Named Entity Recognition using Structured Prediction

1 Introduction

Named Entity Recognition (NER) is an information extraction task and has first been introduced as part of the 6th MUC (Sixth Message Understanding Conference) that was focusing on the extraction of structured information from unstructured text, such as company names from newspaper articles [2]. The most prominent types are generally known under the name of *enamex* types, which are comprised of `Person Names`, `Organisations` and `Locations`. An additional type `Miscellaneous` captures person names outside the classic *enamax* type. Apart from these there is *timex*, which covers date & time expressions and *numex* which is for monetary values & percent. In considering named entities, it is important to distinguish between mention of entities that are non - specific as time expressions, such as *In June*, referring to any possible year or *the prof* as a person name , which itself is not a specific entity, but a deictic reference , which may point back to the mention of a real *NE*, as in the following coreference chain:

(1) $Prof.\ Bateman \Rightarrow he \Rightarrow the\ prof$

Issues in Named Entity Recognition

Among the most common problems for NER is the issue of ambiguity, that is in particular *Polysemy* [2], the property of some lexical representation having more than one possible meaning, and *Metonymy*, which refers to the concept of the part-whole/whole-part relation between two expressions. *Polysemy* could become an issue for NER when the lexical representation of an item could point to two different *NE* types. This is quite frequent with `Person Names` and `Locations`, since many people are named after cities, such as *Paris* or *Georgia*. Often the context will not be disambiguating as in (2).

(2) *Paris is beautiful.*

Metonymy is frequently an issue in literary texts and news data (which is often used in NER), where two items that are in a part-whole relationship, are substituted for each other respectively. Example (3) shows an instance of *whole-part*, where *London* is supposedly substituted for the **Goverment in London**.

(3) ***London** decided to increase the 1200 military personnel involved in Olympic security.*

Methods employed for NER

For *NER*, there exist both rule-based and statisical approaches. Rule-based methods make use of the underlying rules governing languages to extract named entities. However, this approach is quite time-consuming and requires extensive work of computational linguists [2] and although the results are often high in precision, lacks considerably in recall.

In regard to statistical approaches, supervised-learning is the most common method applied in *NER*. Although, there are unsupervised approaches, their performance is not as high as for SL applications.

Prominent algorithms in NER include maximum entropy

2 Structured Prediction

Structured Prediction [1] is a supervised-learning approach and sets itself apart from Non-structured Prediction through the form of its output. Prediction maps an input x to an output y: $x \to y$. Non-structured output is atomic, thus it is binary prediction for a two-class problem and may corresponds to more than one of more than 2 possible labels for a multiclass problem. The output of Structured Prediction is a structured and gives back a sequence/tree.

3 Our structured Perceptron

Our NER structured perceptron for the languages English, German, Dutch and Spanish is trained and tested on the *CoNLL 2003* and *CoNLL 2002* data sets respectively.

Learning Labels of the whole sentence (0) for zero entity

Structure Structured Perceptron with Averaging

Decoding Viterbi algorithm (Markov assumption, only 1 prev. label)

Features

The features employed in the system can be divided into three categories: node, label and gazetteer features. We describe each of the three groups in the following.

Node features These are only present on the word in question: the *Token*, suffix and prefix, capitalisation.

Table 1 shows the various node features with an example respectively.

Label Interaction Features These features register which label has been assigned to the previous token and takes into account the most likely sequence.

Feature	Example
Token	
Suffix	Amster**dam**
Prefix	**San** Sebastian
Captitalized	**B**enetton
Number Pattern	
UPPERCASE	**BENETTON**
POS-tag	Benetton **NNP**
Lemma	

Table 1: Node Features

Gazetteer Features In order to create gazetteer lists for the more common named entities, we designed a *SPARQL* query, that would retrieve entries from *DBPedia* for all languages. The reliability of the respective list is learnt be the perceptron.

4 Experiments/ Evaluation

In the following section we present our experiments and the evalution of our system.

Experiments

Evaluation

For the evaluation of the system we used *Precision* and *Recall* as shown in (4) and (5) respectively. The general formula for the *F-Score* is shown in (6). Since we rate both *Precision* and *Recall* evenly, we use the harmonic mean as shown in (7).

(4) $$Precision = \frac{gold\ tag \bigcap predicted}{predicted}$$

(5) $$Recall = \frac{gold\ tag \bigcap predicted}{gold\ tag}$$

(6) $$F_\beta = (1 + \beta^2) * \frac{precision * recall}{\beta^2 * precision + recall}$$

(7) $$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

5 Discussion of Results

Some Challenges

6 Future Work & Conclusion

References

[1] Xavier Carreras. Learning Structured Predictors. Lecture at Lisbon Machine Learning School 2012, http://lxmls.it.pt/strlearn.pdf, 2012.

[2] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1):3–26, January 2007. Publisher: John Benjamins Publishing Company.