Datenbankpraktikum – Gruppe 3
Abschlusspräsentation

Janis Hamme, Karoly Kemeny, Thomas Neureuther,
Ming Niu, Alex Rubinshteyn, Xinxin Yang

7. Februar 2016

1. Abbildung Semantischer Relationen (Aufgabe 3.2)

These: *"Bei Anfragen nach Adjektiven werden viele Synonyme, aber nur wenige Antonyme gefunden."*

Analyse von 100 Anfragen nach den 100 ähnlichsten Wörtern für zufällig gewählte Adjektive mit WordNet. Verwendet wurde der Amazon Produkt Datensatz mit 200er-Vektoren. Einmal unpartitioniert und unterteilt in 16 Partitionen.

	Unpartitioniert	**Partitioniert**
Adjektive	48.21%	44.78%
Not Found	21.84%	24.59%
Synonyme	1.15%	0.43%
Antonyme	0.20%	0.11%
keine Relation	98.65%	99.57%

Tabelle 1: Analyse mit Wordnet.

Die Analyse (Tabelle 1) bestätigt zwar, dass sehr wenig Antonyme gefunden wurden, allerdings ist auch die Anzahl der Synonyme unter den Adjektiven erstaunlich klein. Die Partitionierung in 16 Partitionen scheint die Ergebnisqualität zusätzlich zu verschlechtern.

weatherproof	**marvelous**	**useless**
waterproof	splendid	superfluous
rainproof	magnificent	irrelevant
ip66	remarkable	pointless
all-weather	wonderful	bloated
durable	fascinating	**meaningless**
fire-resistant	monumental	obsolete
protected	breathtaking	wasted
damage-resistant	**extraordinary**	unreliable
outdoor-rated	phenomenal	low-quality
weather-tight	macabre	out-of-date

Tabelle 2: Auszug aus den Anfrageergebnissen. Jeweils ein Synonym wurde von WordNet erkannt (fettgedruckt).

Betrachtet man allerdings stichprobenartig die Anfrageresultate (Tabelle 2), zeigen sich Schwächen der Analyse mit WordNet:

- Viele Synonyme werden nicht erkannt, da die entsprechenden Relationen fehlen.
- Manche Wörter sind in der Bedeutung sehr ähnlich, aber kein Synonym.
- Nicht alle Wörter sind in der WordNet Datenbank enthalten: Umgangssprache, Markennamen, technische Bezeichnungen, zusammengesetzte Wörter.

Antonyme sind tatsächlich kaum vorhanden. Subjektiv bestätigen die Ergebnisse unsere These.

2. Umsetzung mit Hive und Hadoop (Aufgabe 3.1)

Drei verschiedene Anfragen an den Datenbestand wurden ausgewählt um die Umsetzbarkeit eines Recommenders mit Hive und Hadoop zu evaluieren.

- *Die hundert ähnlichsten Wörter zu einem gegeben Wort bezüglich der Kosinus Distanz.*
- *Die Kosinus Distanz zweier gegebener Wörter.*
- *Die hundert ähnlichsten Wörter zu einem gegebenen Wort, die zusätzlich dieses Wort beinhalten.*

Aufgrund verschiedener Probleme mit dem Hive-Server waren systematische Messungen leider nicht möglich. Manuelle Messungen auf den Amazon Produkt Daten mit 200er-Vektoren geben trotzdem ausreichend Aufschluss auf die Tauglichkeit:

- Laufzeiten in allen Fällen $\geq$ 200s, Varianz nach oben sehr groß.
- Weit entfernt von – für einen Recommender notwendigen – Echtzeitanforderungen.
- Vorteil durch Partitionen gering: Ohne Index trotzdem Full Table Scan notwendig und schlechte Balancierung.
- Ausführungszeit bricht bei Auslastung stark ein.

Um die Performancemessungen dennoch durchführen zu können haben wir ein Hadoop Cluster beim Cloudanbieter AWS angemietet. Wir haben uns für ein Cluster mit 2 virtuellen Cores Intel Xeon E5-2670 und 7,5GB Arbeitsspeicher entschieden. Es waren 1 Master und 2 Slave Knoten konfiguriert. Nach dem Deployment wurden die oben genannten Querys jeweils 5 mal pro Tabelle ausgeführt.

