Beleggüte

Ulf Hamster
Berlin-Brandenburgische Akademie der Wissenschaften

Übersicht

Trainingsdaten erzeugen

- Rankingdaten mit Best-Worst-Scaling sammeln
- Sampling von Satzbeleggruppen
- Verzerrungen durch die Benutzeroberfläche
- Scores aus Rankings erzeugen

Entwurf eines Active Learning Systems

Scoring-Modell entwickeln

Was sind unsere Scores, Labels, …?

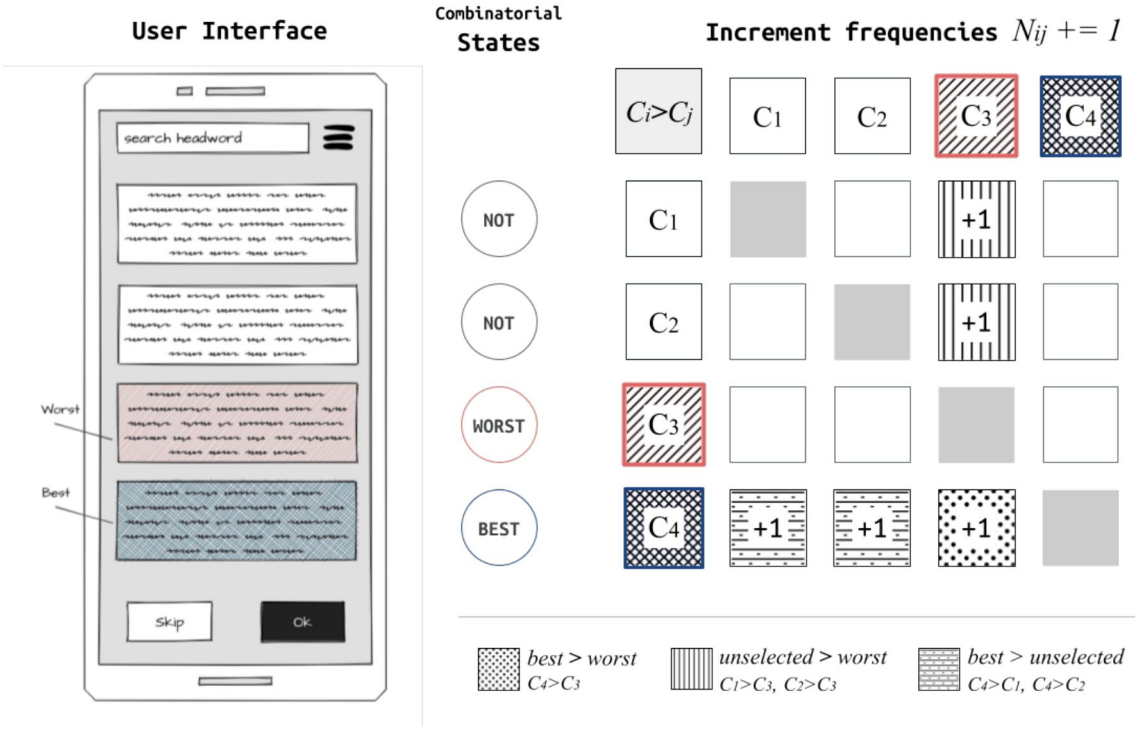
Skalenniveau	1. Moment Operationen	Beschreibung	Modellierung
Verhältnisskala (ratio-scaled)	Mittelwert f < = > + − * /	natürliche 0 existiert nicht. Preis in € (kostenlos), Geschwindigkeit in km/h (Stillstand), Entfernung in m (dieselbe Position), Masse in kg (masselos).	Regression
Intervallskala (interval-scaled)	Mittelwert f < = > + − /	Die Intervalle sind vergleichbar "Alter = Aktuelles Jahr - Geburtsjahr" (aus 2 Verhältniskennz.) "Alter_A - Alter_B" (vergleiche Intervallkennz.) "(Alter_A - Alter_B) / (Alter_C - Alter_D)"	Regression
Ordinalskala (ordinal-scaled)	Median f < = >	Die Abstandswerten zw. Merkmalen sind unbekannt, nicht interpretierbar, ungenau. Schulnoten: sehr gut > gut > … > ungenügend Hierarchien: Besitzer > Chef > … > Meister > …	Learning to Rank *Contrastive Loss Function *Klassifikation (Thermometer Trick) *Regression (Ranks to Scores)
Nominalskala (nominal-scaled)	Modalwert f (Frequenz)	Anzahl der Merkmale ist oft bekannt, z.B. Steuerklassen, Blutgruppe, u.a. Manchmal offen, z.B. Familiennamen, Lesarten, Berufe, u.a.	Klassifikation

