CS565: Intelligent Systems and Interfaces

Words: Morphology

Parsing a word

Semester: Jan – May 2019

Ashish Anand

Associate Professor, Dept of CSE

IIT Guwahati

Announcements

- Scribe
  - Megha Jain, Kushal Dey: 28th Jan Lec

Last Few lectures on

- Finding Collocation
  - Frequency approach
  - Frequency + Rule based approach
  - Statistical approaches

Objective

- What is Morphology?
- Why it matters?
- Different types of Morphology
- Morphological Parsing

Morphology

Parsing a word

Morphology: Exploring structure of words

- Words have structure
  - *Foxes* breaks down into *Fox* and *–es*
  - *Unknowingly* is derived from *knowingly*, which is derived from *knowing*, which in turn is derived from *know*
- Morphology: study of *minimal meaning bearing*, referred to as *morphemes*, units in words
  - *Fox* and *–es* in *Foxes*
  - *Un, know, -ing, -ly* in *Unknowingly*

Why study of Morphology matters ?

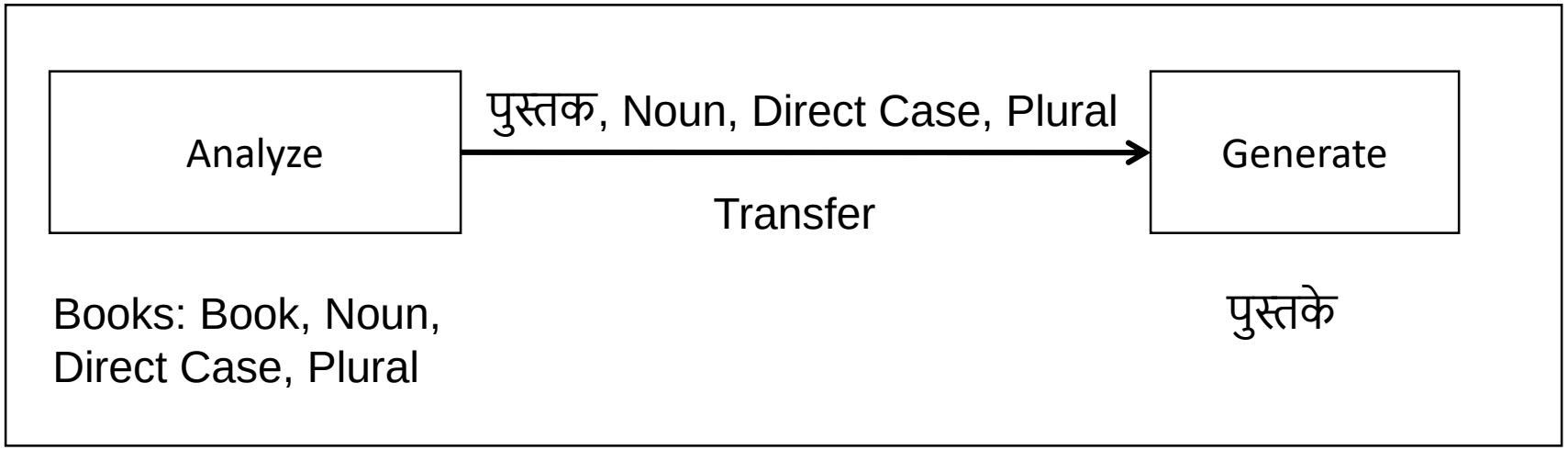
- Information Retrieval (Search)
  - Search for *foxes* must look for both *fox* and *foxes*
  - Morphological rules allow to handle complications, such as
    - Irregular plurals
      - Goose -> geese
      - Fish -> fish
      - Ox -> oxen
    - Spelling rules
      - Fox + PL -> foxes
      - Fly + PL -> flies

Source: http://www.cs.cornell.edu/courses/cs674/2003sp/morphology1.pdf

Why study of Morphology matters ?

