



KLAUSUR ZUM BACHELORMODUL  
„PROBEKLAUSUR COMPUTERLINGUISTISCHE ANWENDUNGEN“  
PROBEKLAUSUR,  
DR. BENJAMIN ROTH  
KLAUSUR AM

VORNAME:	<input type="text"/>
NACHNAME:	<input type="text"/>
MATRIKELNUMMER:	<input type="text"/>
STUDIENGANG:	<input type="checkbox"/> B.Sc. Computerlinguistik, <input type="checkbox"/> B.Sc. Informatik, <input type="checkbox"/> Magister
	<input type="checkbox"/> anderer:

Die Klausur besteht aus **10 Aufgaben**. Die Punktzahl ist bei jeder Aufgabe angegeben. Die Bearbeitungsdauer beträgt **90 Minuten**. Bitte überprüfen Sie, ob Sie ein vollständiges Exemplar erhalten haben. Tragen Sie die Lösungen in den dafür vorgesehenen Raum im Anschluss an jede Aufgabe ein. Falls der Platz für Ihre Lösung nicht ausreicht, benutzen Sie bitte **nur** die ausgeteilten Zusatzblätter! Verwenden Sie einen dokumentenechten Kugelschreiber oder Füller, **keine** Bleistifte. Es sind **keine Hilfsmittel** zugelassen. Geben Sie Programmcode immer in **Python** an. **Sie können Fragen auf Englisch bearbeiten**. Bitte tragen Sie **zuerst**, d.h., bevor Sie die Aufgaben lösen, auf **allen** Seiten Ihren Namen ein und füllen Sie die Titelseite aus.

Aufgabe	mögliche Punkte	erreichte Punkte
1. Naive Bayes Klassifikator	6	
2. Numpy Arrays	4	
3. Hyper-Parameter	4	
4. Matrix Repraesentation in Python	4	
5. ScikitLearn Vektorizers	4	
6. Maximum Entropy Klassifikator	4	
7. Wort-Co-Okkurrenzen	4	
8. Rekurrente Neuronale Netzwerke	8	
9. Keras Layers	9	
10. Relation classification	9	
Summe	56	
Note		

**Einwilligungserklärung (optional)**

Hiermit stimme ich einer Veröffentlichung meines Klausurergebnisses in der Vorlesung „PROBEKLAUSUR Computerlinguistische Anwendungen“ vom unter Verwendung meiner Matrikelnummer im Internet zu.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

NAME:

---

## Aufgabe 1 Naive Bayes Klassifikator

- (a) Wie lautet bei einem Binären Textklassifikator (Naive Bayes) das Entscheidungskriterium?
- (b) Was sind die sogenannten Log-Odds, und wie ergeben sie sich aus dem Entscheidungskriterium?
- (c) Erklären Sie das Konzept der **bedingten Unabhängigkeitsannahme** am Beispiel der Berechnung der Wahrscheinlichkeit für  $P(\text{Text}|\text{Label})$ .

6 PUNKTE

NAME:

---

## Aufgabe 2 Numpy Arrays

Vervollständigen Sie die Funktion `column_average`, die eine Matrix `m` als Argument nimmt, und einen Vektor zurückgibt, der das arithmetische Mittel (=den Durchschnittswert) aller Spalten in `m` enthält. Sowohl die Matrix `m` als auch der Rückgabektor sind Numpy Arrays.

```
import numpy as np

""" Return the average of all columns in an numpy Array
>>> mat = np.arange(6).reshape(3,2)
>>> mat
      array([[0, 1],
             [2, 3],
             [4, 5]])
>>> column_average(mat)
      array([ 0.5,  2.5,  4.5])
"""

def column_average(m):

    return
```

4 PUNKTE

NAME:

---

### Aufgabe 3 **Hyper-Parameter**

Nennen Sie 4 verschiedene Hyperparameter, und für jeden Hyperparameter einen Algorithmus bei dem er anwendbar ist.

4 PUNKTE

NAME:

---

## Aufgabe 4 Matrix Repräsentation in Python

Gegeben die 2 Matrix-Objekte m1 und m2, die durch die Python-Befehle unten erzeugt wurden.

Schätzen Sie das Verhältnis des benötigten Speicherplatzes für m1 und m2, d.h. wieviel mal mehr (oder weniger) Speicher benötigt m1 im Vergleich zu m2.

