Assignment 3: Named Entity Recognition using MALLET

Sourabh Balgi
Sr. No. - 14318
EE Department
M. Tech. (Systems Engineering)
sourabhbalgi@gmail.com

Abstract

This document contains the third assignment report with the details of the methodology, observations, results and plots for **Conditional Random Field models (CRF)**. Different feature set like Word Embeddings *Word2Vec* and POS tagging were used to create different CRF models to observe and compare accuracy, precision and recall. Important observation from this assignment is the CRF model with the sequential models like LSTM and Bi-LSTM provide better accuracies compared to just CRF models.

1 Introduction

Named Entity Recognition (NER) task takes an input sequence of tokens in a sentence and assign tags for each token or phrase in the sequence. Named entity recognition (NER) has proved to be one of the important and challenging tasks in language modeling. There are many applications of NER such as classifying regions of an image, estimating the score in a strategic game, segmenting genes in a strand of DNA, extracting syntax from natural-language text, etc. In all these applications, we wish to predict a vector of random variables given an observed feature vector. A natural way to represent the manner in which output variables depend on each other is provided by graphical models. Graphical models, which include model families such as Bayesian networks, neural networks, factor graphs, Markov random fields, and others, represent a complex distribution over many variables as a product of local factors on smaller subsets of variables.

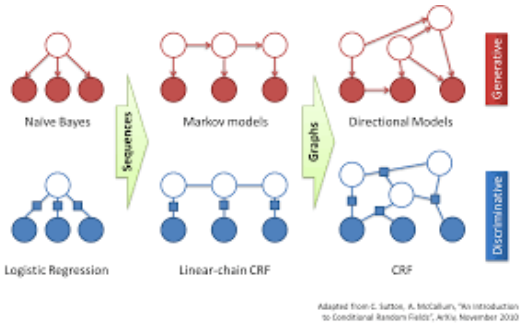

Figure 1: Generative-Discriminative Pairs (Source: Google)

2 Models

CRF using Mallet: *SimpleTagger* class of mallet is a command line interface to the *MALLET* Conditional Random Field (CRF) class. This takes in the training data, makes the CRF model which can be used to predict test data tags. *MALLET* is a Java-based package for statistical natural language processing, document classification, clustering, topic modeling, information extraction, and other machine learning applications to text.

Bi-LSTM with CRF: A sequence to sequence model which implements a Bi-LSTM layer and CRF layer using Mallet. Input is integer encoded sequences which is embedded by the embedding layer and given to the Bi-LSTM and the output of which is fed to CRF layer. The CRF layer gives a probability distribution as its output which is then used to predict the output tag.

3 Data-set Processing

3.1 Data Split

The Dataset available was split into *Train, Dev and Test* in proportions of *0.8, 0.1 and 0.1* respectively. Separate feature files were created for each model.

3.2 Feature Representations

1. Simple Token : Data split for the CRF model to be trained using only the words as the input features.

2. Word Embedding : Data split for the CRF model to be trained using only the *100-dimensional Word2Vec* representation of the words as the input features.

3. Token + Word Embedding : Data split for the CRF model to be trained using the word tokens along with their *100-dimensional Word2Vec* representation as the input features.

4. Token + POS Tag: Data split for the CRF model to be trained using the words and their corresponding *Parts of Speech (POS)* tags as the input features.

4 Evaluation Metric

$$precision = \frac{tp}{(tp + fp)}$$

$$recall = \frac{tp}{(tp + fn)}$$

where tp is the number of true positives, fp the number of false positives and fn the number of false negatives.

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}}$$

Macro Score: Calculate metrics for each label, and find their un-weighted mean. This does not take label imbalance into account.

Weighted Score: Calculate metrics for each label, and find their average, weighted by support.

5 Methodology

5.1 CRF using Mallet

The train, dev and test data splits are made as indicated in Data-set preprocessing part. Different feature representations of Train data are fed into Mallet's *SimpleTagger* class to obtain different CRF models. Each CRF model is validated using the dev data. Evaluation metric for each of the models are calculated and tabulated.

5.2 Bi-LSTM with CRF

Inputs are encoded into integer values. Maximum number of tokens in each sentence is fixed at 60 and short sentences are zero padded. (zero represents unknown token). Tag 'O' is assigned to all the zero paddings. The encoded input is split into batches of size 64 and is given as input to an embedding layer of size 200 which is followed by a Bi-LSTM layer of size 100 and dropout 0.2, the output of which is given to CRF layer. The model is trained for 5 epochs. For each of the epochs train and validation accuracies are calculated. Testing is also done the same way with padded sentences.

6 Results

6.1 CRF using Mallet

Features	F1 (Macro)	F1 (Wtd)
Simple Token	0.506	0.846
Word2Vec(100)	0.766	0.916
Token + Word2Vec(100)	**0.766**	**0.916**
Token + POS Tag	0.638	0.881

Evaluation Metric : All Feature representation

	O	D	T
O	5552	107	68
D	185	341	2
T	164	1	213

Best Confusion Matrix: Token + Word2Vec

6.2 Bi-LSTM with CRF

F1 Score (Macro) - 0.721
F1 Score (Wtd) - 0.970

	O	D	T
O	20808	91	170
D	212	266	58
T	104	8	243

Best Confusion Matrix: Token + Word2Vec

7 Observations

- Word Embedding feature representation provides the best CRF model in Mallet as they capture a lot of semantic as well as syntactic meaning which can be used for identification.
- POS tagging of the tokens didn't provide much improvement in performance.
- Increasing Word Embedding sizes in both models improved the performance.

- Sequential CRF model with Bi-LSTM outperforms the CRF model by Mallet.
- Since the Tag *O, T, D* are skewed, observing accuracy will not provide the best comparisons of the models. So the F1 measures are used for the comparison.

8 References

- MALLET : MAchine Learning for LanguagE Toolkit.
- Log-Linear Models, MEMMs, and CRFs , Michael Collins.
- An Introduction to Conditional Random Fields for Relational Learning , Charles Sutton and Andrew McCallum.