



Maschinelle Sprachverarbeitung

Übung

Aufgabe 6: Gen-Erkennung mit Maschinellen Lernen

Mario Sängner

Problemstellung

- Erkennung von Genen in Texten
 - NEU: Beachtung von Multi-Token-Entitäten (B-/I-protein)

Number	0
of	0
glucocorticoid	B-protein
receptors	I-protein
in	0
lymphocytes	0
and	0
their	0
sensitivity	0
to	0
hormone	0
action	0
.	0
The	0
study	0

Problemstellung

- NEU: Verwendung von Verfahren des Maschinellen Lernen
- Modellierung als Sequenz-Labeling-Problem
 - Ziel: Ordne jedem Wort eines Satzes ein Label zu, welches angibt ob das Wort Beginn oder Teil eines Gens ist
 - Finde die („global“) beste Sequenz von Tags

Number	Of	glucocorticoid	receptors	in	lymphocytes	and	...
↓	↓	↓	↓	↓	↓	↓	
O	O	B-protein	I-protein	O	O	O	...

Problemstellung

- NEU: Verwendung von Verfahren des Maschinellen Lernen
- Modellierung als Sequenz-Labeling-Problem
 - Ziel: Ordne jedem Wort eines Satzes ein Label zu, welches angibt ob das Wort Beginn oder Teil eines Gens ist
 - Finde die („global“) beste Sequenz von Tags

Number	Of	glucocorticoid	receptors	in	lymphocytes	and	...
↓	↓	↓	↓	↓	↓	↓	
0	0	B-protein	I-protein	0	0	0	...
0	0	B-protein	0	0	0	0	

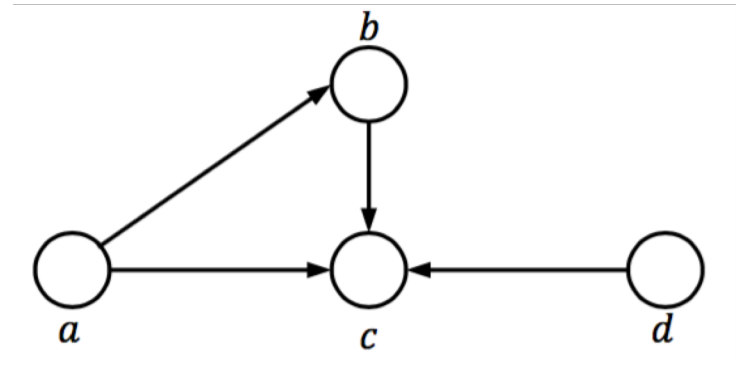
Sequenz-Labeling-Verfahren

- Vielzahl verschiedener Algorithmen für die Vorhersage von Sequenzen
 - Hidden Markov Model (HMM)
 - Conditional Random Fields (CRF)
 - Averaged Perceptron (AP)
 - Structured SVMs (SVM^{struct})
 - Max Margin Markov Networks (M³N)
 - Recurrent Neuronal Networks
- Minimalanforderung: Implementierung eines Ansatzes basierend auf einem CRF
 - Alternative Ansätze bitte mit mir abstimmen!

Conditional Random Fields

- Implementation mit Hilfe von Linear-chain Conditional Random Fields (CRFs)
 - Besondere Form von ungerichteten prob. graphischen Modellen
- Probabilistische graphische Modelle (PGMs)
 - Schematische Darstellung von Wahrscheinlichkeitsmodellen
 - Modellierung von Abhängigkeiten zwischen Zufallsvariablen

$$p(a, b, c, d) = p(a) * p(b|a) * \\ p(c|a, b, d) * p(d)$$



Probabilistische graphische Modelle

- Fehlen einer Kante modelliert bedingte Unabhängigkeit zwischen zwei Variablen
 - Variablen a und b sind bedingt unabhängig hinsichtlich einer dritten Variable c , sofern gilt:

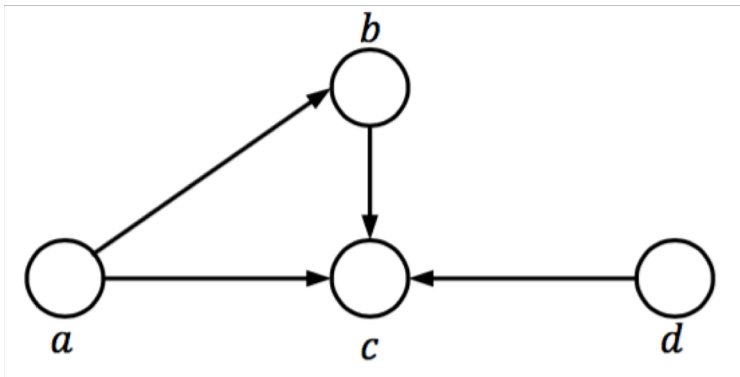
$$p(a, b|c) = p(a|c)p(b|c)$$

- Unabhängigkeitsannahme erlaubt Zerlegung der Wahrscheinlichkeitsverteilung in Faktoren Ψ_i :

