Morphological parsing

- Breaking down words into components and building a structured representation.

  - English:
    - cats → cat +N +Pl
    - caught → catch +V +Past

  - Spanish:
    - vino (came) → venir +V + Perf +3P + Sg
    - vino (wine) → vino +N + Masc + Sg
Morphological parsing

- Exercise: Give an example of an ambiguous word in German and parse two of its meanings into parts.
Exercise: Think of one example of an ambiguous word in German and parse two of its meanings.

- **Weiß:**
  - weiß (white) → white + Adj
  - weiß (know) → to know + V + Present +1P/3P + Sg
Morphological parsing

- Surface segmentation: sequence of substrings whose concatenation is the entire word
  - achievability → achiev + abil + ity

- Canonical segmentation: sequence of standardized segments
  - achievability → achieve + able + ity
Stemming vs. Lemmatization

• Stemming: stripping off word endings (rule-based)
  - foxes $\rightarrow$ fox
  - going $\rightarrow$ go

• Lemmatization: mapping the word to its lemma (lexicon-based)
  - sang, sung $\rightarrow$ sing
  - going, went, goes $\rightarrow$ go
Motivation for morphological parsing

- Information retrieval
  - Normalize verb tenses, plurals, grammar cases

- Machine translation
  - Translation based on the stem
Morphological parsing

• Resources
  - Lexicon
    • List of all stems and affixes
  - Morphotactics
  - Orthographic rules
Morphological parsing

- Resources
  - Lexicon
  - Morphotactics
    - A model of morpheme ordering in a word
    - e.g., plurals are suffixes in English
  - Ortographic rules
Morphological parsing

- Resources
  - Lexicon
  - Morphotactics
  - Orthographic rules
    - Rules for changing in the words when combining morphemes
    - e.g., city → cities
Finite-state automata (FSA)

- FSAs are composed of
  - Vertices (nodes)
  - Arcs (links)

string?
Finite-state automata (FSA)

- FSAs are composed of
  - Vertices (nodes)
  - Arcs (links)
Finite-state lexicon

- Finite state automata (FSA) for English nominal inflection (same word category)

Check example for verbal inflections in Jurafski & Martin book.
Finite-state lexicon

- FSA for derivational morphology (distinct word categories)

\[
\begin{align*}
q_0 & \xrightarrow{\varepsilon} q_1 & \text{un-} & \xrightarrow{\text{adj-root}} q_2 & \text{-er -est -ly} & q_2
\end{align*}
\]

Adjectives:
- cool-er
- small-er
- un-usual-ly
...
Finite-state lexicon

- Exercise: Is it possible to create adjectives that do not exist?
Finite-state lexicon

- Exercise: Is it possible to create adjectives that do not exist?

Incorrect adjectives:
- un-small-er
- orange-er
- small-ly

Solution: classes of roots ($\text{adj-root}_1$, $\text{adj-root}_2$, etc.)
Finite-state transducers (FST)

- FST is a type of FSA which maps between two sets of symbols.
- It is a two-tape automaton that recognizes or generates pairs of strings, one from each type.
- FST defines relations between sets of strings.
Finite-state transducers for NLP

- FST as recognizer
  - Takes a pair of strings and accepts or rejects them
- FST as generator
  - Outputs a pair of strings for a language
- FST as translator
  - Reads a string and outputs another string
  - Morphological parsing: letters (input); morphemes (output)
- FST as relater
  - Computes relations between sets
FST for morphological parsing

• Two tapes
  - Upper (lexical) tape: input alphabet $\Sigma$
    • cat +N +Pl
  - Lower (surface) tape: output alphabet $\Delta$
    • cats
FST for morphological parsing

- goose/geese: g:g o:e o:e s:s e:e
  - Feasible pairs (e.g., o:e) vs. default pairs (g:g)
FST and orthographical rules

- Plural of “fox” is “foxes” not “foxs”
- Consonant double: beg/begging
- E deletion: make/making
- E insertion: watch/watches
- Y replacement: try/tries
- K insertion: panic/panicked
FST and orthographical rules

- Lexical: foxes +N +Pl
- Intermediate: fox^s#
- Surface: foxes
Combination of FST lexicon and rules for generation

Lexical \(\rightarrow \text{fox +N +Pl}\) \n
Intermediate \(\rightarrow \text{fox}^{\text{s##}}\) \n
LEXICON-FST \n
FST\(_1\) \ldots \text{FST}\(_n\) (orthographical rules) \n
Surface \(\rightarrow \text{foxes}\)
FST lexicon and rules

