Computational Linguistics
A.Y. 2018/19 - C. Chesi

Lecture 8

LEXICON, MORPHOLOGY…
AND NON-STANDARD ORTHOGRAPHY

Index

- **Lexicon and Morphology**
  - Organizing lexical entries
  - Two-levels morphology
  - Morphological analysis with Finite State Automata (FSA) and Finite State Transducers (FST)
  - Some simple application and psycholinguistic reality: stemming
- **Input normalization**
  - Intro to orthographic correction
  - Error classification
  - Spell-checking methods
  - T9 and Swipe

References

- **Essential references**
- Jurafsky, D. & Martin, J. H. (2009)
  Speech and Language Processing. Prentice-Hall. (2nd edition)
  http://www.cs.colorado.edu/~martin/slp.html
  Chapter 3
- **Extended references**
  - Koskenniemi, K. (1983)
    Two-level morphology: A general computational model for word-form recognition and production. Helsinki
  - Miller & al. (1993)
    Introduction to WordNet: An On-line Lexical Database. ms.
  - Pustejovsky J. (1995)
    The Generative Lexicon. MIT Press
  - Levin B. (1993)
    English Verb Classes and Alternations. The University of Chicago Press.

"Generative" lexicon

- Include in the lexicon any inflected word as independent and «atomic»?
  - It will be **inefficient**:

	p dqj l#0	vrjq#0	fruu#0	sxii#0
0huh2huh	p dqj l0duh	vrjq0duh	fruudhuh	sxii0duh
0r	p dqj l0r	vrjq0r	fruu0r	sxii0r
0dwr	p dqj l0dwr	vrjq0dwr	-fruu0dwr +fruvr,	sxii0dwr

  - in Turkish (agglutinative language) there would be 600x10^6 entries. In Finnish 10^7
  - It will be **non-informative**:
    - **No relation** among lexical entries (alphabetic order is not interesting)
    - **No processing hints** (nouns = verbs?)

“Generative” lexicon

- A computational lexicon can be conceived as:
  - **Mental lexicon** – the relation among lexical units are psycholinguistically plausible?
  - **Computational lexicon** – is the lexical representation efficient?
- Rule of thumb:
  - Lexical representation must be **explicit** end **independent** (with respect to the application that will use it)
  - **Global structure** of lexical entries is as important as **internal structure**
  - A lexicon must have a sufficient **domain coverage** (consider nearly 400.000 lexical entries)

“Generative” lexicon

- Computational Lexicon evaluation parameters:
  - **Coverage** (sufficient domain extension, and depth, also in terms of featural information)
  - **Extensibility** (how easy it is enriching the lexicon?)
  - **Utility** (single application benefit)
- Remember:
  - completeness does not ensure correctness (neither **psycholinguistic** nor **computational**)
  - psycholinguistic plausibility does not guarantee computational effectiveness (and the way around)

“Generative” lexicon

Single entry structure:

- orthographic/phonetic **information**
- morphology (inherent features like number, gendered...)
- syntactic (POS and more fine grained features: mass/countable, animacy, selection…)
- semantic (semantic relations, useful information for Machine Translation)

CASA (“house”)
<C,A,S,A>
{N, sing, fem ...}
{N com ...}
[house]

E.g. XML coding for the “casa” («house») entry:

```
<word cat=“noun" subcat="common.countable" num="sg" gen="f" sem="c12">
    casa
</word>
```

“Generative” lexicon

XML tagged sentence :

«il presidente Mattarella non ha replicato»

