Using Naive Bayes Sequence Classification for Named Entity Tagging

Muhammad Ahmed Shah
Sannan Tariq

December. 4th 2014

Introduction

Information Extraction systems are a major area of research in the field of Natural Language Processing. IE systems, as the name suggests, are designed to find and understand relevant parts of texts and/or gather information from pieces of texts.

Named Entity Recognition (NER), the focus of this report, is a subtask in the field of Information Extraction. NER systems are designed to examine pieces of text and using pre-determined features, classify the constituent words. These systems can either be used to provide binary results, i.e. tag words with a true or false tag representing whether they are named entities or not, or assign more informative tags to words that differentiate between named entities, i.e. persons, locations or organizations.

This report examines some of the previous work in the field of NER by providing an overview of current state-of-the art systems , and then analyses the effectivness of a NE tagger based on a Naive Bayes sequence classifier.

Previous Work

NER is, in the larger context, a machine learning problem and as such it has a significant statistical component to it. Several statistical models have been employed throughout history to facilitate NER, maximum entropy model (MEM) and hidden Markov model (HMM) are two of the more popular ones. At a time when the only one model was chosen and exlusively applied to the NER task, Chiong and Wei proposed a NER system that would successively apply MEM and HMM on the text (Chiong & Wei, 2006). Their method is two pronged, they conduct a "MEM walkthrough" and then use it to train HMM to make the final tagging. The MEM walkthtough uses MEM to estimate the probability of the <PER>, <LOC> and <ORG> tag being assigned to each word based on a set of features and training data. The probabilities obtained from MEM are then passed to the HMM which performs a document-wide context check for each token to update the pobabilities. This strategy resulted in significantly high F-measure scores across various genres of text.

The strategy of incorporating global information in NER systems has been explored a great deal more. Merhav, Mesquita, Barbosa, Yee and Frieder (2010) propose using domain frequencies, i.e the frequency of occurance of a term in all domains, as features in NER. They conduct an experiment to observe the relation between difference in domain frequencies in entry pairs and propensity of erroneous tags. They concluded from their experiment that the differene in DF scores for a given term serves a viable indicator of the possiblity of erroneous types assigned to entities associated with the said term.

The aforementioned research concentrates on sequence-based tagging however another paradigm in NE tagging is chuck based tagging in which a program called a chunker breaks up the text into smaller segments based on certain features, one of the more common of which are noun phrases, such that each chunk contains one or more NEs. Iwakura and Takamura and Okumura (2013) propose a NER system based on decomposition and concatenation of word chunks. They define the following operators to manipulate chunks: SHIFT for separating the first word of the chunk, POP for separating the last word of the chunk, JOIN for concatenating two word chunks and REDUCE for assigning a NE label to a chunk. By using a combination of these operators to facilitate training and classification the authors achieved gains in processing speed over linear-chain perceptron and semi-Markov perceptron, while maintaining accuracy.

Method

This training and development data used in this report consided of short texts spanning a variety of domains and genres. The text was gold standard annotated with POS, phrase and NER tags. The tests data was an unseen body of text with the same gold standard tags as the training and development data for each word.

The method being explored in this report is Naive Bayes Sequence classification. The feature set includes features for individual tokens, and lexical, phrasal and historical context. The feature sets for each category are documented below:

1. Individual Tokens and POS tags:
   (a) Token
   (b) POS tag is NNP
   (c) First letter is capitalized
   (d) All letters are upper case
   (e) Hyphenation and capitalization
   (f) Spells out a number
   (g) Apostrophy
2. Lexical context:
   (a) Previous/next word
   (b) Previous/next POS tag
   (c) Does a verb occur after the current word
   (d) Is the POS tag of the next word "WP"
3. Phrasal Context:
   (a) Phrase tag of the word itself
   (b) Phrase tag of the previous word
4. Historical context:
   (a) Previous 2 NER tags
   (b) Previous NER tag
   (c) comma

The classifier was run for each of these categories and the results were documented. First we ran the classifier with category 1 features and documented the performance, then we included category 2 features along with category 1 features and documented the performance. This procedure was repeated as category 3 and 4 were included into the feature set.

Results

Below we present the performance of classifier as we progressively include feature categories. The precision, recall and F1 measure are determined for the classifers performance when classifying LOC, PER and ORG labels respectively. We also present the classifier's performance when performing binary classification between named-entities and non-named-entities (I/O tagging). We evaluated the classifer's performance first on the develoment set and then on the test set. The results of both these evaluations are presented below.

Feature Set: Individual Tokens and POS tags

We include the capitalization of the token to detect patterns specific to named entities such as Upper case first letter. We also check if the token is all-caps since it may indicate that it is not a person's name. We check for the posessive since it may indicate the presence of a named entity. To reduce false positives we check if the token spells out a number.

Development Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.86	Precision	0.76	Precision	0.59	Precision	0.84
Recall	0.80	Recall	0.95	Recall	0.70	Recall	0.96
F1-Measure	0.83	F1-Measure	0.84	F1-Measure	0.64	F1-Measure	0.90

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.79	Precision	0.68	Precision	0.57	Precision	0.80
Recall	0.76	Recall	0.93	Recall	0.68	Recall	0.96
F1-Measure	0.77	F1-Measure	0.79	F1-Measure	0.62	F1-Measure	0.87

The classifiers performance for the ORG label remained low, compared to other labels, in both the test and development set. We hypothesise that the reason could be attributed to the relative abundance of LOC and PER labels in the test set which would overwhelm the probabilities for the ORG label.

