

Introdução a NLP e IR

Morphology and Finite State Transducers

Alexandre Rademaker¹

FGV/EMAp

August 10, 2022

¹Olga Zamaraeva, University of Washington

Morphology and Phonology

- ▶ Morphology is the study of decomposing words
 - ▶ *Kim sleep-s.*
 - ▶ *Kim is sleep-ing.*
- ▶ Phonology is the study of sound realization based on **environment**
- ▶ Morphology and Phonology are complex fields
- ▶ For our purposes, we will look at their simplified versions (mostly just Morphology)
- ▶ ...in the space of regular languages and FSM

Morphology

- ▶ Lexicon (stems)
- ▶ affixes:
 - ▶ prefix **un-***cover*
 - ▶ suffix *cover*-**ed**
 - ▶ infix *vin***co** (Latin, “I win”, basic root “vic”)
 - ▶ circumfix **ge-***berg*-**te** (Dutch, “mountain range”, from “berg”, mountain)
- ▶ Templatic morphology
- ▶ Isolating languages
- ▶ Fusion
- ▶ Agglutinating

Morphological classes

Morphological Class	Regularly Inflected Verbs			
stem	walk	merge	try	map
-s form	walks	merges	tries	maps
-ing participle	walking	merging	trying	mapping
Past form or -ed participle	walked	merged	tried	mapped

Morphological Class	Irregularly Inflected Verbs		
stem	eat	catch	cut
-s form	eats	catches	cuts
-ing participle	eating	catching	cutting
preterite	ate	caught	cut
past participle	eaten	caught	cut

Morphologically rich languages: Agglutination

Turkish	English
muva ¹ ffak	successful
muva ¹ ffak-iyet	success
muva ¹ ffak-iyet-siz	unsuccessful (without success)
muva ¹ ffak-iyet-siz-les	to become unsuccessful
muva ¹ ffak-iyet-siz-les-tir	to make one unsuccessful
muva ¹ ffak-iyet-siz-les-tiri-ci	maker of unsuccessful ones
muva ¹ ffak-iyet-siz-les-tiri-ci-les	to become a maker of unsuccessful ones
muva ¹ ffak-iyet-siz-les-tiri-ci-les-tir	to make one a maker of unsuccessful ones
...	...

muva¹ffakiyetsizlestiricilestiriveremeyebileceklerimizdenmissinizcesine

like you would be from those we can not easily make a maker of unsuccessful ones

<https://www.rabiaergin.com/turkish-morphology.html>

Morphologically rich languages: Fusion

Russian	English
zlo	evil (noun:neut, nom, sg)
zla	evil (noun:neut, gen, sg)
zloj	evil (adj., masc, nom, sg)
zlaja	evil (adj., fem, nom, sg)
zlogo	evil (adj., masc, gen, sg)
zlost	anger, malevolence (noun:fem,nom,sg)
zlosti	anger, malevolence (noun:fem,gen,sg)
zlostn-ogo	evil,malignant (adj, masc, gen, sg)
zlostnost	evil, malignancy (noun:fem, nom, sg)
zlostnosti	evil, malignancy (noun:fem, gen, sg)

Fusion vs. Agglutination

Fusion:

- ▶ one affix can combine several features (e.g. Case, Number, Gender)
- ▶ affixes tend to *fuse* with each other and with the base, forming a new base
- ▶ once the affix fuses, it “no longer means what it used to”
- ▶ common in Indo-European languages

Agglutination

- ▶ one affix typically “means” one feature
- ▶ affixes maintain their “meaning” in long words
- ▶ common in Turkic languages

Derivational morphology

- ▶ e.g., *un-*, *re-*, *anti-*, *-ism*, *-ist* etc
- ▶ broad range of semantic possibilities, may change part of speech
- ▶ indefinite combinations
e.g., *antiantidisestablishmentarianism*
anti-anti-dis-establish-ment-arian-ism
- ▶ generally semi-productive: e.g., *escapee*, *textee*, *?dropee*, *?snoree*, **cricketee* (* and ?)
- ▶ zero-derivation: e.g. *tango*, *waltz*

