Named Entity Recognition

Toine Bogers
Text Mining
March 8th, 2005

Overview

- introduction
- approaches
  - hand-crafted
  - machine learning
    - feature extraction
    - feature selection
  - hybrid
- my approach
  - features
  - experiments
  - results

Introduction

Named Entity Recognition (NER) is a combination of concept chunking and labeling those chunks: we wish to identify textual information units that represent people, places, organizations, companies, bands, etc.

De door het Amerikaanse National Hurricane Center als 'zeer gevaarlijk' omschreven orkaan Ivan nadert Cuba. Een overzicht over wat Ivan op de Kaaimaneilanden heeft aangericht, is er nog niet. Gouverneur Bruce Dinwiddy zei maandag dat duizenden mensen dakloos zijn geworden en dat ook belangrijke regeringsgebouwen zijn getroffen.

Why NER?

NER has many applications

- prerequisite for information extraction
- improving information retrieval
  - indexing
  - querying

Intuitively simple?

What's the problem? NER seems intuitively simple for humans. How do we determine whether or not a (string of) word(s) represents a name?

- does the word start with a capital letter? (orthographic characteristics)
- have we seen it before? (lists of names)
- contextual clues

How do we teach this to a computer?

Some problems…

Problems:

- not every word that starts with a capital letter is a name
  ex: *"Soms is dat niet mogelijk ..."*
- no list can ever be complete
  ex: *"Antbeard en zijn bemanning voeren ..."*
  ex: *"Wil je wat te drinken?"*
- context can be misleading
  ex: *"Er was geen land met Henk te bezeilen."*

Two routes in NER

The same presence of deductive and inductive routes in language technology in general is also present in NER:

- deductive: heuristics & handcrafted rules
- inductive: machine learning
- hybrid

Deductive route

Handcrafted NER systems are dependent on the human intuition. Their designers craft a large number of rules that represent human NER intuition and/or rules that are clever tricks and shortcuts to NER.

vb: ***ALS PATROON "titel woord_X"***
DAN woord_X = PERSOON
"Hij had nog nooit van dr. Jansen gehoord."

vb: ***ALS PATROON "provincie woord_X"***
DAN woord_X = LOCATIE
"… in de provincie Tilburg. Dit is …"

Inductive route

Bottom-up searching for patterns and relationships in the data that can be modeled.

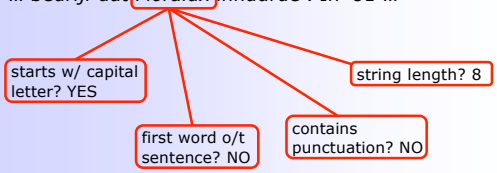
Different stages:

- feature extraction
- feature selection
- algorithm selection
- labeling decisions

Feature extraction

A lot of different features can be extracted for use in (inductively) learning to classify NEs. Every word can be represented with a lot of different features:

"… bedrijf dat Floralux inhuurde . In '81 …"



Feature extraction (2)

We represent the context by sliding a 'window' over the data which is anchored in the focus word.

focus word

"… het bedrijf dat Floralux inhuurde . In '81 bestond…"

left context | right context

"… het bedrijf dat Floralux inhuurde . In '81 bestond…"

"… het bedrijf dat Floralux inhuurde . In '81 bestond…"

"… het bedrijf dat Floralux inhuurde . In '81 bestond…"

Feature extraction (3)

Example instances:

```
Dat  is  verder  opgelaaid  door
Pron  V  Adj  N  Prep
firstCap_YES  firstCap_NO  firstCap_NO  firstCap_NO
firstCap_NO
[...]
isPunctuation_NO  isPunctuation_NO  isPunctuation_NO
isPunctuation_NO
[...]
isURL_NO  isURL_NO  isURL_NO  isURL_NO  isURL_NO
3  2  6  9  4
Dat  is  verd  opge  door
Dat  is  rder  aaid  door
functionWord_YES  functionWord_NO  functionWord_NO
functionWord_NO  functionWord_YES

O
```

Feature selection

Given a set of potential useful features, which subset of that 'produces' the best results on unseen data based on a specific machine learning algorithm?

in other words

What features have the greatest predictive value for a certain problem?