President Mattarella did not replied

```
<node cat="S" id="2015-03-16.1">
    <node cat="NP" role="arg.subj">
      <word cat="D.art.def" agree="m.s" lemma="il">il</word>
      <word cat="N.comm.count" agree="m.s" role="head"
                                          lemma="presidente">presidente</word>
      <node cat="NP" role=“adj.adposition">
      <word cat="NE.per" agree="m.s" role="head"
                          lemma=" Mattarella">Mattarella</word>
      </node>
    </node>
    <word cat="ADV.neg">non</word>
    <node cat="VP" role="head">
      <word cat="V.aux.ind.pres" agree="3.s" role="head"
                          lemma="avere">ha</word>
      <word cat="V.part.past" agree="m.s" role="head"
                          lemma=“replicare">replicato</word>
    </node>
</node>
```

“Generative” lexicon

DTD (Document Type Definition):

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<!ELEMENT node (node|word)*>
<!ELEMENT word (#PCDATA)>
<!ATTLIST expression id CDATA #REQUIRED>

<!ATTLIST node id CDATA #IMPLIED>
<!ATTLIST node cat CDATA #REQUIRED>
<!ATTLIST node subcat CDATA #IMPLIED>
<!ATTLIST node ref CDATA #IMPLIED>
<!ATTLIST node role CDATA #IMPLIED>
<!ATTLIST node agree CDATA #IMPLIED>
<!ATTLIST node lp CDATA #IMPLIED>

