Lecture 6
Sequence Tagging

LTAT.01.001 – Natural Language Processing
Kairit Sirts (kairit.sirts@ut.ee)
20.03.2020
Plan for today

- The sequence tagging problem
  - POS tagging
  - Morphological tagging
  - Named Entity Recognition
- CONLL-U data format
- Sequence tagging methods
- Evaluating Sequence tagging tasks
Sequence tagging problem

- Many NLP problems can be viewed as sequence tagging/labelling.
- Each token in a sequence is assigned a tag/label.
- Labels of tokens are dependent on the labels of other tokens in the sequence, particularly their neighbors.

The cat sat on the mat
Sequence tagging task examples

- POS (part-of-speech) tagging
- Named entity recognition
- Chinese word segmentation
- Shallow parsing (also called chunking)
- Semantic role labeling
POS tagging
POS tags

- Part-of-speech tags, syntactic categories, word classes
- POS tags give information about the word and its neighbors
- Useful for many other NLP tasks: information extraction, syntactic parsing, information retrieval, summarization

Janet will back the bill

Proper noun Modal verb Verb Article Noun
Main POS categories

- Nouns
  - Common nouns – things (chair), events (lecture), abstractions (justice), verb-like terms (swimming) etc
  - Proper nouns – proper names of people (John), countries (Estonia), organizations (University of Tartu) etc

- Verbs – words referring to actions and processes (to draw, to ponder)

- Adjectives – words describing properties or qualities (black, young)

- Adverbs – words modifying (mostly) verbs
  
  Unfortunately, John walked home extremely slowly yesterday
Open and closed class words

Open class words
- Nouns
- Verbs
- Adjectives
- Adverbs

Closed class words
- **Prepositions**: on, under, over
- **Determiners**: a, an, the
- **Pronouns**: she, who, I, others
- **Conjunctions**: and, but, or, as
- **Auxiliary verbs**: can, may, are
- **Particles**: up, down, on, off
- **Numerals**: one, two, first
## Universal POS tags

<table>
<thead>
<tr>
<th>Open class words</th>
<th>Closed class words</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADJ</td>
<td>ADP</td>
<td>PUNCT</td>
</tr>
<tr>
<td>ADV</td>
<td>AUX</td>
<td>SYM</td>
</tr>
<tr>
<td>INTJ</td>
<td>CCONJ</td>
<td>X</td>
</tr>
<tr>
<td>NOUN</td>
<td>DET</td>
<td></td>
</tr>
<tr>
<td>PROPN</td>
<td>NUM</td>
<td></td>
</tr>
<tr>
<td>VERB</td>
<td>PART</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PRON</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SCONJ</td>
<td></td>
</tr>
</tbody>
</table>

Source: [https://universaldependencies.org/u/pos/index.html](https://universaldependencies.org/u/pos/index.html)
POS tagging

- UD POS tags

- WSJ POS tags
Morphological tagging
Morphology

- Morphology studies the internal structure of words

availability NOUN +nom +pl

availabilities

avail +able +ity +es

avail_abil_iti_es
Morphemes

- Morphemes are the smallest units of language that carry a semantic meaning
Lexical and grammatical morphemes

- Lexical morphemes carry themselves a semantic meaning. Most of them can stand on their own.
  - Boy, table, yellow, run, waste etc

- Grammatical morphemes cannot stand on their own. Their role is to modify the meaning of a lexical morpheme or specify the relationships between lexical morphemes
  - -s, -ing, -able, at, in, on
The role of grammatical morphemes

- Overlap with syntax and semantics

“I put the book on the table” vs “Panin raamatu lauale” (in Estonian)

“Giraffe bit the zebra” vs “Kaelkirjak hammustas sebrat” or “Sebrat hammustas kaelkirjak”
Morphological analysis

The task of finding all possible morphological tags for a word

A morphological analysis consists of:
- Lemma/stem
- POS
- Morphological attributes/features

Question:
Should morphological analysis be done in context or can you analyse each word in isolation?

