

HW2 report

Name: Jiawei Luan Email: luanji@oregonstate.edu

0 Sentiment Classification Task and Dataset

Q: why is each of these steps necessary or helpful for machine learning?

First, lowercase wording reduces the complexity of the language. In general, the words are the same no matter their place in a sentence. For example, "For the first time in forever", "I will be there for you". The initial "For" and the "for" in the middle of the sentence are treated as the same.

Second, punctuation can cause different meanings of a sentence, it can be overlooked by the model, so we treat each punctuation as a separate token.

Third, splitting the verb contractions and possessives helps the model understand the negative verb and the relationship between entities.

Fourth, changing the marks of quotes can help models distinguish the direct speech or quote.

1 Naïve Perceptron Baseline

Q1

Addition It creates a new 'svector' and then adds the elements from the 'self', then it iterates over the other 'svector' ('other') and adds the value of common keys together.

Subtraction It uses the logic of addition directly within the `__sub__` method by inversely adding the elements.

Scalar product ($v * c$) is implemented by `__mul__` method. It returned a new 'svector' with each element multiplied by the scalar 'c'.

Dot product dot methods is an operation that multiplies corresponding elements and then sums up those products.

Negation In `__neg__` method, it creates a new 'svector' where each element is the negated value of the corresponding element in the original 'svector'.

Q2

Test() function takes ``devfile`` and ``model`` as input parameters. For each iteration, it converts the sequence into a sparse feature vector(``make_vector``). And it computes the dot product of the feature vector with the model vector. Also, it checks the product of the label (1,-1) and the dot product ≤ 0 , which would determine a misclassification. If there's a misclassification, it increases the err. Then it returns the error rate.

Train() function takes `trainfile`, `devfile`, and `epochs` as input. Epochs means the number of times the training process will iterate over the entire training dataset. For each epoch, it reads the training file. The label is converted to 1 or -1, and the word is split into a list. The `make_vector` function is called to convert the list of words into a sparse vector. The algorithm checks if the current model makes a correct prediction for the current instance. If the product of the label and the dot product of the model and the feature vector are less than or equal to zero, it updates the model by adding the feature vector scaled by the label. After each epoch, the function evaluates the current model on the dev set using the `test()` function. And keeps tracking the best error rate.

Q3

In the `make_vector` function, add:
if bias:

```
v['<bias>'] = 1
```

This allows the model to learn an offset is not tied to any input feature. This can be helpful when the optimal decision boundary does not pass through the origin of the feature space.

```
epoch 1, update 38.8%, dev 36.6%
epoch 2, update 25.1%, dev 34.6%
epoch 3, update 20.7%, dev 33.8%
epoch 4, update 16.7%, dev 31.7%
epoch 5, update 13.8%, dev 34.0%
epoch 6, update 12.1%, dev 31.9%
epoch 7, update 10.3%, dev 30.1%
epoch 8, update 9.2%, dev 30.6%
epoch 9, update 8.4%, dev 31.8%
epoch 10, update 7.0%, dev 31.4%
best dev err 30.1%, |w|=16743, time: 0.6 secs
```

As we can see, the best dev rate was down to 28.9%, which improved the results!

```
epoch 1, update 39.0%, dev 39.6%
epoch 2, update 25.5%, dev 34.1%
epoch 3, update 20.8%, dev 35.3%
epoch 4, update 17.2%, dev 35.5%
epoch 5, update 14.1%, dev 28.9%
epoch 6, update 12.2%, dev 32.0%
epoch 7, update 10.5%, dev 32.0%
epoch 8, update 9.7%, dev 31.5%
epoch 9, update 7.8%, dev 30.2%
epoch 10, update 6.9%, dev 29.8%
best dev err 28.9%, |w|=16744, time: 0.6 secs
```

Q4

Even though the dataset is balanced, there are a few reasons that we should add a bias dimension. Adding bias may let the model fit the training dataset. In real-world scenarios, there are noises of complex sentences, and adding noise can help to model these complexities.

2 Average Perceptron

Q1

Yes, it improved the dev error rate. The dev error rate was down to 26.3%. And it made the dev error rate more stable.

```
epoch 1, updates 0.4%, dev 31.4%
epoch 2, updates 0.6%, dev 27.7%
epoch 3, updates 0.9%, dev 27.2%
epoch 4, updates 1.0%, dev 27.6%
epoch 5, updates 1.2%, dev 27.2%
epoch 6, updates 1.3%, dev 26.7%
epoch 7, updates 1.4%, dev 26.3%
epoch 8, updates 1.5%, dev 26.4%
epoch 9, updates 1.6%, dev 26.3%
epoch 10, updates 1.6%, dev 26.3%
best dev err 26.3%, |w|=15806, time: 79.4 secs
```

Q2

Yes, the total time increased by about 79 seconds.