- Information Retrieval (Stemming)
  - Useful to map all of *learning*, *learns*, *learned* to *learn*

- Machine Translation



Adapted from IIT Bombay lecture slides

Why study of Morphology matters ?

- Efficiency
  - Cannot list all possible forms even in morphological poor language (relatively) English
  - *Productivity* of language

- Morphological rich languages
  - Turkish, Finnish, Indian Languages

Two types of Morphemes

- Stems
  - Main morpheme of the word, supplying the main meaning
  - Example: fox, know

- Affixes
  - Provides additional meanings of various kinds
  - Mainly categorized into four types -
    - Prefix: Un-, Im-
    - Suffix: -s, -es, -ly
    - Infix: Mostly with other language. *–n-* in “vandimi” in Sanskrit; -um- in humingi in Philipine language Tagalog
    - Circumfix: ge-sag-t (meaning: said) in German; past participle of the verb *sagen* (to say)

Concatenative and non-concatenative Morphology

- Concatenative
  - Word is composed by concatenating a number of morphemes
  - Prefixes and Suffixes
- Non-concatenative
  - Combining morphemes is more complex
  - Tagalog Infixation example (hingi + um -> humingi)
  - Templatic morphology
    - Arabic, Hebrew
    - Hebrew: verb constitutes a root (carrying basic meaning) and a template giving ordering of consonants and vowels determining semantic information (active, passive)
    - Example: lmd (learn or study), template: CaCaC -> lamad (he studied)
    - Example: lmd (learn or study), template: CuCaC -> lumad (he was taught)

Two broad classes of Morphology

- Inflection
  - Stem + grammatical morpheme (s)
  - Usually word of the same class and filling some syntactic functions
  - English has simple inflectional morphology, compared to Hindi, Finnish or other European Languages
  - Very productive

- Derivation
  - Stem + grammatical morpheme (s)
  - Usually results in a word of different class and often difficult to guess exact meaning
  - English also has quite complex derivational morphology
  - Relatively less productive (-ation cannot be added to all verbs)

Inflectional Morphology: Example

- Nouns
  - Suffixes for Plural and possessive
- Verbs
  - Suffixes for –s form, -ing participle, past form or –ed participle
  - Walks, walking, walked
- Adjectives
  - Suffixes for comparatives
  - Cheap, cheaper, cheapest

Derivational Morphology: Example

- Nominalization
  - Formation of new nouns, often from verbs or adjectives
  - Organize (v) + -ation
  - Appoint (v) + -ee
  - Silly (ADJ) + -ness

- Adjectives
  - Computation (N) + -al

Derivational Morphology: Example

- Nominalization
  - Formation of new nouns, often from verbs or adjectives
  - Organize (v) + -ation
  - Appoint (v) + -ee
  - Silly (ADJ) + -ness

- Adjectives
  - Computation (N) + -al

Morphological Analysis

- Token -> stem + POS +grammatical features
  - Cats -> Cat +N +PL
- Often non-deterministic
  - Plays -> play +N +PL
  - Plays -> play +V +3SG
- Lemmatization
  - Token -> stem

Parsing the morphological structure

- Goal
  - Given an input word in *surface form*, produce *stem* plus *morphological features* (POS and grammatical features) as an output

- Example Goal: Productive nominal plural (-s) and the verbal progressive (-ing)
  - Input: Cats ; Output: cat +N +PL
  - Input: Eating; Output: eat +V +PRES-PART

Three knowledge resources needed

- Lexicon
  - Repository of words in a language
  - Explicit list is infeasible. Why ?
  - List of stems and affixes with basic information about them
- Morphotactics
  - Rules of morpheme ordering
  - Example: English plural morpheme follows the noun rather than preceding it.
- Orthographic or Spelling Rules
  - Model change in spelling when two morphemes combine
  - Fly -> flies [y -> ie]

Lexicons and Morphotactics

- Structured as
  - List of stems and affixes
  - Representation of the syntactics of morphemes