(something) has lasted
kestnud
kest+nud // _V_ nud, //
kest=nu+d // _S_ pl n, //
kest=nud+0 // _A_ //
kest=nud+0 // _A_ sg n, //
kest=nud+d // _A_ pl n, //
Morphological analysis

- The task of finding all possible morphological tags for a word
- A morphological analysis consists of:
  - Lemma/stem
  - POS
  - Morphological attributes/features

**Question:**
Should morphological analysis be done in context or can you analyse each word in isolation?

**Answer:** Can be done in isolation

(something) has lasted

kestnud
kest+nud // _V_ nud, //
kest=nu+d // _S_ pl n, //
kest=nud+0 // _A_ //
kest=nud+0 // _A_ sg n, //
kest=nud+d // _A_ pl n, //
Morphological tagging/disambiguation

- **Morphological tagging**
  - Predict the morphological tag for each word in context choosing from all possible tags

- **Morphological disambiguation**
  - First perform morphological analysis
  - Then perform morphological disambiguation by choosing for each word the most appropriate analysis

- **“Soft” morphological disambiguation**
  - First perform morphological analyses and then use the analyses to influence the tagging decisions.
Morphological tagging

- Can be treated as a sequence tagging task
- Conceptually very similar to POS tagging – instead of POS tags there are now morphological tags
- The UD (universal dependencies) datasets also contain morphological analyses for many languages
### Morphological tagset sizes in UD corpora

<table>
<thead>
<tr>
<th>Language</th>
<th>Tagset Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arabic</td>
<td>349</td>
</tr>
<tr>
<td>Chinese</td>
<td>31</td>
</tr>
<tr>
<td>Czech</td>
<td>2630</td>
</tr>
<tr>
<td>English</td>
<td>117</td>
</tr>
<tr>
<td>Estonian</td>
<td>662</td>
</tr>
<tr>
<td>Finnish</td>
<td>2052</td>
</tr>
<tr>
<td>French</td>
<td>228</td>
</tr>
<tr>
<td>German</td>
<td>684</td>
</tr>
<tr>
<td>Korean</td>
<td>11</td>
</tr>
<tr>
<td>Russian</td>
<td>693</td>
</tr>
</tbody>
</table>
### Universal morphological features

<table>
<thead>
<tr>
<th>Lexical features</th>
<th>Inflectional features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><em>Nominal</em></td>
</tr>
<tr>
<td>PronType</td>
<td>Gender</td>
</tr>
<tr>
<td>NumType</td>
<td>Animacy</td>
</tr>
<tr>
<td>Poss</td>
<td>NounClass</td>
</tr>
<tr>
<td>Reflex</td>
<td>Number</td>
</tr>
<tr>
<td>Foreign</td>
<td>Case</td>
</tr>
<tr>
<td>Abbr</td>
<td>Definite</td>
</tr>
<tr>
<td></td>
<td>Degree</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The role of computational morphology in NLP

- Still largely an unexplored area. Why?
- Intuitively, modeling morphology should help to reduce the vocabulary sparsity problems

Morphological agreement:
- For instance, verbs and nouns must agree in number
- In German: der Mann geht vs die Männer gehen (the man goes vs men go)

Potentially could be useful for many downstream tasks:
- Machine translation
- Natural language generation
- Language modeling (for speech recognition)
Named Entity Recognition
NER: Named entity recognition

- Find all **named entities** in the text and label their types

At the party **Thursday night** at **Chateau Marmont**, **Cate Blanchet** barely made it up in the elevator.
Types of named entities

<table>
<thead>
<tr>
<th>Type</th>
<th>Tag</th>
<th>Sample Categories</th>
<th>Example sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>People</td>
<td>PER</td>
<td>people, characters</td>
<td>Turing is a giant of computer science.</td>
</tr>
<tr>
<td>Organization</td>
<td>ORG</td>
<td>companies, sports teams</td>
<td>The IPCC warned about the cyclone.</td>
</tr>
<tr>
<td>Location</td>
<td>LOC</td>
<td>regions, mountains, seas</td>
<td>The Mt. Sanitas loop is in Sunshine Canyon.</td>
</tr>
<tr>
<td>Geo-Political</td>
<td>GPE</td>
<td>countries, states, provinces</td>
<td>Palo Alto is raising the fees for parking.</td>
</tr>
<tr>
<td>Entity</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Facility</td>
<td>FAC</td>
<td>bridges, buildings, airports</td>
<td>Consider the Tappan Zee Bridge.</td>
</tr>
<tr>
<td>Vehicles</td>
<td>VEH</td>
<td>planes, trains, automobiles</td>
<td>It was a classic Ford Falcon.</td>
</tr>
</tbody>
</table>
NER

