Lecture 5: Sequence tagging

LTAT.01.001 – Natural Language Processing
Kairit Sirts (kairit.sirts@ut.ee)
13.03.2019
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
- 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
### English POS tags — Penn Treebank tagset

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordin. conjunction</td>
<td><em>and, but, or</em></td>
<td>SYM</td>
<td>symbol</td>
<td><em>+, %, &amp;</em></td>
</tr>
<tr>
<td>CD</td>
<td>cardinal number</td>
<td><em>one, two</em></td>
<td>TO</td>
<td>“to”</td>
<td><em>to</em></td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
<td><em>a, the</em></td>
<td>UH</td>
<td>interjection</td>
<td><em>ah, oops</em></td>
</tr>
<tr>
<td>EX</td>
<td>existential ‘there’</td>
<td><em>there</em></td>
<td>VB</td>
<td>verb base form</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td><em>mea culpa</em></td>
<td>VBD</td>
<td>verb past tense</td>
<td><em>ate</em></td>
</tr>
<tr>
<td>IN</td>
<td>preposition/sub-conj</td>
<td><em>of, in, by</em></td>
<td>VBG</td>
<td>verb gerund</td>
<td><em>eating</em></td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
<td><em>yellow</em></td>
<td>VBN</td>
<td>verb past participle</td>
<td><em>eaten</em></td>
</tr>
<tr>
<td>JJR</td>
<td>adj., comparative</td>
<td><em>bigger</em></td>
<td>VBP</td>
<td>verb non-3sg pres</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>JJS</td>
<td>adj., superlative</td>
<td><em>wildest</em></td>
<td>VBZ</td>
<td>verb 3sg pres</td>
<td><em>eats</em></td>
</tr>
<tr>
<td>LS</td>
<td>list item marker</td>
<td><em>1, 2, One</em></td>
<td>WDT</td>
<td>wh-determiner</td>
<td><em>which, that</em></td>
</tr>
<tr>
<td>MD</td>
<td>modal</td>
<td><em>can, should</em></td>
<td>WP</td>
<td>wh-pronoun</td>
<td><em>what, who</em></td>
</tr>
<tr>
<td>NN</td>
<td>noun, sing. or mass</td>
<td><em>llama</em></td>
<td>WP$</td>
<td>possessive wh-</td>
<td><em>whose</em></td>
</tr>
<tr>
<td>NNS</td>
<td>noun, plural</td>
<td><em>llamas</em></td>
<td>WRB</td>
<td>wh-adverb</td>
<td><em>how, where</em></td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, sing.</td>
<td><em>IBM</em></td>
<td>$</td>
<td>dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>NNP$</td>
<td>proper noun, plural</td>
<td><em>Carolinast</em></td>
<td>#</td>
<td>pound sign</td>
<td>#</td>
</tr>
<tr>
<td>PDT</td>
<td>predeterminer</td>
<td><em>all, both</em></td>
<td>“</td>
<td>left quote</td>
<td>‘ or “</td>
</tr>
<tr>
<td>POS</td>
<td>possessive ending</td>
<td>‘s</td>
<td>”</td>
<td>right quote</td>
<td>” or ”</td>
</tr>
<tr>
<td>PRP</td>
<td>personal pronoun</td>
<td><em>I, you, he</em></td>
<td>(</td>
<td>left parenthesis</td>
<td>[, {, &lt;</td>
</tr>
<tr>
<td>PRP$</td>
<td>possessive pronoun</td>
<td><em>your, one’s</em></td>
<td>)</td>
<td>right parenthesis</td>
<td>[), }, &gt;</td>
</tr>
<tr>
<td>RB</td>
<td>adverb</td>
<td><em>quickly, never</em></td>
<td>,</td>
<td>comma</td>
<td>,</td>
</tr>
<tr>
<td>RBR</td>
<td>adverb, comparative</td>
<td><em>faster</em></td>
<td>.</td>
<td>sentence-final punct</td>
<td>. ?</td>
</tr>
<tr>
<td>RBS</td>
<td>adverb, superlative</td>
<td><em>fastest</em></td>
<td>;</td>
<td>mid-sentence punct</td>
<td>; ... --</td>
</tr>
<tr>
<td>RP</td>
<td>particle</td>
<td><em>up, off</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Estonian POS tags – used in estnltk