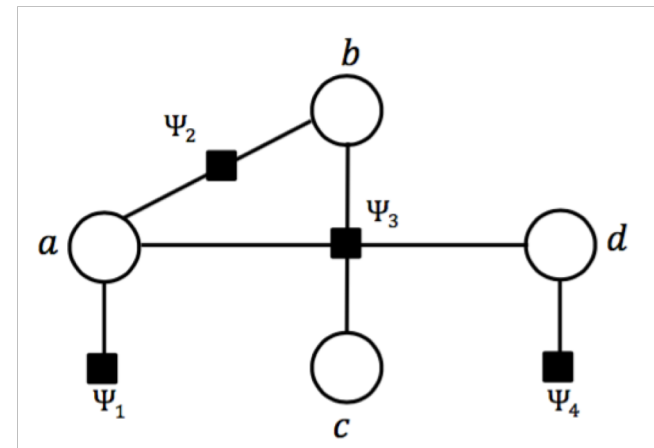
$$p(\mathbf{x}) = \prod_{i=1}^k \Psi_i(\mathbf{x}_i)$$

Probabilistische graphische Modelle

- Visualisierung der Zerlegung durch Faktorgraphen
 - Explizite Modellierung der Faktoren mittels eines bipartiten Graphen



$$\Psi_1 = p(a)$$



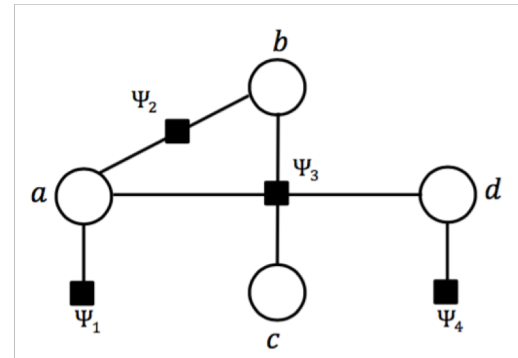
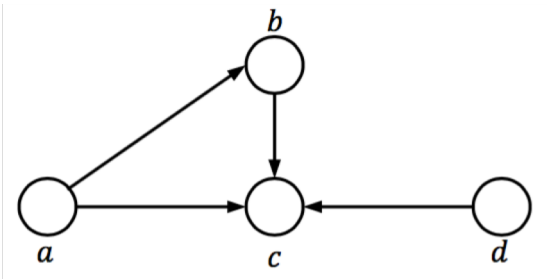
$$\Psi_2 = p(b|a)$$

$$\Psi_3 = p(c|a, b, d)$$

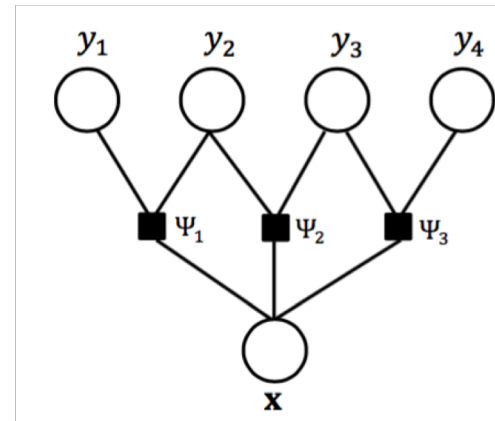
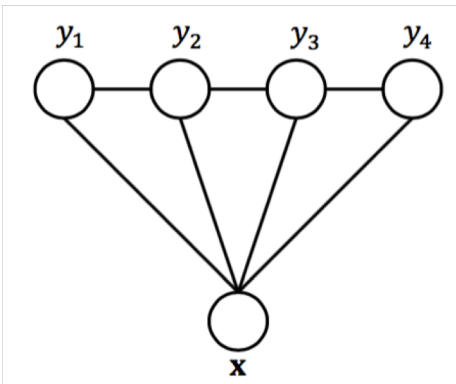
$$\Psi_4 = p(d)$$

Probabilistische graphische Modelle

- Gerichtete Modelle: Modellierung von kausalen Beziehungen



- Ungerichtete Modelle: Modellierung von inhaltlichen Beziehungen



Conditional Random Fields (CRFs)

- Form von ungerichteten probabilistischen graphischen Modellen
 - Konditionierung der Wahrscheinlichkeit einer Ausgabesequenz $\mathbf{y}$ auf die Eingabesequenz $\mathbf{x}$