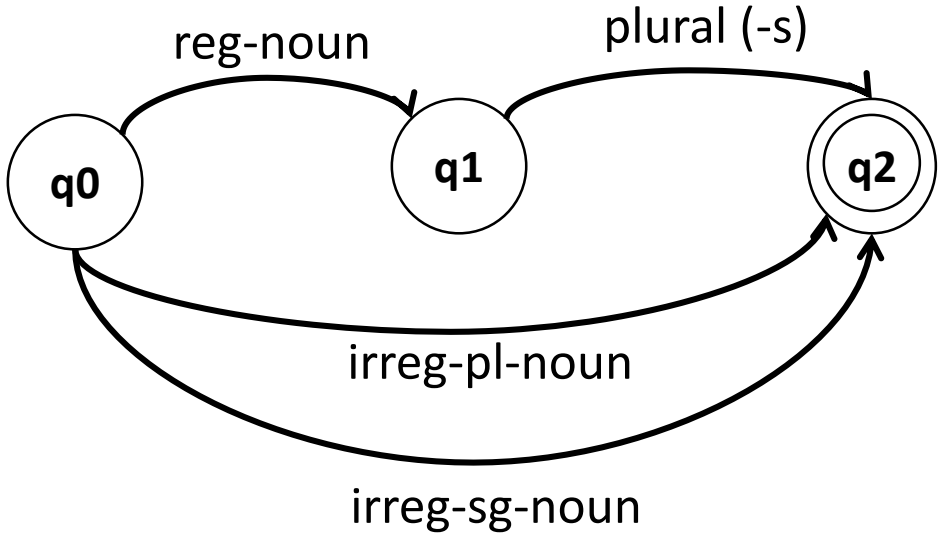
- Represent via a finite-state automaton (FSA)

Example: A FSA for English nominal inflection

Takes regular nouns (reg-nouns) that take regular –s plural.

Also includes irregular noun forms, both singular and plural, that don't take –s

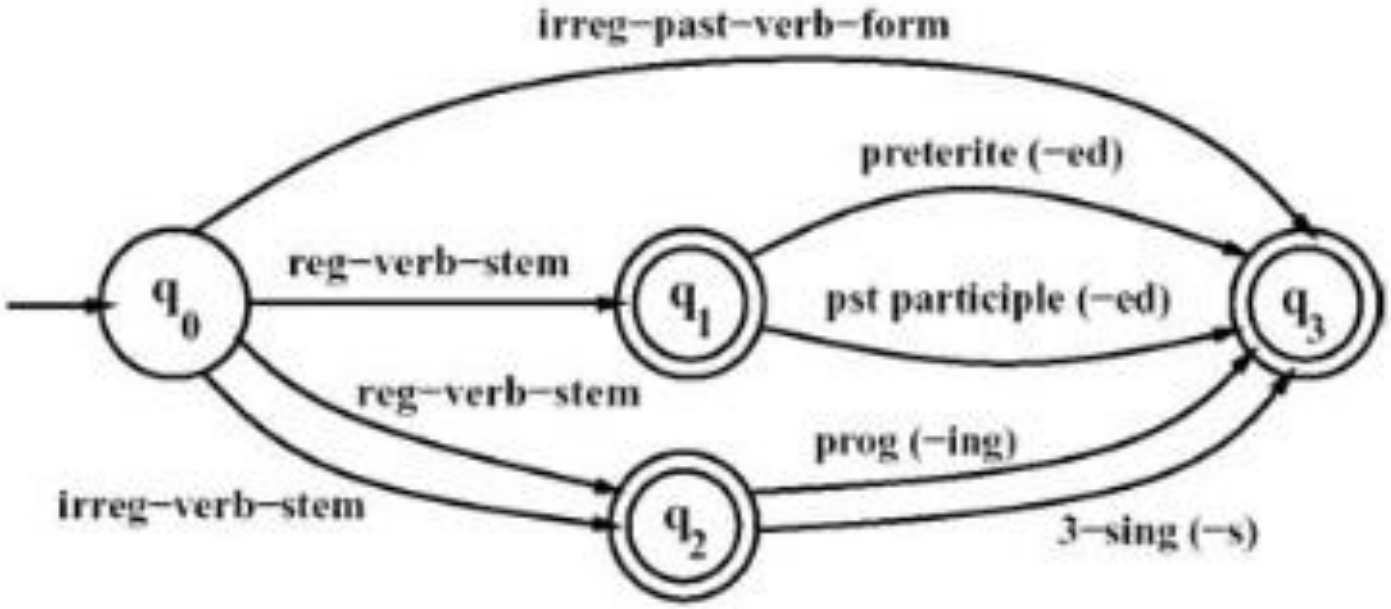
Ignore mistakes like foxes.



