Named Entity Recognition using Structured Prediction

Joachim Daiber
University of Groningen
jodaiber@gmx.de

Carmen Klaussner
University of Groningen
Carmen@wordsmith.de

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 (Nadeau and Sekine, 2007). 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. Table 1 gives an overview of the different NE types.

Named Entity Type	**Example**
Person Names	Greta Garbo, John, Mary
Organisations	Benetton, Coca-Cola Gmbh
Locations	Paris, Australia, Bristol
Miscellaneous	World Cup
Date Expression	01-02-2012, January, 6th, 1998
Time Expression	10 p.m.
Monetary Value	$15, 100 Euro
Percent	30%

Table 1: NE Examples

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* (Nadeau and Sekine, 2007), 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 high maintenance, quite time-consuming and requires extensive work of computational linguists (Nadeau and Sekine, 2007) 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 yet.

Prominent algorithms in NER include maximum entropy

CoNLL data set & baseline

The training and test dataset was taken from the *CoNLL Shared Task 2002* (Tjong Kim Sang and De Meulder, 2003) and the *CoNLL Shared Task 2003* (**?**) for Spanish & Dutch and English & German respectively.

The baseline for *Named Entity Recognition* in the four different languages is visualised in table

Language	Precision	Recall	F_1-measure
Spanish	26.27 %	56.48 %	35.86 %
Dutch	64.38 %	45.19 %	53.10 %
English	71.91%	50.90 %	59.61 ± 1.2 %
German	31.86 %	28.89 %	30.30 ± %1.3

Table 2: NER Baseline

2 Structured Prediction

Structured Prediction (Carreras, 2012) 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:
$x \rightarrow y$.

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 structured and gives back a sequence/tree. Thus, the output could be the complete sentence with its corresponding labels.

(4) *The motor company 'Ford' was founded and incorporated by Henry Ford on June, 16th 1903.*

would trigger the labels:

(5) x The motor company 'Ford' was founded and incorporated by Henry Ford on June, 16th 1903. Furthermore, Structured Prediction is different from similar approaches such as HMM through the combination of features and the use of label interactions.

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 3 shows the various node features with an example respectively.

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

Table 3: Node Features

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

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 (6) and (7) respectively. The general formula for the *F-Score* is shown in (8). Since we rate both *Precision* and *Recall* evenly, we use the harmonic mean as shown in (9).

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

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

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

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

Language	**testA**			**testB**		
	Precision	Recall	F_1-measure	Precision	Recall	F_1-measure
Spanish						
Dutch						
English						
German						

Table 4: NER Structured Prediction Results

5 Discussion of Results

Some Challenges

6 Future Work & Conclusion

References

[Carreras2012] Xavier Carreras. 2012. Learning structured predictors. Lecture at Lisbon Machine Learning School 2012, `http://lxmls.it.pt/strlearn.pdf`.

[Nadeau and Sekine2007] D. Nadeau and S. Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

[Tjong Kim Sang and De Meulder2003] Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.