

## Natuurlijke Taalverwerking II

Erik Tjong Kim Sang  
e.f.tjong.kim.sang(at)rug.nl  
18 May 2009

NLP11 2009

18 May 2009

### Automata theory

- finite state automata (deterministic and nondeterministic)
- regular languages and regular expressions
- finite state transducers
- extension: the replace operator
- weighted automata

Informatiekunde

1

NLP11 2009

18 May 2009

### Applications of automata

- morphological processing
- part-of-speech tagging
- information extraction

Informatiekunde

2

## INFORMATION EXTRACTION

NLP11 2009

18 May 2009

### NLP strives towards language understanding

Our goal is to build a system that understands human language.

This is hard to evaluate. But we have three subtasks we can use as tests:

- Question Answering
- Summarization
- Machine Translation

Informatiekunde

4

NLP11 2009

18 May 2009

### Question Answering

Question Answering involves finding answers on questions in text.

Examples questions from the CLEF 2008 competition

35. Wat is een relikwie?
36. Aan welke universiteit promoveerde Linnaeus in 1735?
37. Wanneer is het Moederdag in België?

Informatiekunde

5

NLP11 2009

18 May 2009

### Finding answers in text: Relikwie



#### Relikwie

**Relikwieën** of **reliken** zijn overblijfselen (Latijn: reliquiae), in het bijzonder overblijfselen die voorwerp van religieuze verering zijn.

Met name in het **hindoelisme**, het **boeddisme** en het **rooms-Katholieke** en **orthodoxe christendom** spelen relikwieën een belangrijke rol. Binnen sommige



Informatiekunde

6

NLP11 2009

18 May 2009

### Finding answers in text: Carolus Linnaeus

zeer **dominee**, wilde dat Linnaeus **theologie** ging studeren. Linnaeus had hierin echter weinig interesse, en uiteindelijk wist een leraar zijn vader te overreden erin toe te stemmen dat Linnaeus in plaats daarvan **geneeskunde** ging studeren.

Tijdens zijn studie verwierf hij een opdracht om de natuurlijke schatten van Lapland te inventariseren. Na zijn onderzoekreis in 1732 door Lapland schreef hij zijn *Flora lapponica*. In 1735 vertrok hij naar Nederland om te promoveren. Op 23 juni promoveerde hij in de geneeskunde op het proefschrift 'Hypothesis nova de februm intermittentium causa' aan de **Universiteit van Harderwijk** (in zes dagen, waarvan drie voor het drukken van het proefschrift). Tijdens zijn jaren in Harderwijk raakte hij bevriend met David de Gorter.

Vervolgens publiceerde hij zijn *Systema Naturae* (1735 in Leiden) waarin

Informatiekunde

7

## Finding answers in text: Moederdag

	Anguilla, Aruba, Australië, Bahama's, Bangladesh, Barbados, België, Belize, Bermuda, Bonaire, Brazilië, Brunel, Canada, Chili, Colombia, Cuba, Curaçao, Cyprus, Denemarken, Duitsland, Ecuador, Estland, Filipijnen, Finland, Ghana, Grenada, Griekenland, Honduras, Hongkong, IJsland, India, Italië, Jamaica, Japan, Kroatië, Letland, Maleisië, Malta, Nederland, Nieuw-Zeeland, Oostenrijk, Pakistan, Peru, Puerto Rico, Singapore, Slowakije, Saint Lucia, Suriname, Taiwan, Trinidad en Tobago, Tjechië,
Tweede zondag in mei	

## How can we find relevant answers?

First, we need to find documents with information relevant to the question (information retrieval).

Next we need to find candidate answers in the texts (information extraction).

The main challenges are dealing with different ways of stating information in the question and in the documents, and combining information present in different locations in the documents.

## Subtasks of information extraction

1. syntactic analysis: part-of-speech tagging and parsing
2. named entity recognition: finding names
3. co-reference resolution: linking references like *he* to names
4. relation detection: finding relations between names
5. event detection: identifying events involving names
6. temporal analysis: discovering times associated with events
7. template filling: finding attributes of predefined events

## Named entity recognition

Named entities are phrases that contain the names of persons, organizations, locations, times and quantities. Example:

**U.N.**<sub>NAME</sub> official **Ekeus**<sub>NAME</sub> heads for **Baghdad**<sub>NAME</sub>.

We are also interested in the types of the named entities

**U.N.**<sub>ORG</sub> official **Ekeus**<sub>PER</sub> heads for **Baghdad**<sub>LOC</sub>.

Some examples of named entity types: person, organization, location, miscellaneous, time and number.

