c}} = \frac{e^c \cdot e^{x_i}}{e^c \cdot \Sigma_j e^{x_j}} = \frac{e^{x_i}}{\Sigma_j e^{x_j}} = softmax(x)_i$$

For each i,

$$softmax(x+c)_i = softmax(x)_i$$

Hence,

$$softmax(x+c) = softmax(x)$$

(b) python code

2 Neural Network Basics

(a) derive:

$$\sigma(x) = \frac{1}{1+e^{-x}}$$

$$\frac{d}{dx}\sigma(x) = \frac{d}{dx}\frac{1}{1+e^{-x}}$$

Denote

$$u = 1+e^{-x}, \text{then } \sigma(u) = \frac{1}{u}$$

We have

$$\frac{d}{dx}\sigma(x) = \frac{d\sigma(u)}{du} \cdot \frac{du}{dx} = -\frac{1}{u^2} \cdot (-e^{-x}) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1+e^{-x}-1}{(1+e^{-x})^2} = \sigma(x) - \sigma^2(x)$$

(b) derive:

We have already known that

$$CE(y,\hat{y}) = -\Sigma_i y_i \log(\hat{y}_i)$$

$$\hat{y} = softmax(\theta) = \frac{e^{\theta_i}}{\Sigma_j e^{\theta_j}}$$

Assume that, for $y_k = 1$ and $y_{\neq k} = 0$,

$$\frac{\partial}{\partial\theta} CE(y,\hat{y}) = -\frac{\partial}{\partial\theta} \Sigma_i y_i \log(\hat{y}_i)$$

When $i = k$,

$$-y_i \log(\hat{y}_i) = -\log(\hat{y}_k)$$

When $i \neq k$,

$$-y_i \log(\hat{y}_i) = 0$$

hence,

$$\frac{\partial}{\partial\theta} CE(y,\hat{y}) = -\frac{\partial}{\partial\theta}\log(\hat{y}_k) = -\frac{\partial}{\partial\theta}\log\frac{e^{\theta_k}}{\Sigma_j e^{\theta_j}} = -\frac{\partial}{\partial\theta}\left(\theta_k - \log\Sigma_j e^{\theta_j}\right) = \frac{\partial}{\partial\theta}\log\Sigma_j e^{\theta_j} - \frac{\partial}{\partial\theta}\theta_k$$

$$\frac{\partial}{\partial\theta} CE(y,\hat{y}) = \frac{1}{\Sigma_j e^{\theta_j}} \cdot \Sigma_l \frac{\partial}{\partial\theta} e^{\theta_l} - \frac{\partial}{\partial\theta}\theta_k$$

$$\frac{\partial}{\partial\theta}CE(y,\hat{y}) = \frac{1}{\Sigma_j e^{\theta_j}} \cdot \Sigma_l \frac{\partial}{\partial\theta} e^{\theta_l} - \frac{\partial}{\partial\theta}\theta_k = \frac{1}{\Sigma_j e^{\theta_j}}\begin{pmatrix}\theta_1\\ \vdots\\ \theta_k\\ \vdots\\ \theta_n\end{pmatrix} - \begin{pmatrix}0\\ \vdots\\ 1\\ \vdots\\ 0\end{pmatrix} = \hat{y} - y$$

(c) derive:

We have already known that

From 1 (a), we have

$$softmax(x+c) = softmax(x)$$

From 2 (a), we have

$$\sigma'(x) = \frac{d}{dx}\sigma(x) = \sigma(x) - \sigma^2(x)$$

From 2 (b), we have

$$\frac{\partial}{\partial\theta}CE(y,\hat{y}) = \hat{y} - y$$

From 2 (c), we have

$$h = sigmoid(xW_1 + b_1) = \sigma(xW_1 + b_1)$$
$$\hat{y} = softmax(hW_2 + b_2)$$

Assume

$$\theta = hW_2 + b_2, hence\ \hat{y} = softmax(\theta)$$
$$u = xW_1 + b_1, hence\ h = \sigma(u)$$

Here, we denote $\cdot$ as dot product and denote $\circ$ as element-wise product.