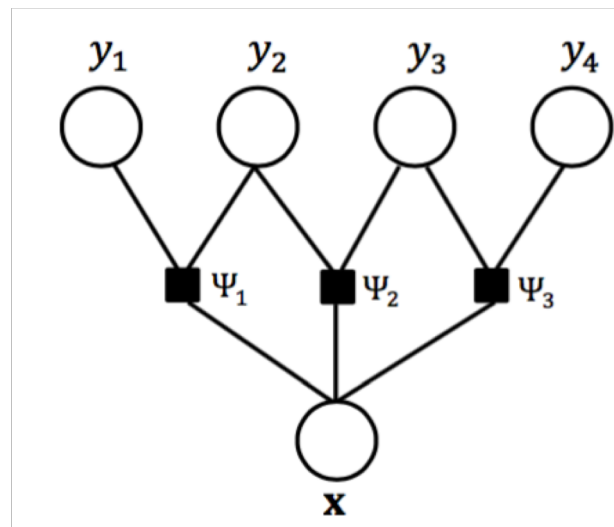
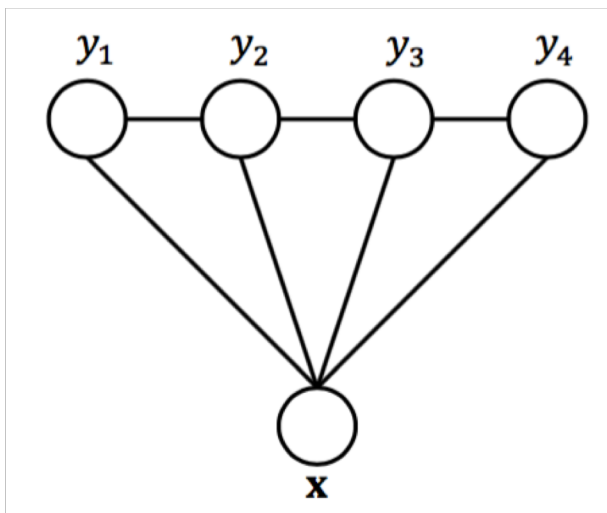
$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Psi_c(\mathbf{x}_c, \mathbf{y}_c)$$

- Keine Unabhängigkeitsannahmen zwischen Eingabevariablen
 - Komplette Eingabesequenz an jeder Stelle zugreifbar
 - $\Rightarrow$ Erfassung komplexer Merkmale möglich!

Conditional Random Fields (CRFs)

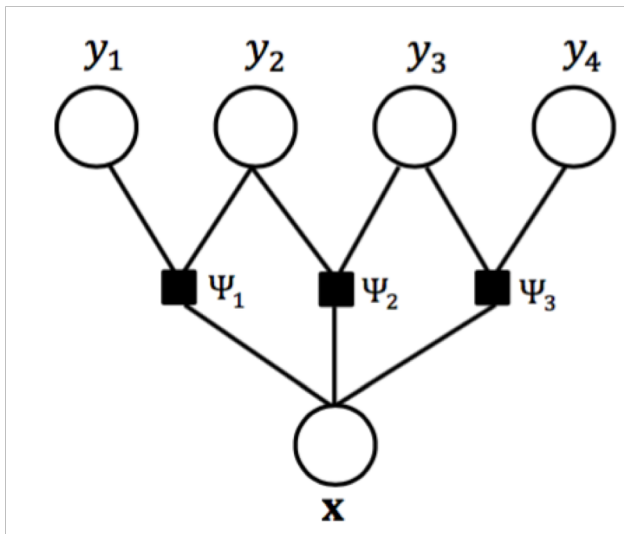
- Festlegung der Graphenstruktur mittels Cliquentemplates
 - Verbreitetste Form: Modellierung als lineare Kette

$$C = \{\Psi_j(y_j, y_{j+1}, x) | j = 1, \dots, n\}$$



Conditional Random Fields (CRFs)

- Modellierung der Wahrscheinlichkeiten mittels Potenzial- bzw. Featurefunktionen auf den Cliques
 - Messung der Kompatibilität der Ein- und Ausgabevariablen eines Faktors



$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C} \Psi_c(\mathbf{x}_c, \mathbf{y}_c)$$

$$\Psi_j(\mathbf{x}, \mathbf{y}) = \exp \left(\sum_{i=1}^m \lambda_i f_i(y_j, y_{j+1}, \mathbf{x}, j) \right)$$

$$Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod_{c \in C} \Psi_c(\mathbf{x}_c, \mathbf{y}'_c)$$

Aufgabenstellung

- Implementierung eines Sequenz-Labeling-Ansatzes zur Erkennung von Genen in Texten
- Verwendung einer bestehenden CRF-Implementierung:
 - MALLET (<http://mallet.cs.umass.edu/index.php>)
 - LingPipe (<http://alias-i.com/lingpipe/demos/tutorial/crf/read-me.html>)
 - BANNER (<https://sourceforge.net/projects/banner/>)
 - FACTORIE (<http://factorie.cs.umass.edu/>)
 - Sklearn-crfsuite (<https://sklearn-crfsuite.readthedocs.io/en/latest/>)
- Implementierungen unterscheiden sich stark hinsichtlich Schwierigkeits- und Dokumentationsgrad!