- Types of named entities correspond to proper names
- Often also **temporal expressions** and **numeric expressions** are extracted

- Types of named entities may depend on the domain
  - Names of genes and proteins in biomedical texts
  - Names of college courses
  - Commercial products
  - Works of art
Citing high fuel prices, [United Airlines ORG] said [Friday TIME] it has increased fares by [$6 MONEY] per round trip on flights to some cities also served by lowercost carriers. [American Airlines ORG], a unit of [AMR Corp. ORG], immediately matched the move, spokesman [Tim Wagner PER] said. [United ORG], a unit of [UAL Corp. ORG], said the increase took effect [Thursday TIME] and applies to most routes where it competes against discount carriers, such as [Chicago LOC] to [Dallas LOC] and [Denver LOC] to [San Francisco LOC].
Named entity recognition

1. Find text segments corresponding to entities
   ● Segmentation ambiguity

2. Classify entities
   ● Type ambiguity

<table>
<thead>
<tr>
<th>Name</th>
<th>Possible Categories</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Person, Location, Political entity, Organization, Vehicle</td>
</tr>
<tr>
<td>Downing St.</td>
<td>Location, Organization</td>
</tr>
<tr>
<td>IRA</td>
<td>Person, Organization, Monetary Instrument</td>
</tr>
<tr>
<td>Louis Vuitton</td>
<td>Person, Organization, Commercial Product</td>
</tr>
</tbody>
</table>
Type ambiguity

- [Washington \_PER\_] was born into slavery on the farm of James Burroughs.
- [Washington \_ORG\_] went up 2 games to 1 in the four-game series.
- Blair arrived in [Washington \_LOC\_] for what may well be his last state visit.
- In June, [Washington \_GPE\_] passed a primary seatbelt law.
- The [Washington \_VEH\_] had proved to be a leaky ship, every passage I made …
NER as Sequence Tagging task

- The standard approach for NER is word-by-word sequence tagging

- The text is tagged with BIO or IO tagging
  - B – start of a named entity
  - I – inside of a named entity
  - O – outside of a named entity

- The tag is coupled with a type label:
  - B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC etc
NER labelling

[American Airlines \text{ORG}], a unit of [AMR Corp. \text{ORG}], immediately matched the move, spokesman [Tim Wagner \text{PER}] said.

- O – outside
- B-ORG – beginning of the Organization entity
- I-ORG – inside of the Organization entity
- B-PER – beginning of the Person entity
- I-PER – inside of the Person entity
- etc
NER labelling

\[\text{American Airlines}_{\text{ORG}}, \text{ a unit of AMR Corp.}_{\text{ORG}}, \text{ immediately matched the move, spokesman Tim Wagner}_{\text{PER}} \text{ said.}\]

<table>
<thead>
<tr>
<th>Word</th>
<th>BIO label</th>
<th>IO label</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>B-ORG</td>
<td>I-ORG</td>
</tr>
<tr>
<td>Airlines</td>
<td>I-ORG</td>
<td>I-ORG</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>a</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>unit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>of</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>AMR</td>
<td>B-ORG</td>
<td>I-ORG</td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>I-ORG</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>immediately</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>BIO label</th>
<th>IO label</th>
</tr>
</thead>
<tbody>
<tr>
<td>matched</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>the</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>move</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>spokesman</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Tim</td>
<td>B-PER</td>
<td>I-PER</td>
</tr>
<tr>
<td>Wagner</td>
<td>I-PER</td>
<td>I-PER</td>
</tr>
<tr>
<td>said</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>.</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Gazetteers

- Dictionaries that contain lists of entities
- Place names, person names etc
- Geographical names: [www.geonames.org](http://www.geonames.org)
- Census data for person names
- Derived from text corpora
- Lists of months, weekdays etc
- Corporations, commercial products etc
- Proteins, genes etc
CONLL-U data format
CONLL-U format

- A standard tabular format for certain type of annotated data
- Each word is in a separate line
- 10 tab-separated columns on each line:
  1. Word index
  2. The word itself
  3. Lemma
  4. Universal POS
  5. Language specific POS
  6. Morphological features
  7.-9. Information related to syntactic information
  10. Any other annotation
# text = They buy and sell books.