Since $\sigma(u)$ is an element-wise function, hence before $\frac{\partial\sigma(u)}{\partial u}$ is an element-wise product.

$$\frac{\partial J}{\partial x} = \left(\left(\frac{\partial J}{\partial\theta}\cdot\frac{\partial\theta}{\partial h}\right)\circ\frac{\partial h}{\partial u}\right)\cdot\frac{\partial u}{\partial x} = \left(\left(\frac{\partial}{\partial\theta}CE(y,\hat{y})\cdot\frac{\partial(hW_2+b_2)}{\partial h}\right)\circ\frac{\partial\sigma(u)}{\partial u}\right)\cdot\frac{\partial(xW_1+b_1)}{\partial x}$$

$$\frac{\partial J}{\partial x} = \left(\left((\hat{y}-y)\,W_2^T\right)\circ\sigma'(u)\right)W_1^T = (\hat{y}-y)W_2^T\circ\sigma'(xW_1+b_1)W_1^T$$

Check the dimension by analysis.

Since J is a scalar, $\frac{\partial J}{\partial x}$ must have the same dimension as x.

Assume the input x is $1\times D_x$, output y is $1\times D_y$, hidden units h is $1\times H$, then

- $(\hat{y}-y)$ has dimension $1\times D_y$
- W_1 has dimension $D_x\times H$
- W_2 has dimension $H\times D_y$
- $xW_1 + b_1$ has dimension $1\times H$
- $\sigma'(xW_1+b_1)$ has dimension $1\times H$

Hence, the dimension of $\frac{\partial J}{\partial x}$ has dimension $(1\times D_y)\times(D_y\times H)\circ(1\times H)\times(H\times D_x) = 1\times D_x$

(d) answer:

The parameters we need are W_1, b_1, W_2, b_2.

Total number is $D_x\times H + 1\times H + H\times D_y + 1\times D_y$

(e) python code

(f) python code

(g) python code

3 word2vec

(a) derive

$J = CE(y, \hat{y}) = \Sigma_i - y_i \log \hat{y}$

When $i = o$,

$$J = -\log \hat{y}_o = -\log p(o|c) = -\log \frac{\exp(u_o^T v_c)}{\Sigma_{w=1}^W \exp(u_w^T v_c)} = -u_o^T v_c + \log \Sigma_{w=1}^W \exp(u_w^T v_c)$$

The gradient of J, w.r.t. v_c

$$\frac{\partial J}{\partial v_c} = -u_o^T + \frac{\partial}{\partial v_c} \log \Sigma_{w=1}^W \exp(u_w^T v_c) = -u_o^T + \frac{\Sigma_{w=1}^W \frac{\partial}{\partial v_c} \exp(u_w^T v_c)}{\Sigma_{x=1}^W \exp(u_x^T v_c)}$$

Assume v_c is a column vector, and u_w^T is a row vector,

$$\frac{\partial J}{\partial v_c} = -u_o^T + \frac{\Sigma_{w=1}^W \frac{\partial}{\partial v_c} \exp(u_w^T v_c)}{\Sigma_{x=1}^W \exp(u_x^T v_c)} = -u_o^T + \frac{\Sigma_{w=1}^W \exp(u_w^T v_c) \cdot u_w^T}{\Sigma_{x=1}^W \exp(u_x^T v_c)}$$

$$\boxed{\frac{\partial J}{\partial v_c} = -u_o^T + \Sigma_{w=1}^W p(w|c) \cdot u_w^T}$$

(b) derive

$$J = -\log \hat{y}_o = -\log p(o|c) = -\log \frac{\exp(u_o^T v_c)}{\Sigma_{w=1}^W \exp(u_x^T v_c)} = -u_o^T v_c + \log \Sigma_{w=1}^W \exp(u_w^T v_c)$$

The gradient of J, w.r.t. u_w

- When $w = o$,

$$\frac{\partial J}{\partial u_w} = -v_c + \frac{\partial}{\partial u_w} \log \Sigma_{w=1}^W \exp(u_w^T v_c) = -v_c + \frac{\Sigma_{w=1}^W \frac{\partial}{\partial u_w} \exp(u_w^T v_c)}{\Sigma_{x=1}^W \exp(u_x^T v_c)}$$

$$\boxed{\frac{\partial J}{\partial u_w} = -v_c + \frac{\exp(u_w^T v_c) \cdot v_c}{\Sigma_{x=1}^W \exp(u_x^T v_c)}}$$