Trainingsdaten sammeln mit Best-Worst Scaling

```
pip install bwsample>=0.7.0
```

Hamster, U. A., (2021). bwsample: Processing Best-Worst Scaling data. Journal of Open Source Software, 6(64), 3324, https://doi.org/10.21105/joss.03324

Extrahiere Paarvergleiche mit Best-Worst-Scaling (BWS)



2 Klicks (Worst, Best)

5 Paarvergleiche (1x Direkt, 4x Indirekt)

N_ij ist eine riesige, erweiterbare Sparse Matrix

https://github.com/satzbeleg/bwsample/blob/main/docs/count.ipynb

Mehr Items = Höherer Ertrag

Via Paarvergleich kann mit 1 Klick, 1 Paarvergleich erhoben werden.

M sei die Anzahl der Items (z.B. angezeigte Satzbelege).
Beim BWS werden mit 2 Klicks, f(M)=2*M-3 Paarvergleiche extrahiert (für M>=3).

Hier gilt “f(beta * M) > beta * f(M)”
=> Increasing Return of Scale
(Manager reden immer gerne über “Skalierbarkeit”. Das ist die Formel dazu …)

Die optimale Lösung wäre demnach M=infinity
… ABER …

… die kognitive Kapazität der Benutzer ist begrenzt

Die Anzahl der Items M, die ein Mensch verarbeiten kann, sei begrenzt [Mil56]

Die Umstände unter denen ein Mensch Information gleichzeitig Aufmerksamkeit schenken kann [Cow01, p. 114]; mögliche Einflussfaktoren seien

a) allgemeine Fähigkeiten der Person,

b) wie neu & komplex sind die Inhalte des Items aus Sicht der Person,

c) wie neu & komplex sind die anzuwendenden Entscheidungskriterien aus Sicht der Person,

c) kognitive Erschöpfung der Person im Laufe des Experiments.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63(2), 81–97. https://doi.org/10.1037/h0043158
Cowan, N. (2001). The magical number 4 in short-term memory: A reconsideration of mental storage capacity. Behavioral and Brain Sciences, 24(1), 87-114. https://doi.org/10.1017/S0140525X01003922

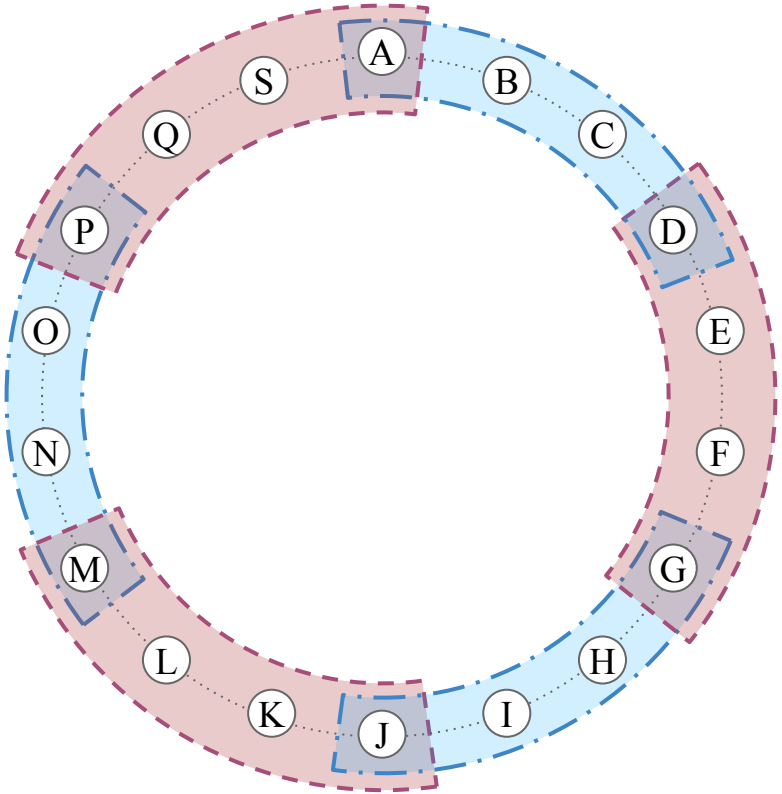
Sampling von Satzbeleggruppen für möglichst lange Vergleichsketten