• Disambiguation
  
  – For some cases, it requires external evidences:
    • I saw two foxes yesterday. (fox +N +Pl)
    • That trickster foxes me every time! (fox +V +3SG)

  – But it can handle local ambiguity (intersection & composition)
    – „asses“ vs. „assess“
FST lexicon and rules

- Intersection & Composition

\[
\text{FST}_{A} (=\text{FST}_{1} \ldots \text{FST}_{n})
\]

- compose

\[
\text{intersect}
\]
Porter Stemmer (Lexicon-Free FST)

- Popular for information retrieval and text categorization tasks
- It is based on a series of simple cascade rules
  - ATIONAL → ATE (relational → relate)
  - ING → ε (motoring → motor)
  - SSES → SS (grasses → grass)
- But it commits many errors:
  - ORGANIZATION → ORGAN
  - DOING → DOE

(http://tartarus.org/martin/PorterStemmer/)
WordNet lemmatizer

- Uses WordNet to find the stem of a word.

WordNet
A lexical database for English

Noun
- S: (n) **small** (the slender part of the back)
- S: (n) **small** (a garment size for a small person)

Adjective
- S: (adj) **small**, **little** (limited or below average in number or quantity or magnitude or extent) "a little dining room"; "a little house"; "a small car"; "a little (or small) group"
- S: (adj) **minor**, **modest**, **small**, **small-scale**, **pocket-size**, **pocket-sized** (relatively moderate, limited, or small) "a small business"; "a newspaper with a modest circulation"; "small-scale plans"; "a pocket-size country"
- S: (adj) **little**, **small** (of children and animals) young, immature) "what a big little boy you are"; "small children"
- S: (adj) **small** (slight or limited; especially in degree or intensity or scope) "a series of death struggles with small time in between"
- S: (adj) **humble**, **low**, **lowly**, **modest**, **small** (low or inferior in station or quality) "a humble cottage"; "a lowly parish priest"; "a modest man of the people"; "small beginnings"
- S: (adj) **little**, **minuscule**, **small** (lowercase) "little a"; "small a"; "e.e.cummings's poetry is written all in minuscule letters"
- S: (adj) **little**, **small** (of a voice) faint) "a little voice"; "a still small voice"
- S: (adj) **small** (have fine or very small constituent particles) "a small misty rain"
- S: (adj) **modest**, **small** (not large but sufficient in size or amount) "a modest salary"; "modest inflation"; "helped in my own small way"
- S: (adj) **belittled**, **diminished**, **small** (made to seem smaller or less (especially in worth)) "her comments made me feel small!"

Adverb
- S: (adv) **small** (on a small scale) "think small"
Use case: BioLemmatizer

- Based on MorphAdorner
- Enriched with biomedical-specific resources (lexicon)
Machine learning-based morphological parsing

- Based on available training data, e.g., from the Morpho Challenge

<table>
<thead>
<tr>
<th>Language</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>baby-sitters</td>
</tr>
<tr>
<td></td>
<td>indoctrinated</td>
</tr>
<tr>
<td></td>
<td>baby_N s_i t V</td>
</tr>
<tr>
<td></td>
<td>er_s +PL</td>
</tr>
<tr>
<td></td>
<td>in_p -doctrine_N</td>
</tr>
<tr>
<td></td>
<td>ate_s +PAST</td>
</tr>
<tr>
<td>Finnish</td>
<td>linuxiin</td>
</tr>
<tr>
<td></td>
<td>makaronia</td>
</tr>
<tr>
<td></td>
<td>linux_N +ILL</td>
</tr>
<tr>
<td></td>
<td>makaroni_N +PTV</td>
</tr>
<tr>
<td>German</td>
<td>choreographische</td>
</tr>
<tr>
<td></td>
<td>zurueckzubehalten</td>
</tr>
<tr>
<td></td>
<td>choreographie_N</td>
</tr>
<tr>
<td></td>
<td>isch +ADJ-e</td>
</tr>
<tr>
<td></td>
<td>zurueck_B zu be</td>
</tr>
<tr>
<td></td>
<td>halt_V +INF</td>
</tr>
<tr>
<td>Turkish</td>
<td>kontrole</td>
</tr>
<tr>
<td></td>
<td>popUlerliGini</td>
</tr>
<tr>
<td></td>
<td>kontrol +DAT</td>
</tr>
<tr>
<td></td>
<td>popUler +DER_lHg</td>
</tr>
<tr>
<td></td>
<td>+POS2S +ACC,</td>
</tr>
<tr>
<td></td>
<td>popUler +DER_lHg</td>
</tr>
<tr>
<td></td>
<td>+POS3 +ACC3</td>
</tr>
</tbody>
</table>

