POS-TAGGING & NER

SEQUENCE LABELING

SEQUENCE LABELING

- Assigning labels (categories) to all elements of a sequence of observations
- Examples:
  - POS tagging: Assigning POS tags to tokens in a text

    Das Auto ist rot. → Das|DET Auto|NOUN ist|VERB rot|ADJ .|PUNCT
  - Chunking: Assigning chunk labels (e.g. NP, VP) to token sequences

    Das Auto ist rot. → [Das$_1$ Auto$_2$] [ist$_1$ rot$_2$] .
  - Named Entity Recognition (NER): assigning NE labels to tokens in a text

    Scholz fährt zum Bundestag nach Berlin. → Scholz|PERS fährt| zum| Bundestag| ORG nach| Berlin|LOC .|

BASELINE POS TAGGING

- „Most frequent tag“ (baseline):
  - Given: tag probabilities for a word $P(t|w)$
  - Tagging: $\underset{t}{argmax}\, P(t|w)$
  - Sufficient for 92% correct tagging; goal: > 96%

MARKOV CHAINS

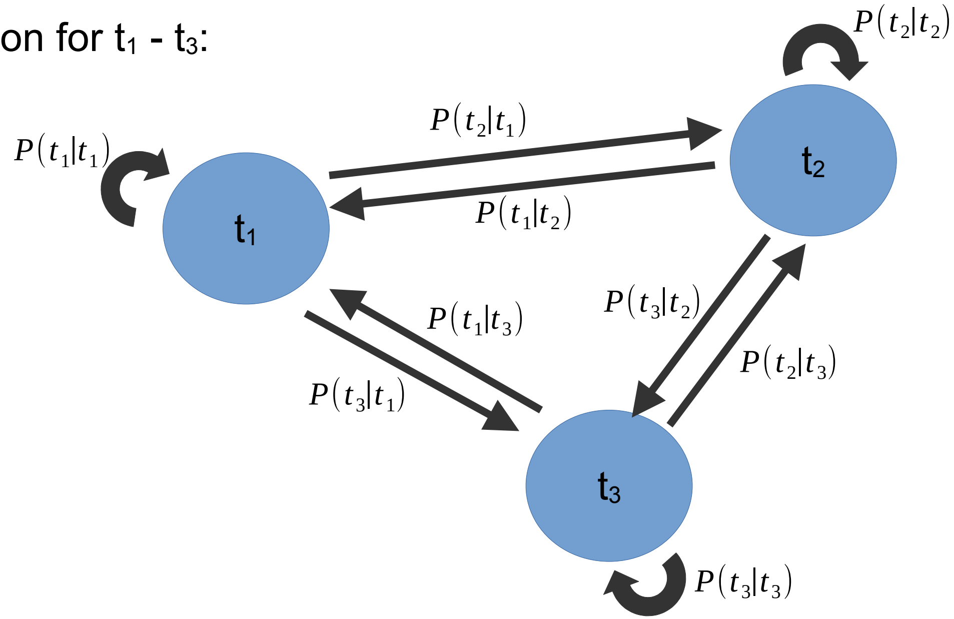
MARKOV CHAINS

- *Ich fahre nach Haus und trinke das* __. → "Haus", "Bier", "zwischen", "!"
- Probability of event depends on a limited (*n*) number of preceding events

  → Markov chain *n*-th order
- e.g. **n = 1** (first order Markov chain)
  - P(Haus|das), P(Bier|das), P(zwischen|das), P(!|das)

MARKOV MODEL I

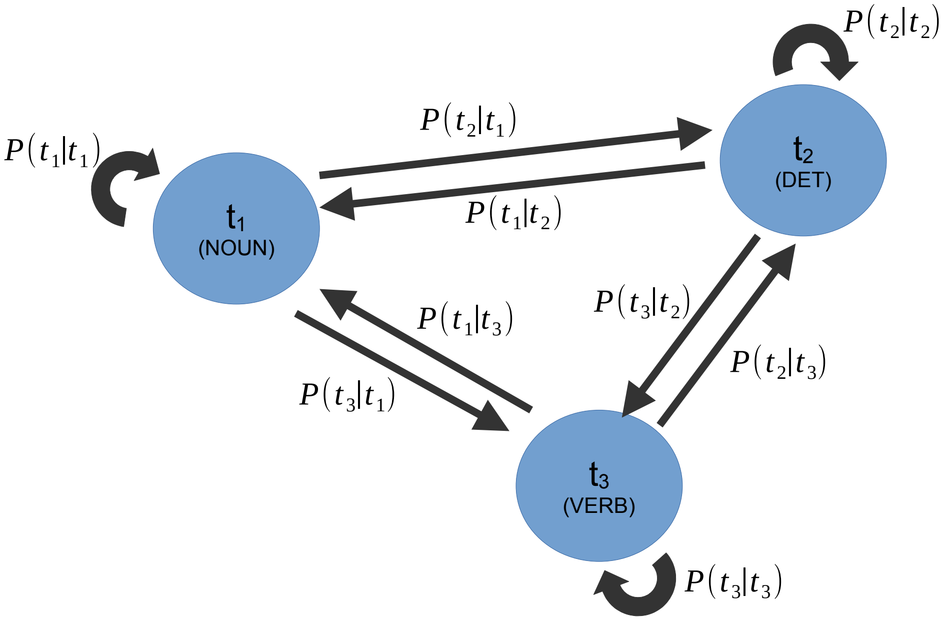
- Assumption: $P(t_i|t_1 \ldots t_{i-1}) = P(t_i|t_{i-1})$
- Automaton for t_1 - t_3:



$$\sum_{i=1}^{n} P(t_j|t_i) = 1$$

MARKOV MODEL II

- Example: tagset { DET, NOUN, VERB }



Transition probabilities

to

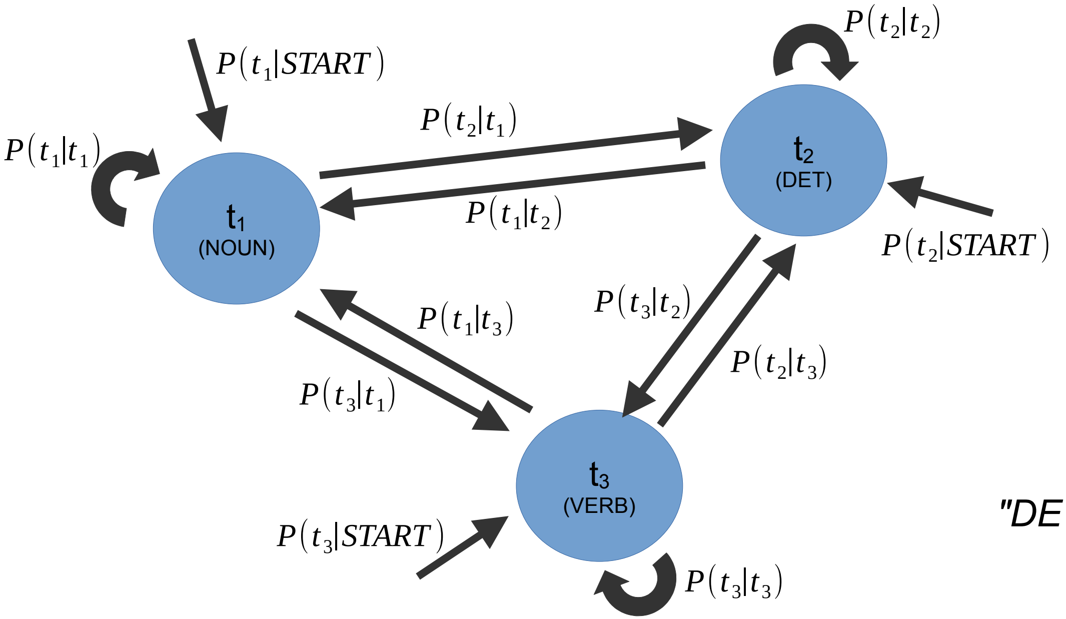
from

	DET	**NOUN**	**VERB**
DET	0,0	0,9	0,1
NOUN	0,1	0,3	0,6
VERB	0,5	0,3	0,2

$$\sum_{i=1}^{n} P(t_j|t_i)=1$$

MARKOV MODEL III

- Example: tagset { DET, NOUN, VERB }



Transition probabilities

to

from

	DET	**NOUN**	**VERB**
START	0,9	0,1	0,0
DET	0,0	0,9	0,1
NOUN	0,1	0,3	0,6
VERB	0,5	0,3	0,2

"DET DET NOUN" vs. *"DET NOUN VERB"*

MARKOV CHAIN I

- Therefore: probability of sequences (here: bigram model) as product of transition probabilities
- P("DET DET NOUN") = P(DET|*START*) * P(DET|DET) * P(NOUN|DET)

  = 0,9 * 0,0 * 0,9 = 0
- P("*DET NOUN VERB")* = P(DET|*START*) * P(NOUN|DET) * P(VERB|NOUN)

  = 0,9 *0,9 * 0,6 = **0,486**

to

from

	DET	**NOUN**	**VERB**
START	0,9	0,1	0,0
DET	0,0	0,9	0,1
NOUN	0,1	0,3	0,6
VERB	0,5	0,3	0,2

MARKOV CHAIN II

- Therefore: probability of sequences (here: bigram model) as product of transition probabilities
- P("DET DET NOUN") = P(DET|*START*) * P(DET|DET) * P(NOUN|DET)

  = 0,9 * 0,0 * 0,9 = 0
- P("*DET NOUN VERB")* = P(DET|*START*) * P(NOUN|DET) * P(VERB|NOUN)

  = 0,9 *0,9 * 0,6 = **0,486**
- POS tagging? Tokens?

to

from

	DET	NOUN	VERB
START	0,9	0,1	0,0
DET	0,0	0,9	0,1
NOUN	0,1	0,3	0,6
VERB	0,5	0,3	0,2

HIDDEN MARKOV MODELS

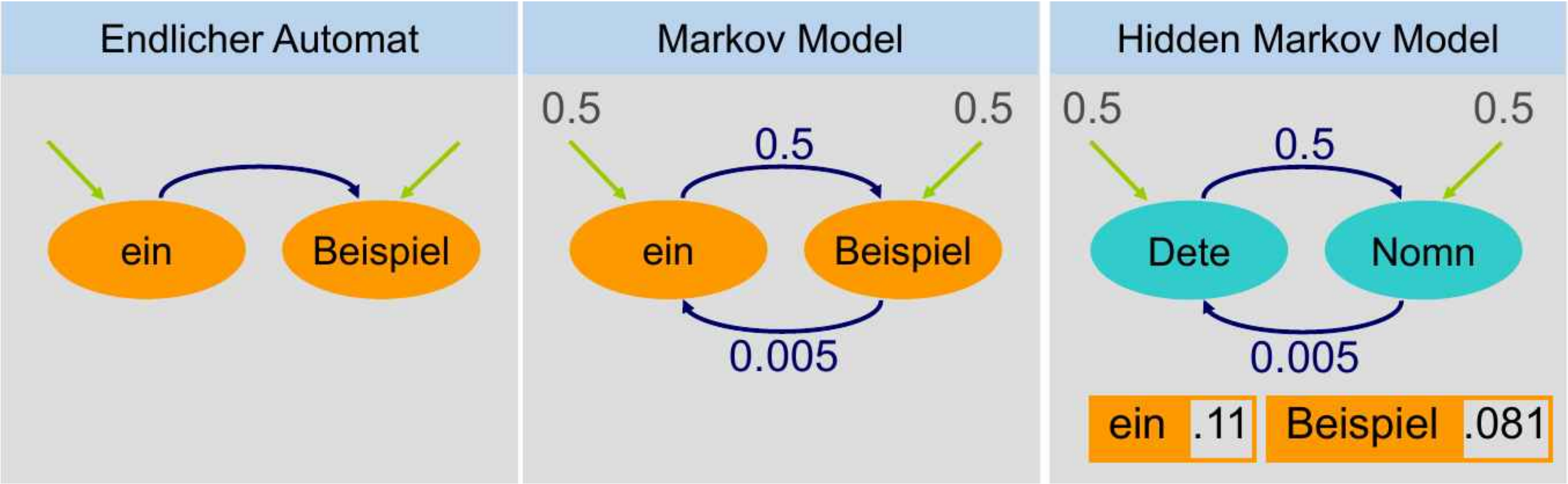
HMM I

- Idea:
  - Observations vs. underlying ("hidden") states
  - e.g. observed tokens vs. POS tags
- HMM with
  - Set of (hidden) states Q (q_1, q_2 … q_n)
  - Transition probabilities between all states of Q
  - Probabilities for q_i being a starting state (*start probabilities*) π (π_1, π_2, … π_n)
  - **Emission probabilities P(o_j|q_i)**
- Sequence of observations O (o_1, o_2 … o_t)

HMM II

- Furthermore:
  - Markov property $P(q_i|q_1 \ldots q_{i-1}) = P(q_i|q_{i-1})$
  - $P(o_i|q_1 \ldots q_n o_1 \ldots o_n) = P(o_i|q_i)$

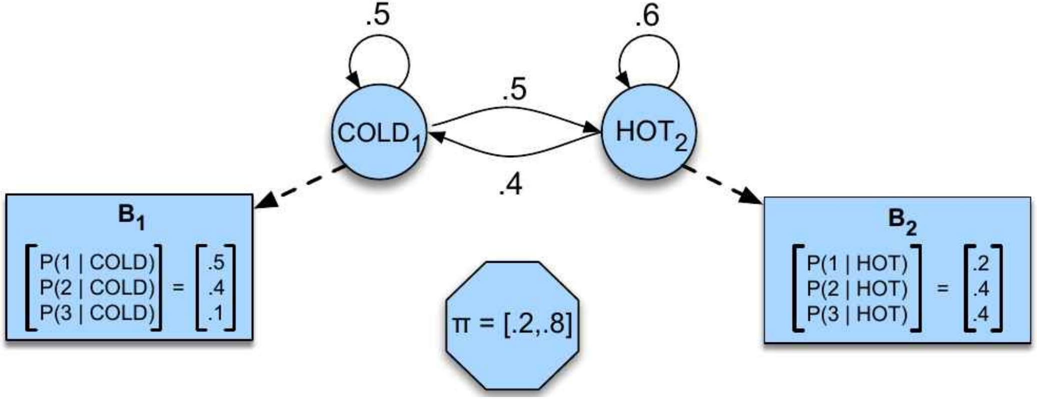
HMM III



- ● Übergangswahrscheinlichkeit
- ● Startzustände
- ● Beobachtung / Emisionen

HMM IV

- Example (by Jason Eisner):
    - Number of ice creams vs. air temperature



Eisner, J. 2002. An interactive spreadsheet for teaching the forward-backward algorithm. Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching NLP and CL.

