Untersuchung maschineller Lernverfahren zur Übersetzung natürlicher Sprache in SQL-Befehle

Kevin Friedl

Problemstellung

Relationale Datenbanken, welche in den 70er eingeführt wurden, stellen heutzutage die häufigste Methode der Datenhinterlegung dar. Um diese Datenbanken anzusprechen wird meist die Sprache SQL verwendet. SQL ist zwar für Endnutzer vorgesehen, dennoch sind die Anfragen welche sich über mehrere Table erstrecken meist zu komplex für die meisten Endnutzer. Auch kann nicht jede Anfrage über eine grafische Oberfläche übersichtlich abgebildet werden.

Problemstellung - SQL

Die dargestellte Query gibt alle Informationen über Studenten der Klasse mit der höchsten Anzahl an Studenten aus. Dies ist zwar sprachlich einfach zu beschreiben, aber das Schreiben dieser Query ist für fortgeschrittene SQL Programmierer.

```
SELECT *
FROM students
WHERE class_id = (
    SELECT id
    FROM classes
    WHERE number_of_students = (
        SELECT MAX(number_of_students)
        FROM classes));
```

Nested Query (aus learnsql.com)

Problemstellung

https://blog.einstein.ai/talk-to-your-data-one-model-any-database/

Interpretierung natürlicher Sprache um daraus SQL Anfragen zu stellen.

Zielsetzung

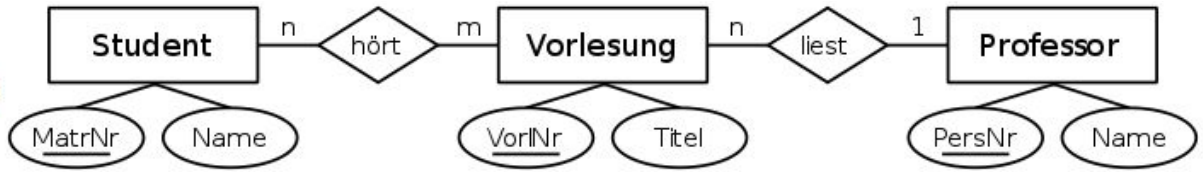
- Neuronale Verfahren erklären
- Schnittstelle in Form einer Webseite und API entwickeln
- Mit Klausuren aus dem Fach Datenbanken prüfen wie Leistungsfähig aktuelle Verfahren sind
- Eine eigene Meinung zu der Nützlichkeit der Verfahren bilden

SQL

SQL besitzt vier Kategorien an Befehlen:

- **Abfrage von Informationen - Data Query Language (DQL)**
- Änderung (Manipulation) von Informationen - Data Manipulation Language (DML)
- Änderung des Schemas - Data Definition Language (DDL)
- Kontrolle der Rechte - Data Control Language (DCL)

SQL



Student

MatrNr	Name
26120	Fichte
25403	Jonas
27103	Fauler

hört

MatrNr	VorlNr
25403	5001
26120	5001
26120	5045

Vorlesung

VorlNr	Titel	PersNr
5001	ET	15
5022	IT	12
5045	DB	12

Professor

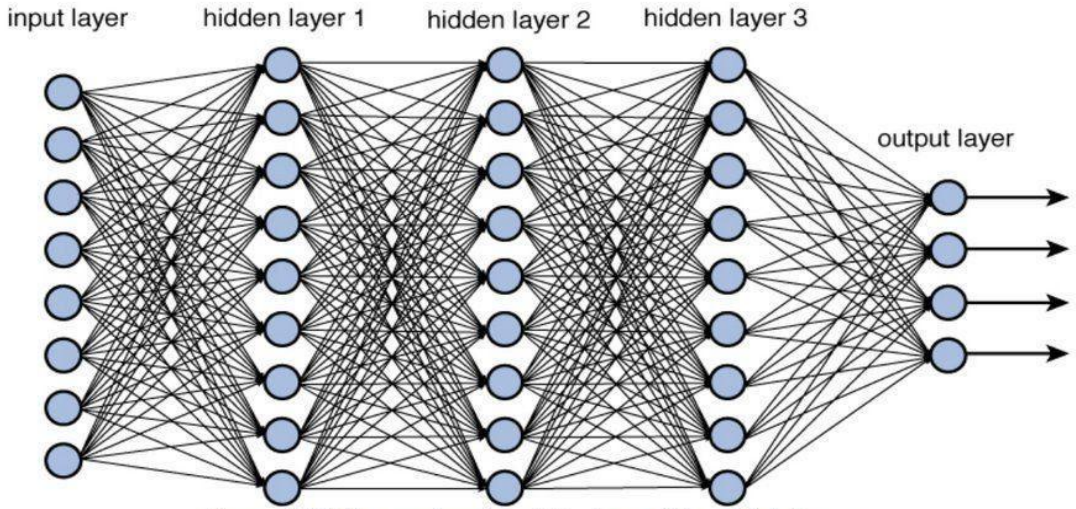
PersNr	Name
12	Wirth
15	Tesla
20	Urlauber