Advantages:

- faster and more efficient extraction/ classification
- better classification results
- better insight into the problem

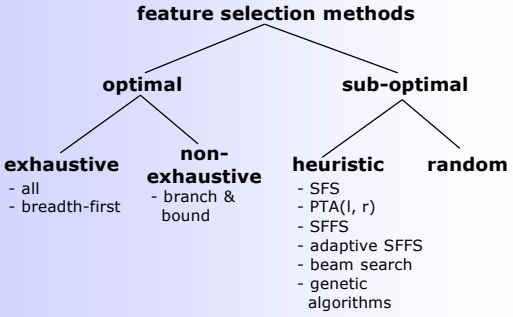
Feature selection (2)

For N features the number of different subsets is 2^N → exhaustive search is impractical

Key elements of a feature selection algorithm:

1. generation/search procedure
   - add or remove features
   - empty starting set, all features, or a random set
2. evaluation function
   - filters
   - wrappers
3. stopping criterion
4. validation procedure

Feature selection (3)



Feature selection (4)

I used the Sequential Forward Floating Selection (SFFS) algorithm for my experiments.

1. see if adding one of the remaining features leads to an improvement in generalization performance
2. add the best of the remaining features to the current set **S**
3. see if leaving out of the features in **S** leads to an improvement
4. remove that 'bad' feature from **S**
5. repeat steps 3 and 4 until they yield no more improvements; then return to step 1

Algorithm selection

- supervised learning
  - eager learners
    - probabilistic (HMMs, maximum entropy)
    - decision trees
    - transformation based learning
    - support vector machines
  - lazy learners
    - memory-based learning
- unsupervised learning (mostly combined with supervised)
- combining classifiers
  - stacking
  - bagging
  - boosting

Hybrid approach

State of the art system by (Mikheev et al., 1999) called *sequence strategy* combines the deductive and the inductive route.

- internal (phrasal) evidence
- external (contextual) evidence ("one sense per discourse", Gale et al., 1992)

Hybrid approach (2)

Five different steps:

1. sure-fire rules
2. first partial matching
3. rule relaxation
4. second partial matching
5. title assignment

State of the art

STATE OF THE ART	F-score
English	~93%
Dutch	~77%
German	~72%
Spanish	~81%
Human performance	96-98%

Lots of other different languages have been targeted as well: Chinese, French, Japanese, Portuguese, Greek, Hindi, Rumanian, Turkish, Norwegian, and so on…

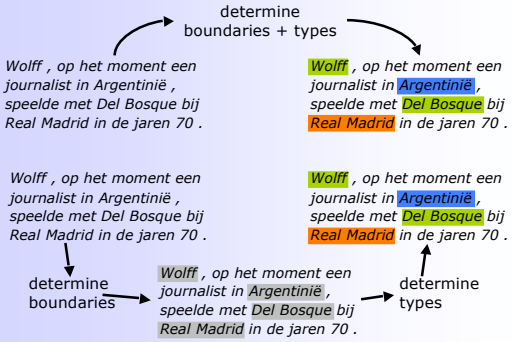
Problem

The sub-optimal feature set and the parameters of the classification algorithm are dependent on each other. Combining these options leads to a very complex and time-consuming search process.

MaxEnt has relatively few parameters, so its combined search is relatively short. Unfortunately, the kNN algorithm has many different parameters so searching this space simply is not practical.

Assumption: the feature set that was sub-optimal for MaxEnt is sub-optimal for kNN as well.

To split or not to split?



My approach

My approach:

- 2 different ML algorithms (MBL (lazy) en maximum entropy (eager))
- pool of potentially useful features
- maximize performance for Dutch

Research questions:

1. What is the best combination of features, algorithm and parameter settings?
2. Should NER take place in 1 or in 2 steps?

