Project Report - NER tagging for Twitter - CMPSCI 585

Apoorva Rao Balevalachilu and Armand Halbert

December 10, 2014

Abstract

The task is to perform named entity recognition on tweets. We provide results in BIO notation. This report will consist of two sections: System building and Analysis.

1 Introduction

NER can be performed for the following tasks:

- Question Answering
- Textual Entailment
- Coreference Resolution
- Computational Semantics

An example of NER is given below:

"Germany's representative to the European Union's veterinary committee Werner Zwingman said on Wednesday consumers should ..."

Germany - B, European - B, Union - I, Werner - B, Zwingman - I, All other tokens - O

Our system uses Conditional Random Fields to make predictions for NER. This model is discriminative. It does not assume that features are independent. The benefit of using a CRF is that while labeling it takes future observations into account.

We have used the starter code provided by Professor Brendan O' Connor and David Belanger, CRFSuite, NLTK, and performed some experiments with the Freebase API.

We achieved an F-score of 0.23644 before the end of the competition, but improved our system significantly after that. At present, our scores are:

```
F = 0.3821, Prec = 0.5761 (462/802), Rec = 0.2859 (462/1616)
(3336 sentences, 46714 tokens, 1616 gold spans, 802 predicted spans)
```

2 System Building

We created a feature extractor that produces the following types of features.

2.1 Lexical or wordform features

1. Lowercased version of the word
2. Is in upper case?
3. Is a digit?
4. Is "retweet or RT"?
5. Is a URL?
6. Is an emoticon?
7. Is an apostrophe s?
8. Is a date?

These were checked using regular expressions in the python code in `simple_fe.py`

2.2 Character Affixes

1. Affixes consisting of first character
2. Affixes consisting of first two characters
3. Affixes consisting of first three characters
4. Suffixes consisting of last three characters
5. Suffixes consisting of last two characters
6. Suffixes consisting of last character
7. Begins with a hashtag i.e. #?
8. Is a mention i.e. begins with ?

These were extracted using simple string manipulation code in python.

2.3 Shape Features

1. Reduce uppercase characters - (r'[A-Z]+','A')
2. Reduce lowercase characters - (r'[a-z]+','a')
3. Reduce lowercase characters - (r'[0-9]+','0')
4. Reduce punctuations - (r'[Â-Za-z0-9]+','$')

We created the shape features using regular expressions in python.

2.4 Positional Offset Features

1. Previous word
2. Next word
3. Word before previous word
4. Word after next word

2.5 Major Extension - Freebase

2.6 Major Extension - POS Taggers

We use an external POS tagger to generate features.