Diagrams: Jurafsky & Martin 2024. Speech and Language Processing (3rd ed. draft).

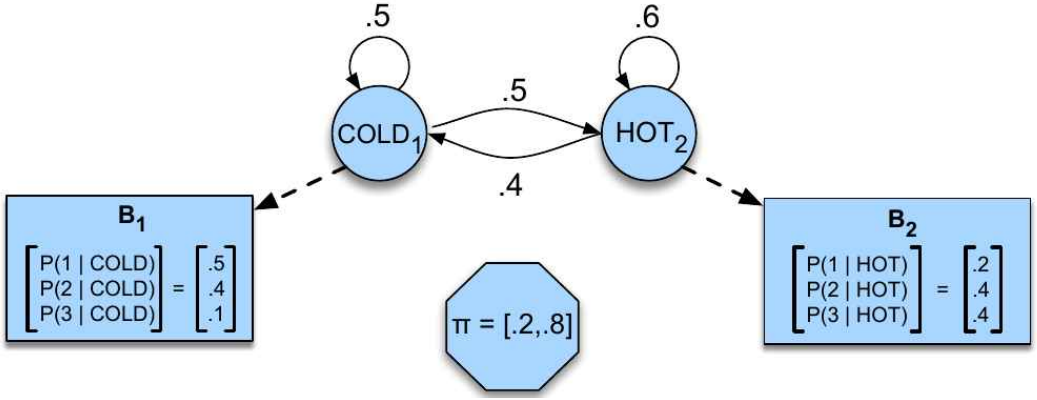
HMM V

- Not necessarily bigram model

  → Encoding in the states (here: {cold, hot} / {c, h})
- Trigram model: {cc, ch, hc, hh}
- ...

HMM VI

- Number of ice creams vs. air temperature


.5
.6
.5
COLD$_1$
HOT$_2$
.4
B_1
$\begin{bmatrix} P(1 \mid COLD) \\ P(2 \mid COLD) \\ P(3 \mid COLD) \end{bmatrix} = \begin{bmatrix} .5 \\ .4 \\ .1 \end{bmatrix}$
$\pi = [.2,.8]$
B_2
$\begin{bmatrix} P(1 \mid HOT) \\ P(2 \mid HOT) \\ P(3 \mid HOT) \end{bmatrix} = \begin{bmatrix} .2 \\ .4 \\ .4 \end{bmatrix}$

- Questions:
  - How likely is a sequence of observations?
  - What is the most likely sequence of states given a sequence of observations?

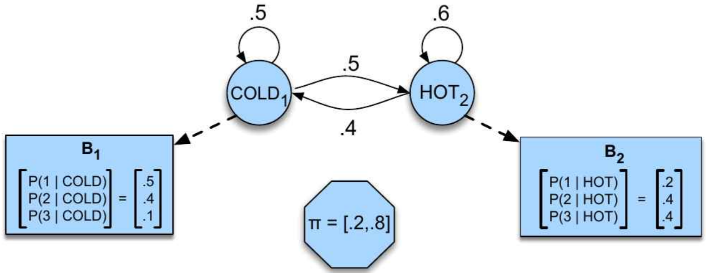
HMM – PROBABILITY OF AN OBSERVED SEQUENCE

- What is the probability of observation "1 2 3"?
- Part of the solution: P(1 2 3, cold cold hot)

  $= \prod P(o_i|q_i) * \prod P(q_i|q_{i-1})$

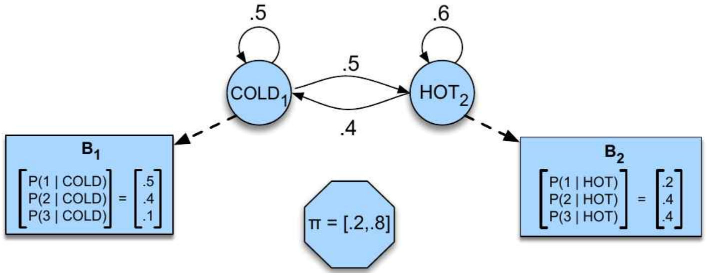
  = P(1|cold) * P(2|cold) * P(3|hot) * P(cold|START) * P(cold|cold) * P(hot|cold)

  = 0,5 * 0,4 * 0,4 * 0,2 * 0,5 * 0,5 = 0,004
- Complete solution: Probabilities for all 8 (2^3) possible state sequences
- Complexity? → *Forward Algorithmus*



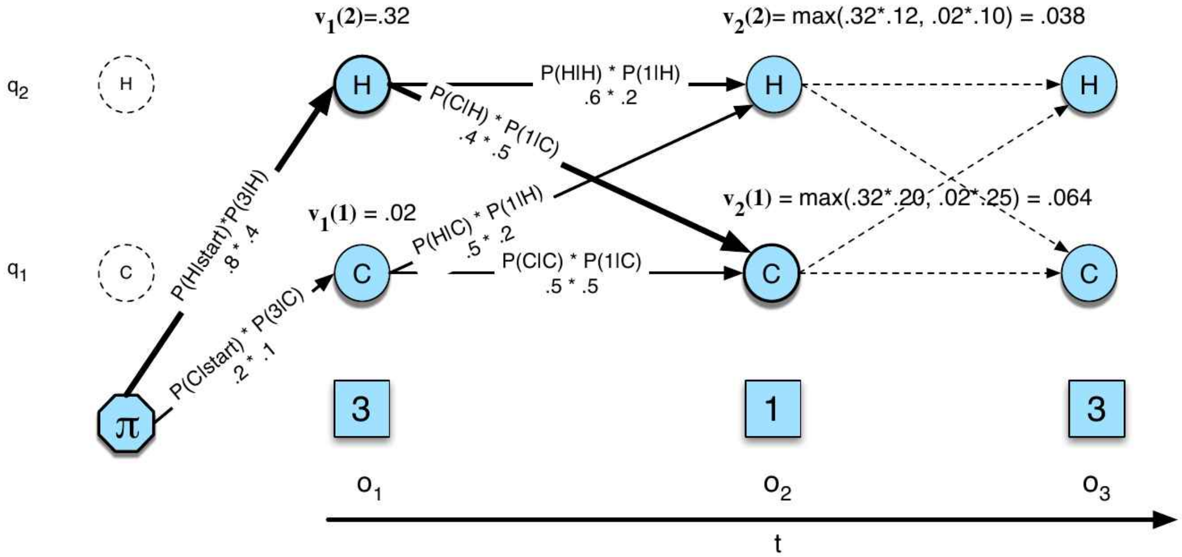
HMM – STATE SEQUENCE FOR OBSERVATION

- What is the most likely sequence of states given a "3 1 3"?
- Naive approach: Compute probabilities of all 8 possible combinations
  → State sequence with highest probability wins
  → $t_1 ... t_n = \text{argmax}_{1\text{-}n}\ P(t_k|t_{k-1}) * P(o_i|t_k)$
- Complexity given large set of states?
- Solution: Viterbi algorithm



HMM – VITERBI

- Compute most likely path for given observations at time t
- Use of already computed partial results for t-1



SHORT SUMMARY HMM

- Markov chains are automata whose transitions are assigned probabilities.
- The sum of the probabilities of the outgoing transitions of a node is 1.
- All states are "final states".
- They accept or generate symbol chains like automata, but in addition provide the probability for the symbol chain.
- The probability of a symbol chain is calculated from the product of the probabilities of the transition paths.
- The states of the symbol chain are not observable. (Hidden)
- Instead, we can observe the words and transitions and estimate the state transitions as parameters (latent variable) by counting them.