<table>
<thead>
<tr>
<th>POS tag</th>
<th>Description</th>
<th>Example</th>
<th>POS tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Adjective</td>
<td>kallis</td>
<td>N</td>
<td>Cardinal number</td>
<td>kaks</td>
</tr>
<tr>
<td>C</td>
<td>Adj., comparative</td>
<td>laiem</td>
<td>O</td>
<td>Ordinal number</td>
<td>teine</td>
</tr>
<tr>
<td>D</td>
<td>Adverb</td>
<td>kõrvuti</td>
<td>P</td>
<td>Pronoun</td>
<td>see</td>
</tr>
<tr>
<td>G</td>
<td>Genitive attribute</td>
<td>balti</td>
<td>S</td>
<td>Common noun</td>
<td>asi</td>
</tr>
<tr>
<td>H</td>
<td>Proper name</td>
<td>Edgar</td>
<td>U</td>
<td>Adj., superlative</td>
<td>pikim</td>
</tr>
<tr>
<td>I</td>
<td>Interjection</td>
<td>tere</td>
<td>V</td>
<td>Verb</td>
<td>lugema</td>
</tr>
<tr>
<td>J</td>
<td>Conjunction</td>
<td>ja</td>
<td>X</td>
<td>Verb complements</td>
<td>plehku</td>
</tr>
<tr>
<td>K</td>
<td>Adposition</td>
<td>kaudu</td>
<td>Y</td>
<td>Abbreviation</td>
<td>USA</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Z</td>
<td>Punctuation</td>
<td>-, /, ...</td>
</tr>
</tbody>
</table>
# 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
  - The: DET
  - cat: NOUN
  - sat: VERB
  - on: ADP
  - the: DET
  - mat: NOUN

- WSJ POS tags
  - The: DT
  - cat: NN
  - sat: VBD
  - on: IN
  - the: DT
  - mat: NN
Tag ambiguity in English

<table>
<thead>
<tr>
<th>Types:</th>
<th>WSJ</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous</td>
<td>44,432 (86%)</td>
<td>45,799 (85%)</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>7,025 (14%)</td>
<td>8,050 (15%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tokens:</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous</td>
<td>577,421 (45%)</td>
<td>384,349 (33%)</td>
</tr>
<tr>
<td>Ambiguous</td>
<td>711,780 (55%)</td>
<td>786,646 (67%)</td>
</tr>
</tbody>
</table>

Figure 10.2 The amount of tag ambiguity for word types in the Brown and WSJ corpora, from the Treebank-3 (45-tag) tagging. These statistics include punctuation as words, and assume words are kept in their original case.

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.-9. Information related to morphology and syntactic information
  10. Any other annotation
# text = They buy and sell books.

<table>
<thead>
<tr>
<th>#</th>
<th>Word</th>
<th>Form</th>
<th>Tag</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>They</td>
<td>they</td>
<td>PRON</td>
<td>PRP</td>
</tr>
<tr>
<td>2</td>
<td>buy</td>
<td>buy</td>
<td>VERB</td>
<td>VBP</td>
</tr>
<tr>
<td>3</td>
<td>and</td>
<td>and</td>
<td>CONJ</td>
<td>CC</td>
</tr>
<tr>
<td>4</td>
<td>sell</td>
<td>sell</td>
<td>VERB</td>
<td>VBP</td>
</tr>
<tr>
<td>5</td>
<td>books</td>
<td>book</td>
<td>NOUN</td>
<td>NNS</td>
</tr>
<tr>
<td>6</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>.</td>
</tr>
</tbody>
</table>

# text = I had no clue.

<table>
<thead>
<tr>
<th>#</th>
<th>Word</th>
<th>Form</th>
<th>Tag</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I</td>
<td>I</td>
<td>PRON</td>
<td>PRP</td>
</tr>
<tr>
<td>2</td>
<td>had</td>
<td>have</td>
<td>VERB</td>
<td>VBD</td>
</tr>
<tr>
<td>3</td>
<td>no</td>
<td>no</td>
<td>DET</td>
<td>DT</td>
</tr>
<tr>
<td>4</td>
<td>clue</td>
<td>clue</td>
<td>NOUN</td>
<td>NN</td>
</tr>
<tr>
<td>5</td>
<td>.</td>
<td>.</td>
<td>PUNCT</td>
<td>.</td>
</tr>
</tbody>
</table>
Universal Dependencies data