Internal structure and ambiguity

Morpheme ambiguity stems and affixes may be individually ambiguous: e.g. *dog* (noun or verb), *+s* (plural or 3persg-verb)

Structural ambiguity: e.g., *shorts* or *short -s*
unionised could be *union -ise -ed* or *un- ion -ise -ed*

Bracketing: *un- ion -ise -ed*

- ▶ *un- ion* is not a possible form, so not *((un- ion) -ise) -ed*
- ▶ *un-* is ambiguous:
 - ▶ with verbs: means 'reversal' (e.g., *untie*)
 - ▶ with adjectives: means 'not' (e.g., *unwise*, *unsurprised*)
- ▶ therefore *(un- ((ion -ise) -ed))*

Using morphological processing in NLP

- ▶ compiling a *full-form* lexicon
- ▶ recognizing and normalizing mini-formal languages (e.g. dates)
- ▶ *stemming* for IR (not linguistic stem)
- ▶ *lemmatization* (often inflections only): finding stems and affixes as a precursor to parsing. *morphosyntax*: interaction between morphology and syntax

Spelling rules

- ▶ English morphology is essentially concatenative
- ▶ irregular morphology — inflectional forms have to be listed
- ▶ regular phonological and spelling changes associated with affixation, e.g.
 - ▶ -s is pronounced differently with stem ending in s, x or z
 - ▶ spelling reflects this with the addition of an e (*boxes* etc)

morphophonology

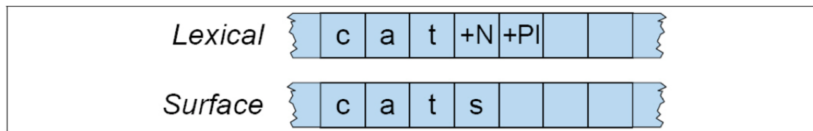
- ▶ in English, description is independent of particular stems/affixes

Lexical requirements for morphological processing

- ▶ affixes, plus the associated information conveyed by the affix
 - ed PAST_VERB
 - ed PSP_VERB
 - s PLURAL_NOUN
- ▶ irregular forms, with associated information similar to that for affixes
 - began PAST_VERB begin
 - begun PSP_VERB begin
- ▶ stems with syntactic categories (plus more)

Morphological parsing

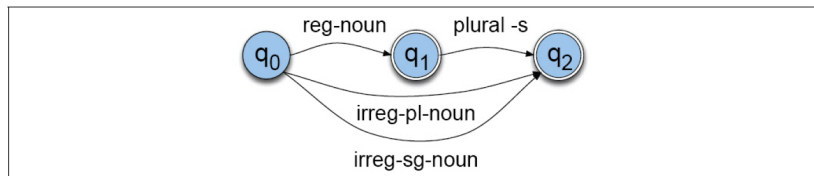
- ▶ Accept/reject strings consisting of morphemes
 - ▶ E.g. for spell-checking
 - ▶ What about just encoding the lexicon as a list of words?
- ▶ Map strings to bundles/sequences of linguistic features
- ▶ Morphological analysis for research support
 - ▶ A parser as a *hypothesis*
 - ▶ Build a parser based on your current understanding of what the language does
 - ▶ Then run it over a corpus of words, see how much it actually parsed and where and why it broke



J&M text, Fig 3.12

FSA for morphological parsing

- ▶ Create FSAs for classes of word stems (word lists).
- ▶ Create FSA for affixes using word classes as stand-ins for the stem word lists.
- ▶ Concatenate FSAs for stems with FSAs for affixes.