FSA for verbal inflection

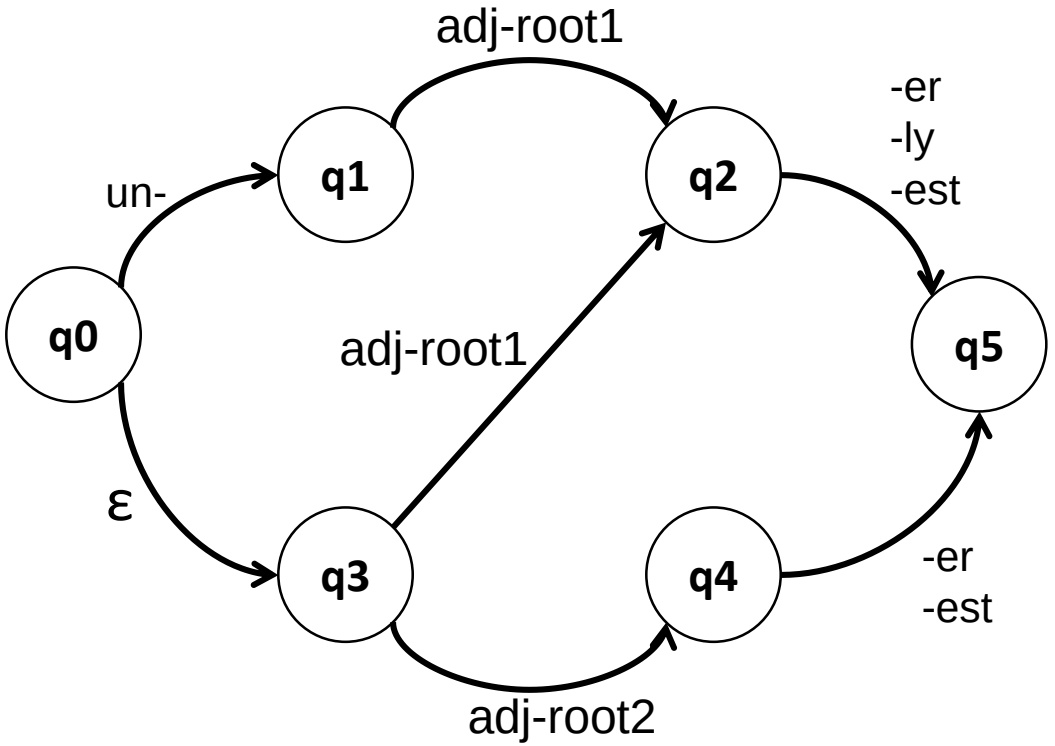
Lexicon
- three stem classes
- four affix classes (-ed past, -ed participle, -ing participle, third singular –s)

- example: reg-verb-stem: walk, talk
- Example: irreg-verb-stem: cut, speak
- Example: irreg-past-verb: caught, ate