• https://universaldependencies.org/

• Manually annotated data in more than 70 languages
  • POS tags
  • Lemmas
  • Morphology
  • Syntax

• All datasets are in CONLL-U format
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>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>
Types of named entities correspond to proper names. Typically 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 low-cost 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

- \text{[Washington}_{\text{PER}}} was born into slavery on the farm of James Burroughs.
- \text{[Washington}_{\text{ORG}}} went up 2 games to 1 in the four-game series.
- Blair arrived in \text{[Washington}_{\text{LOC}}} for what may well be his last state visit.
- In June, \text{[Washington}_{\text{GPE}}} passed a primary seatbelt law.
- The \text{[Washington}_{\text{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]_{ORG}, a unit of [AMR Corp.]_{ORG}, immediately matched the move, spokesman [Tim Wagner]_{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
[American Airlines\textsubscript{ORG}], a unit of [AMR Corp.\textsubscript{ORG}], immediately matched the move, spokesman [Tim Wagner\textsubscript{PER}] said.
[American Airlines\textsubscript{\textit{ORG}}], a unit of [AMR Corp. \textsubscript{\textit{ORG}}], immediately matched the move, spokesman [Tim Wagner\textsubscript{\textit{PER}}] said.
Common features of a (non-neural) NER system

- Identity of the word (word unigram)
- Identities of the neighboring words
- POS of the word
- POS of the neighboring words
- Presence of the word in a gazetteer
- Syntactic chunk label (NP, VP, PP etc)

- Character ngram prefixes <=4
- Character ngram suffixes <=4
- Shape of the word
- Shapes of the neighboring words
- Short shape of the word
- Short shape of the neighboring words
- Presence of hyphen
- Word is all upper case
Word shape features

- Map small letters to $x$
- Map capital letters to $X$
- Map numbers to $d$
- Keep the punctuation marks

<table>
<thead>
<tr>
<th>Word</th>
<th>Long shape</th>
<th>Short shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.M.F.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DC10-30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>L’Occitane</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Word shape features

• Map small letters to $x$
• Map capital letters to $X$
• Map numbers to $d$
• Keep the punctuation marks

<table>
<thead>
<tr>
<th>Word</th>
<th>Long shape</th>
<th>Short shape</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.M.F.</td>
<td>X.X.X</td>
<td>X.X.X</td>
</tr>
<tr>
<td>DC10-30</td>
<td>XXdd-dd</td>
<td>Xd-d</td>
</tr>
<tr>
<td>L’Occitane</td>
<td>X’Xxxxxxxx</td>
<td>X’Xx</td>
</tr>
</tbody>
</table>
Gazetteers

• Dictionaries that contain lists of entities
• Place names, person names etc
• Geographical names: www.geonames.org
• Census data for person names
• Derived from text corpora
• Lists of months, weekdays etc
• Corporations, commercial products etc
• Proteins, genes etc
Evaluating a sequence tagger
Evaluating a POS tagger

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

- Accuracy of OOV words

- Precision and recall per POS label
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:
- False positives:
- False negatives:
- Precision:
- Recall:
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>I-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>
</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>I-ORG</td>
</tr>
<tr>
<td>Corp.</td>
<td>I-ORG</td>
<td>B-ORG</td>
</tr>
</tbody>
</table>

True positives: 1
False positives: 1
False negatives: 1
Precision: 0.5
Recall: 0.5
Sequence tagging approaches
Sequence tagging approaches

1. Hidden Markov Models – old school
2. Log-linear sequence tagger (maximum-entropy model)
3. Conditional random fields
4. Neural network models
Hidden Markov Model

Transition probability

Emission probability

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

$p(w_i|t_i)$
Hidden Markov Model

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

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

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

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

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
Parameter estimation

Transition probabilities

\[ P(t_i | t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})} \]

\[ P(VB | MD) = \frac{C(MD, VB)}{C(MD)} = \frac{10471}{13124} = 0.80 \]