(http://morpho.aalto.fi/events/morphochallenge2010/datasets.shtml)
Conditional random fields (CRF)

- A discriminative undirected probabilistic graphical model for structured prediction

Conditional random fields (CRF)

- Morphology parsing as a classification task
- Linear-chain CRF is to exploit the dependencies between the output variables using a chain structured undirected graph

\[
\text{drivers} \rightarrow \text{driv + er + s}
\]

```
START  B  M  M  E  B  E  S  STOP
<w>    d r i v e r s  </w>
```
Conditional random fields (CRF)

- Features:
  - Left and right substrings, e.g., \{v, iv, riv, driv, \textless w\textgreater driv\} and \{e, er, ers, ers</w>\} for „driver“
  - Rules, such as the following for -ed words („talked“, „played“ and „speed“):
    - position $t$ is a segment boundary if its right context is $ed$ and the left context is not $spe$. 

(http://www.aclweb.org/anthology/W13-3504)
Recurrent neural networks language model (RNNLM)

- Recurrent neural networks (RNN) is a class of NN in which connections between the units form a directed cycle.
- It makes use of sequential information.
- It does not assume independence between input and output.

A recurrent neural network and the unfolding in time of the computation involved in its forward computation. Source: Nature
Long short-term memory (LSTM)

- It is a special kind of RNN that connects previous information to the present task.
- It is capable to learn long-term dependencies and is suitable for sequence learning tasks.

![Diagram of LSTM](http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

*The repeating module in a standard RNN contains a single layer.*
Long short-term memory (LSTM)

- LSTMs usually have four interacting layers (but there are many variations of the architecture).

The repeating module in an LSTM contains four interacting layers.
LSTM for morphological segmentation

- Instead of relying heavily on linguistic knowledge (e.g., CRFs), the NN automatically learns the structure of input sequences and predict morphological boundaries for words.
- Series of window-based LSTM architectures for morphological segmentation.
- Predictions based on both past and future inputs, i.e., left and right neighbors.
- Classification task based on \{B,M,E,S\} classes:

\[
\begin{array}{cccccccc}
<w> & a & c & t & o & r & s & </w> \\
\text{START} & B & M & E & B & E & S & \text{STOP}
\end{array}
\]
LSTM for morphological segmentation

- Simple Window LSTM model considers a new character window and label independently at each step.

Figure 2: Window LSTM Model
LSTM for morphological segmentation

- Multi-Window LSTM model processes an entire word jointly.

![Multi-Window LSTM model diagram](http://iiis.tsinghua.edu.cn/~weblt/papers/window-lstm-morph-segmentation.pdf)

Figure 3: Multi-Window LSTM model
LSTM for morphological segmentation

- The model first makes a forward pass to process the sequence in the normal order.
- Then adopts an additional backward pass to process it in reverse order.
- With these bidirectional passes, the network is able to learn even more fine-grained features from the input words and corresponding label sequences.

Figure 4: Bidirectional Multi-Window LSTM model
Summary

- Morphological parsing
- Methods:
  - Finite-state automata & lexicon
  - Finite-state transistors
  - Machine learning
    - Training data & features
    - Sequential algorithms, e.g., CRFs and RNN-LSTM
Exercise

- Project:
  - Could morphological parsing support your project?
  - Choose a morphological parser and try it in your document collection. Manually check a sample of the results.
Tools

- **FS-based morpha**: [https://github.com/knowitall/morpha](https://github.com/knowitall/morpha)
- **WordNet lemmatizer**: [http://search.cpan.org/~tpederse/WordNet-Similarity-2.05/lib/WordNet/stem.pm](http://search.cpan.org/~tpederse/WordNet-Similarity-2.05/lib/WordNet/stem.pm)
- **MorphAdoner**: [http://morphadorner.northwestern.edu/morphadorner/](http://morphadorner.northwestern.edu/morphadorner/)
- **CLEAR parser**: [https://code.google.com/archive/p/clearparser/](https://code.google.com/archive/p/clearparser/)
- **NLP DotNet (on-line)**: [http://nlpdotnet.com/services/Morphparser.aspx](http://nlpdotnet.com/services/Morphparser.aspx)
- **Morphisto (German)**: [https://code.google.com/archive/p/morphisto/](https://code.google.com/archive/p/morphisto/)
Further reading

- NLP book: Chapter 3
- DL book: Chapter 10
  - http://www.deeplearningbook.org/contents/rnn.html
- Other references:
  - BioLemmatizer (good overview of various lemmatizers): https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3359276/
  - morpha: http://dl.acm.org/citation.cfm?id=973922