APPLICATION: POS TAGGING

POS TAGGING? A REMINDER…

- Original text:

  A relevant document will describe marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales.

- Brill tagger (using Penn Treebank Tagset):

  *A/**DT** relevant/**JJ** document/**NN** will/**MD** describe/**VB** marketing/**NN** strategies/**NNS** carried/**VBD** out/**IN** by/**IN** U.S./**NNP** companies/**NNS** for/**IN** their/**PRP$** agricultural/**JJ** chemicals/**NNS** ,/**,** report/**NN** predictions/**NNS** for/**IN** market/**NN** share/**NN** of/**IN** such/**JJ** chemicals/**NNS** ,/, or/**CC** report/**NN** market/**NN** statistics/**NNS** for/**IN** agrochemicals/**NNS** ,/**,** pesticide/**NN** ,/, herbicide/**NN** ,/, fungicide/**NN** ,/, insecticide/**NN** ,/, fertilizer/**NN** ,/**,** predicted/**VBN** sales/**NNS** ,/**,** market/**NN** share/**NN** ,/, stimulate/**VB** demand/**NN** ,/, price/**NN** cut/**NN** ,/**,** volume/**NN** of/**IN** sales/**NNS** ./**.***

POS TAGGING: BIGRAM HMM TAGGER

- States: POS tags, observations: word tokens
- Training:
  - Create dictionary with all word types and their tags based on a corpus
  - Generate probabilities:
    - $P(Word_i|Tag_k)$
    - $P(Tag_k|Tag_{k-1})$ with Tag_{k-1} predecessor of Tag_k (→ **bi**gram)
- „Local“ tagging
  - For each $Word_i$: select Tag_k which maximises $P(Word_i|Tag_k) * P(Tag_k|Tag_{k-1})$

POS TAGGING: BIGRAM HMM TAGGER – EXAMPLE

- People/NNS are/VBZ expected/VBN to/TO **queue**/VB at/IN the/DT registry/NNS
- The/DT police/NN is/VBZ to/TO blame/VB for/IN the/DT **queue**/NN

- to/TO **queue**/? (TO = "infinitive marker to")
- the/DT **queue**/?

- argmax$_k$ $P(t_k|t_{k-1}) * P(w_i|t_k)$
  - w_i = Word token i in sequence, t_k = possible tags for “queue”

POS TAGGING: BIGRAM HMM TAGGER – EXAMPLE II

- People/NNS are/VBZ expected/VBN to/TO **queue/???** at/IN the/DT registry/NNS
- The/DT police/NN is/VBZ to/TO blame/VB for/IN the/DT **queue**/NN

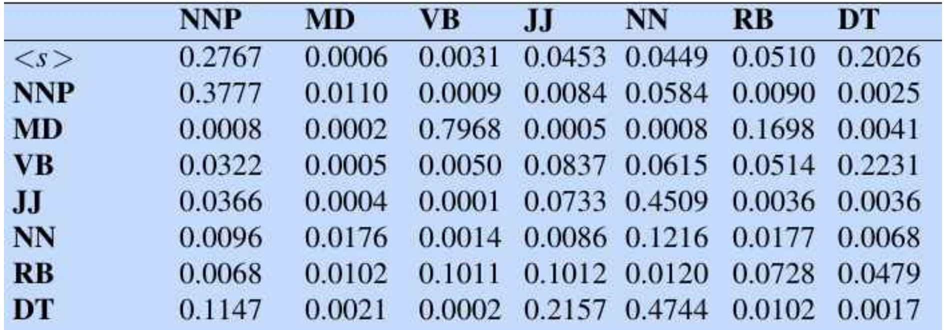
- $\text{argmax}_k\ P(t_k|t_{k-1}) * P(w_i|t_k)$

- $P(t_k|t_{k-1})$? → $|(t_{k-1}t_k)| / |t_{k-1}|$
- $P(w_i|t_k)$? → $|(w_it_k)| / |t_k|$

- Example:
  - P(NN|TO) * P(queue|NN) = 0,021 * 0,00041 → 0,000007
  - P(VB|TO) * P(queue|VB) = 0,34 * 0,00003 → 0,00001

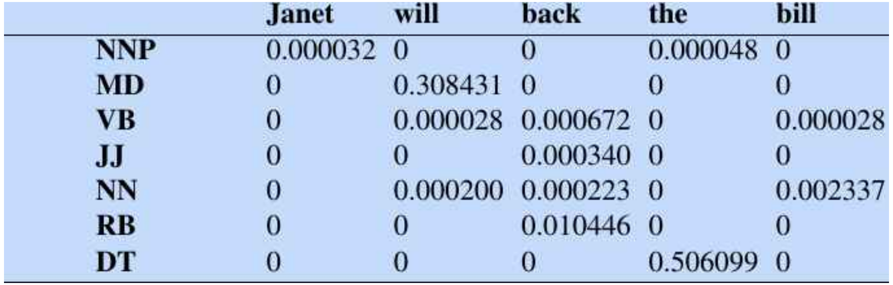
POS TAGGING: VITERBI

Transitions (with start <s>):



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

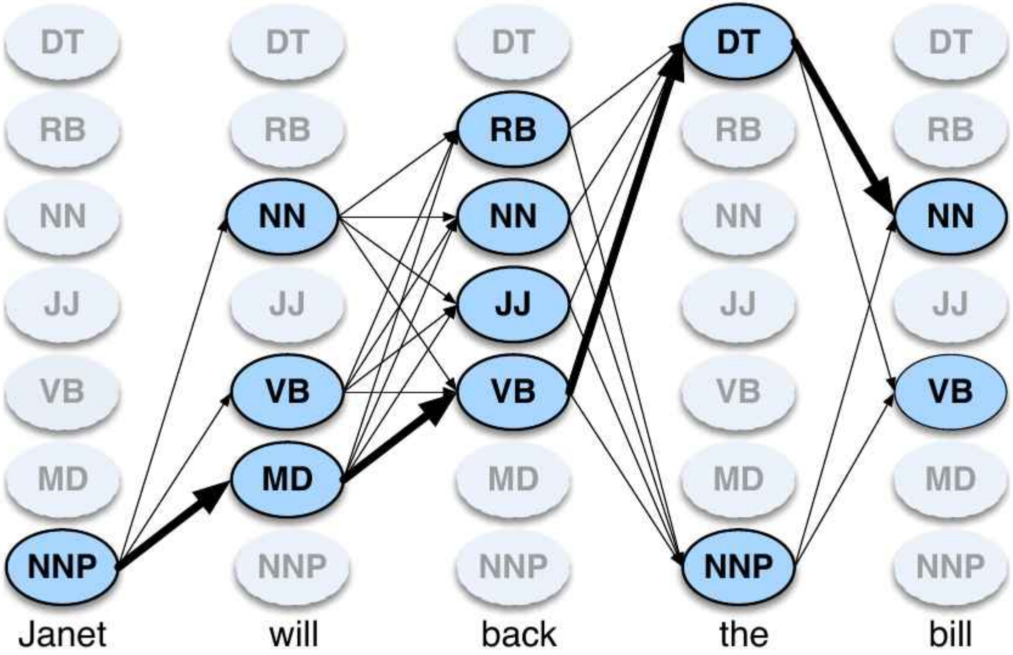
Emissions:

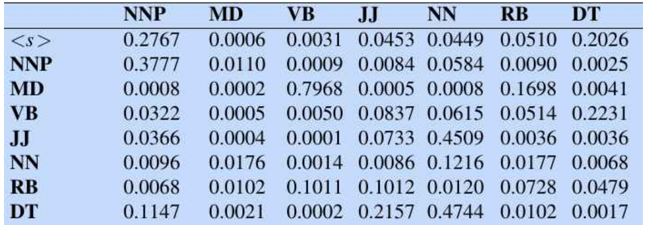


	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

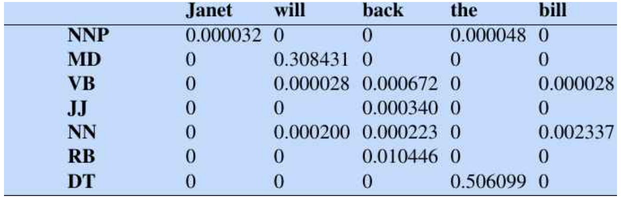
Jurafsky & Martin 2024. Speech and Language Processing (3rd ed. draft).

POS TAGGING: VITERBI





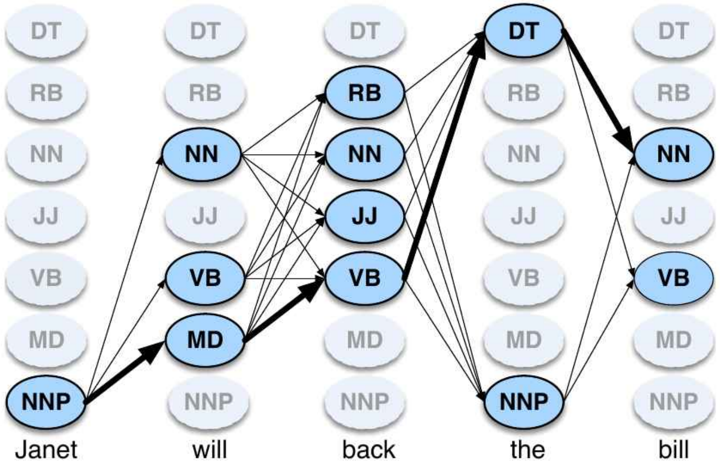
	NNP	MD	VB	JJ	NN	RB	DT
$<s>$	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176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

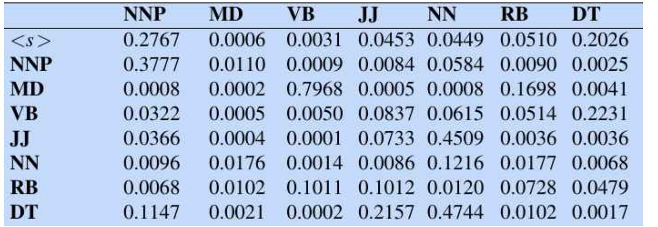


	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

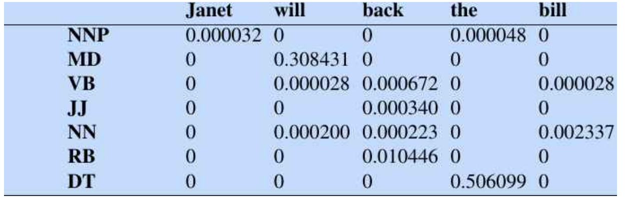
Jurafsky & Martin 2024. Speech and Language Processing (3rd ed. draft).

POS TAGGING: VITERBI – EXAMPLES





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



	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Janet|MD = P(MD|START) * P(Janet|MD) = 0,0006 * 0 = 0
Janet|NNP = P(NNP|START) * P(Janet|NNP) = 0,3777 * 0,000032 = 0,0000120864

Janet|NNP will|NN = 0,0000120864 * P(NN|NNP) * P(will|NN) = 0,0000120864 * 0,0584 * 0,0002 = 0,00000000014
Janet|NNP will|MD = 0,0000120864 * P(MD|NNP) * P(will|MD) = 0,0000120864 * 0,011 * 0,308431 = 0,000000041

PROBLEM: UNKNOWN WORDS

- What to do if w_i not found in the training corpus?
  - $P(w_i|t_k)$?
- Different approaches:
  - Tag distribution in the corpus for words with frequency 1
  - Use only $P(t_k|t_{k-1})$
  - Morphology or other criteria
    - Upper-/lowercase, hyphens in word, numbers in word …

WHY STILL RULE / TRANSFORMATION BASED POS TAGGING?

- Sakiba et al. 2020 "A Memory-Efficient Tool for Bengali Parts of Speech Tagging"

  “*Although there are different existing studies on Bengali parts of speech tagging [...] none of these studies consider memory optimization technique*“

- Gamit et al. 2019 "A Review on Part-Of-Speech Tagging on Gujarati Language"

  Advantage rule-based approaches: “*cost efficient*“, „*high precision*“; disadvantages: “*demands deep knowledge of the domain as well as a lot of manual work*“

- Naren & Ganesan 2019 "Rule based POS Tagger for Sanskrit"

  “*Unavailability of considerable amount of annotated corpora of sound quality for South-Asian languages like Sanskrit and tokenization of joined words are the major challenges.*“

- …

APPLICATION:
NAMED ENTITY RECOGNITION

NAMED ENTITIES

- Typs:
  - Persons
  - Geographical names (locations, countries, rivers, mountains, …)
  - Organisation names (companies, administration, …)
  - Product names
- Plus:
  - Date / time
  - Monetary expressions
  - Distances, weights, …

NER – NAMED ENTITY RECOGNITION

- Helpful: Lists of places, companies, organizations (...) that are as complete as possible
- No complete list of person names, but somehow complete lists of forenames and surenames

- Useful: fixed structures
  - e.g. Name = [Title +] Forename + Surename

- Comparable: description of numbers/weights using regular expressions
  - **All** rules languages dependent
  - Example.: 2022-09-08 = 8.9.2022 or 9.8.2022?