Hamster, U. A. (2021, March 9). Extracting Pairwise Comparisons Data from Best-Worst Scaling Surveys by Logical Inference. https://doi.org/10.31219/osf.io/qkxej

Überlappend

Sample Satzgruppen, sodass jede Satzgruppe ein Beispiel (Satzbeleg) enthält, was genau 1x auch in einer anderen Satzgruppe enthalten ist.

- `I=4` Items pro Satzgruppe (BWS set)
- `S=6` Satzgruppen
- `M = (I-1)*S = 3*6 = 18` Beispiele




```
import bwsample as bws
samples = bws.sample(
    examples, n_sets=6, n_items=4, method='overlap')
```

https://github.com/satzbeleg/bwsample/blob/main/docs/sample.ipynb

Input & Befehl

```
# Satzbelege jeweils als JSON
examples = [
    {"id": "id1", "data": "data..."},
    {"id": "id2", "data": ["other", "data"]},
    {"id": "id3", "data": {"key", "value"}},
    {"id": "id4", "data": "lorem"},
    {"id": "id5", "data": "ipsum"},
    {"id": "id6", "data": "blind"},
    {"id": "id7", "data": "text"}
]

# Erzeuge Satzgruppen
samples = bws.sample(
      examples, n_sets=2, n_items=4, method='overlap')
```

Ergebnis

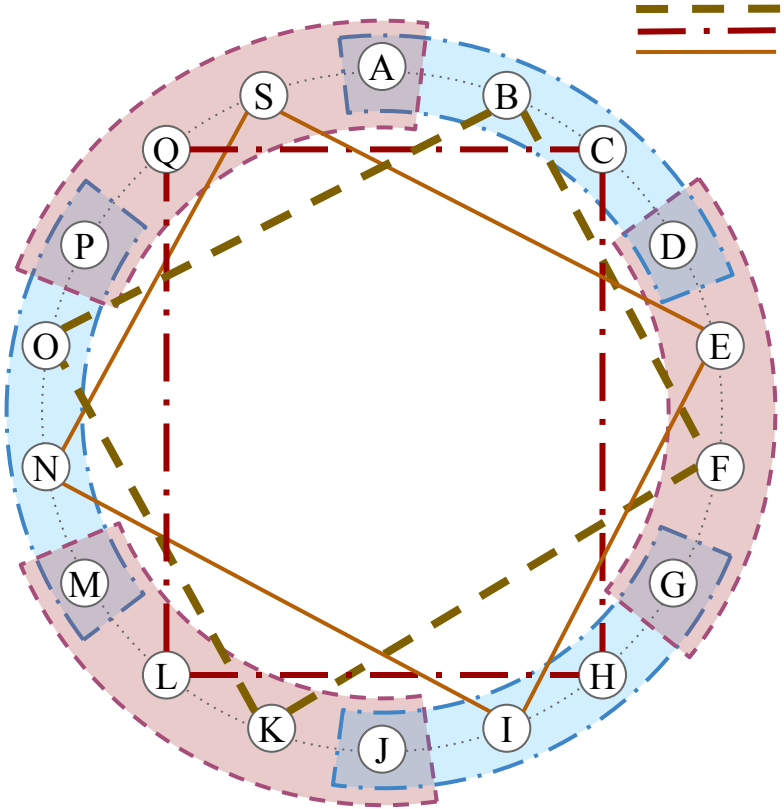
```
[
    [
        {'id': 'id1', 'data': 'data...'},
        {'id': 'id2', 'data': ['other', 'data']},
        {'id': 'id3', 'data': {'value', 'key'}},
        {'id': 'id4', 'data': 'lorem'}
    ],
    [
        {'id': 'id1', 'data': 'data...'},
        {'id': 'id4', 'data': 'lorem'},
        {'id': 'id5', 'data': 'ipsum'},
        {'id': 'id6', 'data': 'blind'}
    ]
]
```