Beispielhafte SQL Relationen (aus Wikipedia: SQL)

SQL - Verknüpfungen

```
SELECT Vorlesung.VorlNr, Vorlesung.Titel, Professor.PersNr, Professor.Name
FROM Professor, Vorlesung
WHERE Professor.PersNr = Vorlesung.PersNr;
```

Die in der obigen Abbildung dargestellte SQL Anfrage gibt Vorlesungsnummer, Vorlesungstitel, Personalnummer und Name von allen Professoren, welche einer Vorlesung zugeteilt wurden, aus.



Grundlagen neuronaler Netzwerke

Die blauen Kreise stellen Neuronen dar und die Linien die Verbindungen zwischen ihnen. Beim “Lernen”, also das Trainieren eines neuronalen Netzes, werden die zu lernenden Daten auf den Input-Layer und die Lösung auf den Output-Layer gelegt.

Tiefes neuronales Netzwerk graphisch dargestellt (aus: towardsdatascience.com)

Grundlagen neuronaler Netzwerke

BERT

- Es wurde ein Satz übergeben bei dem zwei Worte maskiert wurden
  Eingabe: The [Maskiert1] brown fox [Maskiert2] over the lazy dog
  Geforderte Ausgabe: [Maskiert1] = quick, [Maskiert2] = jumped

- Es wurden zwei Sätze übergeben und BERT soll angeben, ob der zweite Satz nach dem ersten folgt
  Eingabe: Satz A: Max is a cool dude. Satz B: He lives in San Francisco
  Geforderte Ausgabe: True

BERT wurde mit 800 Millionen Wörtern aus dem Google BooksCorpus und 2 500 Millionen Wörtern des englischen Wikipedia trainiert.

Grundlagen neuronaler Netzwerke

LSTM

- “Long short-term memory” (kurz LSTM) ist eine Art von Neuronen, welche es ermöglicht beliebig lange Eingaben zu verarbeiten
- Wie der Name verrät, kann das neuronale Netzwerk sich Informationen merken

Datensätze

WikiSQL

- Der größte Datensatz mit rund 87 000 exemplarischen Beispielen von natürlicher Sprache
- Die Fragen wurden auf Datenbanken (ca. 24 000) von Wikipedia angewandt
- In der ersten Phase entwickelt ein Arbeiter eine natürlichsprachliche Frage, basierend auf einer SQL Anfrage, welche auf der Datenbank generiert wird. Diese SQL Anfrage wird zufällig generiert
- Danach wird die erstellte Frage mit der Antwort von zwei Arbeitern verglichen
- Wenn beide Arbeiter der Meinung sind, dass diese stimmt, wird sie in WikiSQL übernommen

Datensätze

WikiSQL

Da verschiedene SQL Befehle für den gleichen Output sorgen, unterscheidet der WikiSQL Benchmark unter:

- Der logischen Form (logical form)
- Dem ausgeführten Ergebnis (execution accuracy)

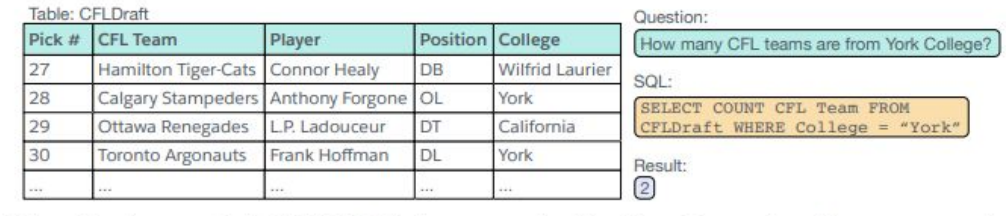


Table: CFLDraft

Pick #	CFL Team	Player	Position	College
27	Hamilton Tiger-Cats	Connor Healy	DB	Wilfrid Laurier
28	Calgary Stampeders	Anthony Forgone	OL	York
29	Ottawa Renegades	L.P. Ladouceur	DT	California
30	Toronto Argonauts	Frank Hoffman	DL	York
...	...	...	...	...

Question:
How many CFL teams are from York College?

SQL:
```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = "York"
```

Result:
2

Beispielhafter Auszug aus dem WikiSQL (Zhong et al.)