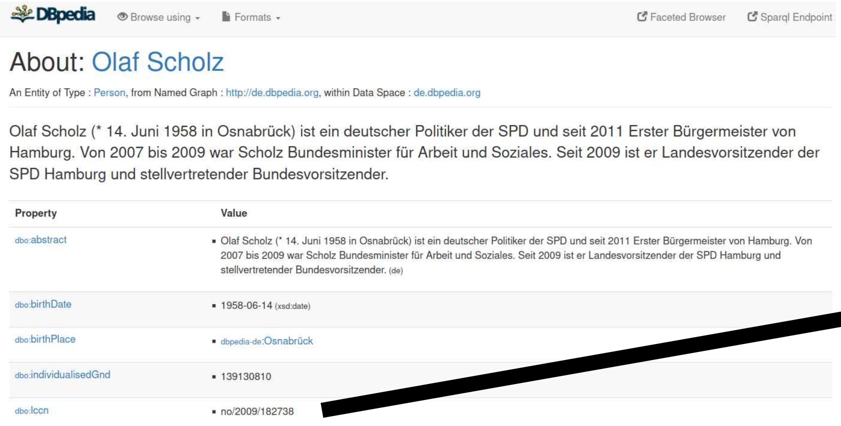
NER – LANGUAGE DEPENDENCY

- Patrick McKenzie („Falsehoods Programmers Believe About Names“):
  - *10. People’s names are written in any single character set.*
  - *11. People’s names are all mapped in Unicode code points.*
  - *12. People’s names are case sensitive.*
  - *15. People’s names do not contain numbers.*
  - *16. People’s names are not written in ALL CAPS.*

https://www.kalzumeus.com/2010/06/17/falsehoods-programmers-believe-about-names

RESSOURCES FOR PERSON NAMES: WIKIPEDIA & GND

- Lists of 100.000s persons from Wikipedia (de.wikipedia.org: ~900.000)
  - Using Wikipedia categories
  - Using IDs from different sources
- Example:

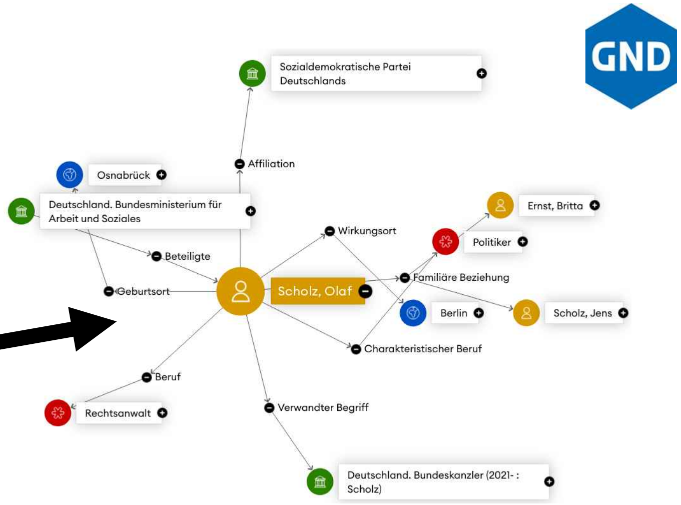


About: Olaf Scholz

An Entity of Type : Person, from Named Graph : http://de.dbpedia.org, within Data Space : de.dbpedia.org

Olaf Scholz (* 14. Juni 1958 in Osnabrück) ist ein deutscher Politiker der SPD und seit 2011 Erster Bürgermeister von Hamburg. Von 2007 bis 2009 war Scholz Bundesminister für Arbeit und Soziales. Seit 2009 ist er Landesvorsitzender der SPD Hamburg und stellvertretender Bundesvorsitzender.

Property	Value
dbo:abstract	▪ Olaf Scholz (* 14. Juni 1958 in Osnabrück) ist ein deutscher Politiker der SPD und seit 2011 Erster Bürgermeister von Hamburg. Von 2007 bis 2009 war Scholz Bundesminister für Arbeit und Soziales. Seit 2009 ist er Landesvorsitzender der SPD Hamburg und stellvertretender Bundesvorsitzender. (de)
dbo:birthDate	▪ 1958-06-14 (xsd:date)
dbo:birthPlace	▪ dbpedia-de:Osnabrück
dbo:individualisedGnd	▪ 139130810
dbo:lccn	▪ no/2009/182738



RESSOURCES FOR PERSON NAMES: WIKIPEDIA & GND

- More:
  - Virtual International Authority File VIAF
  - Library of Congress Control Number LCCN
  - SELIBR (Schweden)
  - Bibliothèque nationale de France BNF
  - BIBSYS (Norwegen)
  - National Diet Library NDL (Japan)
  - ...

RESSOURCES FOR PERSON NAMES: LISTS

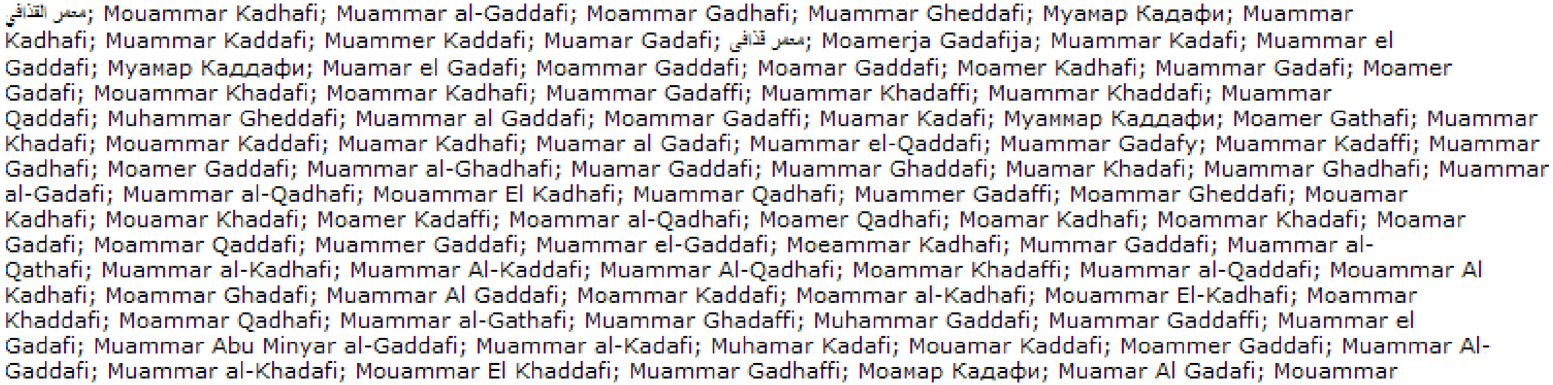
- Phone books or similar
    - Cleaning? ("Salon Brigitte")
- Directory of forenames

JRC-NAMES

- "JRC-Names is a highly multilingual named entity resource for person and organisation names (called 'entities') developed by the European Commission's Joint Research Centre (JRC). JRC-Names consists of large lists of names and their many spelling variants (up to hundreds for a single person), including across scripts (Latin, Greek, Arabic, Cyrillic, Japanese, Chinese, etc.)."
- "The resource is the by-product of the Europe Media Monitor family of applications, which has been analysing up to 220,000 news reports per day, since 2004. EMM recognises names mentioned in the news in over twenty languages and decides automatically for each newly found name whether it belongs to a new entity or whether it is a spelling variant of a previously known entity. This resource allows EMM users to display news about people or organisations even if their names are spelt differently or if the news articles are written in different languages and scripts."