- When $w \neq o$,

$$\frac{\partial J}{\partial u_w} = \frac{\partial}{\partial u_w} \log \Sigma_{w=1}^W \exp(u_w^T v_c) = \frac{\Sigma_{w=1}^W \frac{\partial}{\partial u_w} \exp(u_w^T v_c)}{\Sigma_{x=1}^W \exp(u_x^T v_c)}$$

$$\boxed{\frac{\partial J}{\partial u_w} = \frac{\exp(u_w^T v_c) \cdot v_c}{\Sigma_{x=1}^W \exp(u_x^T v_c)}}$$

(c) derive:

$J = -\log(\sigma(u_o^T v_c)) - \Sigma_{k=1}^K \log\left(\sigma(-u_k^T v_c)\right)$

The gradient of J, w.r.t. v_c

$$\frac{\partial J}{\partial v_c} = \frac{\partial}{\partial v_c}\left\{-\log(\sigma(u_o^T v_c)) - \Sigma_{k=1}^K \log\left(\sigma(-u_k^T v_c)\right)\right\}$$

$$\frac{\partial J}{\partial v_c} = -\frac{\partial}{\partial v_c} \log(\sigma(u_o^T v_c)) - \Sigma_{k=1}^K \frac{\partial}{\partial v_c} \log\left(\sigma(-u_k^T v_c)\right)$$

Where

$$\frac{\partial}{\partial v_c}\log\left(\sigma(u_o^T v_c)\right) = \frac{1}{\sigma(u_o^T v_c)}\sigma(u_o^T v_c)\left(1-\sigma(u_o^T v_c)\right)\cdot u_o = \left(1-\sigma(u_o^T v_c)\right)\cdot u_o$$
$$\frac{\partial}{\partial v_c}\log\left(\sigma(-u_k^T v_c)\right) = \frac{1}{\sigma(-u_k^T v_c)}\sigma(-u_k^T v_c)\left(1-\sigma(-u_k^T v_c)\right)\cdot(-u_k) = \left(1-\sigma(-u_k^T v_c)\right)\cdot(-u_k)$$

Hence

$$\boxed{\frac{\partial J}{\partial v_c} = -\left(1-\sigma(u_o^T v_c)\right)\cdot u_o + \Sigma_{k=1}^K\left(1-\sigma(-u_k^T v_c)\right)\cdot u_k}$$

The gradient of J, w.r.t. u_w

- When $w = o$,

$$\frac{\partial J}{\partial u_w} = \frac{\partial}{\partial u_w}\left\{-\log\left(\sigma(u_o^T v_c)\right) - \Sigma_{k=1}^K\log\left(\sigma(-u_k^T v_c)\right)\right\}$$
$$\frac{\partial J}{\partial u_w} = -\frac{\partial}{\partial u_w}\log\left(\sigma(u_o^T v_c)\right) - \Sigma_{k=1}^K\frac{\partial}{\partial u_w}\log\left(\sigma(-u_k^T v_c)\right)$$

Where

$$\frac{\partial J}{\partial u_w}\log\left(\sigma(u_o^T v_c)\right) = \frac{1}{\sigma(u_o^T v_c)}\sigma(u_o^T v_c)\left(1-\sigma(u_o^T v_c)\right)\cdot v_c = \left(1-\sigma(u_o^T v_c)\right)\cdot v_c$$
$$\Sigma_{k=1}^K\frac{\partial}{\partial u_w}\log\left(\sigma(-u_k^T v_c)\right) = 0, \text{since } o \notin \{1,2,\dots,K\}$$

Hence,

$$\boxed{\frac{\partial J}{\partial u_w} = \left(\sigma(u_o^T v_c)-1\right)\cdot v_c}$$

- When $w \neq o$ and $w \in \{1,2,\dots,K\}$,

$$\frac{\partial J}{\partial u_w} = \frac{\partial}{\partial u_w}\left\{-\log\left(\sigma(u_o^T v_c)\right) - \Sigma_{k=1}^K\log\left(\sigma(-u_k^T v_c)\right)\right\}$$
$$\frac{\partial J}{\partial u_w} = -\frac{\partial}{\partial u_w}\log\left(\sigma(u_o^T v_c)\right) - \Sigma_{k=1}^K\frac{\partial}{\partial u_w}\log\left(\sigma(-u_k^T v_c)\right)$$

