

Inspiziert von dem NYT first said - Twitterbot, haben wir im Zuge des Kurses *Legal Data Science* versucht, einen OGH - Twitterbot zu bauen.

@OGH\_first\_said ist ein Twitteraccount, der neue Wörter in aktuellen OGH Entscheidungen findet und diese tweetet.

Der Bot durchsucht das RIS nach neuen OGH Entscheidungen und vergleicht sie mit einer Datenbank von Wörtern, die sich aus allen Entscheidungen seit 1946 zusammensetzt.

Seit 1946 gibt es 136.245 Entscheidungen des Obersten Gerichtshofes.

Wir wollten wissen, inwiefern sich die Sprache im Laufe der Zeit verändert hat, ob heute überhaupt noch Begriffe in den Entscheidungen vorkommen, die in der Vergangenheit noch nie verwendet worden sind und wie oft dies der Fall ist.

Unser Projekt haben wir zunächst in drei Arbeitsschritte unterteilt:

- 1) get data
- 2) process data
- 3) tweet results

Im ersten Schritt geht es darum, eine Datenbank zu erstellen.

Dazu haben wir über die RIS API die benötigten Daten heruntergeladen, diese in das passende Format umgewandelt und alle Sonderzeichen entfernt.

Im Zuge dessen sind einige Schwierigkeiten aufgetreten:

Zum einen war es nicht möglich, mehr als 100 Einträge pro request herunterzuladen.

Um das zu lösen, haben wir einen while loop so oft hintereinander ausgeführt, bis alle Entscheidungen heruntergeladen waren.

Weiters wurde ab einer gewissen Anzahl an requests die verwendete IP-Adresse von den RIS Servern kurzzeitig geblockt, weshalb wir eine kurze Pause nach jeder Anfrage implementiert haben.

Im zweiten Schritt war nun das Ziel, die Entscheidungen zu verarbeiten. Hierbei kommt erneut ein loop zur Anwendung der eine Entscheidung nach der anderen ab arbeitet. Die Entscheidungstexte werden in einzelnen Wörter zerlegt und mittels einer Funktion werden zunächst alle Wörter in kleingeschriebene umgewandelt.

Weiters werden stopwords und nicht aus Buchstaben bestehende Teile aussortiert. Was dabei übrig bleibt wird zu einer Liste hinzugefügt. Diese Liste wird nun über einen for loop mit einer bestehenden Liste an bereits einmalig vorkommenden Wörtern, die als .txt hinterlegt ist, verglichen. Für den Fall, dass dieses Wort nicht in dieser Liste vorkommt wird es zur Liste hinzugefügt und ausgegeben.

Im letzten Schritt haben wir uns mit der Veröffentlichung der Ergebnisse durch einen Twitter Bot beschäftigt. Zuerst haben wir einen Account bei Twitter registriert und einen Developer Account beantragt. Über den Developer Account konnten wir auf die Twitter API zugreifen.

Die Twitter API, bzw die generierten API- und Access-Keys bilden hierbei die Schnittstelle zwischen unserem Python Script und dem Twitter Account. Mithilfe der Python Library *Tweepy* und den Keys konnten wir den ersten Tweet des Accounts über Python absenden. Um den ganzen Abfrage- und Analyse Prozess der Entscheidungen zu automatisieren, haben wir schlussendlich noch eine Funktion in AWS Lambda erstellt. Diese Funktion, welche unseren Code enthält, wird mithilfe eines EventBridge - Auslösers mehrmals täglich ausgeführt, sollte eine neue Entscheidung ein neues Wort enthalten, wird es sofort über unseren Twitter Account publiziert.