## Issues in named entity recognition

How do we deal with entities of multiple words?

→ IOB tags like in:

In **New**<sub>B-LOC</sub> **York**<sub>-LOC</sub> wees **Balkenende**<sub>B-PER</sub> **Bos**<sub>B-PER</sub> terecht.

Is there anything else that makes named entity recognition hard?

→ Ambiguity! *Washington* can be person, organization and location

What do we need for building named entity recognizers?

→ Machine learners and training data

How well do named entity recognizers perform?

→ 95% correct for English; 80% correct for Dutch

## Extraction patterns

We perform relation and event extraction with extraction patterns: predefined phrases which contain the information which we are looking for.

Examples:

- X promoveert in Y
- X in Y promoveerde

## How well do these patterns work?

Google: *Linnaeus promoveert in \**

- Harderwijk (7x)
- 1735 (1x)

Google: *Linnaeus in \* promoveerde*

- Harderwijk (1x)
- 1735 (3x)

## Observations about the extraction patterns

The patterns do not generate perfect results. We still need to filter the output (look for locations).

These *lexical* patterns are very specific (require specific verb forms and sentence format). It would be better to look for *syntactic* patterns: verb == promoveren with location attribute

The patterns do not manage to find a lot of information (low recall): with these two, Google finds nothing relevant for Albert Einstein!

The patterns only work for a specific kind of information.

## Techniques for obtaining extraction patterns

- manually defining specialized patterns (Hearst, 1992)
- automatically finding specialized patterns (Lin & Pantel, 2001)
- automatically finding general patterns (Banko et al., 2007)

## Automatic discovery of extraction patterns

The example patterns are very specific and we need a lot of them in order to extract a reasonable amount of information.

However, we can extract useful patterns automatically by searching for relevant pairs information of information.

Example: Google: *Mozart \* 1756*

- X geboren Y (2x)
- X Salzburg Y (2x)
- X werd in Y geboren (1x)
- X componist Y (1x)

## How do we select the best patterns?

We don't select the best patterns but use everything.

A machine learning system can combine the information obtained from all patterns and then assign scores to the extracted information (based on the performance of the patterns on the training examples).

## Case study

A hypernym of a word W is another word H such that H both covers the meaning of W and has a broader meaning.

Example: *furniture* is a hypernym of *table*

Our goal is to extend a lexical resource with new words which have a hypernym which is already present in the resource.

We will approach this task as an information extraction task.

## Approach

We will use different methods for finding and classifying new words X:

- Search in a text corpus for phrases *Y zoals X* (such as)
- Search in a text corpus for phrases *Y en X* (and)
- Search in a text corpus for learned pattern
- Search on the web for phrases *Y en X* (and)

## Evaluation

We evaluate by comparing the output of each method with the contents of the lexical resource (Dutch part of EuroWordNet).

The **precision** of a method is the fraction of discovered pairs of known words (Hyponym,Word) which are related according to the lexicon.

The **recall** of a method is the fraction of related pairs in the lexicon (Hyponym,Word) that the method is able to discover.

## Examples of patterns that were discovered

Precision	Recall	Pattern
0.375	0.00137	N-pl , vooral N-pl ( <i>especially</i> )
0.300	0.00133	N-pl , waaronder N-pl ( <i>among which</i> )
0.258	0.00120	N-pl , waaronder N-sg ( <i>among which</i> )
0.250	0.00196	N-pl of ander N-pl ( <i>or other</i> )
0.244	0.00418	N-pl zoals N-sg ( <i>such as</i> )
0.220	0.00259	N-pl zoals N-pl ( <i>such as</i> )
0.213	0.00809	N-pl en ander N-pl ( <i>and other</i> )
0.205	0.00387	N-pl , zoals N-pl ( <i>such as</i> )
0.184	0.00396	N-pl , zoals N-sg ( <i>such as</i> )
0.158	0.00394	N-sg en ander N-pl ( <i>and other</i> )

## Results

Method	Precision	Recall
corpus: <i>N zoals N</i>	0.22	0.0068
corpus: learned patterns	0.36	0.020
corpus: <i>N en N</i>	0.31	0.14
web: <i>N en N</i>	0.39	0.31
morphological approach	0.54	0.33

The morphological method splits compounds: *tafeltennistafel* has a suffix part *tafel* and the approach generates this string as candidate hypernym.

