SAKI SS19 Homework 3

Author: Joachim Wagner

Program code: https://github.com/audiofrequenz/oss-saki-ss19-exercise-3.git

# Summary

Aufgabe 3 umfasste das erstellen einer automatisierten Lagereinheit mittels reinforcement learning. Hierbei wurde sowohl mit einem 2x2 Lager wie auch mit einem 2x3 Lager gearbeitet. Das Lager soll autonom mittels einem Roboter mit roten, blauen und weißen Containern verwaltet werden.

Zu beginn der Aufgabe war es notwendig sich einen Überblick über die benötigten Zustände zu verschaffen. Diese werden folgend zur Erarbeitung der Übergangswahrscheinlichkeitsmatrix benötigt. Diese Matrix wird mittels der Wahrscheinlichkeit der Änderung von einen Zustand in den nächsten Zustand erstellt. Es resultieren x (n\*n) Matritzen. „n“ ist hierbei die Anzahl der Zustände, x ist die Anzahl der Aktionen die durchgeführt werden können. In diesem Fall umfassen die Aktionen die Lagerposition in welchem ein Ablegen oder Aufnehmen eines Containers geschehen soll.

Um die Übergangswahrscheinlichkeit eines Zustandes in den nächsten zu definieren gibt es nachdem definiert wurde, welche Übergänge möglich sind, mehrere Möglichkeiten. Zum einen kann eine Gleichverteilung angenommen werden, zum anderen kann mittels eines Trainingssets an Daten eine Wahrscheinlichkeitsbestimmung geschehen. In unserem Fall war die Wahrscheinlichkeit, dass ein roter Container abgelegt oder geholt werden soll höher als die der beiden anderen Container-Arten. Diese so ermittelte Wahrscheinlichkeit wird folgend in die Übergangsmatrix eingetragen. Eine Reihe einer solchen n\*n Matrix darf in der Summe nur 1 ergeben. Um dies zu gewährleisten wurden die Summen überprüft und je nach über- oder unterschreiten entsprechend angepasst.

Nach der Ermittlung der Übergangswahrscheinlichkeits Matritzen wurden die Belohnungen definiert, welche sowohl positiv als auch negativ ausfallen kann. Reinforcement Learning benötigt diese Belohnungen, um die optimale Strategie zu errechnen und damit einhergehend eine möglichst hohe Belohnung zu erreichen.

Die Auswahl der Belohnungen hat Auswirkung auf das Verhalten des Reinforcement Algorithmus. Zunächst wurde die Auswahl so gewählt, dass sowohl der nächste freie Platz, im Falle des Ablegens eines Containers, als auch der erste Container mit der richtigen Farbe im Falle des Aufnehmens eines Containers den höchsten Reward bekommt.

Nachfolgend wurde diese Art der Belohnung überarbeitet, um die Leistung des Algorithmus zu testen.

Hierfür wurde….

Zum Trainieren des Algorithmus werden neben den Übergangswahrscheinlichkeiten und den Belohnungen noch die maximale Anzahl der Durchgänge sowie der Diskontierungsfaktor eingestellt. Dies dient dazu dass während des Lernprozess verhindert werden soll in eine unendliche Schleife zu gelangen. Die Wahl des Diskontierungsfaktors wurde im ersten Fall des Belohnungssystem mit 0,2 gewählt, spielte aber keine Rolle für das Ergebnis.

Als Algorithmen wurden zur Auswertung ValueIteration wie auch PolicyIteration gewählt.

Einen Test der angelernten Algorithmen wurde mittels eines Testdatensatzes gewährleistet. Hierfür wurde die errechnete Strategie der beiden Reinforcement Algorithmen angewendet um von einen Zustand in den nächsten zu gelangen.

Einen Überblick über die Performance wurden sowohl über die Zustände, die im Lagerhaus erreicht werden, ermöglicht, wie auch mittels der Anzahl der Schritte die ein Roboter benötigt anhand der Container Position. Da dies allein keine Aussage trifft ob ein Algorithmus gute Ergebnisse liefert wurde ein Greedy Algorithmus implementiert, welcher dieselben Aufgaben tätigen musste.

