Report

1. Introduction

The purpose of the report is to illuminate the implementation of NER to find the category of title in a text file. And it will cover feature analysis and the process of improving the trainer model.

2. Feature Analysis

(1) Word's Affix

It is an obvious feature: some titles (e.g. doctor, minister) has especially affix (e.g. 'er', 'or', 'ist', 'an', 'ive', 'ant'). If only using this feature, the f1-measure value could be more than 30%, so it is a good feature.

(2) Abbreviation

To some extent, abbreviation could facilitate to decide whether a token is a title or not (e.g. 'CEO', 'Mr.', 'Dr.'). After observing the training data, the chance of an abbreviation title take place in sentence is very low, but the chance to be a title is high.

(3) Current Word's itself

In NER problem, current word always has a great power to improve the whole precision, but it may lead to overfitting.

(4) Previous Word's Tag and Next Word's Tag

These two feature will improve F1-measure's value by some extent. The reason to take this into consider is that if the current word is a title, in some scope, its surrounding word could belong to this title.

(5) Current Word's Tag

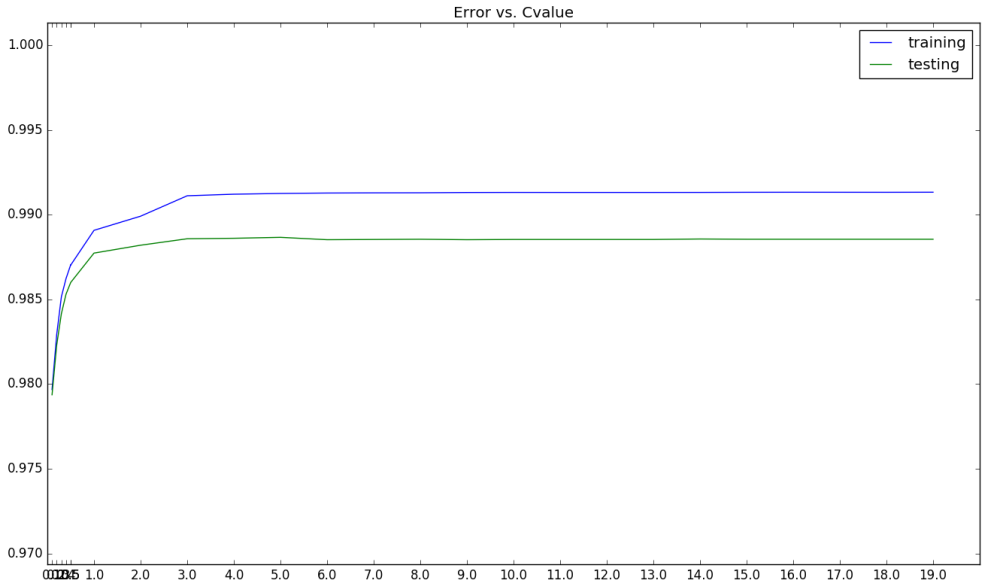
It is a very common rule, that a title's tag is highly likely to be 'NN' or 'NNS' or 'NNP' or 'NNPS'. And because it is very common, so its precision may not be high, but this feature could be bind with other features. In my implementation, if a word satisfied Word's Affix feature, it need to still check the word's tag to avoid some mistakes decision.

single feature test (test training data, set class weight to balanced):

	F1-measure value
Word's Affix	0.344253632761
Abbreviation	0.0159547092126
Current Word's itself	0.90966938719
Previous Word's tag	0.101476962265
Next Word's tag	0.135315222963

3. Improving the trainer model

(1) using k-fold cross validation to find a suitable C parameter of logistic regression (set k = 10).



As shown in the above picture, the optimize C parameter is around 3. So I set this value in trainer.

(2) binding some features together.

- binding Word's Affix with current word's Token.
- binding Abbreviation with current word's Token.