Where

$$\frac{\partial}{\partial u_w}\log\left(\sigma(u_o^T v_c)\right) = 0, \text{since } w \neq o$$
$$\Sigma_{k=1}^K\frac{\partial}{\partial u_w}\log\left(\sigma(-u_k^T v_c)\right) = \frac{1}{\sigma(-u_k^T v_c)}\sigma(-u_k^T v_c)\left(1-\sigma(-u_k^T v_c)\right)\cdot(-v_c) = \left(\sigma(-u_k^T v_c)-1\right)\cdot v_c$$

Hence,

$$\boxed{\frac{\partial J}{\partial u_w} = \left(1-\sigma(-u_k^T v_c)\right)\cdot v_c}$$

- When $w \neq o$ and $w \notin \{1,2,\dots,K\}$,

$$\frac{\partial J}{\partial u_w} = 0$$

(d) derive:

For $J_{skip-gram}$, we have

$$\frac{\partial J}{\partial v_c} = \sum_{-m\le j\le m, j\neq 0}\frac{\partial F\left(w_{c+j}, v_c\right)}{\partial v_c}$$
$$\frac{\partial J}{\partial w_{c+j}} = \sum_{-m\le j\le m, j\neq 0}\frac{\partial F\left(w_{c+j}, v_c\right)}{\partial w_{c+j}} = \frac{\partial F\left(w_{c+j}, v_c\right)}{\partial w_{c+j}}, \text{where } -m \le j \le m, j \neq 0$$

For J_{CBOW}, we have

$$\frac{\partial J}{\partial v_{c+j}} = \frac{\partial F(w_c, \hat{v})}{\partial v_{c+j}} = \frac{\partial F(w_c, \hat{v})}{\partial \hat{v}} \cdot \frac{\partial \hat{v}}{\partial v_{c+j}}, \text{where } -m \leq j \leq m, j \neq 0$$

$$\frac{\partial J}{\partial w_c} = \frac{\partial F(w_c, \hat{v})}{\partial w_c}$$

