Natuurlijke Taalverwerking
Natural Language Processing

Gosse Bouma and Begoña Villada

3e trimester 2002/2003

Overview

1. Spelling Rules,
2. Replace,
3. Verbal Inflection.

Spelling Reform

- (Dutch) Words should be spelled the way they are pronounced,
- In the new Dutch Spelling,
  - ⋆ `x` will be written as `ks`
  - ⋆ `qu` will be written as `kw`,
  - ⋆ `c` will be written as `s` or `k` (depending on pronunciation),
  - ⋆ `ch` will be written as `g`,
  - ⋆ `isch` will be written as `ies`.

Spelling Reform

extra	$\longrightarrow$	ekstra
frequentie	$\longrightarrow$	frekwentie
centraal	$\longrightarrow$	sentraal
camera	$\longrightarrow$	kamera
lach	$\longrightarrow$	lag
automatisch	$\longrightarrow$	automaties
exotoxine	$\longrightarrow$	eksotoksine
accu	$\longrightarrow$	akku
accent	$\longrightarrow$	aksent
acquit	$\longrightarrow$	akkwit

Rules for Spelling Changes

- Replace x by ks,
- First Attempt (wrong):
  - `[[?* , x:[k,s] ]*, ?*]`
  - xerox ⟶ kseroks, kserox, xeroks, xerox
  - Replacement is obligatory!
- Second Attempt (ok):
  - `[[(? -x) * , x:[k,s] ]*, (? -x) *]`
  - `{ (? -x), x:[k,s]}*` (shorter)

Context-Sensitive Rules

- Replace c with s if followed by e or i
  - `cent, politici`
- First Attempt:
  - `{ ? - c, [c:s, {e,i}] }*`
  - cent ⟶ sent,
  - politicus ⟶ no output

Context-Sensitive Rules 2

- Replace c followed by e or i with s, elsewhere with k

```
{ ? -c,
  [c:s, {e,i}],
  [c:k, ? -{e,i}] }*
```

  - cent ⟶ sent,
  - politicus ⟶ politikus,
  - akku ⟶ akcu

Third Attempt, Double C

```
{ ? - c,
  [c:s,{e,i}],
  [c:k,{ ? -{e,i,c},
         [c:s,{e,i}],
         [c:k,? -{e,i}]  }  ]  }*
```

- accu ⟶ akku,
- accent ⟶ aksent,
- accccu ⟶ akkcku

Fourth Attempt, Double C

```
{? -c, [c:s, {e,i}], [c:k, ? -{e,i}]}*
                  o
{? -c, [c:s, {e,i}], [c:k, ? -{e,i}]}*
```

- accu → akku,
- accent → aksent,
- accccu → akkkku

Multiple C Rules

- isch → ies,
  - ⋆ Must obligatory replace a string of 4 characters, but leave untouched all other 1-4 sequences ???
- ch → g
- c/k/s rule
- Order specific rules before more general rules

Fifth Attempt, Double C

```
{? - c,
 [c:k*,{[c:s,{e,i}],
        [c:k,? -{e,i,c}] } ] }*
```

- accu → akku,
- accent → aksent,
- accccu → akkkku

Multiple C Rules

```
[[ ?*, [i,s,c,h]:[i,e,s]]*, ?*]
              o
         ~ $[i,s,c,h]
              o
     [[ ?*, [c,h]:g]*, ?*]
              o
           ~ $[c,h]
              o
          cks_rule
```

Intermezzo: Reg Ex's without Kleene *

- Automata for languages definable without Kleene * or + have interesting properties (Yli Jyrä, EACL 2003)
- Can you define the language `a*` without using Kleene *, +, or $?
- `~[{[],~[]},? -a,{[],~[]}]`

Intermezzo: Information Extraction with Reg Ex's

- Question Answering:
  - ⋆ Find web-pages which contain the answer to a question in natural language,
  - ⋆ When was Michael Boogerd born?