<!ATTLIST word id CDATA #IMPLIED>
<!ATTLIST word cat CDATA #REQUIRED>
<!ATTLIST word subcat CDATA #IMPLIED>
<!ATTLIST word ref CDATA #IMPLIED>
<!ATTLIST word agree CDATA #IMPLIED>
<!ATTLIST word role CDATA #IMPLIED>
<!ATTLIST word lemma CDATA #IMPLIED>
<!ATTLIST word lp CDATA #IMPLIED>
<!ATTLIST word sem CDATA #IMPLIED>
```

“Generative” lexicon”

Simple text, corpus-extracted, lexicon (Tab-Separated Values, TSV):

Token	**type/lemma**	**cat**	**agree**
il	il	D.art.def	m.s
presidente	presidente	N.comm.count	m.s
non	non	ADV.neg	
ha	avere	V.aux.ind.pres	3.s
commentato	commentare	V.part.past	m.s

“Generative” lexicon

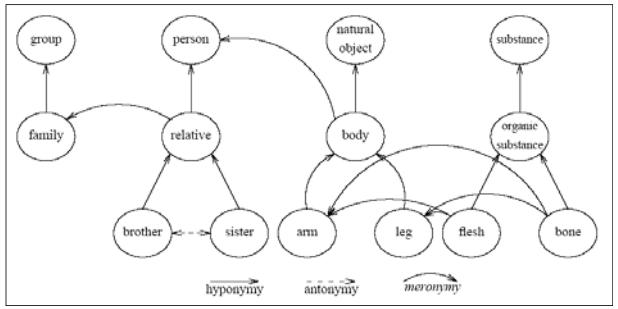
- **Global** lexicon structure
  - Subcategorization (and alternation Levin 1993) might be correlated with **semantic class**
  - We could draw immediate **inferences** on the basis of the hierarchical organization of the items in an ontology (**part_of**, **member_of**…)

Global structure of the lexicon: semantic networks

- **Wordnet** (Miller 90)
  - Example of semantic network (goal: organizing the lexicon on the basis of the meaning rather than orthography) based on the following principles:
    - Relation among **nouns** (hierarchy and inheritance), **verb** (implicatures), **adjectival** and **adverbials** (oppositions) (but **no functional words** are included)
    - Every lexical concept (**synset**) can be represented using **their synonims** (other synsets)
    - Kind of **relations** used:
      - **Hyponymy** (relation between a more concept general (bird) and a more specific one «robin»; “robin” is an hyponym of “bird”)
      - **Hypernymy** (inverse of hyponymy)
      - **Meronimy** (part_of)...
    - Representation of the «semantic ambiguity» problem: **polysemy** (*cane* = animal and *cane* = part of a gun, are **two distinct synsets** in wordnet)

Global structure of the lexicon: semantic networks

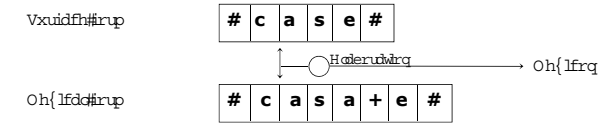
- **Semantic Relations example** (Miller 1993)



Morphology – the theoretical model

- **Goal**: recognizing a well-formed string and decompose it in morphemes

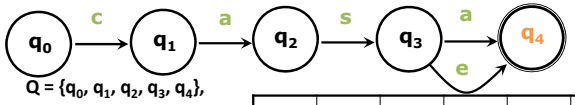
- **Theoretical model:**



Morphological analysis with FSA

- An FSA can be used for **recognizing** or **generating** a lexical item, but also for **representing** the lexicon.

- FSA recognizing *casa* and its plural:



Q = {q_0, q_1, q_2, q_3, q_4},
Σ = {c, a, s, e, #},
Q_0={q_0},
F ={q_4},
δ =

	t_3	t_4	t_5	t_6	t_7
f	t_4				
d		t_5		t_7	
v			t_6		
h				t_7	

Morphological analysis FSA and two-level morphology

- **FSA limits**

  no memory: it is not possible to **associate** a **structural description** to an element recognized as belonging to the lexicon, simple FSAs are not sufficient (since it does not exist an external memory, there is no way to keep track of the derivation).

- Koskenniemi (83) **two-level morphology**: a **lexical level** and a **superficial one** that must be put in a specific relation one with the other.

- We use **Finite-State Transducers** (**FST**) to do so.

Koskenniemi, K. (1983) *Two-level morphology: A general computational model for word-form recognition and production*. Publication 11, University of Helsinki, Department of General Linguistics, Helsinki.

Morphological analysis with FSTs

- **Finite-State Transducers** (**FST**, o **Transducers**) **<Q, Σ, q_0, F, δ>**:
  - **Σ** = finite, non-null alphabet, input special (complex) chars *of the form i:o* where *i* are symbols of the input alphabet *I* and *o* are symbols of the output alphabet *O*. *Σ ∈IxO*. ε (the null element) can be included both in *I* and in *O*
  - **δ** = is defined as *(q , q', i : o)* and it represents a transition matrix putting in relation a state *q* (start) with a state *q'* (arrival) if the *i : o* relation is defined. δ is then a relation from *Q x Σ* to *Q*
- FSAs define a formal language (set of strings); FSTs define relations among languages.

Morphological analysis with FSTs