Datensätze

Spider

Besteht aus 200 verschiedenen Datenbanken mit 10 181 Fragen (Yu et al.)

- 138 unterschiedliche Domänen
- 5693 komplexe SQL Anfragen (Sortier-, Join- oder Gruppierungs-anforderungen)
- Verschiedene Schwierigkeitsgrade
- Von 11 Studenten der Yale Universität

Datensätze

Spider

Easy

What is the number of cars with more than 4 cylinders?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Meidum

For each stadium, how many concerts are there?

```
SELECT T2.name, COUNT(*)
FROM concert AS T1 JOIN stadium AS T2
ON T1.stadium_id = T2.stadium_id
GROUP BY T1.stadium_id
```

Hard

Which countries in Europe have at least 3 car manufacturers?

```
SELECT T1.country_name
FROM countries AS T1 JOIN continents
AS T2 ON T1.continent = T2.cont_id
JOIN car_makers AS T3 ON
T1.country_id = T3.country
WHERE T2.continent = 'Europe'
GROUP BY T1.country_name
HAVING COUNT(*) >= 3
```

Extra Hard

What is the average life expectancy in the countries where English is not the official language?

```
SELECT AVG(life_expectancy)
FROM country
WHERE name NOT IN
   (SELECT T1.name
    FROM country AS T1 JOIN
    country_language AS T2
    ON T1.code = T2.country_code
    WHERE T2.language = "English"
      AND T2.is_official = "T")
```

Figure 3: SQL query examples in 4 hardness levels.

SQL Anfragen Beispiel (aus der Spider Veröffentlichung)

Datensätze

Spider

Die Auswahl viel auf Spider:

- Sehr praxisnah - viele Datenbanken aus unterschiedlichen Domänen, somit wird ein generalisiertes Lernen erzielt
- Hohes wissenschaftliches Interesse
- Anspruchsvoll für aktuelle neuronale Netzwerke - hohe Komplexität der zu lernenden Anfragen

Neuronale Netze

Baseline Modelle

Seq2Seq

- Ist ein maschinelles Lernverfahrern, welches Texte in Texte umwandelt
- Ursprünglich für die Übersetzung von Sprachen entwickelt

SQLNet

- Nutzt ein Verfahren bei dem nicht die ganze SQL-Query, sondern nur die Parameter vorhergesagt werden

TypeSQL

- Die übergebene Anfrage wird vorverarbeitet, indem jedem Wort eine Kategorie zugeordnet wird

Neuronale Netze

Baseline Modelle

	Test					Dev
	Easy	Medium	Hard	Extra Hard	All	All
Example Split						
Seq2Seq	22.0	7.8	5.5	1.3	9.4	10.3
Seq2Seq+Attention (Dong and Lapata, 2016)	32.3	15.6	10.3	2.3	15.9	16.0
Seq2Seq+Copying	29.3	13.1	8.8	3.0	14.1	15.3
SQLNet (Xu et al., 2017)	34.1	19.6	11.7	3.3	18.3	18.4
TypeSQL (Yu et al., 2018)	47.5	38.4	24.1	14.4	33.0	34.4
Database Split						
Seq2Seq	11.9	1.9	1.3	0.5	3.7	1.9
Seq2Seq+Attention (Dong and Lapata, 2016)	14.9	2.5	2.0	1.1	4.8	1.8
Seq2Seq+Copying	15.4	3.4	2.0	1.1	5.3	4.1
SQLNet (Xu et al., 2017)	26.2	12.6	6.6	1.3	12.4	10.9
TypeSQL (Yu et al., 2018)	19.6	7.6	3.8	0.8	8.2	8.0

Zusammenfassung der Leistung der verschiedenen Baseline Modelle (aus der Spider Veröffentlichung)

Neuronale Netze

Fortgeschrittene neuronale Netze

RAT-SQL (Relation-Aware Schema Encoding and Linking for Text-to-SQL Parsers)

- Übergibt die Datenbankbeziehungen an das Netzwerk
- Ein verbessertes Verfahren, um die Spalten den Worten aus der Frage zuzuordnen