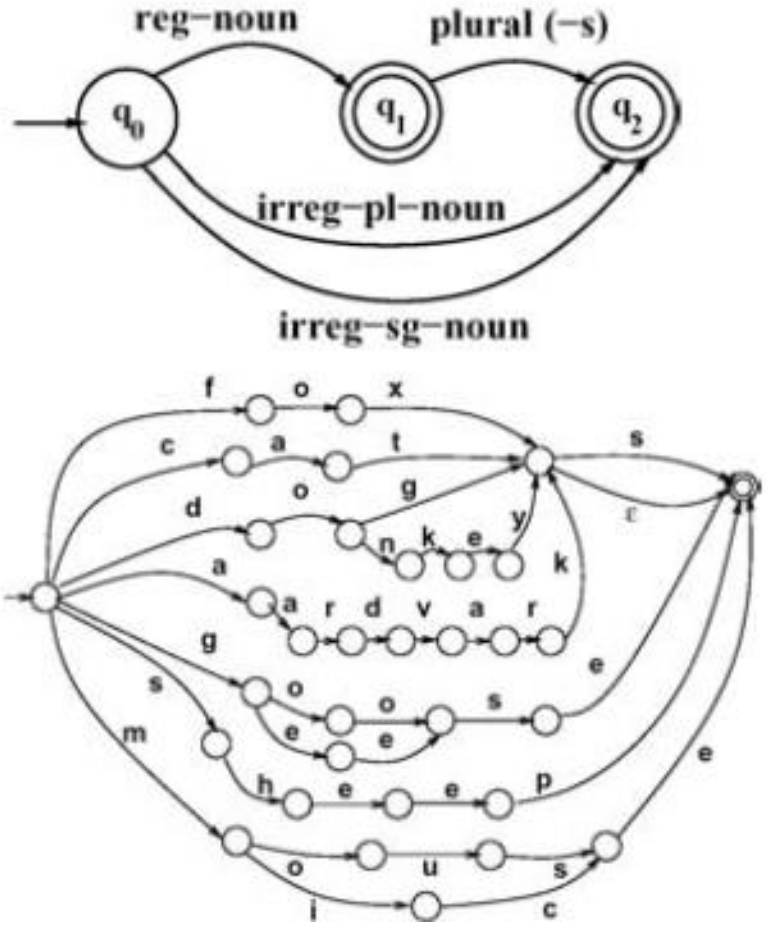
FSA for derivational morphology

Type	Properties	Examples
adj-root1	Occur with un- and -ly	happy, real
Adj-root2	Can't occur with un- and -ly	big, red



FSA for morphological recognition

- Goal: Use FSA to determine whether an input strings of letters makes up a legitimate English word