<table>
<thead>
<tr>
<th>#</th>
<th>Word 1</th>
<th>Word 2</th>
<th>POS 1</th>
<th>POS 2</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>They</td>
<td>they</td>
<td>PRON</td>
<td>PRP</td>
<td>Case=Nom</td>
</tr>
<tr>
<td>2</td>
<td>buy</td>
<td>buy</td>
<td>VERB</td>
<td>VBP</td>
<td>Number=Plur</td>
</tr>
<tr>
<td>3</td>
<td>and</td>
<td>and</td>
<td>CONJ</td>
<td>CC</td>
<td>_</td>
</tr>
<tr>
<td>4</td>
<td>sell</td>
<td>sell</td>
<td>VERB</td>
<td>VBP</td>
<td>Number=Plur</td>
</tr>
<tr>
<td>5</td>
<td>books</td>
<td>book</td>
<td>NOUN</td>
<td>NNS</td>
<td>Number=Plur</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>.</td>
<td>_</td>
</tr>
</tbody>
</table>

# text = I had no clue.

<table>
<thead>
<tr>
<th>#</th>
<th>Word 1</th>
<th>Word 2</th>
<th>POS 1</th>
<th>POS 2</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>I</td>
<td>PRON</td>
<td>PRP</td>
<td>Case=Nom</td>
</tr>
<tr>
<td>2</td>
<td>had</td>
<td>have</td>
<td>VERB</td>
<td>VBD</td>
<td>Number=Sing</td>
</tr>
<tr>
<td>3</td>
<td>no</td>
<td>no</td>
<td>DET</td>
<td>DT</td>
<td>PronType=Neg</td>
</tr>
<tr>
<td>4</td>
<td>clue</td>
<td>clue</td>
<td>NOUN</td>
<td>NN</td>
<td>Number=Sing</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>.</td>
<td>_</td>
</tr>
</tbody>
</table>
Universal Dependencies data

- [https://universaldependencies.org/](https://universaldependencies.org/)
- Manually annotated data in more than 70 languages
  - POS tags
  - Lemmas
  - Morphology
  - Syntax
- All datasets are in CONLL-U format
Sequence tagging methods
Sequence tagging approaches

1. Hidden Markov Models – old school
2. Log-linear sequence tagger (logistic regression at each step)
3. Conditional random fields

4. Recurrent neural networks
Hidden Markov Model

$p(t_i | t_{i-1})$ Transition probability

$\mathbf{t}_1 \rightarrow \mathbf{t}_2 \rightarrow \mathbf{t}_3 \rightarrow \mathbf{t}_4$

$p(w_i | t_i)$ Emission probability

$\mathbf{w}_1 \rightarrow \mathbf{w}_2 \rightarrow \mathbf{w}_3 \rightarrow \mathbf{w}_4$
Hidden Markov Model

\[ \hat{t}_1^n = \arg \max_{t_1^n} P(t_1^n | w_1^n) \]

\[ = \arg \max_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)} \]

\[ = \arg \max_{t_1^n} P(w_1^n | t_1^n)P(t_1^n) \]

\[ = \arg \max_{t_1^n} \prod_{i=1}^{n} P(w_i | t_1)P(t_i | t_{i-1}) \]

Application of the Bayes rule:

\[ P(y|x) = \frac{P(x|y)P(y)}{P(x)} \]

Argmax does not depend on the denominator, thus can be omitted.

Factor the probabilities according to the dependencies in the graphical model.
Viterbi algorithm

- A dynamic programming algorithm that enables to find the most likely tag sequence.
- Create a table $V$ with $N+2$ rows and $T$ columns:
  - $N$ – the number of tags
  - $T$ – the length of the sequence/sentence
- Initialize the first column:
  - For each tag $t$ in the tagset compute: $V[t, 1] = P(t|\text{start})P(w_1|t)$
- For each column $j = 2$ to $T$ in the table $V$:
  - For each tag $t$ in the tagset compute:
    $$V[t, j] = \max_{t'} V[t', j - 1]P(t|t')P(w_j|t)$$
- Finally trace the sequence with maximum probability from the end until the beginning
Janet will back the bill
Log-linear sequence tagging

\[ \hat{t}_1^n = \arg \max_{t_1^n} P(t_1^n | w_1^n) \]

\[ = \arg \max_{t_1^n} \prod_{i=1}^{n} P(t_i | t_{i-1}, w_i) \]

\[ = \arg \max_{t_1^n} \prod_{i=1}^{n} P(t_i | t_1^{i-1}, \mathbf{w}_1^n) \]
Log-linear sequence tagging