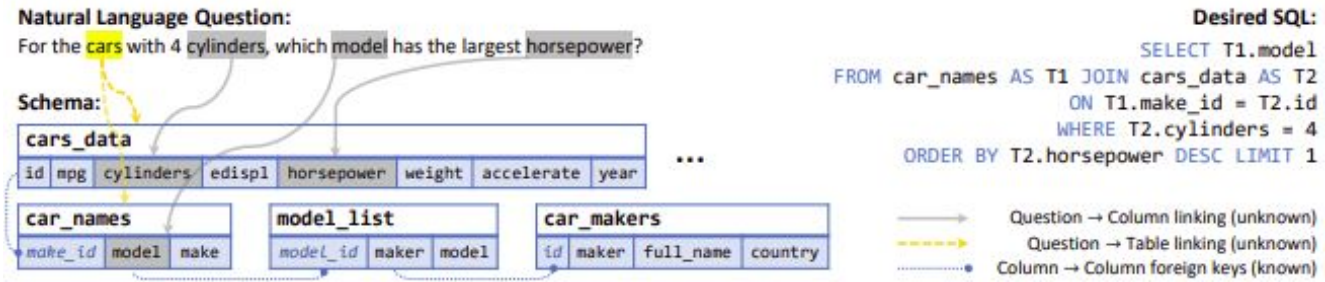
Performance auf Spider:

Mit BERT liegt der bei 65.6 % im Test-Set.

Neuronale Netze

Fortgeschrittene neuronale Netze

RAT



Beispiel einer schwierigen Spider Frage (aus der RAT-SQL Veröffentlichung)

Neuronale Netze

Fortgeschrittene neuronale Netze

GAP (Generation-Augmented Pre-training)

- Baut auf RAT-SQL auf
- Erweitert den Trainingsdatensatz durch selbst generierte Daten
  - Es wird eine verbesserte Spaltenzuordnung erreicht
  - Verschachtelte Fragen sollen zuverlässiger vorhergesagt werden

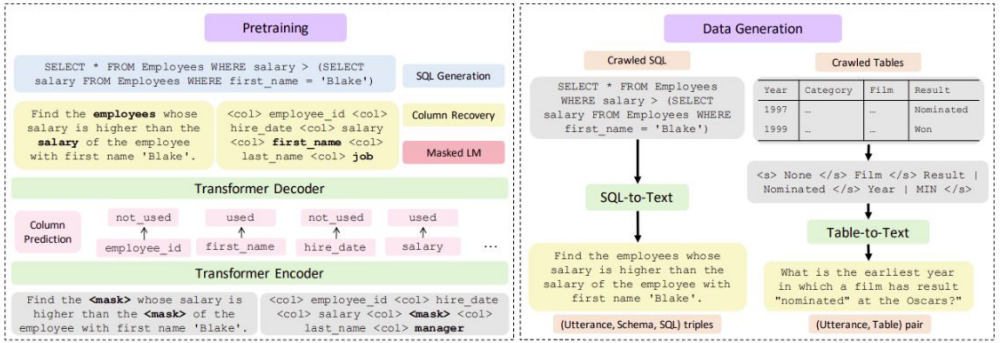
Performance auf Spider:

Mit BERT liegt der bei 69.7 % im Test-Set.

Neuronale Netze

Fortgeschrittene neuronale Netze

GAP



Aufbau von GAP Pretraining Komponenten (aus der GAP Veröffentlichung)

Neuronale Netze

Prototypische Implementierung

Demo: Bachelorarbeit - Untersuchung maschineller Verfahren zur Übersetzung natürlicher Sprache in SQL-Befehle

Diese Webseite dient zur Veranschaulichung des GAP-text2SQL Netzwerkes.

Funktionsweise: Stellen Sie bitte eine Anfrage und fügen Sie eine **.sqlite Datei** ein.

Eine Datei muss nicht angegeben werden. Es wird dann die zuletzt übergebene .sqlite Datenbank genutzt.

Bekannter fehler: Der Begriff 'terminal' erscheint in der Aussagabe. Grund hiefür ist das GAP keine Eingabewerte in der Ausgabe unterstützt.

Der Debug-Modus ist aktiviert. Fehler werden als Traceback ausgegeben.

.sqlite Datenbank: (optional) Choose File No file chosen

Ihre Anfrage: In englisch Vorhersagen

Prediction:

```
Query: where is the best restaurant in bay area for american food ?

Columns: *, id, campus, location, county, year, campus, year, campusfee, year, campus, degrees, campus, discipline, year, undergraduate, graduate, c

SELECT campuses.campus
FROM   campuses
WHERE  campuses.campus = 'terminal'
   and campuses.county = 'terminal'
   and faculty.faculty = (SELECT max(faculty.faculty)
                          FROM   campuses
                          WHERE  campuses.campus = 'terminal'
                             and campuses.county = 'terminal')
```