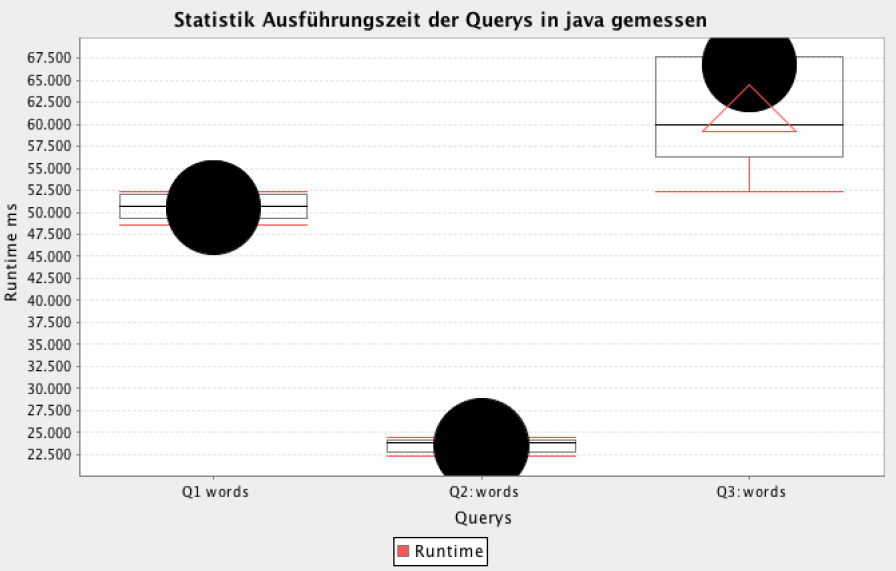

Abbildung 1: Performance Hive Querys auf Tabelle Words (unpartitioniert)

Die Messungen unterstreichen die bereits gezogenen Schlüsse.

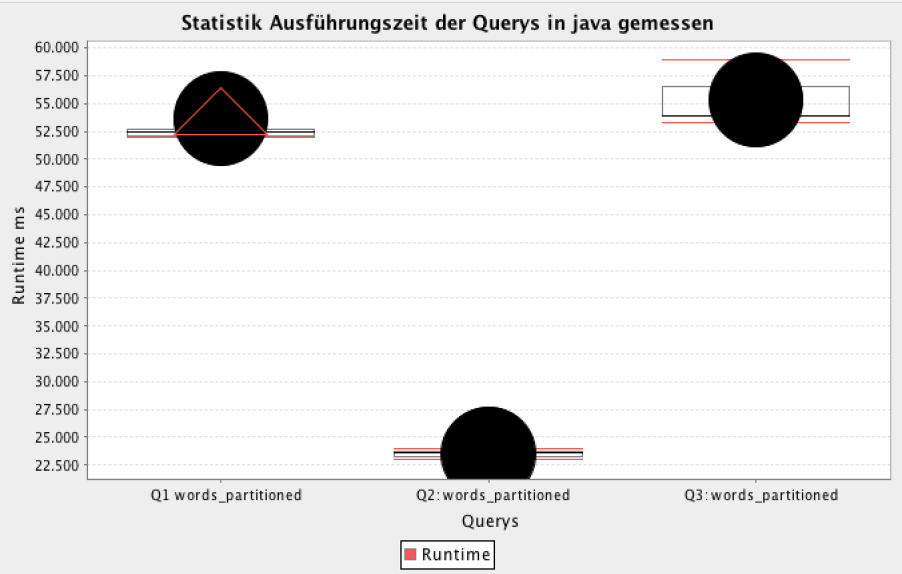

Abbildung 2: Performance Hive Querys auf Tabelle WordsPartitoned

3. Fazit (Aufgabe 3.3)

Word2Vec

- Scheint grundsätzlich benutzbare Technik für ein Recommender-System zu sein.
- Voraussetzungen: ein themenbezogener Datenbestand als Eingabe (z.B. Amazon Produkt Daten oder Reviews)
- Abbildung von semantischen Relationen: Bewertung anhand einer einzelnen These mit WordNet sehr schwierig.
  - Subjektiv liefert Word2Vec aber keine schlechten Resultate.
  - Bessere Möglichkeit zur systematischen Bewertung der Ergebnisqualität müsste gefunden werden.

Hive und Hadoop

- Nicht geeignet, da für einen Recommender zu langsam.
  - Datenbestand mit wenigen GB würde auch problemlos in Hauptspeicher passen. Noch wesentlich kleiner im Binärformat.
  - Hadoop mit Map-Reduce eigentlich eher für mehrere Terabyte große Daten sinnvoll.
- Nicht geeignet, da scheinbar sehr Instabil (zumindest bei uns). Ein Recommender muss mehrere Anfragen gleichzeitig abarbeiten können.
- Nicht geeignet für Clustering, da Mächtigkeit von Map-Reduce sehr eingeschränkt.
  - Balancieren von Partitionen vermutlich schlecht umsetzbar.