Feature Engineered Named Entity Recognition

Anish Acharya
University of Texas at Austin

1 Introduction

Named Entity Recognition(NER) is a fundamental NLP task where the objective is to identify the named entities in a piece of text. In this work, we focus on designing powerful features for NER, which leads to reasonably good accuracy without the need of a complex model. We tried our approach on CONLL-2003 NER dataset which has four class of named entities: person, organization, location, and miscellaneous. We just focus on identifying instances of the person label in isolation for this work.

2 Information Encoding

Word Encoding
(a) Word Indicator BOW style 1-hot representation in the vocab space. Extremely **sparse** nature makes learning difficult and slow.
(b) SVD Word Embedding To combat with sparsity we take a k-rank $k = 300$ SVD of the BOW representation of training set offline.
(c) Pre-trained Word Embedding Glove pre-trained embedding per word

Context Encoding
We designed two types of context features:
(a) window context average of previous , current and next word encoding with different lengths $n \in [3, 5]$
(b) one sided context Average word encoding of either prev or next n words $n \in [1, 2]$

Syntactic Information Encoding
(a) POS Indicator 1-hot sparse representation of the Parts of speech of the token.
(b) case Indicator We considered two types of casing indicators.
(a) Set if If the first char of the token is upper case and it's not the first word,
(b) Set if all the chars of the current token are uppercase.

3 Modelling Choices

For all the experiments we used a simple Feed-forward neural network with 2 hidden layers with 16 and 4 neurons with Exponential Linear Unit (ELU) activation.
Scaled Loss Function We formulated the objective as minimizing Binary Cross Entropy(BCE) loss. One particular challenging aspect for this particular task was the steep imbalance between the classes while only about 10% of the data had positive labels. In these imbalanced settings it is very easy to learn an all-zero classifier. The loss function was modified using a scaling factor $\alpha = \frac{card(y==1)}{card(y==0)}$ to up-weight the minority class.
Cyclically Annealed Learning Rate We designed a cyclically annealed LR schedule to combat this. The learning rate exponentially decays until a pre-chosen lower bound and then we hot start(anneal) the LR in the hope to jump out of local minima.

4 Results

We experimented with different combinations of features described in 2. While, compact representation of SVD based/ pretrained BOW feature improved the performance multi-fold the main performance boost comes from context features. While we noticed windowed context works better than one-sided context.

Features	Acc.	Prec	Recall	F1
WI	97.35	94.11	60.38	73.56
WI-SVD	98	95.41	70.65	81.18
WE	98.04	96.93	70.27	81.48
WE+CL1	98.39	96.07	76.84	85.39
WE+CR1	98.26	92.34	78.12	84.63
WE+CL1+CR1	98.45	95.86	78.06	86.04
WE+CW1	98.57	88.94	87.62	88.27
WE + CL1 + CR1+ PI + UI	98.45	94.45	81.07	87.25
WE + CW1 + PI + UI	98.93	91.97	90.31	91.14
WE+ CW1 +CL1 + CR1 + PI + UI	**99**	**91.18**	92.63	**91.9**

Table 1: Performance on CoNLL 2003 Shared Task on Named Entity Recognition (dev set) for different feature combinations. Legends:WI: word indicator, WE: Word Embedding, CL1: Context Left size=1 (w-1), CW1: Context window size 1 each side (w-1)(w)(w+1), PI: POS indicator, UI: Uppercase Indicator