Language and Computation

week 4, Tuesday, February 4, 2014

Tamás Biró
Yale University
`tamas.biro@yale.edu`
`http://www.birot.hu/courses/2014-LC/`

Practical matters

- **Post-reading:** JM 3, 23.1.1, 4.1-4.3
- **Pre-reading:** JM 5.1-5.4 (eventually: chapter 7)
- **Python:** this week H 3 and 4; next week H 5.
- Section: chance to practice reading pseudo-codes.

Today

- A short note on FS phonology and morphology (more to come in March)
- Minimal Edit Distance
- Document classification with cosine metrics
- Intro to machine learning

Finite-state phonology and morphology

FSTs and Regular Relations

Given a finite input alphabet Σ and a finite output alphabet Δ:

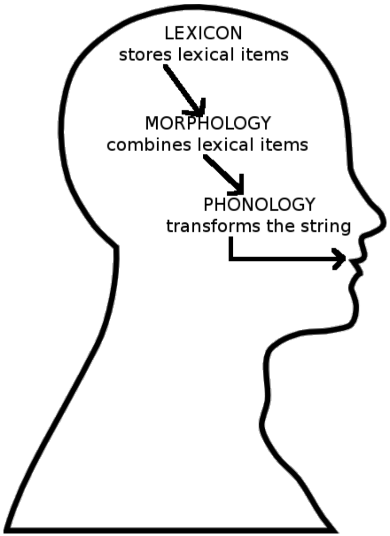
Let relation $\mathcal{R}$ be $\subseteq (\Sigma^* \times \Delta^*)$

- FST as **translator**: maps (some) strings $\in \Sigma^*$ onto strings $\in \Delta^*$.
- FST as **recognizer**: accepts string pairs $\in \mathcal{R}$, rejects if $\notin \mathcal{R}$.
- FST as **generator**: outputs string pairs $\in \mathcal{R}$, does not produce if $\notin \mathcal{R}$.
- FST as **set relater**: defines relation $\mathcal{R}$.

(Almost FSA over alphabet $\{(a : b) | a \in \Sigma, b \in \Delta\}$. Why not exactly?)

Finite-state phonology and morphology: Natural language phonology as a *regular relation*?

- SPE phonology (Chomsky and Halle (1968): *The Sound Pattern of English*)
  context-sensitive rules map */underlying form/* → [*surface form*]



/l/ Devoicing

$$/\mathrm{l}/ \;\rightarrow\; [\mathrm{\widehat{\mathring{l}l}}] \;/\; \begin{bmatrix}+\text{consonant}\\ -\text{voice}\end{bmatrix} \_\_\_$$

Partially devoice /l/ after a voiceless consonant.

/l/ Dentalization

/l/ → [ɫ̪] / ___ θ

/l/ *is rendered as velarized and dental before* [θ].

/l/ Velarization

/l/ → [ɫ] / ___ $]_{\text{word}}$

/l/ *is velarized word-finally.*

file	*slight*	*wealth*	*listen*	
/faɪl/	/slaɪt/	/wɛlθ/	/ˈlɪsən/	underlying forms
—	sl̥͡laɪt	—	—	/l/ Devoicing
—	—	wɛɫ̪θ	—	/l/ Dentalization
faɪɫ	—	—	—	/l/ Velarization
[ˈfaɪɫ]	[sl̥͡laɪt]	[ˈwɛɫ̪θ]	[ˈlɪsən]	surface forms

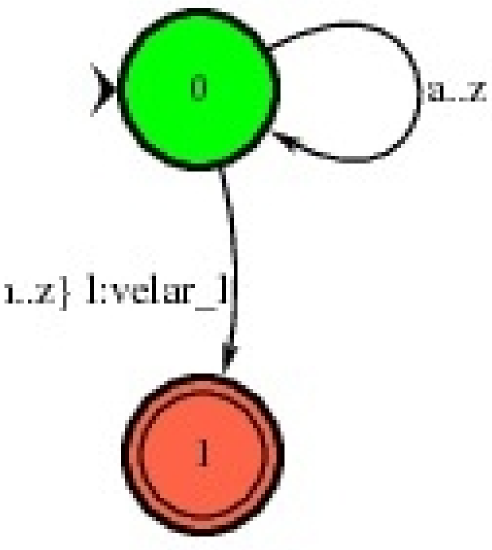
(Bruce Hayes (2009). *Introductory Phonology*, pp. 29-30.)

- SPE rules are context sensitive, but define a regular relation (modulo. . .): Johnson (1972), Kaplan and Kay (1994). Cf. *Two-level phonology* by Koskenniemi (1983).

- *Optimality Theory* also defining a regular relation? Sometimes, cf. Frank and Satta (1998), etc.

Finite-state phonology and morphology

/l/ is velarized word-finally:



`/ (a:a..z:z)* (a:a..k:k, m:m..z:z, l:velar_l) $ /`

Finite-state phonology and morphology

Natural language phonology as a *regular relation*?

- Language technology (e.g., spell checkers):

  a cascade of FS-lexicon, FS-morphology and FS-phonology;

  stemming with and without a lexicon (*Porter stemmer*);

  tokenization; error correction.

- Spelling suggestions?

Words not recognized by `ispell`: FSA, stemmer, tokenization.