Emission probabilities

\[ P(w_i | t_i) = \frac{C(t_i, w_i)}{C(t_i)} \]

\[ P(will | MD) = \frac{C(MD, will)}{C(MD)} = \frac{4046}{13124} = 0.31 \]
Viterbi decoding

• The goal is to find the most likely sequence of hidden tags
• Viterbi decoding is a dynamic programming algorithm

• Dynamic programming:
  • The task consists of a recursion that has repeated calls for the same inputs
  • Break the task into sub-problems and store the results of each sub-problems
  • When the result of a sub-problem is later needed again then use the stored results and do not compute it all over

• Dynamic programming algorithms are quite often used in NLP:
  • Viterbi decoding – for finding the sequence with maximum probability
  • Expectation-maximization – for estimating the parameters in latent variable model
Figure 10.7  A schematic of the tagging task for the sample sentence, showing the ambiguities for each word and the correct tag sequence as the highlighted path through the hidden states.
Viterbi algorithm for a sentence

• Create a table V with N+2 rows and T columns:
  • N – the number of tags
  • T – the length of the sequence/sentence

• Initialise 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
### Figure 10.5
The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

<table>
<thead>
<tr>
<th></th>
<th>NNP</th>
<th>MD</th>
<th>VB</th>
<th>JJ</th>
<th>NN</th>
<th>RB</th>
<th>DT</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;s &gt;</td>
<td>0.2767</td>
<td>0.0006</td>
<td>0.0031</td>
<td>0.0453</td>
<td>0.0449</td>
<td>0.0510</td>
<td>0.2026</td>
</tr>
<tr>
<td>NNP</td>
<td>0.3777</td>
<td>0.0110</td>
<td>0.0009</td>
<td>0.0084</td>
<td>0.0584</td>
<td>0.0090</td>
<td>0.0025</td>
</tr>
<tr>
<td>MD</td>
<td>0.0008</td>
<td>0.0002</td>
<td>0.7968</td>
<td>0.0005</td>
<td>0.0008</td>
<td>0.1698</td>
<td>0.0041</td>
</tr>
<tr>
<td>VB</td>
<td>0.0322</td>
<td>0.0005</td>
<td>0.0050</td>
<td>0.0837</td>
<td>0.0615</td>
<td>0.0514</td>
<td>0.2231</td>
</tr>
<tr>
<td>JJ</td>
<td>0.0366</td>
<td>0.0004</td>
<td>0.0001</td>
<td>0.0733</td>
<td>0.4509</td>
<td>0.0036</td>
<td>0.0036</td>
</tr>
<tr>
<td>NN</td>
<td>0.0096</td>
<td>0.0176</td>
<td>0.0014</td>
<td>0.0086</td>
<td>0.1216</td>
<td>0.0177</td>
<td>0.0068</td>
</tr>
<tr>
<td>RB</td>
<td>0.0068</td>
<td>0.0102</td>
<td>0.1011</td>
<td>0.1012</td>
<td>0.0120</td>
<td>0.0728</td>
<td>0.0479</td>
</tr>
<tr>
<td>DT</td>
<td>0.1147</td>
<td>0.0021</td>
<td>0.0002</td>
<td>0.2157</td>
<td>0.4744</td>
<td>0.0102</td>
<td>0.0017</td>
</tr>
</tbody>
</table>

### Figure 10.6
Observation likelihoods $B$ computed from the WSJ corpus without smoothing.

<table>
<thead>
<tr>
<th></th>
<th>Janet</th>
<th>will</th>
<th>back</th>
<th>the</th>
<th>bill</th>
</tr>
</thead>
<tbody>
<tr>
<td>NNP</td>
<td>0.000032</td>
<td>0</td>
<td>0</td>
<td>0.000048</td>
<td>0</td>
</tr>
<tr>
<td>MD</td>
<td>0</td>
<td>0.308431</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>VB</td>
<td>0</td>
<td>0.000028</td>
<td>0.000672</td>
<td>0</td>
<td>0.000028</td>
</tr>
<tr>
<td>JJ</td>
<td>0</td>
<td>0</td>
<td>0.000340</td>
<td>0.000097</td>
<td>0</td>
</tr>
<tr>
<td>NN</td>
<td>0</td>
<td>0.000200</td>
<td>0.000223</td>
<td>0.000006</td>
<td>0.002337</td>
</tr>
<tr>
<td>RB</td>
<td>0</td>
<td>0</td>
<td>0.010446</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DT</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.506099</td>
<td>0</td>
</tr>
</tbody>
</table>