Jedes Item 2-mal

Sample Satzgruppen, sodass jedes Beispiel (Satzbeleg) in 2 Satzgruppen enthalten ist.

- `I=4` Items pro Satzgruppe (BWS set)
- `S=6` überlappende Satzgruppen
- `Z = (I-2)*S/I = (2*6)/4 = 3` zusätzliche Satzgruppen

```
import bwsample as bws
samples = bws.sample(
    examples, n_sets=6, n_items=4, method='twice')
```



https://github.com/satzbeleg/bwsample/blob/main/docs/sample.ipynb

Random Sampling der Top-N Satzbelege

Vorgehen

1. Sortiere die Ersatzbelege nach dem höchsten Score
2. Wähle die Top-N Satzbelege aus
3. Shuffle die ausgewählten N Satzbelege

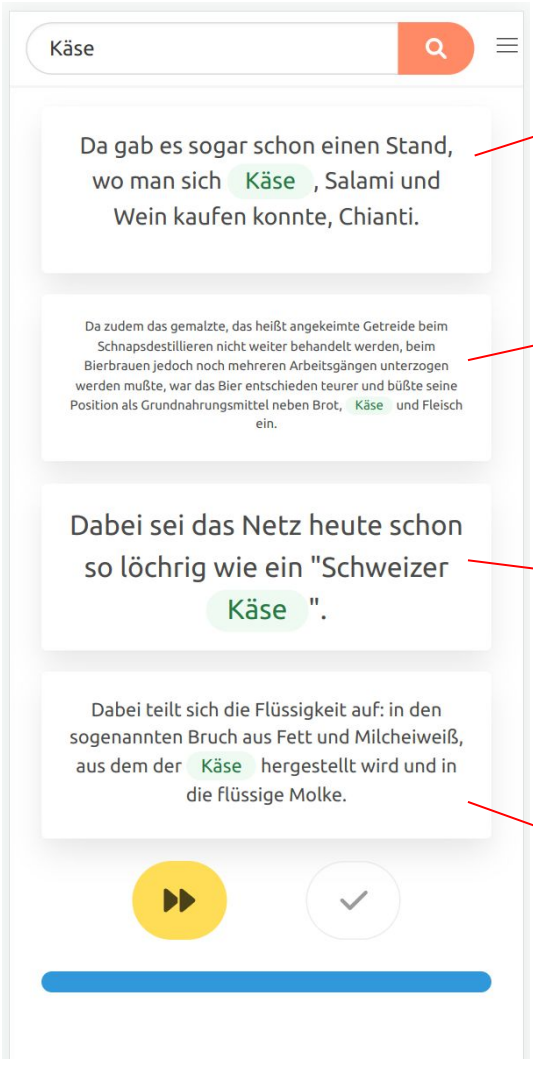
Systemverhalten

- 100% “Exploitation” (RL) => Belege mit niedrigen Scores verschwinden für immer
- Gegenmaßnahme: Sample X% zufällige Belege unabh. vom Score (“Exploration” [RL], “Mutation” [GA])

SQL Query mit N=100

```sql
SELECT tbl2.*
FROM (
    SELECT sentence_id
         , sentence_text
         , lemma
         , score
    FROM tbl
    ORDER BY score DESC
    LIMIT 100
) tbl2
ORDER BY RANDOM()
;
```

Verzerrungen durch die Benutzeroberfläche



eher mit Finger erreichbar

Wird sich wirklich jeder den langen Text mit der kleinen Schrift sorgfältig durchlesen?