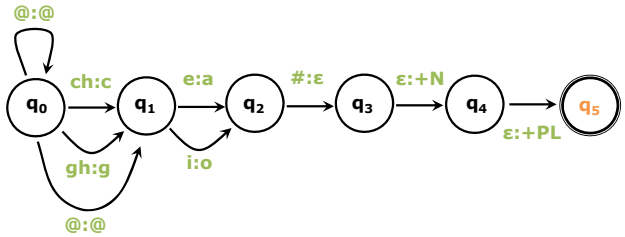
- FSTs can be used as **recognizers**, **generators**, **translators**, correlators **among sets**.
- Some formal property of the FSTs:
  - **Inversion** (defined as T^{-1}): input and output labels can be inverted
  - **composition**, if T_1 maps I_1 to O_1 and T_2 transduces from I_2 to O_2, $T_1 \circ T_2$ maps I_1 into O_2.

Morphological analysis with FSTs

- Example of **inflectional morphology** approach: define a FST describing plural inflection in Italian.
  - **Problem representation**
    examples: casa > case; donna > donne; gatto> gatti; ago > aghi; sacco > sacchi ...
  - **Generalizations/intuitions**
    feminine nouns get as plural inflection «*e*», «i» for masculine. *c* and *g* become *ch* and *gh* respectively.
  - **formalization**
    regular case: masculine noun > @:@ c|g|@:ch|gh|@ o:i
    feminine noun > @:@ c|g|@:ch|gh|@ a:e
    irregular case: uomo > @:@ o:i #:n #:i
  - **implementation**
    feminine noun > @:@ c|g|@:ch|gh|@ e:a #:ε #:+N #:+PL
    e.g. case -> casa +N +PL (c:c a:a s:s e:a #:ε #:+N #:+PL)

Morphological analysis with FSTs

Approximation of a FST for describing plural morphology in Italian:



Inadequacy of FSTs (and FSA) to express any morphological phenomena

- There are languages that present morphological derivations more complex than the one described. Such phenomena fall in the class of what we call **non-concatenative morphology**
- **Tagalog** (Philippine dialect) uses **infixes** in the middle of a word:
  um (marks the agent) + **hingi** («lend») =
  h-um-ingi («lend to someone»)
- **Semitic languages**, **template morphology**:
  consonant roots (CCC) **lmd** («learn») + inflection by vocalic schemes (CVCVC) =
  lamad («learned»)
  lumad («was learned»)

On inadequacy of FSTs (and FSA)

- **Problems**:
  - **Non-determinism** (multiple transition from the same state *q* might be pursued; *ε* transitions)
  - **Inadequacy** (e.g. non-concatenative morphology)
  - **Order** of application of FSAs

Morphological analysis: some application

- **Information extraction**
  (web, unstructured corpora/digital archives)
- **Keywords** expansion:
  (hotels in Florence = *(hotel AND Florence) OR (hotels AND Florence)*)
- **Stemming**
  retrieving the word root (***stem***) we can refine queries and make them more tolerant

Morphological analysis: some application

- **Porter Stemming** Algorithm

  simple set of cascade FSTs like:
  - ATIONAL -> ATE (e.g. relational -> relate)
  - ING -> ε (talking -> talk)
- pros e cons:
  - **ipergeneralization** (Krovetz 93)
    e.g. organization > organ, generalization > generic,
  - **Non captured exception** :
    matrices > matrix or European > Europe.
  - stemming is useful only with expansive research
    (no standard information retrieval)

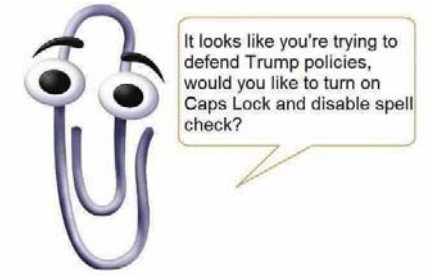
Morphological analysis: psycholinguistic plausibility

- **How is the mental lexicon structured?**
  - **full listing hypothesis –** *runs and run*, are two lexical entries in the mental lexicon (no internal morphological structure)
  - **minimum redundancy** – only morphemes are encoded in the mental lexicon; accessing an inflected lexical item requires accessing distinct morphemes and combination rules

Morphological analysis: psycholinguistic plausibility

- **Evidence for a structured lexicon**
  - **Priming Effects** (Stanners ad al. 79)
    irregular inflections: *happiness*, *happily* no priming with the root *happy* Vs.
    regular inflections: *pouring* > *pour*
  - **Semantic affinity** (Marslen-Wilson 94)
    government > *govern*
  - **Pronunciation errors** (Fromkin e Ratner 98)
    easy enoughly*** vs ˅*easi**ly** enough*
- This suggests that some information of the morhemic structure of the word should be encoded in our mental lexicon.

Spell Checking and Correction



Error Classification

- Different levels, different strategies/resources:
  - **lexical**
  - **syntactic**
  - **semantic**
  - **pragmatic**