$V[t, j] = \max_{t'} V[t', j - 1] P(t | t') P(w_j | t)$
Log-linear sequence tagging

HMM

Log-linear

Log-linear sequence tagging

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

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

\[ = \arg \max_{t_1^n} \prod_{i=1}^{n} P(t_i | t_{i-1}, w_1^n) \]
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 in neural models):

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

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

\n
\[
\begin{align*}
t_i &= \text{VB and } w_{i-2} = \text{Janet} \\
t_i &= \text{VB and } w_{i-1} = \text{will} \\
t_i &= \text{VB and } w_i = \text{back} \\
t_i &= \text{VB and } w_{i+1} = \text{the} \\
t_i &= \text{VB and } w_{i+2} = \text{bill} \\
t_i &= \text{VB and } t_{i-1} = \text{MD} \\
t_i &= \text{VB and } t_{i-1} = \text{MD} \text{ and } t_{i-2} = \text{NNP} \\
t_i &= \text{VB and } w_i = \text{back and } w_{i+1} = \text{the}
\end{align*}
\]

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 well-dressed

prefix($w_i$) = w
prefix($w_i$) = we
prefix($w_i$) = wel
prefix($w_i$) = well
suffix($w_i$) = ssed
suffix($w_i$) = sed
suffix($w_i$) = ed
suffix($w_i$) = d
has-hyphen($w_i$)
word-shape($w_i$) = x-xxxx-xxxxxxxx
short-word-shape($w_i$) = x-x

Neural sequence tagging

• Instead of manual feature extraction learn latent feature representations
• All standard components are there:
  • Word embeddings
  • Character embeddings
  • bi-directional recurrent network
  • Softmax output

Source: Ling et al., 2015. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation
More complex recurrent cells

• RNN

• LSTM

• GRU
LSTM – Long Short Term memory

• 3 gates

\[ \text{gate}_{\text{forget}} = \sigma(W_{fx}X_t + W_{fh}h_{t-1} + b_f) \]
\[ \text{gate}_{\text{input}} = \sigma(W_{ix}X_t + W_{ih}h_{t-1} + b_i) \]
\[ \text{gate}_{\text{out}} = \sigma(W_{ox}X_t + W_{oh}h_{t-1} + b_o) \]

• Three equations to update the cell

\[ \tilde{C} = \tanh(W_{cx}X_t + W_{ch}h_{t-1} + b_c) \]
\[ C_t = \text{gate}_{\text{forget}} \cdot C_{t-1} + \text{gate}_{\text{input}} \cdot \tilde{C} \]
\[ h_t = \text{gate}_{\text{out}} \cdot \tanh(C_t) \]
GRU – Gated Recurrent Unit

• 2 gates

\[
gate_r = \sigma(W_{rx}X_t + W_{rh}h_{t-1} + b) \\
gate_{update} = \sigma(W_{ux}X_t + W_{uh}h_{t-1} + b)
\]

• Two update equation

\[
\tilde{h}_t = \tanh(W_{hx}X_t + W_{hh} \cdot (gate_r \cdot h_{t-1}) + b) \\
h_t = (1 - gate_{update}) \cdot h_{t-1} + gate_{update} \cdot \tilde{h}_t
\]
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 normalised

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

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

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

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

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

Example from: Toutanova et al., 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network
Figure adopted from: https://www.researchgate.net/figure/263324470_fig20_Figure-1-Linear-chain-conditional-random-fields-modelBlack-nodes-represent-observable
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.
Beam search

- Beam size = 3

<START> time flies like an arrow

BOS NOUN NOUN VERB VERB NOUN VERB
VERB NOUN VERB ADP VERB ADP ADP
VERB NOUN ADP ADP DET DET DET
NOUN NOUN NOUN NOUN PROPN PROPN VERB