Großer Text ist besser lesbar und fällt mehr auf

gut mit dem Daumen erreichbar

GermEval 2022 Task

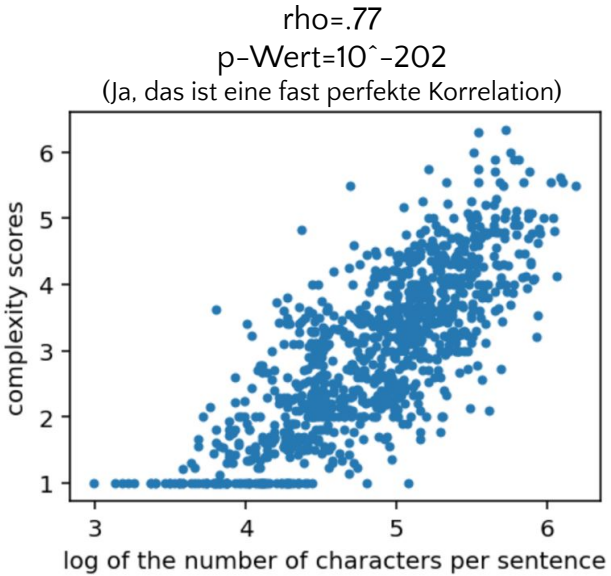
“complexity score” – Sprachlerner sollten bewerten wie “komplex” sie einen Satzbeleg wahrnehmen.

Im Studienverlauf entwickelten Teilnehmer die Zeichenlänge als Bewertungsheuristik (=Augen messen Bildschirmkontrast)

Anreizproblem:
Warum sollte sich ein Teilnehmer Mühe geben?

Gegenmaßnahme:
Sampling von Satzbeleggruppe mit ähnlicher Zeichenlängen.

Merke:
Teure Trainingsdaten können nur Wissen/Kalküle/Insights enthalten, die Annotatoren bei ihrer Entscheidung berücksichtigen.



Ulf A. Hamster. 2022. Everybody likes short sentences - A Data Analysis for the Text Complexity DE Challenge 2022. In Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text, pages 10–14, Potsdam, Germany. Association for Computational Linguistics.
https://aclanthology.org/2022.germeval-1.2/

Scores aus Rankings erzeugen

Hamster, U. A. (2021, April 1). Pairwise comparison based ranking and scoring algorithms. https://doi.org/10.31219/osf.io/ev7fw
Python Code: https://github.com/satzbeleg/bwsample/blob/main/bwsample/ranking.py

(a) Einfache Verhältniskennzahlen

(1) Vergleiche die Häufigkeit von Paar (i,j) versus (j,i)

$$\mu_{ij} = \frac{N_{ij}}{N_{ij} + N_{ji}} \; \forall \; i, \; j$$

Matrixschreibweise:

$$\mathbf{M} = \frac{\mathbf{N}}{\mathbf{N} + \mathbf{N}^T}$$

(2) Summiere die Zeilen auf

$$s_j = \sum_i \mu_{ij}$$

(3a) Score als MinMax-Scaling der s_j Werte

(3b) Sortiere Indizes j nach aufsteigenden Werten s_j (Rankings) $j^* = \text{sort}(s_j)$
und weise eine gleichverteilte Werte [a=0, b=1] hinzu.

(b) p-Werte des Pearson's Chi-Squared Tests

Alle widersprüchlichen Häufigkeiten N_ij und N_ji können als Ergebnis eines Münzwurf-Experiments betrachtet werden wofür der **Binomial-Test** anzuwenden ist. (Wir verwenden den Chi-Squared Test weil die Berechnung effizienter ist.)