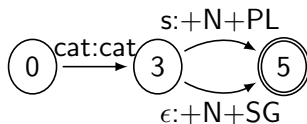


Finite State Transducers

Analyzing (*parsing*) a word morphologically:

cat is *cat* $[+N, +SG]$,

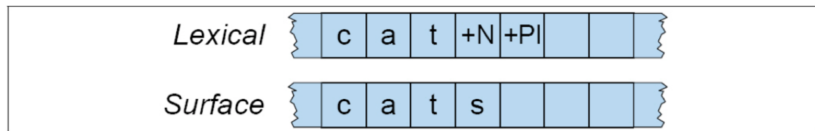
cats is *cat* $[+N, +PL]$



FST and FSA

- ▶ FSA define regular expressions
- ▶ FST define **regular relations**
- ▶ FST use two alphabet sets
- ▶ The **transition** function relates input to states
- ▶ the **output** function relates input to output

Visualizing FST



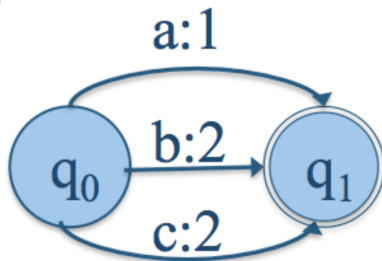
J&M text, Fig 3.12

upper and lower tapes

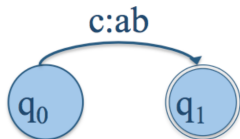
Regular relations

- ▶ Regular language: a set of strings
- ▶ Regular relation: a set of **pairs** of strings:
 - ▶ E.g., Regular relation = $\{a:1, b:2, c:2\}$
 - ▶ Input $\Sigma = \{a,b,c\}$ Output = $\{1, 2\}$

FST:

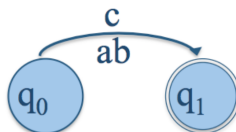


FST conventions

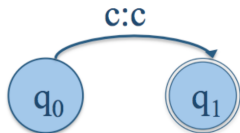


Complex input element

=

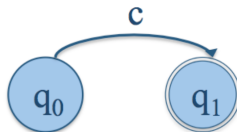


Divided into upper and lower

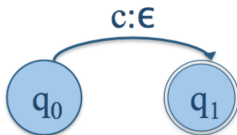


Default pair

=



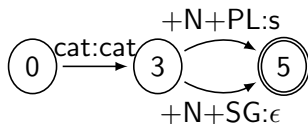
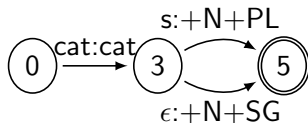
Default pair - shortcut



c on upper, nothing on lower

Inversion

- ▶ Inversion of an FST switches input and output labels
- ▶ Thus we can turn a *parser* into a *generator*
- ▶ Parsing:
 - ▶ Input: cat
 - ▶ Output: cat+N+SG
- ▶ Generating:
 - ▶ Input: cat+N+PL
 - ▶ Output: cats



Composition

► example:

► $T1 = \{a:1\}$

► $T2 = \{1:one\}$

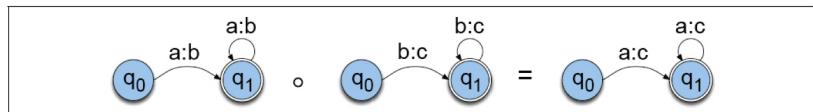
► $T1 \cdot T2 = \{a:one\}$

► $T2(T1(a)) = one$

► Note that order matters: $T1(T2(a)) \neq one$

► Take a minute: what is $T1(T2(a))$?