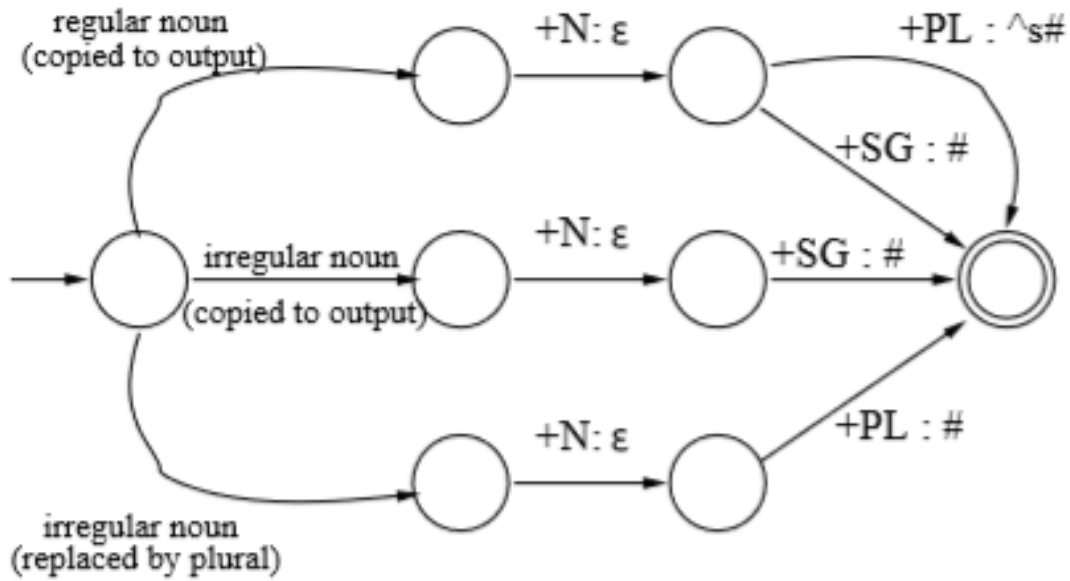


Morphological Analyzer

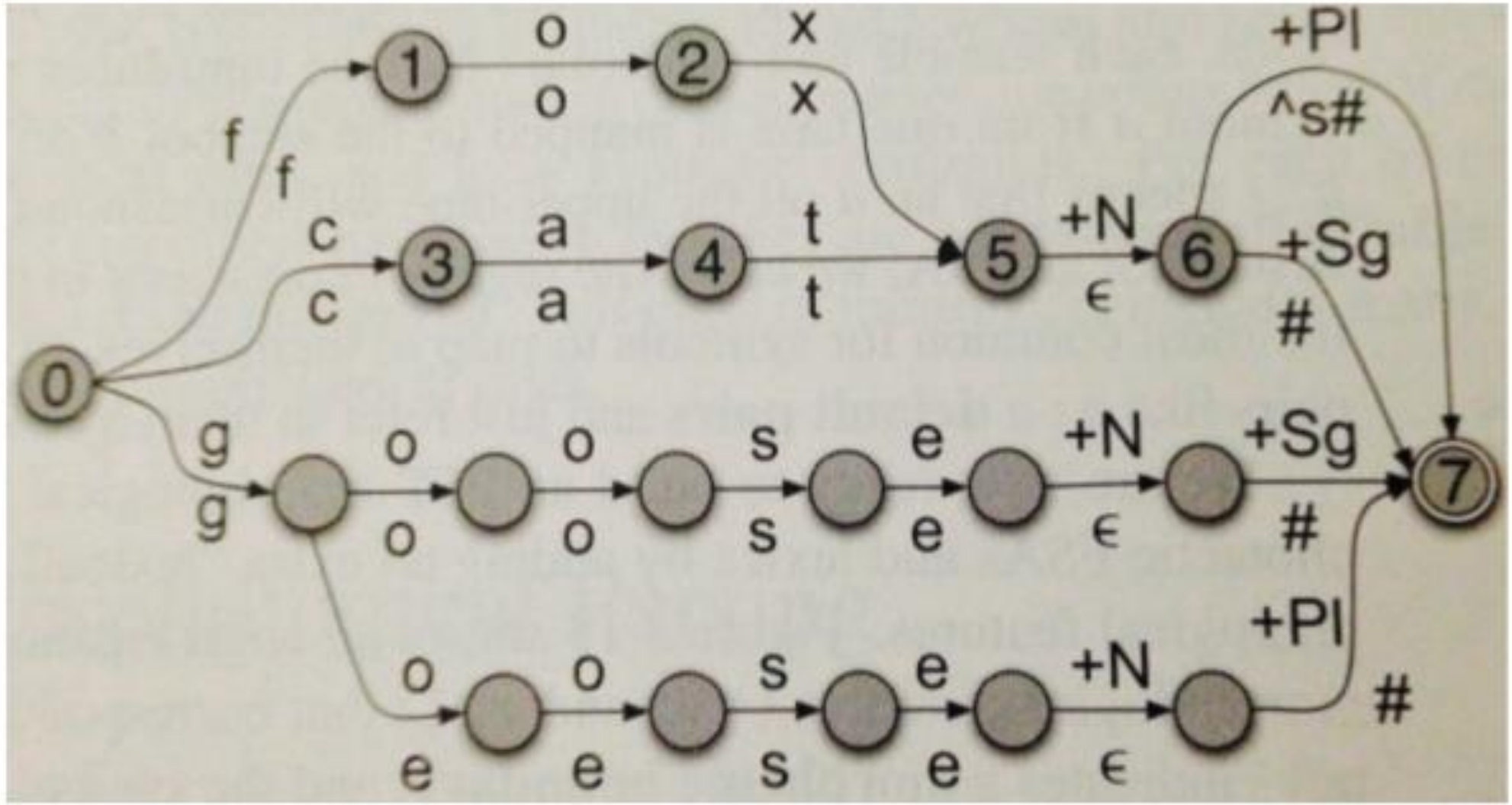
- FSAs can be used for morphological recognizers
- Morphological analyser produce output
  - Input: cats
  - Output: cat +N +PL
- Finite state Transducer to model two level morphology
  - Lexical level: concatenation of morphemes
  - Surface level: actual spelling of the word
  - Alphabets of complex symbols