A look at the data

Several conferences with a shared task about NER

- MUC (English)
- CoNLL (English, Dutch, German, Spanish)
- MET (Japanese)

2002 dataset contains tokenized text, POS tags and NE tags:

```
-DOCSTART- -DOCSTART- O
Dat Pron O
is V O
in Prep O
Italië N B-LOC
of Conj O
Engeland N B-LOC
misschien Adv O
geen Pron O
probleem N O
. Punc O
```

Some statistics:

# docs	480
# instances	309686
# NEs	19901

Feature extraction

Twenty different features were extracted for each word in the 2-1-2 window
→ total of (20 × 5) + 1 = 101 features

- orthographic (`firstCap`, `allCaps`, `internalCaps`, `allLowercase`, `containsDigit`, `containsDigitAndAlpha`, `onlyDigits`, `isPunctuation`, `containsPunctuation`, `isHyphenated`)
- word type (`firstSentenceWord`, `isInitial`, `quotedText`, `isURL`, `functionWord`)
- statistical (`wordLength`)
- morphological (`prefix`, `suffix`)

Experiments

Some characteristics:

- 2 different algorithms
- 3 + 1 kinds of 'problems'
  - ♦ determine boundaries & types in 1 step
  - ♦ determine boundaries separately
  - ♦ determine types separately
    - based on predicted boundaries
    - based on perfect boundaries (ceiling performance)
- all 100 features together
- 2 × (3 + 1 + 1) = 10 experiments

Experiments (2)

- add 4 seedlist features → 10 experiments
- 2nd stage stacking for the best 2 approaches
- error analysis → error-correcting rules and apply to the best 2 approaches

Results (feature selection)

	One-shot	NEChunk	CatImperf	CatPerf
allLowercase_L2		7		
allLowercase_L1	4			
containsPunctuation_FOCU	5			2
firstCap_L1		2		
firstCap_FOCUS	1	1		
firstCap_R1		4		
prefix_L2	8		5	
prefix_L1			4	
prefix_FOCUS	3	3	2	3
prefix_R1				4
suffix_L1	6	5		
suffix_FOCUS	2	6	1	1
suffix_R1	7			
wordLength_FOCUS			3	

Results (basic)

BASIC EXPERIMENTS	TiMBL	MaxEnt
one-step, selected features	60.03	57.40
one-step, all features	70.58	56.42
two-step, only chunking	88.58	82.86
two-step, imperfect chunking	57.44	61.17
two-step, perfect chunking	70.07	67.16

Seedlists

Presence of a word in a list was added as a binary feature.

"… bedrijf dat Floralux inhuurde . In …"

in LOC list?	NO	NO	NO	NO	NO	NO
in ORG list?	YES	NO	YES	NO	NO	NO
in PER list?	NO	NO	NO	NO	NO	NO
in MISC list?	NO	NO	YES	NO	NO	NO

List features were added in a separate round to better measure their influence.

Intermezzo: seedlists

Can be extracted from the web:

Class	Content	Number of names
PERSON	international male first names	1,219
PERSON	international female first names	4,275
PERSON	Dutch first names	5,177
PERSON	international last names	31,821
PERSON	Dutch last names	806
PERSON	total	40,618
LOCATION	Dutch cities/villages	4,819
LOCATION	English country names	163
LOCATION	Dutch country names	231
LOCATION	total	5,135
ORGANIZATION	Dutch non-profit organizations	570
ORGANIZATION	USA companies	1,292
ORGANIZATION	Dutch companies	4,845
ORGANIZATION	Dutch media	783
ORGANIZATION	total	7,312
ALL	total	53,065

Intermezzo: seedlists (2)

Automatically tag names in big corpus:

PERSON	717,512
LOCATION	494,683
ADJECTIVAL	210,581
ORGANIZATION	91,163

Find n-gram patterns with a decision tree:

een [FOCUS] vrachtwagen	a [FOCUS] truck	ADJECTIVAL
burgemeester van [FOCUS] ,	mayor of [FOCUS] ,	LOCATION
PvdA en [FOCUS] .	Socialist party and [FOCUS] .	ORGANISATION
staatssecretaris [FOCUS]	state secretary [FOCUS]	PERSON