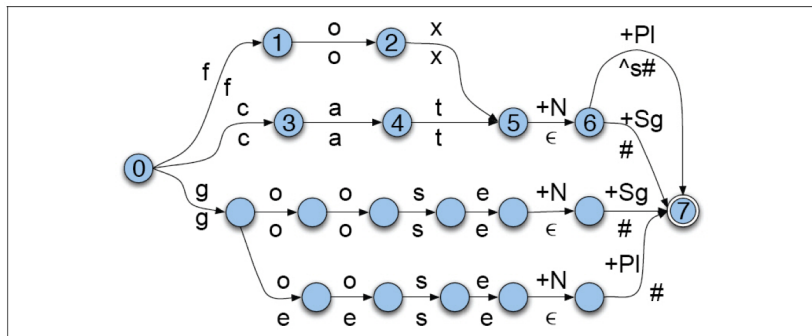
► Composition is used for complex morphological analysis (e.g. semitic languages)



Morphological parsing with FST

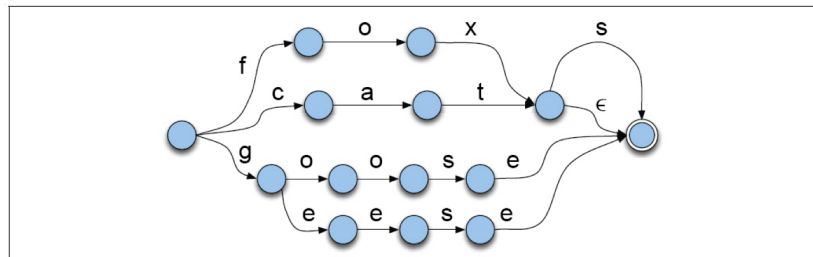
- ▶ A very influential approach since Koskenniemi (1983)
- ▶ AKA “two-level morphology”
- ▶ an example of a *symbolic* approach

Recognizing/analyzing complex words



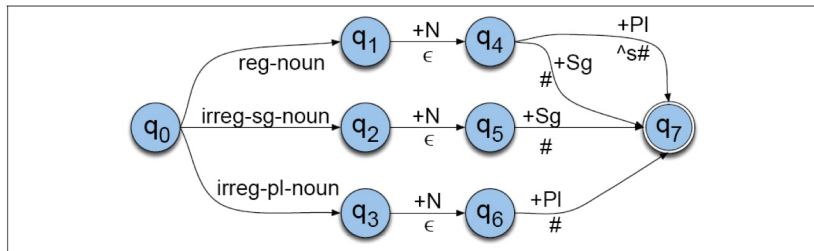
If I only wanted to *recognize*, what would I need?

Recognizing/analyzing complex words



Just an FSA

Identifying word classes



Inflectional classes: Indo-European

LATIN DECLENSIONS

bencrowder.net • Last modified 16 June 2016

1ST DECLENSION

aqua, -ae, F. *water*

	SINGULAR	PLURAL
NOM	aqua	aquae
GEN	aquae	aquarum
DAT	aquae	aquis
ACC	aquam	aquas
ABL	aqua	aquis

3RD DECLENSION I-STEM

civis, -is, M. *citizen*

	SINGULAR	PLURAL
NOM	civis	civēs
GEN	civis	civium
DAT	civī	civibus
ACC	civem	civēs
ABL	cive	civibus

mare, -is, N. *sea*

	SINGULAR	PLURAL
NOM	mare	maria
GEN	maris	marium
DAT	marī	maribus
ACC	mare	maria
ABL	marī	maribus

2ND DECLENSION

servus, -ī, M. *slave*

	SINGULAR	PLURAL
NOM	servus	servī
GEN	servī	servōrum
DAT	servō	servīs
ACC	servum	servōs
ABL	servō	servīs

dōnum, -ī, N. *gift*

	SINGULAR	PLURAL
NOM	dōnum	dōna
GEN	dōnī	dōnōrum
DAT	dōnō	dōnīs
ACC	dōnum	dōna
ABL	dōnō	dōnīs

4TH DECLENSION

fructus, -ūs, M. *fruit*

	SINGULAR	PLURAL
NOM	fructus	fructūs
GEN	fructūs	fructuum
DAT	fructuī	fructibus
ACC	fructum	fructūs
ABL	fructū	fructibus

cornū, -ūs, N. *horn*

	SINGULAR	PLURAL
NOM	cornū	cornua
GEN	cornūs	cornuum
DAT	cornū	cornibus
ACC	cornū	cornua
ABL	cornū	cornibus