- Verhältniskennzahl: $\mu = \frac{N_{ij}}{N_{ij} + N_{ji}}$
- Nullhypothese: $H_0 : \mu = 0.5$ *(Satzbelege i und j seien gleich gut)*
- Alternativhypothese: $H_a : \mu > 0.5$

(1) Berechne den p-value des Chi-Squared Tests (Je kleiner p-Werte, desto "signifikanter" wird H0 abgelehnt)
Und berechne folgende Metrik

$$x_{ij} = \begin{cases} 1 - p, & \text{if } N_{ij} > N_{ji} \\ 0, & \text{otherwise} \end{cases} \quad \forall\ i,\ j$$

(2) & (3) siehe vorige Folie

Elegantere Methoden

(c) Eigenvalues als Scores (Saaty, 2003)

(d) Simuliere als Transitionmatrix

(e) BTL-Modell via Maximum-Likelihoodschätzung (Bradly & Terry, 1952)

Saaty, T. L. (2003). Decision-making with the AHP: Why is the principal eigenvector necessary. European Journal of Operational Research, 145(1), 85–91. https://doi.org/10. 1016/S0377-2217(02)00227-8
Bradley, R. A., & Terry, M. E. (1952). Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. Biometrika, 39(3/4), 324–345. https://doi.org/10.2307/2334029

Sind einfache Methoden “gut genug”

manuell nachgeschaut:
Numerisch instabile Werte
(Eigenwert-Methode)

Korrelation mit dem
Vergleichsdatensatz
=> Alle Methoden funktionieren

	(b)	(c)	(d)	(e)	(f)	time
(a) simple ratio	.9997	-0.0133	1.	1.	1.	1.000
(b) approx p-value	-	-0.0110	1.	1.	1.	0.730
(c) eigenvector	-	-	1.	1.	1.	0.733
(d) transitions	-	-	-	1.	1.	12.077
(e) BTL model	-	-	-	-	1.	0.743
(f) scores from [KM17]	-	-	-	-	-	-

Transitionmatrix schätzen ist sehr teuer

deutlicher Speed-Up durch Sparse-Matrix

Table 2: *Spearman rank correlations between the five procedures and the reference scores. The computation run time is represented as a multiple of the run time for computing the simple ratios based scores.*

Online Learning (A, B) vs. Einmaliges Training (D, C, E)

