NLP

Wiki based Named entity recognition

-By Ayushi Gupta 20162312

1. Introduction

Named-entity recognition (NER) (also known as entity identification, entity chunking and entity extraction) is a subtask of information extraction that seeks to locate and classify elements in text into pre-defined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

Full named-entity recognition is often broken down, conceptually and possibly also in implementations,as two distinct problems: detection of names, and classification of the names by the type of entity they refer to (e.g. person, organization, location and other).

NE recognises entities in text, and classifies them in some way, but it does not create templates, nor does it perform co-reference or entity linking, though these processes are often implemented alongside NE as part of a larger IE system.

NE is not just matching text strings with pre-defined lists of names. It only recognises entities which are being used as entities in a given context.

Now a day's machine learning approach is commonly used for NER because training is easy. There are various machine learning approaches for NER such as CRF (conditional Random Fields),MEMM(Maximum Entropy Markov Model), SVM(Support Vector Machine) and HMM(Hidden Markov Model) and dictionary based approach. Among all these HMM, being the most promising, has not been explored in its full potential for NER.

In this project Hidden Markov Model (HMM) based approach is used to identify the named entities. The main idea behind the use of HMM model for building NER system is that it is language independent and we can apply this system for any language domain. In addition we also explore the SVM approach and Best Tag approach to perform comparative analysis over the results obtained.

Mostly the researchers use hybrid NER system which take advantages of both rule-based and statistical approaches so that the performance of NER system can be improved.

2. Applications of NER

NE involves identification of proper names in texts, and classification into a set of predefined categories of interest.

Three universally accepted categories: person, location and organisation.

Other common tasks: recognition of date/time expressions, measures (percent, money, weight etc), email addresses etc.

Other domain-specific entities: names of drugs, medical conditions, names of ships, bibliographic references etc.

- Named entities can be indexed, linked off, etc.
- Sentiment can be attributed to companies or products
- A lot of IE relations are associations between named entities
- A lot of IE relations are associations between named entities
- For question answering, answers are often named entities.

3. DataSet

The dataset used in this project is CoNELL 2003 dataset.

Dataset is of the form :- word | Pos tag | Chunk Tag | NER tag.

The CoNLL-2003 shared task data files contain four columns separated by a single space. Each word has been put on a separate line and there is an empty line after each sentence. The first item on each line is a word, the second a part-of-speech (POS) tag, the third a syntactic chunk tag and the fourth the named entity tag. The chunk tags and the named entity tags have the format I-TYPE which means that the word is inside a phrase of type TYPE. Only if two phrases of the same type immediately follow each other, the first word of the second phrase will have tag B-TYPE to show that it starts a new phrase. A word with tag O is not part of a phrase.

Partition for training and testing is 75:25

4. Pseudo Code:

Procedure followed on the DataSet is as follows:

PART1: Training

1. Read tagged corpus
2. Tokenize the tagged corpus
3. Find all bi-grams in training dataset.
4. Find list of all words, tags, chunks and named entities in training corpus.
5. Compute $P(NE_i|NE_{i-1})$ (transition probability)
6. Compute $P(NE_i|w_i)$ (transition probability)
7. Compute $P(t_i|t_{i-1})$ (transition probability)
8. Compute $P(w_i|t_i)$ (emission probability)
9. Compute $P(ch_i|ch_{i-1})$ (probability of chunk given previous chunk)
10. Compute $P(ch_i|t_i)$ (probability if chunk given tag)
11. Find probability of every Named Entity being a starting Named Entity (Applied Add-1 smoothing while computing this field)
12. Find probability of every chunk being starting chunk (Applied Add-1 smoothing while computing this field)

Testing

1.Receive a set of testing documents

2.Run model inference to label each token

3.Appropriately output the recognized entities

EXPERIMENTAL RESULTS:

CoNELL 2003 corpus was selected to compute the efficiency of viterbi algorithm.

Total Number of training sentences : 14041

Total Number of testing sentences : 3453

Efficiency : 87.04%

S.No	**Inputs**	**Accuracy**
1.	Generalized Method: Emission Probability :**$P(W_i\|NE_i)$** Transition Probability : **$P(NE_i\|NE_{i-1})$** States : Named-Entities Observations : Words	87.0442554108 %
2.	Emission Probability :**$P(W_i,POS_i\|NE_i)$** Transition Probability : **$P(NE_i\|NE_{i-1})$** States : Named-Entities Observations : (Word, tag) tuple.	85.7908904921 %
3.	Emission Probability :**$P(POS_i\|NE_i)$** Transition Probability : **$P(NE_i\|NE_{i-1})$** States : Named-Entities Observations : Words	85.8059653279 %