Minimal Edit Distance

Metric or distance

Given a set X, the function $d : X \times X \rightarrow \mathbb{R}$ is a **distance metric** iff the following are satisfied for all $a, b, c \in X$:

- $d(a, b) \geq 0$ (non-negativity)
- $d(a, b) = d(b, a)$ (symmetry)
- $d(a, b) = 0$ if and only if $a = b$ (identity of indiscernibles, or coincidence axiom)
- $d(a, b) + d(b, c) \geq d(a, c)$ (subadditivity, or triangle inequality)

Edit Distance, Levenshtein Distance

```
I N T E * N T I O N
| | | | | | | | | |
* E X E C U T I O N
d s s   i s
```

```
i n t e n t i o n
                    ← delete i
n t e n t i o n
                    ← substitute n by e
e t e n t i o n
                    ← substitute t by x
e x e n t i o n
                    ← insert u
e x e n u t i o n
                    ← substitute n by c
e x e c u t i o n
```

Levenshtein distance in *dialectometry*

`http://us.english.uga.edu/lamsas/`

1162 informants from 483 communities. 151 different items.

`http://urd.let.rug.nl/nerbonne/papers/lavis2004.pdf`
pp. 12 and 14.

NC, VA, WV, DC + MD and DE for comparison: 283 field work sites, 57,833 phonetic transcriptions of words and brief phrases (roughly 243 per site).

`http://urd.let.rug.nl/nerbonne/papers/lamsas-lex.pdf`
p. 19.

Minimum Edit Distance

function MIN-EDIT-DISTANCE(*target*, *source*) **returns** *min-distance*

$n \leftarrow$ LENGTH(*target*)
$m \leftarrow$ LENGTH(*source*)
Create a distance matrix *distance[n+1,m+1]*
Initialize the zeroth row and column to be the distance from the empty string
  distance[0,0] = 0
  for each column i **from** 1 **to** n **do**
    distance[i,0] $\leftarrow$ *distance*[i-1,0] + *ins-cost*(*target*[i])
  for each row j **from** 1 **to** m **do**
    distance[0,j] $\leftarrow$ *distance*[0,j-1] + *del-cost*(*source*[j])
for each column i **from** 1 **to** n **do**
  for each row j **from** 1 **to** m **do**
    distance[i,j] $\leftarrow$ MIN(*distance*[$i-1,j$] + *ins-cost*($target_{i-1}$),
        distance[$i-1,j-1$] + *sub-cost*($source_{j-1}$,]$target_{i-1}$),
        distance[$i,j-1$] + *del-cost*($source_{j-1}$))
return *distance*[n,m]

n	*9*	8	9	10	11	12	11	10	9	8
o	*8*	7	8	9	10	11	10	9	8	9
i	*7*	6	7	8	9	10	9	8	9	10
t	*6*	5	6	7	8	9	8	9	10	11
n	*5*	4	5	6	7	8	9	10	11	10
e	*4*	3	4	5	6	7	8	9	10	9
t	*3*	4	5	6	7	8	7	8	9	8
n	*2*	3	4	5	6	7	8	7	8	7
i	*1*	2	3	4	5	6	7	6	7	8
#	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**

n	9	↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙←↓ 12	↓ 11	↓ 10	↓ 9	↙ **8**
o	8	↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↙←↓ 11	↓ 10	↓ 9	↙ **8**	← 9
i	7	↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9	↙ **8**	← 9	← 10
t	6	↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙←↓ 9	↙ **8**	← 9	← 10	←↓ 11
n	5	↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ **8**	↙←↓ 9	↙←↓ 10	↙←↓ 11	↙↓ 10
e	4	↙ 3	← 4	↙← **5**	← **6**	← 7	←↓ 8	↙←↓ 9	↙←↓ 10	↓ 9
t	3	↙←↓ 4	↙←↓ **5**	↙←↓ 6	↙←↓ 7	↙←↓ 8	↙ 7	←↓ 8	↙←↓ 9	↓ 8
n	2	↙←↓ **3**	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙←↓ 8	↓ 7	↙←↓ 8	↙ 7
i	**1**	↙←↓ 2	↙←↓ 3	↙←↓ 4	↙←↓ 5	↙←↓ 6	↙←↓ 7	↙ 6	← 7	← 8
#	**0**	1	2	3	4	5	6	7	8	9
	#	**e**	**x**	**e**	**c**	**u**	**t**	**i**	**o**	**n**

Comparing documents with n-grams

Task: document categorization/classification

Many documents entering a news agency, to be classified by

- language
- topic
- author
- genre
- political preference
  etc.

Machine learning: the basic idea

Task: given set X (e.g., of [possible] documents),

a set Y of tags (e.g., of languages, of topics, of authors, etc.),

and a **training set** $\{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\} \in (X \times Y)^n$,

find a method that maps any $x \in X$ onto Y,

so that the performance of the model on a **test set** be maximal.

to be refined!

A text as

- a meaning, a message
- as a series of sentences
- a string of words
- a bag of words
- a series of n-grams:
  - a string of n characters / letters / words / etc.
  - overlapping or non-overlapping

Vector Space Model and the Cosine Metric

- $f(w_i, D)$: frequency of word / n-gram w_i in document D.
- Given document D, create vector $(f(w_1, D), f(w_2, D), \ldots f(w_n, D))$
- Distance of two vectors: use their **cosine distance** (normalized dot product):

$$d(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n} a_i \cdot b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}}$$

- For each $y \in Y$, create reference vector D_y.
  To categorize document D, find closest reference vector.

See you on Thursday!