Intermezzo: Information Extraction with Reg Ex's

- Information Extraction, Text Mining:
  - ⋆ Search the Web for specific information (names (of politicians, soccer players, ...), (mail, e-mail, telephone) addresses, lists of publications, ...)

Intermezzo: Information Extraction with Reg Ex's

- Finding the birth date for Person:
- Seed: someone for which you know the birthdate,
- Michael Boogerd : 28 mei 1972

Google: Boogerd, 1972

- Michael Boogerd werd geboren op 28 mei 1972 in Den Haag
- Michael Boogerd 28 mei 1972 Den Haag
- Michael Boogerd Geb. 28. Mai 1972
- BOOGERD, Michael (Hol) - b. 5/28/1972
- Boogerd ... Fecha de nacimiento: 28-05-1972

Question Answering with Reg Ex's

- Questions for birth dates for Person are now found on pages
  - ⋆ containing Person (Google)
  - ⋆ and a match for the Reg Ex
- Use several Seeds to get general patterns,
- Extract patterns automatically (Machine Learning)

From Seeds to Reg Ex's

- Create a Reg Ex which matches the answers,
- where Seed Name and Answer are replaced by generic patterns,

```
person werd geboren op date
macro(person,[A-Z,a-z*, ,A-Z,a-z*])
macro(date,[0-9,0-9, ,maand, ,0-9,0-9,0-9,0-9])
```

The Replace Operator

- Writing spelling rules by hand is difficult,
- `replace(A:B, LftContext, RghtContext)`:
  - ⋆ Obligatorily replace all A's between LftContext and RghtContext, by B.
  - ⋆ A:B is a RegEx defining an arbitrary transducer,
  - ⋆ LftContext and RghtContext are RegEx's for a recognizer

C-rules with Replace

```
replace([s,c,h]:[e,s],[i],[])
               o
  replace([c,h]:g,[],[])
               o
  replace(c:s,[],{i,e})
               o
   replace(c:k,[],[])
```

Hyphenation

- Insert a hyphen between a two syllables,
- Maximizing the onset of the second syllable

  `replace([]:-, syllable, syllable)`

- alfabet $\longrightarrow$ al-fa-bet
- aap $\longrightarrow$ a-ap

Replace Left to Right

- Replace works from left to right,

  `replace(a:b,a,[])`

- aa $\longrightarrow$ ab
- aaa $\longrightarrow$ aba

Replace Longest Match

- Replace performs longest match:
  - It replaces the longest substring in the input matching the target
- replace([[]:@, nucleus, []:@],[],[])
- aap $\longrightarrow$ @aa@p , * @a@@a@p

Hyphenation

```
replace([[]:@, nucleus, []:@],[],[])
              o
 replace([]:-, [@,coda^],[onset^,@]
              o
      replace(@:[],[],[])
```

- alfabet → @a@lf@a@b@e@t → al-fa-bet
- aap → @aa@p → aap

Verbal Inflection

- A regular (weak) verbal root in Dutch can be inflected with
  - +t (3rd person singular form),
  - +en (plural and infinitive),
  - +Te (singular past tense),
  - +Ten (plural past tense)

Verbal Inflection

Root		werk	raad
1st pers sing	(ik)	werk	raad
3rd pers sing	(hij,zij)	werkt	raadt
plural	(wij, jullie)	werken	raden
sing past tense	(ik, hij, zij)	werkte	raadde
plur past tense	(wij, jullie)	werkten	raadden

Examples

Lexical	Surface	Lexical	Surface
loop+t	loopt	werk+en	werken
brand+t	brandt	maak+en	maken
ga+t	gaat	zie+en	zien
zet+t	zet		
bof+t	boft	werk+Te	werkte
leev+t	leeft	ren+Te	rende

Exercise 2

- Define a transducer which takes the root form of a verb plus an ending as input, and produces the written (surface) form.