Introduction to Computational Linguistics

Morphology (and a just a little Phonology)
Finite State Transducers

Olga Zamaraeva
Ling/CSE472
University of Washington
April 9, 2019
> how many hours per hw?

- Useful discussion on Canvas about e.g. Assignment 1
  - Thanks to people who post!
- Next time: A guest lecture about intonation in speech
- Assignment 2 is written and will be published soon
- Assignment 5 already published

Finish last slide deck!
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 the environment
  - */t/ → *[dx]* / ķ _ V
  - dx is the flap
  - Which level of language complexity looks at something in an 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 vinco (Latin, “I win”, basic root “vic”)
  - circumfix ge-berg-te (Dutch, “mountain range”, from “berg”, mountain)
- Templatic morphology
- Isolating languages
- Fusion
- Agglutinating languages
Morphological classes

<table>
<thead>
<tr>
<th>Morphological Class</th>
<th>Regularly Inflected Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>stem</td>
<td>walk, merge, try, map</td>
</tr>
<tr>
<td>-s form</td>
<td>walks, merges, tries, maps</td>
</tr>
<tr>
<td>-ing participle</td>
<td>walking, merging, trying, mapping</td>
</tr>
<tr>
<td>Past form or -ed participle</td>
<td>walked, merged, tried, mapped</td>
</tr>
</tbody>
</table>

\[ J \& M \]  

\[ caught \]
Morphological classes

<table>
<thead>
<tr>
<th>Morphological Class</th>
<th>Irregularly Inflected Verbs</th>
</tr>
</thead>
<tbody>
<tr>
<td>stem</td>
<td>eat</td>
</tr>
<tr>
<td>-s form</td>
<td>eats</td>
</tr>
<tr>
<td>-ing participle</td>
<td>eating</td>
</tr>
<tr>
<td>preterite</td>
<td>ate</td>
</tr>
<tr>
<td>past participle</td>
<td>eaten</td>
</tr>
</tbody>
</table>
Morphologically rich languages: Agglutination

<table>
<thead>
<tr>
<th>Turkish</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>muvaffak</td>
<td>successful</td>
</tr>
<tr>
<td>muvaffak-iyet</td>
<td>success</td>
</tr>
<tr>
<td>muvaffak-iyet-siz</td>
<td>unsuccessful (without success)</td>
</tr>
<tr>
<td>muvaffak-iyet-siz-les</td>
<td>to become unsuccessful</td>
</tr>
<tr>
<td>muvaffak-iyet-siz-les-tir</td>
<td>to make one unsuccessful</td>
</tr>
<tr>
<td>muvaffak-iyet-siz-les-tiri-ci</td>
<td>maker of unsuccessful ones</td>
</tr>
<tr>
<td>muvaffak-iyet-siz-les-tiri-ci-les</td>
<td>to become a maker of unsuccessful ones</td>
</tr>
<tr>
<td>muvaffak-iyet-siz-les-tiri-ci-les-tir</td>
<td>to make one a maker of unsuccessful ones</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

muvaffakiyetsizlestiricestiriveremeyebileceklerimizdenmissinizcesine

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

<table>
<thead>
<tr>
<th>Russian</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>zlo</td>
<td>evil (noun: neut, nom, sg)</td>
</tr>
<tr>
<td>zla</td>
<td>evil (noun: neut, gen, sg)</td>
</tr>
<tr>
<td>zloj</td>
<td>evil (adj., masc, nom, sg)</td>
</tr>
<tr>
<td>zlaja</td>
<td>evil (adj., fem, nom, sg)</td>
</tr>
<tr>
<td>zlogo</td>
<td>evil (adj., masc, gen, sg)</td>
</tr>
<tr>
<td>zlost</td>
<td>anger, malevolence (noun: fem, nom, sg)</td>
</tr>
<tr>
<td>zlosti</td>
<td>anger, malevolence (noun: fem, gen, sg)</td>
</tr>
<tr>
<td>zlostn-ogo</td>
<td>evil, malignant (adj, masc, gen, sg)</td>
</tr>
<tr>
<td>zlostnosta</td>
<td>evil, malignancy (noun: fem, nom, sg)</td>
</tr>
<tr>
<td>zlostnosti</td>
<td>evil, malignancy (noun: fem, gen, sg)</td>
</tr>
</tbody>
</table>
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”
  - (see Russian dataset)
  - common in Indo-European languages