- Remember that an error my be a **true error**, or a **system error**: e.g. at the lexical level a word could not be present in the lexicon, but present in the language (**system error**) or the result of a typo or a wrong belief on the orthography (**true error**)

Kinds of error correction approaches

- Main idea: **pattern matching** against lexical items stored in the repository + some heuristics to find the most suitable candidates for substitution
  - **Symbolic methods**
    (good representation of the problem)
  - **Sub-symbolic methods**
    (Machine Learning approach)

Error correction

- **Minimal Distance** (Damerau 64, Wagner 74)
  - the best alternative is the one that minimizes the number of insertions, deletions, substitutions and switching of chars
  - Compare the form with any possible transformation of this form and verify if there are alternatives present in the lexicon
  - This is **very inefficient**, at worst, we should compare the derived form with any lexical item in the vocabulary!

Error correction

- **Similarity Key** (SOUNDEX algorithm, Odell e Russel 1918, Davidson 1962)
  - Procedure: extract a key from the wrong form; extract the keys from all lexical items; compare the keys and provide as correction the lexical items sharing the same key
  - **Key** = first letter of the word + sequence of numbers associated to the chars (according to some frequency calculus); «0» items (vowels) and repeated numbers are reported only once

Yrz hcv	e/#i/#s/#y	R wkhu#frqvrqdqw
3	4	5

Example:

casa = c020 > c2; *csa* = c20 > c2

Error correction

- Improving the **Similarity Key** (Pollock e Zamorra, SPEEDCOP, 84)
  two keys for any word:
  - **skeleton key** = first letter of the word + consonants in the given order without repetition + vowels without repetition (e.g. gambero = gmbraeo);
  - **omission key** = consonants, without repetition ordered by frequency (error corpus) followed by vowels ordered in the same way

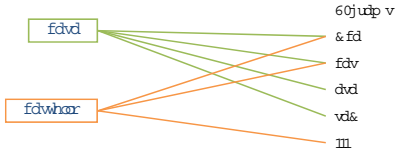
This is because

1. vowels order is often preserved
2. statistically, errors are located not at the beginning, but at the end of the word

94% of word errors are corrected this way, between 74% and 88% of all the errors present in the corpus

Error correction

- **N-grams** (Kohonen 80; DeHer 82; Angell et al. 83; DeSmedt e VanBerkel 88)
  - **word** = set of overlapping substrings (n-grams)
    example:
    casa = #c + ca + as + sa + a# (2-grams)
    strumento = #st str tru rum ume men ent nto to# (3-grams)
  - **Vocabulary** = indexed n-grams table; every index represent a word in the lexicon.
    The set of overlapping n-grams indicate an activation.



Error correction in specific contexts: T9

- **Keypad constraints in SMS typing** (Silfverberg e al. 1999)

- **Some classic solutions:**

	F	D	V	D	wrw
P xoxlOsuhvv	50505	5	:0:0:0:	5	<
wzr0nh|	506	504	:07	504	;
W<	5	5	:	5	7

Error correction in specific contexts: T9

	def	def	stuv	def
W<	5	5	:	5

- **Linguistic resources needed for T9**
  - Dictionary
  - Frequencies (e.g. typing 6-6, "ON" will be preferred to "NO". This is determined on the basis of statistical observations: in this case, the British National Corpus. The alternative choice is required about 5% of the time in English T9!)
- **Non linguistic resources to evaluate the model efficiency**
  - Fitts' Law (modeling rapid, goal-directed movements)
- **Results** (in Words Per Minutes, wpm)
  - Multi-press: 25-27 wpm
  - Two-key: 22-25 wpm
  - T9: 41-46 wpm

Today's key concepts

- What is a **Computational Lexicon**
  - Single entry structure (morpho-syntactic features)
  - Global structure (Wordnet)
- How do we deal with **morphological analysis**
  - Two-level morphology and FST
  - Some application (stemming)
  - The psycholinguistic plausibility of the model
- Input normalization and **spell-checking**
  - Error classification
  - Standard approach to spell correction (minimal distance, similarity keys, n-grams)
  - The case of T9