Aufgabenstellung

- Freie Gestaltungsmöglichkeiten bei der Feature-Auswahl
 - Wörterbucheinträge als Features?
 - Verwenden von POS-Tags?
 - Stemming / Lemmatization?
 - Groß- und Kleinschreibung beachten?
 - Kontextwörter beachten?
 -
- Anwendung von weiteren Heuristiken möglich
 - Gen-Wörterbuch als Post-Filter?
 - Filtern von häufigen False Positives / Negatives?
 - ...

Bereitgestellte Ressourcen

- Gen-Wörterbuch mit multi-token Genen
 - https://hu.berlin/ue_masprach1819_ue6_gen_dict
- Annotierter Trainings- und Validierungskorpus
 - https://hu.berlin/ue_masprach1819_ue6_train
 - https://hu.berlin/ue_masprach1819_ue6_val
- Evaluationsskript / -programm
 - https://hu.berlin/ue_masprach1819_ue6_eval

```
java -jar uebung6-eval.jar \  
    <goldstandard-file> <prediction-file>
```

Ausgabe Evaluationsprogramm

Mismatches

Kennzahlen

Document: 3965

FP (IL-2 with its receptor,607745,607767)

FN (IL-2,607745,607749)

Document: 224

FN (endo-beta-1,4-glucanase-encoding genes,35209,35247)

FN (celCCC,35261,35267)

Document: 538

FP (polysaccharide lyase family 1),84516,84547)

FN (pectate lyase superfamily,84484,84509)

FN (polysaccharide lyase family 1,84516,84545)

True Positives: 2437

False Positives: 1434

False Negatives: 4316

Precision: 0.629553

Recall: 0.360877

F1 Score: 0.458773

Programmaufruf

- Unterstützung eines Trainings- und einen Annotationsmodus
- Trainingsmodus:
 - Lernen des Modells basierend auf den Beispielen aus „*training_file*“
 - Speicherung des Modells unter „*model_name*“ (im aktuellen Ausführungsverzeichnis)

```
java -jar uebung6-gruppeX.jar train \  
    training_file model_name
```

```
python uebung6-gruppeX.py train \  
    training_file model_name
```

Aufbau des Trainingskorpus

- Korpus liegt im CONLL-Format vor
 - Ein Token je Zeile (inkl. POS-Tag und Label)

```
-DOCSTART- X X O

The DT O
P-ITIM-compelled JJ B-NP
multi-phosphoprotein JJ I-NP
complex JJ I-NP
binds NNS O
to TO O
and CC O
activates NNS O
SHP-2 NN B-NP
, , O
...

-DOCSTART- X X O
...
```

Programmaufruf

- Annotationsmodus:
 - Laden des gelernten Modells aus „*model_name*“
 - Einlesen der zu annotierenden Instanzen aus „*input_file*“
 - Anwendung des Sequenz-Labeling-Verfahrens
 - Ausgabe der annotierten Instanzen in „*output_file*“

```
java -jar uebung6-gruppeX.jar annotate \  
    model_name input_file output_file
```

```
python uebung6-gruppeX.py annotate \  
    model_name input_file output_file
```

Ein- und Ausgabeformat

Eingabe

```
-DOCSTART- X X O

The DT O
P-ITIM-compelled JJ O
multi-phosphoprotein JJ O
complex JJ O
binds NNS O
to TO O
and CC O
activates NNS O
SHP-2 NN O
, , O
which WDT O
in IN O
turn NN O

-DOCSTART- X X O
...
```

Ausgabe

```
-DOCSTART- X X O

The O
P-ITIM-compelled B-NP
multi-phosphoprotein I-NP
complex I-NP
binds O
to O
and O
activates O
SHP-2 B-NP
, O
which O
in O
turn O

-DOCSTART- X X O
...
```

Abgabe

- Abgabe eines ZIP-Archivs uebung6-gruppeX.zip
 - Ausführbares Programm und dessen Quellcode
 - Auflistung aller Dependencies (Python)
 - Ergebnis auf dem Validierungsdatensatz
- Testet Euer Programm vor der Abgabe!
 - F1-Score auf den Validierungsdaten muss ≥ 0.40 sein
- Abgabe bis spätestens 10.02.2019, 23:59 Uhr über:
 - https://hu.berlin/ue_masprach1819_ue6

Wettbewerb

- Die Lösung mit dem höchsten F1-Score gewinnt
 - Verwendung des Evaluationsprogramms
 - Laufzeit des Programms spielt keine Rolle!
- Die besten 3 Teams erhalten 6/4/2 Punkte