- Each factorized component $P(t_i|t_1^{i-1}, w^n_1)$ is a local multiclass logistic regression classifier.
- Features can be extracted from the currently predicted tag, the tags predicted for the previous words and from the whole word sequence.
- The probability distribution for one local model has the log-linear form (softmax):

$$P(t_i|t_1^{i-1}, w^n_1) = \frac{\exp(\nu \cdot f(t_i, t_1^{i-1}, w^n_1))}{\sum_{t \in T} \exp(\nu \cdot f(t, t_1^{i-1}, w^n_1))}$$
Word and tag feature templates

\[
\langle t_i, w_{i-2} \rangle, \langle t_i, w_{i-1} \rangle, \langle t_i, w_i \rangle, \langle t_i, w_{i+1} \rangle, \langle t_i, w_{i+2} \rangle, \\
\langle t_i, t_{i-1} \rangle, \langle t_i, t_{i-2}, t_{i-1} \rangle, \\
\langle t_i, t_{i-1}, w_i \rangle, \langle t_i, w_{i-1}, w_i \rangle \langle t_i, w_i, w_{i+1} \rangle,
\]

\[t_i = \text{VB and } w_{i-2} = \text{Janet}\]
\[t_i = \text{VB and } w_{i-1} = \text{will}\]
\[t_i = \text{VB and } w_i = \text{back}\]
\[t_i = \text{VB and } w_{i+1} = \text{the}\]
\[t_i = \text{VB and } w_{i+2} = \text{bill}\]
\[t_i = \text{VB and } t_{i-1} = \text{MD}\]
\[t_i = \text{VB and } t_{i-1} = \text{MD and } t_{i-2} = \text{NNP}\]
\[t_i = \text{VB and } w_i = \text{back and } w_{i+1} = \text{the}\]
Feature templates for unknown words

\( w_i \) contains a particular prefix (from all prefixes of length \( \leq 4 \))
\( w_i \) contains a particular suffix (from all suffixes of length \( \leq 4 \))
\( w_i \) contains a number
\( w_i \) contains an upper-case letter
\( w_i \) contains a hyphen
\( w_i \) is all upper case
\( w_i \)'s word shape
\( w_i \)'s short word shape
\( w_i \) is upper case and has a digit and a dash (like \( CFC-12 \))
\( w_i \) is upper case and followed within 3 words by Co., Inc., etc.

Features for the word \textbf{well-dressed}

\[
\begin{align*}
\text{prefix}(w_i) &= w \\
\text{prefix}(w_i) &= we \\
\text{prefix}(w_i) &= wel \\
\text{prefix}(w_i) &= well \\
\text{suffix}(w_i) &= ssed \\
\text{suffix}(w_i) &= sed \\
\text{suffix}(w_i) &= ed \\
\text{suffix}(w_i) &= d \\
\text{has-hyphen}(w_i) \\
\text{word-shape}(w_i) &= xxxx-xxxxxxx \\
\text{short-word-shape}(w_i) &= x-x
\end{align*}
\]
Greedy decoding

- Proceed from left to right and make predictions

**PRO:**
- Very quick

**CON:**
- Local solutions, future cannot influence the present
Label bias problem

- Occurs because the models are locally normalized

\[
P(\text{MD|START}) > P(\text{NN|START})
\]

\[
P(\text{TO|NN}) = \text{large}
\]

\[
P(\text{TO|MD}) = \text{small}
\]

\[
P(\text{TO|to}) = 1
\]

\[
P(\text{TO|to, t(will)}) = P(\text{TO|to}) = 1
\]

\[
P(\text{TO|to, MD}) = 1
\]

\[
P(\text{TO|to, NN}) = 1
\]
Conditional Random Field - CRF

\[ \hat{t}_1^n = \arg\max_{t_1^n} \frac{\exp \sum_{i=1}^n \rho(t_i, t_{i-1}^i, w_1^n)}{Z} \]
Log-probability loss

- Locally normalized model

\[ \mathcal{L} = \log \prod_{i=1}^{n} p(t_i | t_1^{i-1}, w_1^n) = \sum_{i=1}^{n} \log p(t_i | t_1^{i-1}, w_1^n) \]

\[ = \sum_{i=1}^{n} \log \frac{\exp(f(t_i, t_1^{i-1}, w_1^n))}{\sum_{t'} \exp(f(t', t_1^{i-1}, w_1^n))} \]

\[ = \sum_{i=1}^{n} f(t_i, t_1^{i-1}, w_1^n) - n \log \sum_{t'} \exp(f(t', t_1^{i-1}, w_1^n)) \]