Q3

The results are:

Top positive features:

engrossing
rare
French
unexpected
cinema
provides
wonderful
pulls
triumph
treat
skin
powerful
culture
refreshingly
beautiful
delightful
open
dots
still
affection

Top negative features:

boring
dull
generic
too
fails
routine
instead
badly

bad
ill
tv
problem
worst
flat
unless
guy
attempts
neither
worse
suffers

Most of them are the common positive and negative words that we expected in the movie review. However, there are words like 'french', 'provides', 'tv', 'dots', 'too', which don't have sentiment meaning. They might often appear with other positive/negative words, so that the model misunderstood them.

Q4

Negative examples that are strongly predicted as positive:

Review: ` in this **poor remake** of such a well loved classic , parker exposes the **limitations of his skill** and the **basic flaws**

Review: mr wollter and ms seldhal give strong and convincing performances , but **neither reaches into the deepest recesses** of o unearh the quaking essence of passion , grief and fear

Review: bravo reveals the true intent of her film by carefully selecting interview subjects who will construct a portrait of minantly charitable it **can only be seen as propaqanda**

Review: how much you are moved by the emotional tumult of fran ois and mich le 's relationship **depends a lot on how** interesti you find them

Review: for all of its insights into the dream world of teen life , and its electronic expression through cyber culture , the **quarter to anyone seeking** to pull a cohesive story out of its 2 1 2 hour running time

Those part(orange) are not directly negative words but represents the negative sentiment. Also, there are words like 'love', 'bravo', 'carefully selecting' are positive words that showed in the negative context, which may lead to misclassification.

Positive examples that are strongly predicted as negative:

Review: the thing about guys like evans is this you 're never quite sure where self promotion ends and the truth begins but as you watch the movie , you 're too interested to care

Review: neither the funniest film that eddie murphy nor robert de niro has ever made , showtime is nevertheless efficiently amusing for a go od while before it collapses into exactly the kind of buddy cop comedy it set out to lampoon , anyway

Review: even before it builds up to its insanely staged ballroom scene , in which 3000 actors appear in full regalia , it 's waltzed itself into the art film pantheon

Review: if i have to choose between gorgeous animation and a lame story (like , say , treasure planet) or so so animation and an exciting , clever story with a batch of appealing characters , i 'll take the latter every time

Review: return to never land may be another shameless attempt by disney to rake in dough from baby boomer families , but it 's not half bad

Same, there are also many negative words that appear in the positive context. For example: 'neither', 'shameless attempt', 'lame story'...

What's more, in some reviews, they use contrast to express a positive opinion. E.g. I'll take the latter every time. That may lead to misclassification as well.

In addition, the complex sentence is hard to model as well. The sentences are not straightforward.

3 Pruning the vocabulary.

Q1

After neglecting the one-count words, the dev error rate was improved. And the best error rate is 25.9%.

Q2

Yes, my model size has been shrunk. 15806->8425, the model size was nearly reduced by half. The model was simpler. It prevents overfitting.

Q3

Yes, the update % changed a little bit. The update% was slightly higher each epoch. the slight increase in update percentage doesn't seem problematic and might reflect a more robust learning process.

Q4

The training time was reduced by about 38 seconds.

Q5

In my case, it didn't improve the dev error rate.

```
epoch 1, updates 0.4%, dev 31.1%
epoch 2, updates 0.7%, dev 29.2%
epoch 3, updates 0.9%, dev 28.7%
epoch 4, updates 1.1%, dev 28.0%
epoch 5, updates 1.3%, dev 27.9%
epoch 6, updates 1.5%, dev 26.6%
epoch 7, updates 1.6%, dev 26.8%
epoch 8, updates 1.8%, dev 26.6%
epoch 9, updates 1.9%, dev 26.6%
epoch 10, updates 2.1%, dev 26.6%
best dev err 26.6%, |w|=5934, time: 28.3 secs
```

4 Try some other learning algorithms with sklearn.

Q1

I am using the MLPClassifier which is provided by sklearn. It is a multi-layer perceptron classifier. I am using the numpy array. I was turning labels into numpy array, and turned text data into numerical vectors.

Q2

The dev error was 26.00%. And the running time was 39.39 seconds.

5 Deployment

The best dev error rate is 25.9%. And I achieved this result by using average perceptron, which is the same as Q3, and I set to neglect one-count word.

6 Debriefing

1. Approximately how many hours did you spend on this assignment?
8 hours.
2. Would you rate it as easy, moderate, or difficult?
Medium to hard.
3. Did you work on it mostly alone, or mostly with other people?
I worked on it with one of my classmates. (10%)
4. How deeply do you feel you understand the material it covers (0%–100%)?
100%.
5. Any other comments?
The lecture is amazing, and I have learned a lot.