Die Ergebnisse der Algorithmen wurden schlussendlich miteinander verglichen.

# Evaluation

Die Evaluation der Algorithmen ergab im Falle der „nächst möglichen“ Verteilung der Belohnung dasselbe Ergebnis wie der Greedy Algorithmus (Abb. 1).

Die Ursache ist hierbei, dass sowohl Greedy als auch Value/Policy Iteration den nächsten freien Platz oder den nächsten Verfügbaren richtig gefärbten Container, die meiste Bedeutung zumessen.

Zu sehen ist dies an der Abfolge der Zustände, die alle drei Algorithmen einnehmen (Abb. 2).

Aufgrund dessen wurde das Belohnungssystem wie in Summary beschrieben angepasst. Daraus folge, dass sowohl die Schrittanzahl, die der Roboter durchlaufen muss als auch die Zustände, die er gegenüber des Greedy Algorithmus vollzieht, sich ändern (Abb. 3).

# Screenshot

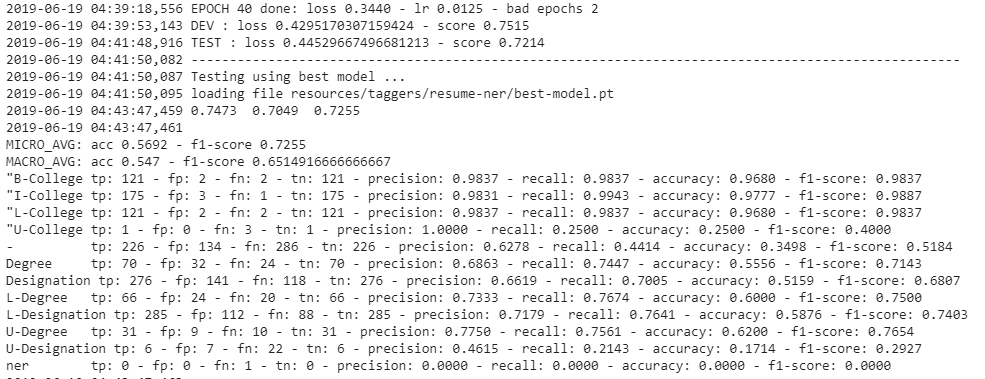


Abbildung 5 Evaluation Flair Character-Embedding

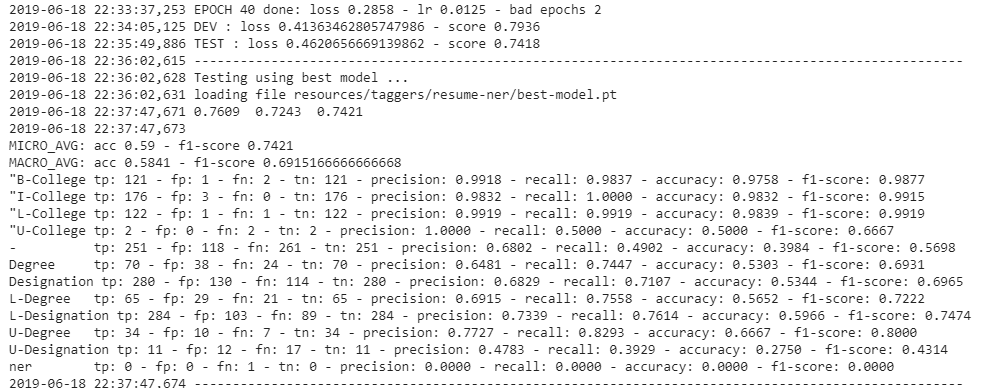


Abbildung 4 Evaluation Flair Word-Embedding



Abbildung 3 Prediction Scores spaCy

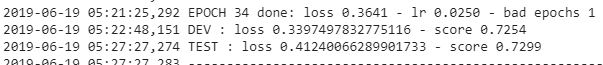


Abbildung 6 f1-Score combinated Embeddings



Abbildung 2 f1-Score spaCy



Abbildung 1 Accuracy spaCy