Update der Häufigkeitsmatrix, z.B. neue Rankings der User

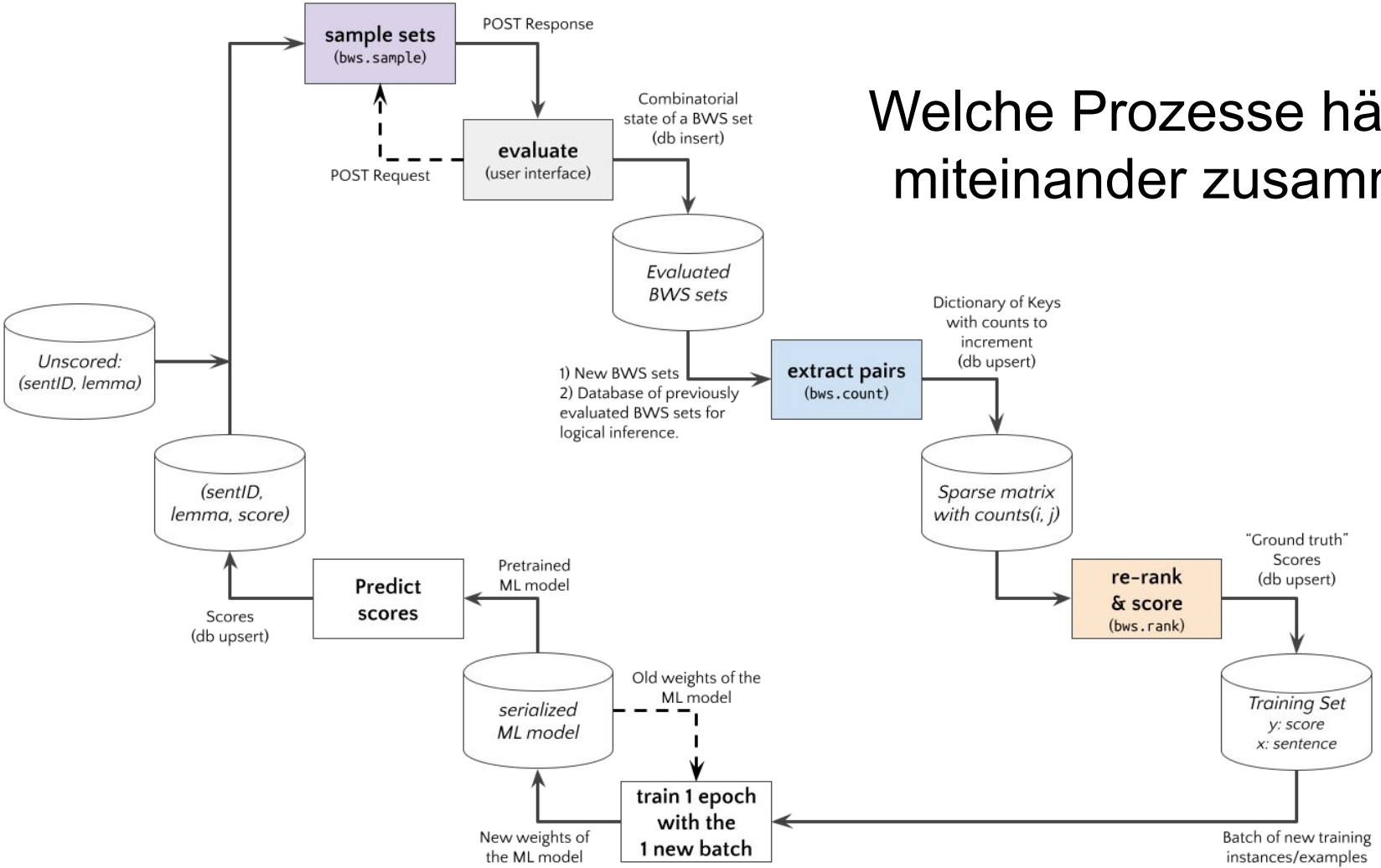
- Bedarf eine komplette Neuberechnung (C) des Eigenvalue-Problems, (D) Schätzung des Transitionmatrix, oder (E) MLE-Schätzung des BTL-Modells.
- Bei (A) & (B) müssen nur die betroffenen Paare (i,j) aktualisiert werden

Wenn man nur 1-mal eine Studie auswerten möchte …
=> Training mit allen Daten möglich, z.B. BTL-Modell

Wenn das Training inkrementell, online, in Etappen erfolgt …
=> Keep It Simple & Stupid (KISS), z.B. elementweise Ops statt Matrixmultiplikation

Entwurf eines Active Learning Systems

Welche Prozesse hängen miteinander zusammen?

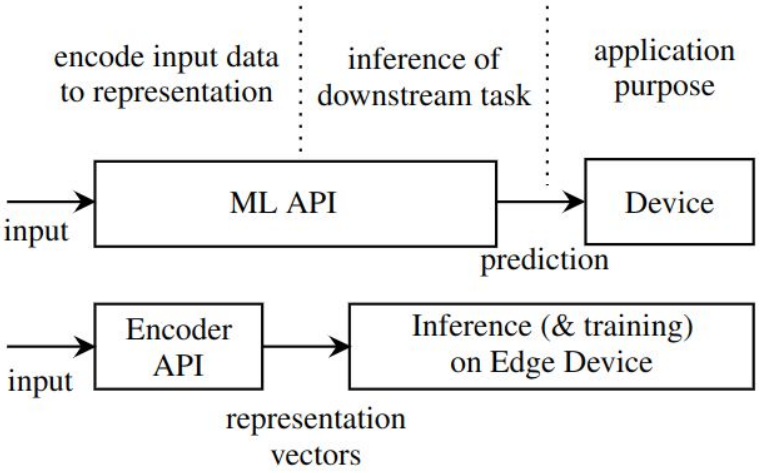


Wo sollen Prozesse stattfinden?

