Project 2 Word Embedding for adjectives

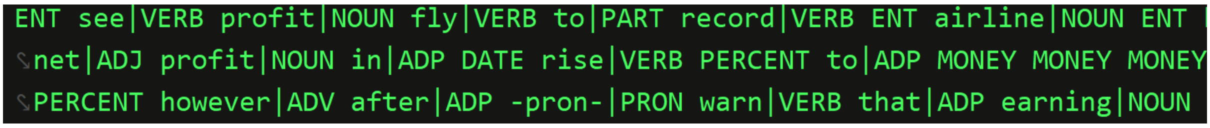
Yang Qiu, z5128684

1 Data Processing and Training Data Generation

Raw text is got from BBC_data.zip, and spaCy is used for implement linguistic features.

The original text contains a lot of entities such as person name, organization name, country name and etc. The information is redundant and can be reduced. All the entity text is replaced by the token.ent_tag_. And all the number is replaced by NUM. And the url is replaced by URL.

For other token, token is reduced to its lemma and a tag is appended at the end of token, such as best -> ‘best|ADJ’, supports -> ‘support|VERB’. The purpose is to add extra information. The punctuation is removed as well. And symbols such as ‘$’ is replaced by SYM. At last, all the upper case is reduced to lowercase. The final form is shown in Figure.1.



```
ENT see|VERB profit|NOUN fly|VERB to|PART record|VERB ENT airline|NOUN ENT
net|ADJ profit|NOUN in|ADP DATE rise|VERB PERCENT to|ADP MONEY MONEY MONEY
PERCENT however|ADV after|ADP -pron-|PRON warn|VERB that|ADP earning|NOUN
```

Figure 1.

2 Model Training

The data processed is trained to produce word ve4ctor.

Here are the parameters the final version of training model is using.

2.1 Tunable Parameters

batch_size	128	**vocabulary_size**	9000
skip_window	2	**learning_rate**	0.002
num_samples	4	**Number of Negative Samples**	200

For vocabulary size, the unique vocabulary size after preprocessing is around 17000. And the word count =1 is 5000. And word count =2 is 3000. Those are infrequency word. Therefore, those are removed from vocabulary and replaced by UNK. 9000 becomes a reasonable choice.

2.2 Fixed Parameters

Embedding_dimensions	200	**Loss function**	sampled_softmax_loss
Number of iterations	100001	**Optimization method**	AdamOptimizer

3 Training Result

After training, the adjectives vector is written into ‘adjective_embeddings.txt’. There are around 1650 adjectives in the file. Test the trained model by using genism and ground truth. There is average above 9 hits which is relatively good result for sample_softmax_loss function.