Intermezzo: seedlists (3)

Automatically tag names in big corpus:

PERSON	717,512
LOCATION	494,683
ADJECTIVAL	210,581
ORGANIZATION	91,163

Find *n*-gram patterns with a decision tree:

een [FOCUS] vrachtwagen	a [FOCUS] truck	ADJECTIVAL
burgemeester van [FOCUS] ,	mayor of [FOCUS] ,	LOCATION
PvdA en [FOCUS] .	Socialist party and [FOCUS] .	ORGANISATION
staatssecretaris [FOCUS]	state secretary [FOCUS]	PERSON

Intermezzo: seedlists (4)

- Automatic bootstrapping of seedlists using large seed (Buchholz & Van den Bosch, 2000)
- Automatic bootstrapping using just a handful of very well-known names of all types (Bill Gates, McDonald's, Pittsburgh) (Cucerzan & Yarowsky, 1999)
- Problem: overlap between lists (Washington)

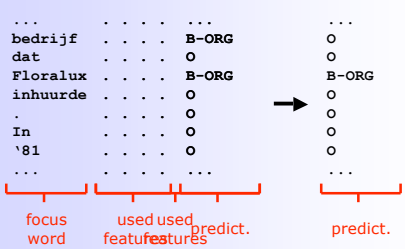
Results (seedlists)

BASIC EXPERIMENTS	TiMBL	MaxEnt
one-step, selected features	60.03	57.40
one-step, all features	70.58	56.42
two-step, only chunking	88.58	82.86
two-step, imperfect chunking	57.44	61.17
two-step, perfect chunking	70.07	67.16

WITH SEEDLIST FEATURES	TiMBL	MaxEnt
one-step, selected features	64.21	61.81
one-step, all features	69.35	58.20
two-step, only chunking	86.26	83.92
two-step, imperfect chunking	68.52	60.19
two-step, perfect chunking	70.13	65.36

Second stage stacking

Use the (best) predictions of the previous round(s) as information for the next classifier.



Results (up to stacking)

dataset/problem	F-score	algorithm
chunking & labeling in 1 step with suboptimal features	64.21	kNN
chunking & labeling in 1 step with all features	70.58	kNN
separate chunking	88.58	kNN
labeling based on the predicted chunks	68.52	kNN
labeling based on the perfect chunks	71.65	kNN

Post-processing

Error analysis → two kinds of rules:

- 'impossible' predictions; pattern such as "**B-LOC I-PER I-LOC**" are illegal
  ex: ***IF PATTERN "B-X I-Y I-X"***
  THEN NEW PATTERN "B-X I-X I-X"
- contextual clues; if a **B-LOC** is often marked as **B-ORG**, it can be corrected by looking at the context
  ex: ***IF PATTERN "deelstaat B-ORG"***
  THEN NEW PATTERN "deelstaat B-LOC"

Results (post-processing)

Improvement after applying the rules:

70.58 → 71.90 (all features; 1 step)

71.65 → 76.30 (perfect boundaries; 2 steps)

Conclusions

- kNN better than MaxEnt
- useful features
  - ♦ morphological (prefix, suffix) → types & chunks
  - ♦ orthographic (first letter capitalized) → chunks
  - ♦ seedlists → types, NOT chunks
  - ♦ stacking
- feature selection lead to better results for MaxEnt
- assumption unjustified: the MaxEnt feature set is not sub-optimal for kNN

Conclusions (2)

- combining machine learning en handcrafted rules is successful
- NER in 2 steps seems to be better than 1 step
  - ♦ upper limit is higher (76.30 > 71.90)
  - ♦ 2 step approach usually performed better
  - ♦ however: best approach was 1 step with kNN
  - ♦ but: the lack of feature selection for kNN is probably responsible

Questions?

References & links

http://ilk.uvt.nl/~tbogers/research/ner.html