- Agglutination
  - one affix typically “means” one feature
  - affixes maintain their “meaning” in long words
  - common in Turkic languages
## Templatic morphology


<table>
<thead>
<tr>
<th>Arabic</th>
<th>English</th>
</tr>
</thead>
<tbody>
<tr>
<td>فعل</td>
<td>کتب</td>
</tr>
<tr>
<td>CCC</td>
<td>کتب</td>
</tr>
<tr>
<td>مفعول</td>
<td>مکتب</td>
</tr>
<tr>
<td>mCCwC</td>
<td>مکتب</td>
</tr>
<tr>
<td></td>
<td>(something written)</td>
</tr>
<tr>
<td></td>
<td>“kttwb”</td>
</tr>
<tr>
<td></td>
<td>(wrote)</td>
</tr>
</tbody>
</table>
Phonology

- Palatalization in Russian:
  - pojehal (I/you/(s)he went)
  - pojeha/-i (they/you/we went)

- Optimality theory (Prince and Smolensky, 1993)
  - an influential approach in today's phonology
  - filtering through all possible forms to get the *optimal* one
  - operate with *violable constraints*
  - e.g. which candidate form violates fewer constraints? pick that one
  - can be modeled as finite-state
  - ...under some conditions
Phonology

- Palatalization in Russian:
  - pojehal (I/you/(s)he went)
  - pojeha\(\hat{i}\)-i (they/you/we went)

\[ c \rightarrow c^j / _-_i \]

- Optimality theory (Prince and Smolensky, 1993)
  - an influential approach in today's phonology
  - filtering through all possible forms to get the *optimal* one
  - operate with *violable constraints*
  - e.g. which candidate form violates fewer constraints? pick that one
  - can be modeled as finite-state
  - ...under some conditions
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

- 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:**

![Finite State Transducer Diagram]

- **Example FST**
  - Initial state: \( q_0 \)
  - Transitions:
    - \( a:1 \) from \( q_0 \) to \( q_1 \)
    - \( b:2 \) from \( q_0 \) to \( q_1 \)
    - \( c:2 \) from \( q_0 \) to \( q_0 \)
FST conventions

Complex input element

Default pair

Default pair - shortcut

c on upper, nothing on lower

Divided into upper and lower
Inversion

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

\begin{align*}
\text{Input: cat} & \quad \xrightarrow{\text{cat:cat}} \quad \text{Output: cat+N+SG} \\
\text{Input: cat+N+PL} & \quad \xrightarrow{\text{cat:cat}} \quad \text{Output: cats}
\end{align*}
Composition

- example:
  - $T_1 = \{a:1\}$
  - $T_2 = \{1:\text{one}\}$
  - $T_1 \cdot T_2 = \{a:\text{one}\}$
  - $T_2(T_1(a)) = \text{one}$

- Note that order matters: $T_1(T_2(a)) \neq \text{one}$

- Take a minute: what is $T_1(T_2(a))$?

- 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

End of class
Identifying word classes
## Inflectional classes: Indo-European

### Latin declensions

#### 1st declension
- **aqua**, -ae, f. water
  - **Singular**
    - **Nom:** aqua
    - **Gen:** aquae
    - **Dat:** aquae
    - **Acc:** aquam
    - **Abl:** aqua
  - **Plural:**
    - **Nom:** aquae
    - **Gen:** aquarum
    - **Dat:** aquae
    - **Acc:** aquae
    - **Abl:** aquae

#### 2nd declension
- **servus**, -i, m. slave
  - **Singular**
    - **Nom:** servus
    - **Gen:** servi
    - **Dat:** servī
    - **Acc:** servum
    - **Abl:** servō
  - **Plural**
    - **Nom:** servi
    - **Gen:** servorum
    - **Dat:** servi
    - **Acc:** servōs
    - **Abl:** servōs