Lexikografen sollen ihre eigenen, individuellen Scoring-Modelle trainieren.

Rankingdaten können an zentralen Server gesendet, aber müssen nicht (z.B. DSGVO)

Es wird zwar ein zentrales Scoring-Modelle mit den gespendeten Rankingdaten trainiert, aber es ist lediglich das Startmodell für die benutzerspezifischen Modelle im Edge-Device



Flow-Diagram für Inference & Training auf Edge Device

Planungen werden nun so detailliert, dass sie beim Programmieren hilfreich sind.

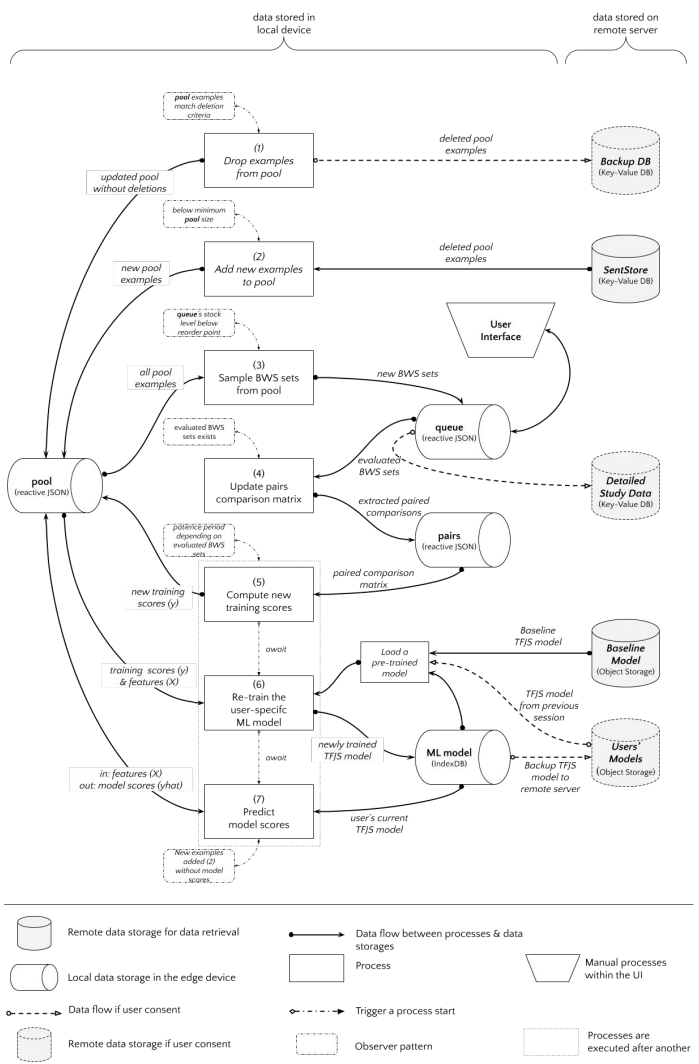
Implementation mit TFJS in der Web-App (Vue.js)

Begrenzter Speicher => Wir müssen Items löschen (& sichern)

Begrenzte Rechenkapazität => So viel wie möglich serverseitig vorberechnen & cachen

Re-Training parallel im Hintergrund, aber muss dennoch “flott” sein (z.B. wenige Epochs)

https://github.com/satzbeleg/evidence-app/blob/main/src/components/bestworst/interactivity.js#L852



Scoring-Modell entwickeln

Starte mit einem einfachen & interpretierbaren Modell

Manuelles Features Engineering
https://github.com/satzbeleg/evidence-features#features-overview

1181 Features + 1x Linear Layer (Simple Regression)

- einfach zu implementieren & und zu debuggen
- Rechenkapazitäten eines SmartPhones sollen immer ausreichen

XAI-Modell, z.B. Random Forest

- die Features sollten interpretierbar sein
- zum Beispiel ein SBert liefert zwar 768 Elemente, aber ist zusammen nur als “Semantik” interpretierbar.

Kompliziertere NN-Modelle (nächste Woche)