Feature Set: Individual Tokens and POS tags, and lexical context

We check the previous and next tokens and their POS tags to establish patterns of the contextual arrangement of tokens. We also check if a verb occurs after the current token in the next 5 tokens, since we believe this to be indicative of a named entity, most likely an organization or a person.

Development Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.65	Precision	0.89	Precision	0.59	Precision	0.82
Recall	0.90	Recall	0.89	Recall	0.76	Recall	0.97
F1-Measure	0.75	F1-Measure	0.89	F1-Measure	0.66	F1-Measure	0.89

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.58	Precision	0.84	Precision	0.52	Precision	0.77
Recall	0.85	Recall	0.83	Recall	0.75	Recall	0.98
F1-Measure	0.69	F1-Measure	0.83	F1-Measure	0.61	F1-Measure	0.86

Adding lexical context to the feature set resulted in mixed results. While the performance on the LOC label suffered it improved significantly on the PER tag. The performance on the ORG tag improved in the development set but remained almost constant in the test set while the I/O tagging performance did not vary significantly.

We think that this might be because the lexical context, in many cases. is similar for all the three named entities in the test and development data.

Feature Set: Individual Tokens and POS tags, lexical and phrasal context

To determine the phrasal context we check if the token is in a noun phrase and has an NNP tag since this is the only way that it will be a named entity. We also check if the next phrase after the current one is a verb phrase, since this may indicate that the token is a person or organization or if the current phrase is preceeded by a prepositional phrase, which may be indicative of a location.

Development Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.63	Precision	0.88	Precision	0.56	Precision	0.79
Recall	0.89	Recall	0.89	Recall	0.74	Recall	0.96
F1-Measure	0.74	F1-Measure	0.88	F1-Measure	0.64	F1-Measure	0.87

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.85	Precision	0.84	Precision	0.54	Precision	0.76
Recall	0.85	Recall	0.83	Recall	0.74	Recall	0.97
F1-Measure	0.68	F1-Measure	0.83	F1-Measure	0.62	F1-Measure	0.85

The addition of phrasal context proved to be ineffective. The performance across all tags remained static or decreased insignificantly.

Feature Set: Individual Tokens and POS tags, historical, lexical and phrasal context

We check upto previous 2 NER tags because we think that if any named entity occurs in the last two tags the current token is very likely of the same type. We base our observation on patterns like, <NE> <O> <NE>, e.g. China and Japan, and <NE><NE>, e.g. Kemal Oflazer and Saudi Arabia. We also include the presence of a comma as a seperate feature since if a comma is used between 2 NEs they are most like of the same type.

Development Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.67	Precision	0.90	Precision	0.66	Precision	0.83
Recall	0.91	Recall	0.91	Recall	0.81	Recall	0.97
F1-Measure	0.77	F1-Measure	0.90	F1-Measure	0.73	F1-Measure	0.89

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.60	Precision	0.88	Precision	0.61	Precision	0.79
Recall	0.89	Recall	0.87	Recall	0.80	Recall	0.98
F1-Measure	0.72	F1-Measure	0.87	F1-Measure	0.69	F1-Measure	0.87

The inclusion of historical context led to improvement an overall imrpovement in the classifier's performance. We believe that this increase in performance results from the formation of patterns in NER tags such as if 2 NEs are separated by a comma then they are very likely to have the same NE tag. It also allows us to train the classifier to associate specific lexical and phrasal features with each tag.

Below we present some examples of the cases when our classifier produced incorrect results:

Group O I-ORG

C O I-ORG

Hiroshige I-PER I-ORG

Yanagimoto I-PER I-PER

To compare our results we used results from a NE tagger provided by nltk and the Stanford NE tagger.

NLTK tagger

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.63	Precision	0.72	Precision	0.47	Precision	0.83
Recall	0.54	Recall	0.75	Recall	0.32	Recall	0.72
F1-Measure	0.58	F1-Measure	0.74	F1-Measure	0.38	F1-Measure	0.68

Stanford tagger

Test Set

LABEL	LOC	LABEL	PER	LABEL	ORG	LABEL	ANY
Precision	0.82	Precision	0.92	Precision	0.62	Precision	0.94
Recall	0.65	Recall	0.77	Recall	0.82	Recall	0.93
F1-Measure	0.72	F1-Measure	0.84	F1-Measure	0.70	F1-Measure	0.92

Conclusion

We believe that our experiment illustrates the significance of contextual features in NER tagging. From our experiments we observe the historical context to be the most significant feature relative to the baseline individualized token features and the phrasal context to be the least significant. This is probably due to the fact that including history allows us to directly leverage contextual gold standard data to train our classifier instead of just indirectly extracting features from sentences. Our final classifier was able to perform significantly better than the native NLTK tagger in all cases which suggests that perhaps the NLTK tagger does not consider such context when tagging the tokens. Compared to the Stanford NE tagger, our performance was almost the same, with the Stanford tagger performing better in terms of precision while our our tagger performed better in most cases as far as recall was concerned.