Automatic Generation of Fine-Grained Named Entity Classes

Aspasia Beneti, Wael Hammoumi, Eric Hielscher,
Martin Mueller, David Persons
LTP 2006
conducted by: Erik Tjong Kim Sang

03 February 2006

Fine-Grained Named Entity Classification

A named entity recognizer (NER) identifies named entities in texts and assigns broad classes to them:

U.N./ORG official Kouznetsov/PER arrested in New York/LOC.

We expand an NER by generating and assigning additional and more specific types:

- U.N./ORG: International_organization, United_Nations
- Kouznetsov/PER: Russian, diplomat, U.N._official
- New York/LOC: State_of_the_United_States

Motivation

We want to expand the standard named entity classes (person, organization and location) because we need more specific classes in automatic question answering (QA):

1. The submarine Kursk was part of which Russian fleet?
2. What country did the winner of the Miss Universe 2000 competition represent?
3. What was the nickname for the 1998 French soccer team?

Question source: TREC-QA competition 2005.

Motivation

We want to expand the standard named entity classes (person, organization and location) because we need more specific classes in automatic question answering (QA):

1. The submarine Kursk was part of which Russian fleet?
2. What country did the winner of the Miss Universe 2000 competition represent?
3. What was the nickname for the 1998 French soccer team?

Question source: TREC-QA competition 2005.

Entity Classes to Look For

PERSON: occupation, nationality, title . . .

LOCATION: type (city, state, continent, street, museum . . .)

ORGANIZATION: location, type (party, company, sports team . . .)

OTHER TYPES: e-mail address, date, time, url . . .

Three Approaches for Extraction of Fine-Grained Categories

1. Rule approach
2. Encyclopedia approach (using Wikipedia)
3. Machine learning approach

Three Approaches for Extraction of Fine-Grained Categories

1. Rule approach
2. Encyclopedia approach (using Wikipedia)
3. Machine learning approach

Uniform Input/Output Format:

- Input: XML output generated by open source NER *LingPipe*
- Output: stand-off XML annotation

Rule Approach

(David Persons, Martin Mueller)

Goals

The goal of this module is to construct a certain number of rules which are able to tag certain words.
A simple example:

President <Person> Bush < /Person>

$\Longrightarrow$ Bush is a president

Methods

- Regular expressions to match certain structures.
- Use of lists of for example occupations to identify certain words.

Regular expressions

LingPipe does not tag words with simple structures such as e-mail addresses and dates:

- d.persons@planet.nl: \S+@\S+.\S{2,3}
- 2006-02-03: \d{2,4}[/.-]\d{1,2}[/.-]\d{1,2}

Lists

We use lists to classify certain words, for example in the sentence:

President Bush of the USA

- President is found in occupation list.
- USA is found in country list

$\Longrightarrow$ Bush is a president
$\Longrightarrow$ Bush is from the USA

The location tag

If LingPipe tags a words as a location:

- If found in country list it is a country.
- Else it is a town.

Results

- Precision: 72
- Recall: 20

Encyclopedia approach (using Wikipedia)

(Wael Hammoumi, Eric Hielscher)

Wikipedia Categories Approach

- Wikipedia groups articles into categories
  - Categories appear at the bottom of each article page
- Idea is to extract these and mark up name entities with them
- Must filter out uninteresting or Wikipedia-internal categories
- Use a scoring system with configurable thresholds
  - Features of categories assigned hand-tweaked weights
  - Categories then ranked based on total score

Goals of Wikipedia Approach

- Mark up name entities with as many relevant categories as possible
- Minimize appearance of useless categories

System Performance

- Two Behavior Parameters
  - Maximum number of categories to return
  - Minimum score a kept category can receive
- Two Performance Metrics
  - First metric requires score to be positive and limits total categories to 5 (maximizes precision)
  - Second accepts all categories returned (maximizes recall score)

Results

- A common test text was used by all three systems
- With thresholds imposed:
  - 54% (35/65) recall score
  - 57% (35/61) precision
- With all categories accepted:
  - 58% (38/65) recall score
  - 43% (38/89) precision

Example

- Categories retrived for name entity “UN”:

Category	Score
Limited geographic scope	0.20
Nobel Peace Prize winners	0.18
United Nations	0.15
Wikipedia Style Guidelines	0.00
1945 establishments	-0.45
Law-related articles lacking sources	-9.80

Machine Learning approach

(Aspasia Beneti)

Learning Module

- Method: TiMBL learner
- It implements several memory based algorithms and distance metrics
- Domain: word entities marked as LOCATION

Classes

1. CONTINENT
2. COUNTRY
3. CITY
4. CAPITAL
5. GENERAL

Features

1. Any class word in the name entity?
2. Word ending
3. Is it an abbreviation?
4. The previous word
5. The word before the previous
6. The following word
7. Any class word in the previous/next 4 words?

An example

Text: The UK, France and Germany are holding urgent talks with Iran on the stand-off over its nuclear program.

Classes: country, city, continent

Word Entity: France

Features: ??? nce — UK The and ??? COUNTRY

The learning approach...

- At the moment there are 333 training instances
- Recall: 30% Precision: 25%
- It can be very accurate once there are enough training data BUT
- it highly depends on the number and the nature of the data SO
- it requires a lot of human effort

Future work

- Rule based:
  - Come up with more regular expressions
  - Enrich the list of subcategories
- Wikipedia:
  - Intelligent access to categories in disambiguation pages
  - Methods for relevant category filtering
- Text Learning:
  - Expand domain
  - Collect more training data

Conclusion

- System that combines all modules is fully operational
- Each module solves part of the NE classification

On-line Demo:

`http://ilps.science.uva.nl/~erikt/cgi-bin/ltp2006.cgi`