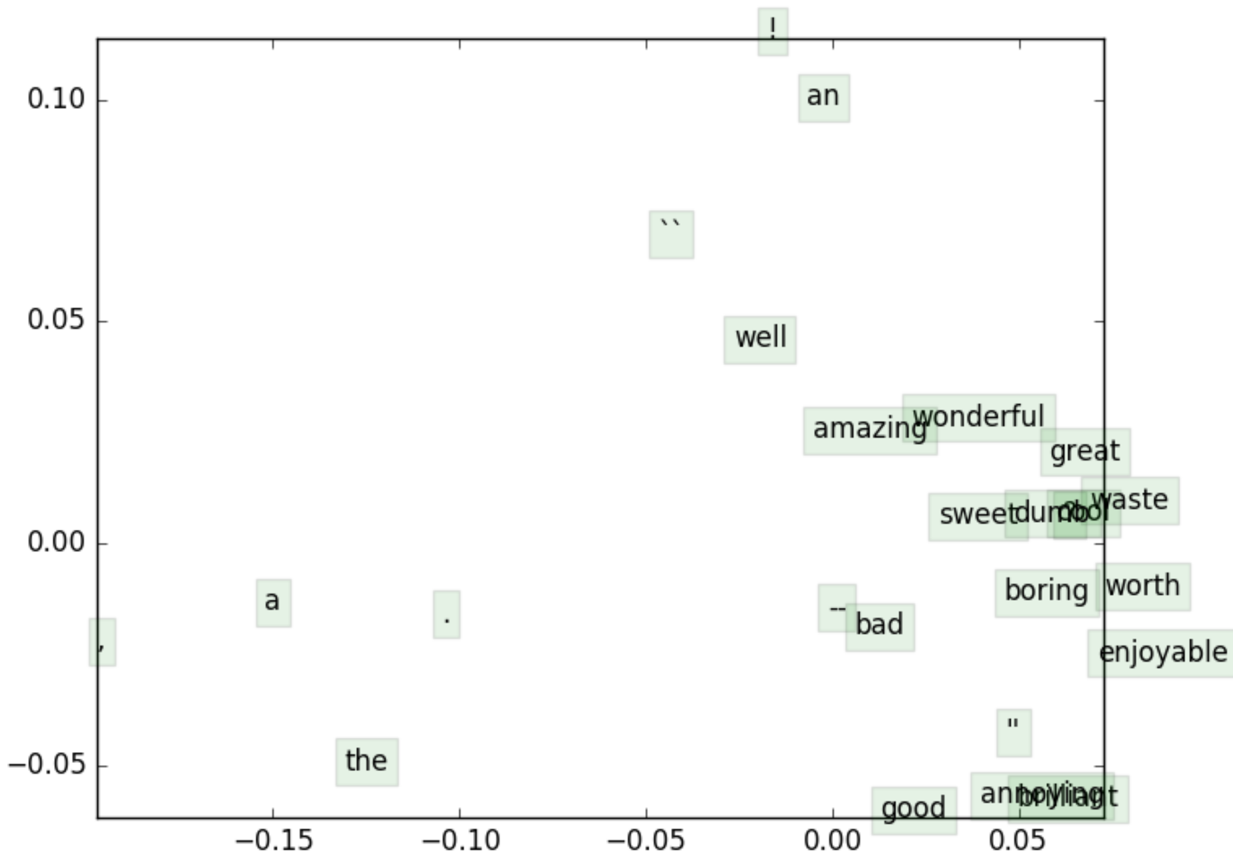
(e) python code

(f) python code

(g) answer:

From the following picture, we can observe:

- Words with similar meaning cluster together.
- Words with different meaning are far from each other.
- Words(Strings) with different part of speech are far from each other.



(h) python code

4 Sentiment Analysis

(a) python code

(b) answer:

Regularization prevents model from overfitting to the training set.

(c) python code

```
maxVal = results[0]["dev"]
idx = 0
for i in xrange(len(results)):
        if results[i]["dev"] > maxVal:
                maxVal = results[i]["dev"]
                idx = i
bestResult = results[idx]
```

(d) answer:

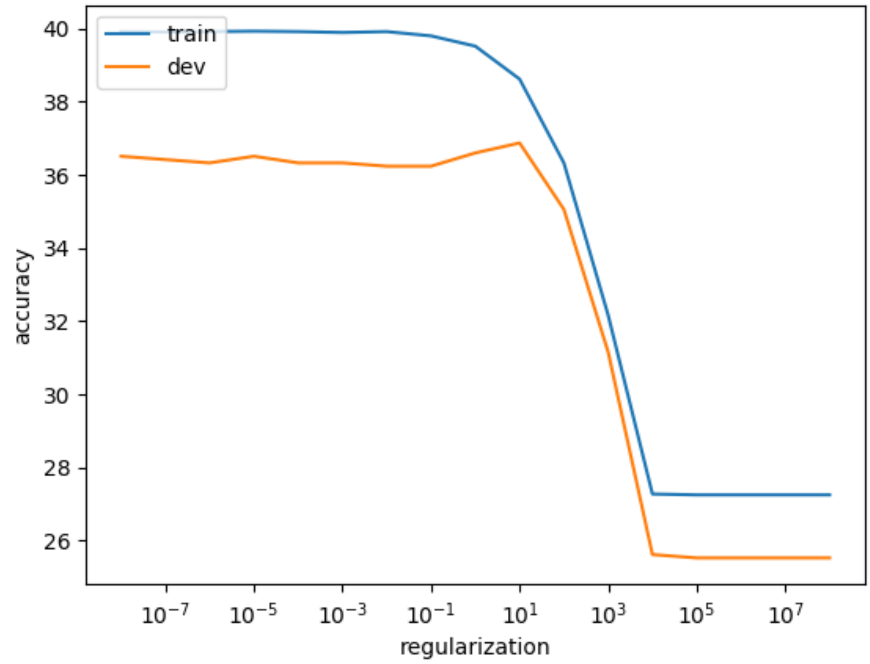
Why does the pretrained vectors perform better?

- The sample set is larger.
- The sample set comes from Wikipedia, which is more standard.
- GloVe algorithm Can capture complex patterns beyond word similarity.