3RD DECLENSION

rēx, rēgis, M. *king*

	SINGULAR	PLURAL
NOM	rēx	rēgēs
GEN	rēgis	rēgum
DAT	rēgī	rēgibus
ACC	rēgem	rēgēs
ABL	rēge	rēgibus

corpus, corporis, N. *body*

	SINGULAR	PLURAL
NOM	corpus	corpora
GEN	corporis	corporum
DAT	corporī	corporibus
ACC	corpus	corpora
ABL	corpore	corporibus

5TH DECLENSION

rēs, rei, F. *thing*

	SINGULAR	PLURAL
NOM	rēs	rēs
GEN	rei	rērum
DAT	rei	rēbus
ACC	rem	rēs
ABL	rē	rēbus

diēs, diēi, M. *day*

	SINGULAR	PLURAL
NOM	diēs	diēs
GEN	diēi	diērum
DAT	diēi	diēbus
ACC	diem	diēs
ABL	diē	diēbus

Inflectional classes: Bigger picture

- ▶ Nobody really needs to **look for** inflectional classes in IE languages
 - ▶ particularly not in Latin...
- ▶ But there are many languages in the world for which the exact morphological behavior is not yet fully understood
- ▶ Why is it important that we learn about it and describe it?

Example: Abui [abz] (Alor island in Indonesia)

Form	Gloss	Condition
Ø-	stem alone	I
Ca-	patient (PAT)	II
Ce-	location (LOC)	III
Cee-	benefactive (BEN)	III
Co-	recipient (REC)	IV
Coo-	goal (GOAL)	IV

Table 2: Prefix forms and glosses; Condition I is stem attested bare.

Stem	I	II	III	IV	Class
<i>fil</i> 'pull'	+	+	+	+	A (1111)
<i>kaanra</i> 'complete'	+	+	+	+	A (1111)
<i>kafia</i> 'scratch'	+	-	+	+	B (1011)
<i>yaa</i> 'go'	+	-	+	+	B (1011)
<i>mpang</i> 'think'	+	-	-	+	C (1001)
<i>bel</i> 'pull out'	-	+	+	+	D (0111)
<i>luk</i> 'bend'	-	-	+	+	E (0011)

Table 3: Examples of Abui verb classes

Probabilistic morphological parsing

- ▶ Train a model on a large training corpus
- ▶ E.g. the corpus contains pairs of surface and underlying strings
 - ▶ (like that same cats/cat+N+PL pair)
- ▶ morpheme boundaries can be inferred statistically
- ▶ Neural nets very successful
- ▶ Not an option when there is no training data
- ▶ Other limitations?

Using FSTs

- ▶ FSTs assume *tokenization* (word boundaries) and words split into characters. One character pair per transition!
- ▶ Analysis: return character list with affix boundaries, so enabling lexical lookup.
- ▶ Generation: input comes from stem and affix lexicons.
- ▶ One FST per spelling rule: either compile to big FST or run in parallel.
- ▶ FSTs do not allow for internal structure:
 - ▶ can't model *un- ion -ize -d* bracketing.
 - ▶ can't condition on prior transitions, so potential redundancy

Foma

- ▶ <https://fomafst.github.io/>
- ▶ Tutorial 1 (basic):<https://github.com/mhulden/foma/blob/master/foma/docs/simpleintro.md>
- ▶ Tutorial 2 (more advanced):
<https://fomafst.github.io/morphut.html>

Concluding comments

- ▶ English is an outlier among the world's languages: very limited inflectional morphology.
- ▶ English inflectional morphology hasn't been a practical problem for NLP systems for decades.
- ▶ Limited need for probabilities, small number of possible morphological analyses for a word.
- ▶ Lots of other applications of finite-state techniques: fast, supported by toolkits, good initial approach for very limited systems.