JRC-NAMES

- Many language variants
- "Winner" (as of 2011): Muammar Gaddafi with 413 variants:



معمر القذافي; Mouammar Kadhafi; Muammar al-Gaddafi; Moammar Gadhafi; Muammar Gheddafi; Муамар Кадафи; Muammar Kadhafi; Muammar Kaddafi; Muammer Kaddafi; Muamar Gadafi; معمر قذافي; Moamerja Gadafija; Muammar Kadafi; Muammar el Gaddafi; Муамар Каддафи; Muamar el Gadafi; Moammar Gaddafi; Moamar Gaddafi; Moamer Kadhafi; Muammar Gadafi; Moamer Gadafi; Mouammar Khadafi; Moammar Kadhafi; Muammar Gadaffi; Muammar Khadaffi; Muammar Khaddafi; Muammar Qaddafi; Muhammar Gheddafi; Muammar al Gaddafi; Moammar Gadaffi; Muamar Kadafi; Муаммар Каддафи; Moamer Gathafi; Muammar Khadafi; Mouammar Kaddafi; Muamar Kadhafi; Muamar al Gadafi; Muammar el-Qaddafi; Muammar Gadafy; Muammar Kadaffi; Muammar Gadhafi; Moamer Gaddafi; Muammar al-Ghadhafi; Muamar Gaddafi; Muammar Ghaddafi; Muamar Khadafi; Muammar Ghadhafi; Muammar al-Gadafi; Muammar al-Qadhafi; Mouammar El Kadhafi; Muammar Qadhafi; Muammer Gadaffi; Moammar Gheddafi; Mouamar Kadhafi; Mouamar Khadafi; Moamer Kadaffi; Moammar al-Qadhafi; Moamer Qadhafi; Moamar Kadhafi; Moammar Khadafi; Moamar Gadafi; Moammar Qaddafi; Muammer Gaddafi; Muammar el-Gaddafi; Moeammar Kadhafi; Mummar Gaddafi; Muammar al-Qathafi; Muammar al-Kadhafi; Muammar Al-Kaddafi; Muammar Al-Qadhafi; Moammar Khadaffi; Muammar al-Qaddafi; Mouammar Al Kadhafi; Moammar Ghadafi; Muammar Al Gaddafi; Moammar Kaddafi; Moammar al-Kadhafi; Mouammar El-Kadhafi; Moammar Khaddafi; Moammar Qadhafi; Muammar al-Gathafi; Muammar Ghadaffi; Muhammar Gaddafi; Muammar Gaddaffi; Muammar el Gadafi; Muammar Abu Minyar al-Gaddafi; Muammar al-Kadafi; Muhamar Kadafi; Mouamar Kaddafi; Moammer Gaddafi; Muammar Al-Gaddafi; Muammar al-Khadafi; Mouammar El Khaddafi; Muammar Gadhaffi; Моамар Кадафи; Muamar Al Gadafi; Mouammar

Ralf Steinberger et al. 2011. JRC-NAMES: A Freely Available, Highly Multilingual Named Entity Resource. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pages 104–110, Hissar, Bulgaria. Association for Computational Linguistics.

NER – RECOGNIZING NAMES AS CLASSIFICATION TASK

- Maybe useful features:
  - Text context of the potential name
    - Forename before potential surename, but no DET
    - *in, of, to* before potential location, but no DET
- String similarity
  - String similarity (if *Obermayer* surename, maybe *Obermeyer* is too?)
  - Common word suffixes (like -*stadt*, -*walde* etc.)
- ...

NER AS SEQUENCE LABELING I

- Problem: Named Entities often consist of multiple tokens
- e.g. *Angela Dorothea Merkel fährt zum Deutschen Bundestag am Berliner Tiergarten in Berlin*

  →

  Angela|PER Dorothea|PER Merkel|PER fährt zum Deutschen|LOC Bundestag|ORG am Berliner|LOC Tiergarten|LOC in| Berlin|LOC

- e.g. IO format (inside-outside):

  Angela|*I-PER* ***Dorothea***|*I-PER* ***Merkel***|*I-PER* ***fährt***|*O* ***zum***|*O* ***Deutschen***|*I-ORG* ***Bundestag***|*I-ORG* ***am***|*O* ***Berliner***|*I-LOC* ***Tiergarten***|*I-LOC* ***in***|*O* ***Berlin***|*I-LOC*

NER AS SEQUENCE LABELING II

Solution: separate marking of NE beginning

- IOB/BIO format (inside-outside-beginning):

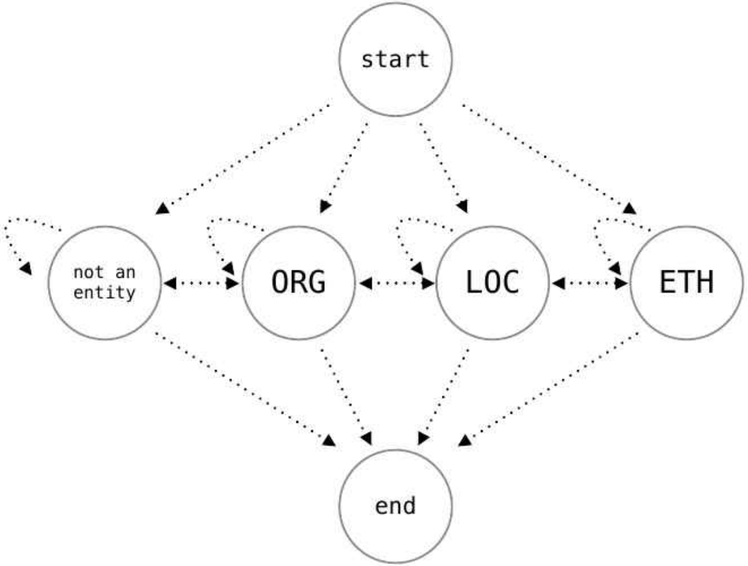
  Angela|*B-PER* ***Dorothea***|*I-PER* ***Merkel***|*I-PER* ***fährt***|*O* ***zum***|*O* ***Deutschen***|*B-ORG* ***Bundestag***|*I-ORG* ***am***|*O* ***Berliner***|*B-LOC* ***Tiergarten***|*I-LOC* ***in***|*O* ***Berlin***|*B-LOC*

- BIOES format (E = End, S = Single)

  Angela|*B-PER* ***Dorothea***|*I-PER* ***Merkel***|*E-PER* ***fährt***|*O* ***zum***|*O* ***Deutschen***|*B-ORG* ***Bundestag***|*E-ORG* ***am***|*O* ***Berliner***|*B-LOC* ***Tiergarten***|*E-LOC* ***in***|*O* ***Berlin***|*S-LOC*

NER AS SEQUENCE LABELING III

- Implementation: analogous to POS tagging

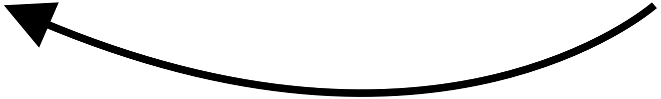


Author: Jesse Anderton

NER AS SEQUENCE LABELING IV

- Problems?
  - Encoding complex features only in transition & emission probabilities
  - What to do with OOV?
  - Tolerance to linguistic varieties (language registers)?
  - Only locale or consecutive features

Will|? *you|PRON marry|VERB me|PRON* ***?|***_PUNCT_



MORE FEATURES?
(INCL. NON-MARKOV)

EXAMPLE: STANFORD NAMED ENTITY RECOGNIZER

- Supported classes: locations, persons, organisations, monetary, percentages, dates, time
- 8 supported languages (Arabic, Chinese, German, English, French, Italian, Hungarian, Spanish)
- Part of *Stanford CoreNLP*
- Implementation: Conditional Random Fields

https://nlp.stanford.edu/software/CRF-NER.html

CONDITIONAL RANDOM FIELDS

- Probability of a sequence Y (POS-/NE tags) for a sequence X (word tokens) as sum of features
- (local) Feature functions describe desired relationships
- Often: $f_k(x) \rightarrow \{0,1\}$

CONDITIONAL RANDOM FIELDS

- e.g.