- Kevin Friedl

Prototypische Implementierung von GAP als Webinterface

Neuronale Netze

Prototypische Implementierung

```python
from flask import Flask, request, render_template
from flask_cors import CORS
from sql_formatter.core import format_sql

app = Flask(__name__)
CORS(app)

def postProcessing(prediction, query):
    numbers = [int(s) for s in query.split() if s.isdigit()]
    prediction = format_sql(prediction)
    if numbers:
        prediction = prediction.replace("'terminal'", str(numbers[0]))
    return prediction

@app.route("/predict", methods=['GET', 'POST'])
def predict():
    query = request.form["query"]
    file = request.files["file"]
    if file:
        file.save("data/sqlite_files/singer/singer.sqlite")
    db_schema = load_db()
    columns = ""
    for column in db_schema.columns:
        columns += column.unsplit_name + ", "
    predictionRaw = infer(query, db_schema)
    code = postProcessing(predictionRaw, query)
    return "Query: " + query + "\n\nColumns: " + columns + "\n\n" + code

@app.route("/", methods=['GET', 'POST'])
def index():
    if not request.form:
        return render_template("index.html")
    else:
        prediction = predict()
        return render_template("index.html", prediction = prediction)
```

GAP Webserver Code

```
{% if prediction %}
<h3>Prediction: </h3>
<code style="white-space: pre">{{ prediction }}</code>
{% endif %}
```

Ausschnitt der index.html

Neuronale Netze

Leistung auf Datenbank Klausuren:

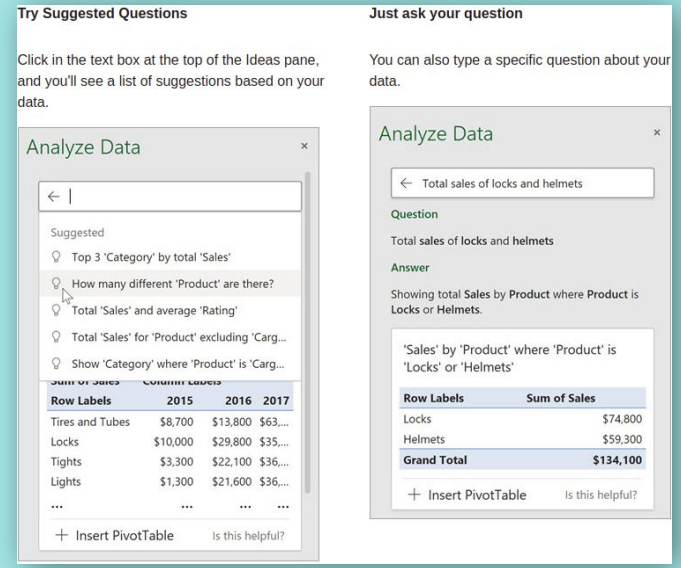
Insgesamt ist zu sagen, dass GAP maximal 2 der 4 Fragen, welche pro Klausur gestellt werden, teilweise richtig beantwortet. Da diese einfacher sind als die Fragen 3 und 4, erreicht GAP somit ca. 10% bis 30% der Punkte.

Frage orginal	Ausgabe aller Fahrzeuge vom Typ 'Cabrio', deren Kennzeichen mit 'N' beginnt.
Frage übersetzt	Output of all vehicles of the type 'Cabrio' with license plates beginning with 'N'.
GAP	SELECT vehicles.vehiclesnumber FROM vehicles join types WHERE types.type = 'terminal' and vehicles.licenseplate like 'terminal'
Gold	SELECT f.* FROM Fahrzeuge f JOIN Typen t ON f.TNr = T.TNr WHERE t.Typ = 'Cabrio' AND f.Kennzeichen LIKE 'N%'

Frage orginal	Geben Sie die Liste aller Kunden aus, die bisher noch kein Fahrzeug gebucht haben.
Frage übersetzt	Output the list of all customers who have not yet booked a vehicle.
GAP	SELECT customers.customersnumber FROM customers WHERE customers.customersnumber not in (SELECT bookings.customernumber FROM bookings)
Gold	SELECT KNr, Name FROM Kunden WHERE KNr NOT IN (SELECT KNr FROM Buchungen)

Neuronale Netze

Ausblick



Excel's natürlichsprachliche Eingabe (aus techcommunity.microsoft.com)

Eigene Meinung

Die Leistung von GAP bei Klausuraufgaben ist schlecht. Zudem ist die Zuverlässigkeit nicht gewährleistet. Wie es bei neuronalen Netzen generell der Fall ist, kann eine kleine Änderung der Eingabedaten für ein komplett anderes Ergebnis sorgen. Das heißt, dass inhaltlich gleiche Fragen aufgrund der Satzstellung für andere Ergebnisse sorgen können. Somit sehe ich **aktuelle Verfahren als unausgereift** an.

Danke für’s zuhören!