- Global model

\[ \mathcal{L} = \log p(t_1^n | w_1^n) = \log \frac{\exp \sum_{i=1}^{n} \rho(t_i, t_1^{i-1}, w_1^n)}{\sum_{t_1^n \in T} \exp \sum_{i=1}^{n} \rho(t'_i, t_1^{i-1}, w_1^n)} \]

\[ = \sum_{i=1}^{n} \rho(t_i, t_1^{i-1}, w_1^n) - \log \sum_{t_1^n \in T} \exp \sum_{i=1}^{n} \rho(t'_i, t_1^{i-1}, w_1^n) \]
Beam search

- Beam search is a heuristic breadth-first search algorithm to find the approximate best solution (Viterbi was exact algorithm).
- Beam search can be also used to approximate the partition function of the CRF model.
- The main idea:
  - While decoding, store only a limited number of partial sequences that constitute a beam.
  - Expand the sequences in the beam with all possible next steps and keep only the most probable.
  - Repeat, until the sequences are fully decoded.
Evaluating a sequence tagger
Evaluating a POS/morphological tagger

- Accuracy = \frac{\text{Number of correctly predicted tags}}{\text{Number of all words}}

- Accuracy of OOV words

- Precision and recall per POS/morphological label
  - Aggregate with micro- or macro-averaging
Evaluating a NER system

- Use precision, recall and F-score
- Evaluation on the **entity** level, not the word level

<table>
<thead>
<tr>
<th>Word</th>
<th>True label</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>B-ORG</td>
<td>O</td>
</tr>
<tr>
<td>Airlines</td>
<td>I-ORG</td>
<td>O</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>a</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>unit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>of</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>AMR</td>
<td>B-ORG</td>
<td>B-ORG</td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>I-ORG</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>True positives:</th>
<th>False positives:</th>
<th>False negatives:</th>
<th>Precision:</th>
<th>Recall:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. True positives: The number of correctly predicted labels.
2. False positives: The number of incorrectly predicted labels.
3. False negatives: The number of missed labels.
4. Precision: The ratio of correctly predicted labels to all predicted labels.
5. Recall: The ratio of correctly predicted labels to all actual labels.
Evaluating a NER system

- Use precision, recall and F-score
- Evaluation on the **entity** level, not the word level

<table>
<thead>
<tr>
<th>Word</th>
<th>True label</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>B-ORG</td>
<td>O</td>
</tr>
<tr>
<td>Airlines</td>
<td>I-ORG</td>
<td>O</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>a</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>unit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>of</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>AMR</td>
<td>B-ORG</td>
<td>B-ORG</td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>I-ORG</td>
</tr>
</tbody>
</table>

True positives: 1
False positives: 0
False negatives: 1

Precision: 1
Recall: 0.5
Evaluating a NER system

- Consider **segmentation errors**
- True entity: ORG(1, 2), predicted entity ORG(1, 1)

<table>
<thead>
<tr>
<th>Word</th>
<th>True label</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>B-ORG</td>
<td>B-ORG</td>
</tr>
<tr>
<td>Airlines</td>
<td>I-ORG</td>
<td>O</td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>a</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>unit</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>of</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>AMR</td>
<td>B-ORG</td>
<td>B-ORG</td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>I-ORG</td>
</tr>
</tbody>
</table>

True positives:
False positives:
False negatives:
Precision:
Recall:
Evaluating a NER system

- Consider **segmentation errors**
- True entity: ORG(1, 2), predicted entity ORG(1, 1)

<table>
<thead>
<tr>
<th>Word</th>
<th>True label</th>
<th>Predicted</th>
<th>True positives: 1</th>
<th>False positives: 1</th>
<th>False negatives: 1</th>
<th>Precision: 0.5</th>
<th>Recall: 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>American</td>
<td>B-ORG</td>
<td>B-ORG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Airlines</td>
<td>I-ORG</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>,</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>unit</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>of</td>
<td>O</td>
<td>O</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AMR</td>
<td>B-ORG</td>
<td>B-ORG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>I-ORG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>