## Lessons learned from this study

The more (positive) patterns are used, the better the results will be.

Applying the methods to a larger collection of documents not only improves the recall scores but also the precision scores.

Always compare smart methods with simple methods. The latter might perform better than you expected.

## Information Extraction Competitions

- MUC (Message Understanding Conference, 1987–1997)  
First competition on tasks like named entity recognition, information extraction and coreference resolution
- ACE (Automatic Content Extraction, 1999–)  
Yearly competition with different tasks: detection of entities, events and relations <http://nist.gov/speech/tests/ace/>
- TREC (1992–) and CLEF (2000–)  
Yearly text processing tasks with among others question answering <http://trec.nist.gov/> and <http://clef-campaign.org/>

## Examples of extraction systems

- GATE (University of Sheffield)  
Java toolkit for processing text with among others an information extraction component. Targeted at researchers <http://gate.ac.uk/>
- OpenCalais (Reuters)  
Enriches documents with semantic metadata. Targeted at web masters <http://opencalais.com/>
- START (MIT)  
Question answering system developed for research purposes <http://start.csail.mit.edu/>

## DBpedia

DBpedia is an effort to extract factual information from the online encyclopedia Wikipedia, for example names and relations

Relations are extracted from infoboxes and category information

Relations are stored in RDF triplets: subject, predicate and object  
Example: document<sub>123</sub> has\_creator person<sub>456</sub>

Present size: 2.6 million items and 274 million relations (triples)

Address: <http://dbpedia.org/>

## Information extraction with finite-state techniques

Basic lexical extraction rules can be implemented with finite-state techniques in a straight-forward way.

Examples:

- LOCATION-ADJ hoofdstad LOCATION
- PERSON ( YEAR - YEAR )
- directeur van COMPANY , PERSON

## Extraction rules in the FSA toolkit

```
macro (capital,
      replace([[ : ['C', 'A', 'P', 'I', 'T', 'A', 'L'] ,
                space: '_' , location-adj , space: '_' ,
                [h, o, f, d, s, t, a, d]: [], space: [] ,
                location , space], [ ] , [ ] ) .
```

Example: Zweedse hoofdstad Stockholm → CAPITAL\_Zweedse\_Stockholm

Notes:

- We need lists of terms (country adjectives, locations, ...)
- A separate postprocessing macro will remove all other text

## Practical problem with the toolkit

Information extraction rules like the one on the previous slide result (character level) are tedious to formulate.

So we will do the lab session by defining rules in a programming language.

You may write software in your favorite programming language. Choices in 2008: Perl (8), Java (4) and C++ (2).

## Interesting topics to work on

We need to restrict ourselves to specific information, for example:

- general definitions
- people facts: birth days, death days, country, profession, ...
- location facts: area, inhabitants, capital, president, ...
- organization facts: boss, founder, location head quarters, ...
- event parts: location, date, purpose, ...
- other: abbreviation expansion, ...

## Background material

More information about these topics can be found in:

- Daniel Jurafsky and James Martin, *Speech and Language Processing*. Prentice-Hall, 2009, chapter 22.
- M. Banko, M. Cafarella, S. Soderland, M. Broadhead and O. Etzioni, Open Information Extraction from the Web. In: *Proceedings of IJCAI-07, 2007*.
- Erik Tjong Kim Sang and Katja Hofmann, Automatic Extraction of Dutch Hypernym-Hyponym Pairs. In: *Proceedings of CLIN-2006*, Leuven, Belgium, 2007.
- Valentin Jijkoun, Jori Mur en Maarten de Rijke, Information Extraction for Question Answering: Improving Recall Through Syntactic Patterns. In: *Proceedings of COLING 2004*, Geneva, Switzerland, 2004.

## What is next?

- final lab (information extraction, Wednesday 20 May)
- third Nestor quiz (available Wednesday 20 May)

## Course wrap-up

We have examined finite-state automata, finite-state transducers and their application in natural language processing.

Finite-state methods work fine for representing linguistic knowledge.

However, finite-state solutions for practical problems tend to become big. Automatic methods for building them (like transformation-based error-driven learning) are very useful.

**THE END**