(Nehmen Sie an, dass für eine Zelle mit Wert  $\neq 0$  in einer Sparse-Matrix 3-mal soviel Speicher benötigt wird, wie für eine Zelle in einer Dense Matrix.)

```
>>> from scipy import sparse
>>> import numpy as np
>>> rnd = np.random.RandomState(seed=123)
>>> m1 = rnd.uniform(low=0.0, high=1.0, size=(100, 100))
>>> m1[m1<0.99] = 0
>>> m2=sparse.csr_matrix(m1)
>>> m2.nnz
100
```

4 PUNKTE

NAME:

---

**Aufgabe 5 ScikitLearn Vektors**

Gegeben der unten erstellte DictVectorizer D und Numpy Array X. Was ist der Inhalt von X?  
(Geben Sie konkrete Zahlwerte an).

```
>>> from sklearn.feature_extraction import DictVectorizer
>>> v = DictVectorizer(sparse=False)
>>> D = [{'foo': 2, 'bar': 1}, {'foo': 1, 'baz': 3}]
>>> X = v.fit_transform(D)
>>> v.get_feature_names()
['bar', 'baz', 'foo']
```

4 PUNKTE

NAME:

---

**Aufgabe 6 Maximum Entropy Klassifikator**

Die Softmax Funktion nimmt einen Vektor  $\vec{x}$  als Argument, und liefert einen Vektor  $\vec{y}$  als Funktionswert zurück.

(a) Welche Eigenschaften hat der zurückgelieferte Vektor immer (im Vergleich zum Argumentvektor)?

(b) Geben Sie die Formel zur Berechnung Softmax-Funktion an, und beschreiben Sie alle verwendeten Variablen.

4 PUNKTE

NAME: \_\_\_\_\_

## Aufgabe 7 Keras Layers

- Entscheiden Sie für jede der folgenden Modell-Architekturen in Keras, für welche der angegebenen Aufgaben sie geeignet ist. Geben Sie jeweils eine kurze Begründung.

```
from keras.models import *  
from keras.layers import *
```

- Architektur 1**

```
model = Sequential()  
model.add(Embedding(input_dim=10000, output_dim=300))  
model.add(Bidirectional(LSTM(units=300, return_sequences=False)))  
model.add(GlobalMaxPooling1D())  
model.add(Dense(units=1, activation='sigmoid'))
```

- ☐ keine mögliche oder keine sinnvolle Architektur
- ☐ Sequenz-Tagging
- ☐ Multiklassen-Klassifikation
- ☐ Binäre Klassifikation
- ☐ Regression

**Begründung:**

- Architektur 2**

```
model = Sequential()  
model.add(Dense(units=1, activation='linear'))
```

- ☐ keine mögliche oder keine sinnvolle Architektur
- ☐ Sequenz-Tagging
- ☐ Multiklassen-Klassifikation
- ☐ Binäre Klassifikation
- ☐ Regression

**Begründung:**



NAME:

---

- **Architektur 3 (Hinweis:** TimeDistributed() wendet eine nicht-sequenzielle Layer einzeln auf jeden Zeitschritt in einer Sequenz an)

```
model = Sequential()  
model.add(Embedding(input_dim=10000, output_dim=300))  
model.add(Conv1D(filters=100, kernel_size=3, padding='same'))  
model.add(TimeDistributed(Dense(units=12, activation='softmax')))
```

- ☐ keine mögliche oder keine sinnvolle Architektur
- ☐ Sequenz-Tagging
- ☐ Multiklassen-Klassifikation
- ☐ Binäre Klassifikation
- ☐ Regression

**Begründung:**

9 PUNKTE

NAME:

---

## Aufgabe 8 **Rekurrente Neuronale Netzwerke**

- (a) Beschreiben Sie kurz die Eingabe sowie alle Layers der in der Vorlesung vorgestellten einfachen RNN Architektur (Vorhersage von Satz-Sentiment). (Je 1-2 Sätze je Layer.) Ergänzen Sie die Beschreibung durch eine Skizze.

NAME:

---

(b) Betrachten Sie folgenden zwei Alternativen, um einem Keras-Modell eine LSTM Layer hinzuzufügen (z.B. nach einer Embedding Layer).

- Für welche computerlinguistischen Anwendungen ist Variante (1) nötig, für welche Variante (2). Nennen Sie je ein Beispiel.

(1) ...  
`model.add(LSTM(units=100))`  
...

(2) ...  
`model.add(LSTM(units=100, return_sequences=True))`  
...

8 PUNKTE

NAME:

---

## Aufgabe 9 Relation classification

- (a) Definieren Sie die Aufgabe der Relationsvorhersage (Relation classification). Was ist die Eingabe, was die vorherzusagende Ausgabe? (3-6 Sätze)
- (b) Beschreiben Sie kurz einen merkmalsbasierten Ansatz zur Relationsvorhersage. (3-6 Sätze)
- (c) Beschreiben Sie kurz einen Ansatz zur Relationsvorhersage mit neuronalen Netzwerken. (3-6 Sätze)

NAME:

---

**Aufgabe 10 Wort-Co-Okkurrenzen**Betrachten Sie den Satz: **she told me she would**

Zählen Sie (von Hand) die Anzahl der Co-Okkurrenz-Vorkommen (co-occurrence counts) aller Wortpaare in einem Kontext mit maximalem Abstand von 1 (sowohl links als auch rechts), nach dem in der Vorlesung vorgestellten Schema. Stellen Sie das Ergebnis als Matrix dar und beschriften Sie alle Zeilen und Spalten der Matrix entsprechend.

4 PUNKTE