#### 3rd declension
- **rēx**, rēgis, m. king
  - **Singular**
    - **Nom:** rēx
    - **Gen:** rēgis
    - **Dat:** rēgī
    - **Acc:** rēgem
    - **Abl:** rēge
  - **Plural**
    - **Nom:** corporis
    - **Gen:** corporī
    - **Dat:** corporibus
    - **Acc:** corpore
    - **Abl:** corporibus

- **corpus**, corporis, n. body
  - **Singular**
    - **Nom:** corpus
    - **Gen:** corporī
    - **Dat:** corporibus
    - **Acc:** corporum
    - **Abl:** corporibus

#### 3rd declension I-stem
- **civis**, -is, m. citizen
  - **Singular**
    - **Nom:** civis
    - **Gen:** civī
    - **Dat:** civīm
    - **Acc:** civem
    - **Abl:** cive
  - **Plural**
    - **Nom:** civēs
    - **Gen:** civium
    - **Dat:** civibus
    - **Acc:** civibus
    - **Abl:** civibus

- **mare**, -is, n. sea
  - **Singular**
    - **Nom:** mare
    - **Gen:** mari
    - **Dat:** mari
    - **Acc:** mariam
    - **Abl:** mario
  - **Plural**
    - **Nom:** maria
    - **Gen:** marium
    - **Dat:** maribus
    - **Acc:** maria
    - **Abl:** maribus

#### 4th declension
- **frōctus**, -ūs, m. fruit
  - **Singular**
    - **Nom:** frōctus
    - **Gen:** frōctūs
    - **Dat:** frōctūm
    - **Acc:** frōctum
    - **Abl:** frōctūs
  - **Plural**
    - **Nom:** frōctūs
    - **Gen:** frōctum
    - **Dat:** frōctibus
    - **Acc:** frōctibus
    - **Abl:** frōctibus

- **cornū**, -ūs, n. horn
  - **Singular**
    - **Nom:** cornū
    - **Gen:** cornūs
    - **Dat:** cornūs
    - **Acc:** cornua
    - **Abl:** cornibus
  - **Plural**
    - **Nom:** cornua
    - **Gen:** cornuum
    - **Dat:** cornibus
    - **Acc:** cornua
    - **Abl:** cornibus

#### 5th declension
- **rēs**, rei, f. thing
  - **Singular**
    - **Nom:** rēs
    - **Gen:** rērum
    - **Dat:** rebus
    - **Acc:** rem
    - **Abl:** re
  - **Plural**
    - **Nom:** diēs
    - **Gen:** diērum
    - **Dat:** diēbus
    - **Acc:** diem
    - **Abl:** diē

- **diēs**, diēi, m. day
  - **Singular**
    - **Nom:** diēs
    - **Gen:** diēi
    - **Dat:** diērum
    - **Acc:** diem
    - **Abl:** diē
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)

<table>
<thead>
<tr>
<th>Form</th>
<th>Gloss</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Φ-</td>
<td>stem alone</td>
<td>I</td>
</tr>
<tr>
<td>Ca-</td>
<td>patient (PAT)</td>
<td>II</td>
</tr>
<tr>
<td>Ce-</td>
<td>location (LOC)</td>
<td>III</td>
</tr>
<tr>
<td>Cee-</td>
<td>benefactive (BEN)</td>
<td>III</td>
</tr>
<tr>
<td>Co-</td>
<td>recipient (REC)</td>
<td>IV</td>
</tr>
<tr>
<td>Coo-</td>
<td>goal (GOAL)</td>
<td>IV</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Stem</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>fil ‘pull’</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>A (1111)</td>
</tr>
<tr>
<td>kaanra ‘complete’</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>A (1111)</td>
</tr>
<tr>
<td>kafia ‘scratch’</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>B (1011)</td>
</tr>
<tr>
<td>yaa ‘go’</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>B (1011)</td>
</tr>
<tr>
<td>mpang ‘think’</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>C (1001)</td>
</tr>
<tr>
<td>bel ‘pull out’</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>D (0111)</td>
</tr>
<tr>
<td>luk ‘bend’</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>E (0011)</td>
</tr>
</tbody>
</table>

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?
foma (Assignment 2)

- https://fomafst.github.io/
- Tutorial 1 (basic): https://github.com/mhulden/foma/blob/master/foma/docs/simpleintro.md