--yourvectors				--pretrained			
Reg	Train	Dev	Test	Reg	Train	Dev	Test
1.00E-08	31.016	32.516	30.452	1.00E-08	39.923	36.512	37.014
1.00E-07	31.016	32.516	30.407	1.00E-07	39.911	36.421	36.968
1.00E-06	31.016	32.516	30.407	1.00E-06	39.923	36.331	36.968
1.00E-05	31.016	32.516	30.452	1.00E-05	39.934	36.512	37.014
1.00E-04	31.016	32.698	30.362	1.00E-04	39.923	36.331	37.014
1.00E-03	31.156	32.698	30.271	1.00E-03	39.899	36.331	37.104
1.00E-02	30.946	32.334	29.910	1.00E-02	39.923	36.240	37.195
1.00E-01	30.290	31.880	29.819	1.00E-01	39.806	36.240	37.149
1.00E+00	28.897	29.609	27.149	1.00E+00	39.525	36.603	37.330
1.00E+01	27.247	25.522	23.077	1.00E+01	38.624	36.876	37.692
1.00E+02	27.247	25.522	23.032	1.00E+02	36.330	35.059	35.701
1.00E+03	27.247	25.522	23.032	1.00E+03	32.163	31.153	30.588
1.00E+04	27.247	25.522	23.032	1.00E+04	27.271	25.613	23.122
1.00E+05	27.247	25.522	23.032	1.00E+05	27.247	25.522	23.032
1.00E+06	27.247	25.522	23.032	1.00E+06	27.247	25.522	23.032
1.00E+07	27.247	25.522	23.032	1.00E+07	27.247	25.522	23.032
1.00E+08	27.247	25.522	23.032	1.00E+08	27.247	25.522	23.032
Best regularization value: 1.00E-04 Test accuracy (%): 30.361991				Best regularization value: 1.00E+01 Test accuracy (%): 37.692308			

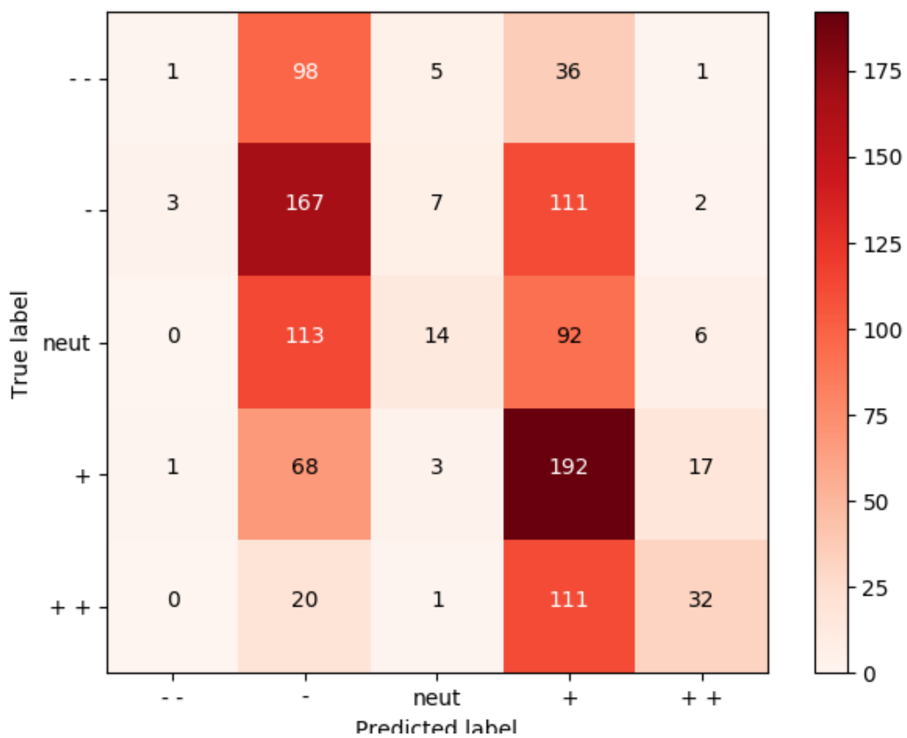
(e) answer:

With the increment of regularization value, the accuracy increases slightly. After regularization value passes 10, the accuracy decreases significantly. The overly large value of regularization takes over the model and reduces the effectiveness of the model.



(f) answer:

The predicted result is not absolutely accurate.
The prediction of (++) and (--) are relatively more accurate.
The prediction of (--), (neut) and (++) are not so accurate.



(g) answer:

	True	Predicted	Sentence
(1)	3	1	we know the plot 's a little crazy, but it held my interest from start to finish.
(2)	4	1	manages to transcend the sex, drugs and show-tunes plot into something far richer.
(3)	1	3	a subject like this should inspire reaction in its audience; the pianist does not.

(1) The word “but” in this sentence change the meaning of the first half.
(2) The second sentence maybe doesn’t take the verb “transcend” into consideration.
(3) The latter half of the third sentence turns over the sentiment of the former half.