Consider the problem of translating a lexical form like 'fox+N+PL' into an intermediate form like 'fox ^ s # ', taking account of irregular forms like goose/geese.

We can do this with a transducer of the following schematic form:



We treat each of +N, +SG, +PL as a single symbol.
The 'transition' labelled $+PL : \hat{}s\#$ abbreviates three transitions:
$+PL : \hat{}, \quad \epsilon : s, \quad \epsilon : \#.$


0
1
2
3
4
5
6
7
f
f
o
o
x
x
c
c
a
a
t
t
+N
ϵ
+Pl
^s#
+Sg
#
g
g
o
o
o
o
s
s
e
e
+N
ϵ
+Sg
#
o
e
o
e
s
s
e
e
+N
ϵ
+Pl
#

Current Status

- Learning from data
  - Unsupervised and supervised parsing
- Good Resource
  - SIGMORPHON workshop and associated challenges

SIGMORPHON 2019: 16th SIGM

https://sigmorphon.github.io/workshops/2019/

SIGMORPHON 2019: 16th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology

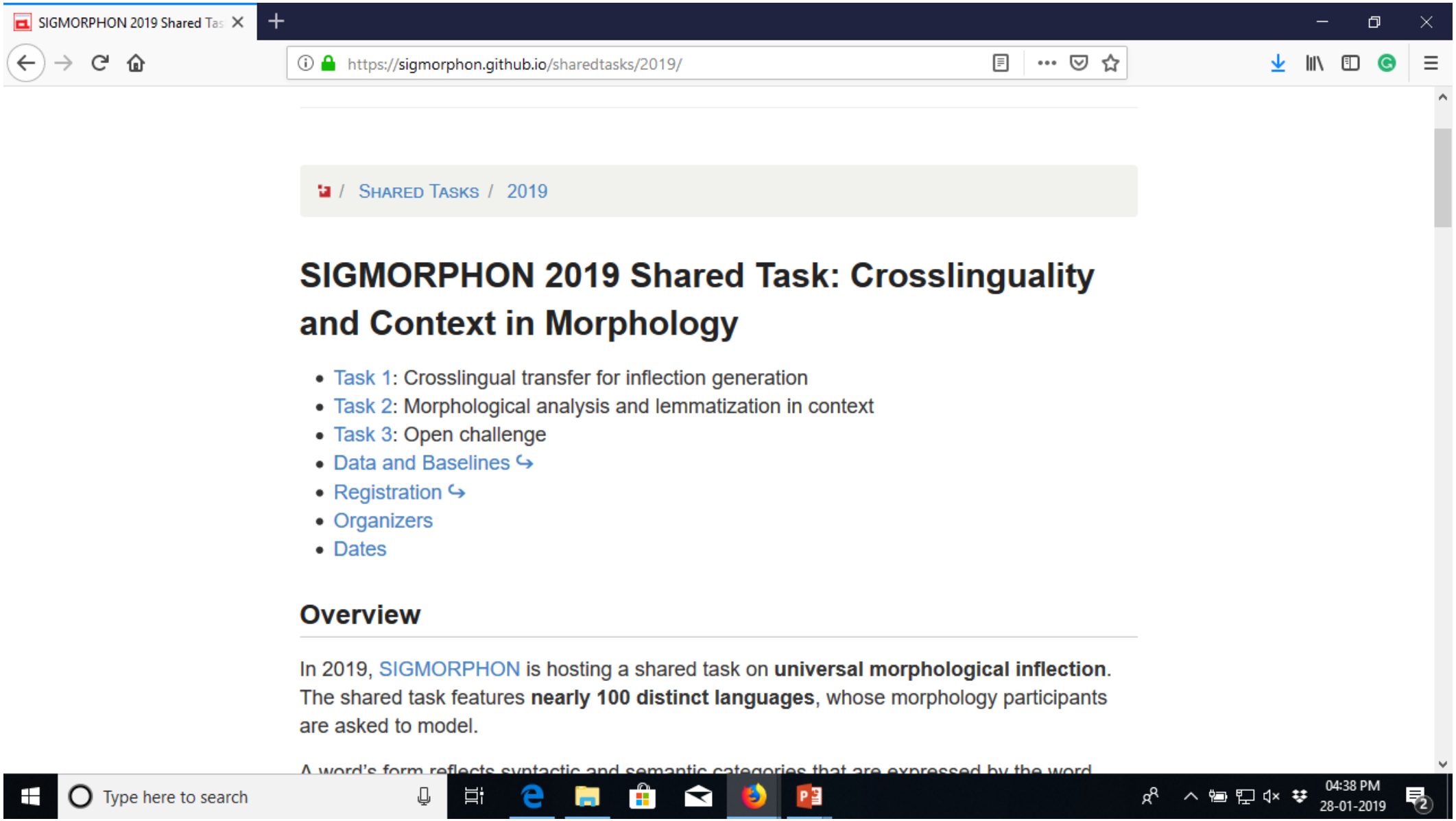
co-located with ACL 2019
Florence, Italy

SIGMORPHON aims to bring together researchers interested in applying computational techniques to problems in morphology, phonology, and phonetics. Work that addresses orthographic issues is also welcome. Papers will be on substantial, original, and unpublished research on these topics, potentially including strong work in progress. Appropriate topics include (but are not limited to) the following as they relate to the areas of the workshop:

- New formalisms, computational treatments, or probabilistic models of existing linguistic formalisms
- Unsupervised, semi-supervised, or machine learning of linguistic knowledge
- Analysis or exploitation of multilingual, multi-dialectal, or diachronic data
- Integration of morphology, phonology, or phonetics with other NLP tasks
- Algorithms for string analysis and manipulation, including finite-state methods
- Models of psycholinguistic experiments
- Approaches to orthographic variation
- Approaches to morphological reinflection
- Corpus linguistics
- Machine transliteration and back-transliteration

Type here to search

04:37 PM
28-01-2019


SIGMORPHON 2019 Shared Tas
https://sigmorphon.github.io/sharedtasks/2019/
/ Shared Tasks / 2019
SIGMORPHON 2019 Shared Task: Crosslinguality and Context in Morphology
Task 1: Crosslingual transfer for inflection generation
Task 2: Morphological analysis and lemmatization in context
Task 3: Open challenge
Data and Baselines
Registration
Organizers
Dates
Overview
In 2019, SIGMORPHON is hosting a shared task on universal morphological inflection. The shared task features nearly 100 distinct languages, whose morphology participants are asked to model.
Type here to search
04:38 PM
28-01-2019

References

- Chapter 3 [SLP: 2nd Ed.]