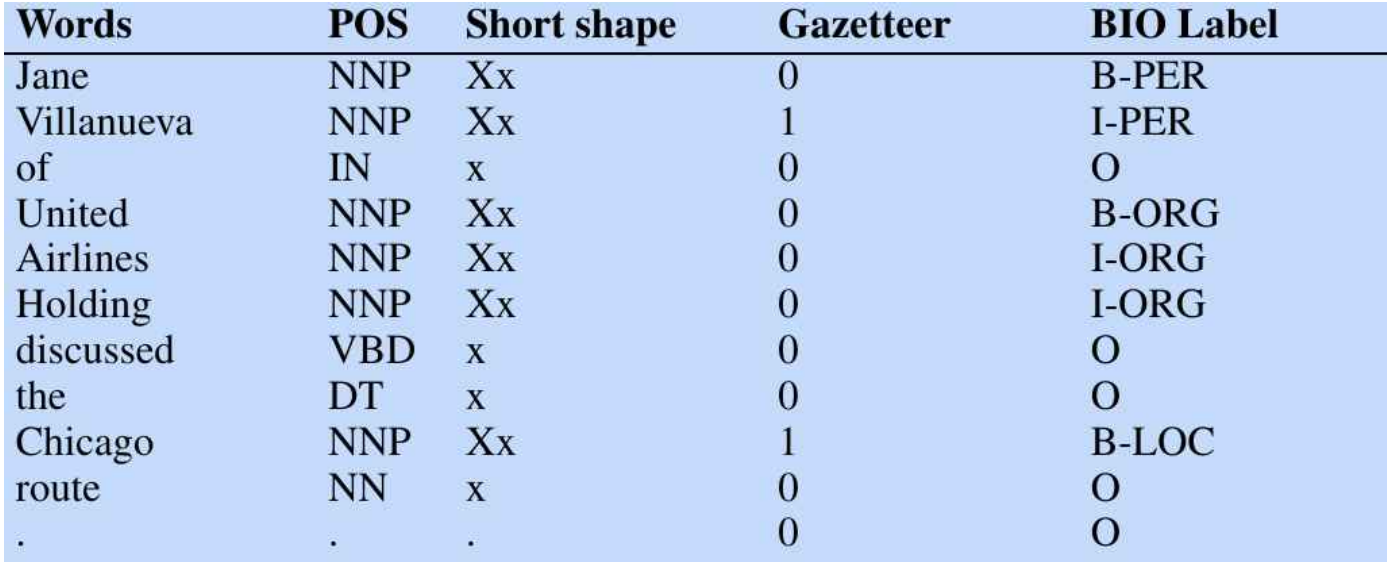
	Das|DET	*Auto|NOUN*	*vergessen|VERB*	*?|PUNCT*
– f_1: $\|w_i\| < 4$, w_i = DET	1	0	0	0
– f_2: w_i = "^ver", w_i = VERB	0	0	1	0
– f_3: w_{i-2} = DET, w_i = NOUN	0	0	0	0

	*Das|**ADJ***	*Auto|**PROPN***	*vergessen|VERB*	*?|PUNCT*
– f_1: $\|w_i\| < 4$, w_i = DET	0	0	0	0
– f_2: w_i = "^ver", w_i = VERB	0	0	1	0
– f_3: w_{i-2} = DET, w_i = NOUN	0	0	0	0

CONDITIONAL RANDOM FIELDS

- Flexible features:
  - Word prefixes/suffixes (e.g. „-ung“ in German → NOUN)
  - Use of external resources (gazetteers, name lists)
  - Lower/upper case
  - Words in same sentence (and their properties)
  - Word position (in sentence)
  - …
  - Any combination (e.g. "Upper case & not at start of sentence → NOUN or PROPN")
    → Basis: templates

CONDITIONAL RANDOM FIELDS – FEATURES IN A SENTENCE



Words	POS	Short shape	Gazetteer	BIO Label
Jane	NNP	Xx	0	B-PER
Villanueva	NNP	Xx	1	I-PER
of	IN	x	0	O
United	NNP	Xx	0	B-ORG
Airlines	NNP	Xx	0	I-ORG
Holding	NNP	Xx	0	I-ORG
discussed	VBD	x	0	O
the	DT	x	0	O
Chicago	NNP	Xx	1	B-LOC
route	NN	x	0	O
.	.	.	0	O

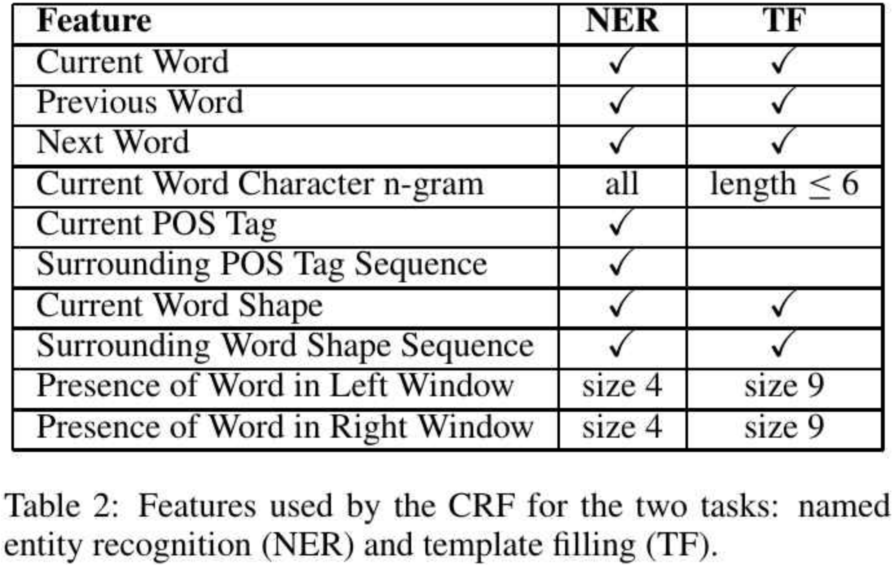
Jurafsky & Martin 2024. Speech and Language Processing (3rd ed. draft).

CONDITIONAL RANDOM FIELDS – TOY SAMPLE

- Sentence: "*Das Auto fährt*", POS tags = { DET, NOUN, VERB }
  - f_1 (x_i = "Auto" and y_i = "NOUN"), w_1 = 10
  - f_2 (y_i = "NOUN" and y_{i-1} = "DET"), w_2 = 5
  - f_3 ($|x_i|$ > 5 and y_i = "VERB"), w_3 = 5
  - f_4 (x_i = "fährt" and y_i = "VERB"), w_4 = 5
  - f_5 (y_i = "VERB" and y_{i-i} = "NOUN"), w_4 = 2

- *Das*|DET *Auto*|NOUN *fährt*|VERB
  → 10 + 5 + 0 + 5 + 2 = 22
- *Das*|DET *Auto*|NOUN *fährt*|NOUN
  → 10 + 5 + 0 + 0 + 0 = 15

CONDITIONAL RANDOM FIELDS



Feature	**NER**	**TF**
Current Word	✓	✓
Previous Word	✓	✓
Next Word	✓	✓
Current Word Character n-gram	all	length ≤ 6
Current POS Tag	✓	
Surrounding POS Tag Sequence	✓	
Current Word Shape	✓	✓
Surrounding Word Shape Sequence	✓	✓
Presence of Word in Left Window	size 4	size 9
Presence of Word in Right Window	size 4	size 9

Table 2: Features used by the CRF for the two tasks: named entity recognition (NER) and template filling (TF).

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL 2005), pp. 363-370.