Vandalismus-Detektion von OpenStreetMap-Namen

Student

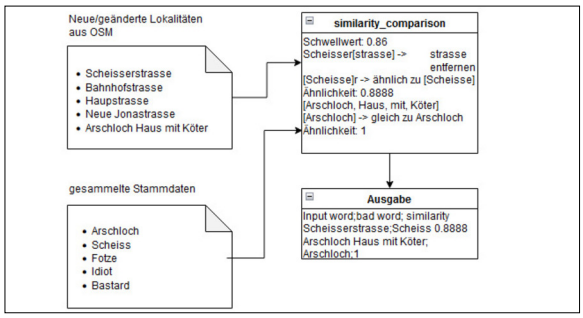
Davide Ferrara

Aufgabenstellung: Diese Arbeit gliedert sich in zwei Teile: Teil 1 beschäftigt sich mit der Zusammenstellung einer Datensammlung von Lokalitäten (Ortsnamen) im deutschsprachigen Raum, die in OpenStreetMap (OSM) als "Bad Edits" (=unerwünschte Änderungen) identifiziert wurden. Teil 2 besteht aus der Programmierung von Funktionen in Python, die die Ähnlichkeit von eingegebenen Wörtern mit Wörtern aus den "Bad Words"-Listen überprüfen. Damit soll festgestellt werden, ob es sich bei der Änderung in OSM um einen möglicherweise tatsächlich existierenden Ort handelt oder nicht. Im Verdachtsfall wird das Wort in eine CSV-Datei geschrieben, die dann manuell überprüft werden kann.

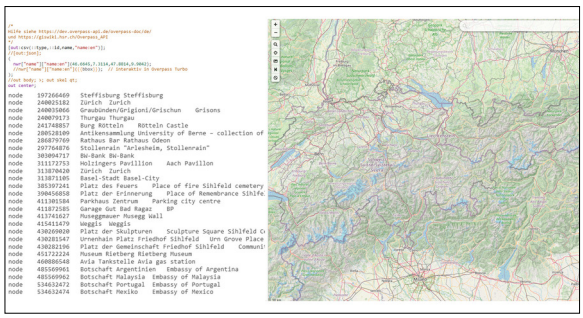
Vorgehen / Technologien: Im ersten Teil wurden mehrere Datensammlungen jeweils in Deutsch (de) und Englisch (en) erstellt. Aus dem Korpus "deTenTen 20" (de) des Portals Sketchengine wurden ca. 270 Wörter und aus Kaggle (en) ca. 1600 Wörter mit vulgären Ausdrücken zusammengestellt. 20 Bad Edits wurden aus OSM gesammelt, und zwar mit Hilfe des Online-Tools OSMCha, das sogenannte "Changesets" filtert und ausgibt. Die sehr geringe Zahl von 20 gefundenen Bad Edits ist vor allem darauf zurückzuführen, dass OSM über Mechanismen verfügt, die einen Grossteil der unerwünschten Bad Edits abfangen, bevor sie weitergegeben werden. Zu Test- und Trainingszwecken wurden zusätzlich ca. 1200 deutschsprachige Namen mit dem OSM-Online-Tool Overpass-Turbo aus OSM extrahiert. Dabei kann das Untersuchungsgebiet eingeschränkt werden (Abb. 2). Im zweiten Teil wurden verschiedene Algorithmen mit Zeichenketten-Metriken verglichen, die normierte Werte zwischen 0.0 (ungleich) und 1.0 (gleich) zurückgeben. Implementiert wurden die Algorithmen Jaro-Winkler (JW), Levenshtein (LV), Smith-Waterman (SM) und Cosinus-Distanz (CD). Der am besten geeignete Algorithmus wurde evaluiert und ein geeigneter Schwellenwert heuristisch bestimmt.

Ergebnis: Im Folgenden sind die Ergebnisse für vier Testmetriken aufgeführt (Abb. 3 links): F1-Score: SM 0.99, CD 0.99, JW 0.99, LV 0.99; "False Positives Rate": SM 0.94, CD 0.85, JW 0.66, LV 0.0; "False Discovery Rate": SM 0.68, CD 0.89, JW 0.96, LV 0.99; "True Negative Rate": SM 0.84, CD 0.92, JW 0.92 LV 0.84. Die "True Negative Rate" und vor allem die F1-Scores sind bei allen Algorithmen ähnlich. Bei der False "Discovery Rate" und der "False Positives Rate" schneidet der LV-Algorithmus am besten ab. Der SM-Algorithmus funktioniert bei der Erkennung von Wortähnlichkeiten nicht so gut; wahrscheinlich weil er entwickelt wurde, um lange DNA-Sequenzen zu finden. Der CD-Algorithmus scheint ebenfalls besser geeignet, um ganze Sätze zu vergleichen. Der JW-Algorithmus, eine Weiterentwicklung des LV-Algorithmus, schneidet etwas besser ab als die CD- und der SM-Algorithmen. Im Gegensatz zum LV-Algorithmus legt der JW-Algorithmus mehr Wert auf gleiche Wortanfänge, was für längere Wortketten oder Texte geeignet ist. Der LV-Algorithmus bestimmt, wie viele Mutationen durchgeführt werden müssen, um zu dem zu vergleichenden Wort zu gelangen. Er hat Vorteile bei kurzen Zeichenketten was für das vorliegende Problem am geeignetsten zu sein scheint. Mit ca. 1300 Testwörtern und einem heuristisch ermittelten Schwellenwert von 0.86 lieferte er alle True-Positives und nur 4 False-Negatives (Abb. 3 rechts). Zu beachten ist, dass in Abb. 3 rechts die Sekundärachse (rechts) für alle Metriken gilt ausser für die True Positives, für welche die linke Primärachse verwendet wurde.

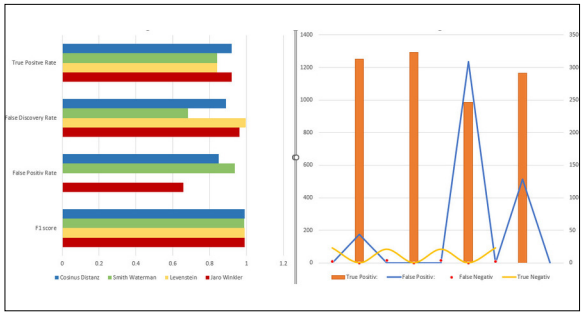
Datenflussdiagramm: Wörter zur Überprüfung auf "Bad Edits", Bad Edits-Erkennung und Ausgabe als CSV-Datei
Eigene Darstellung



Das Overpass-Tool mit einer typischen Abfrage von Lokalitäten aus von OSM über ein Teilgebiet der Schweiz
12.12.2023, https://overpass-turbo.osm.ch/



Metriken Anlyse False-Positive/True-Negative/Discovery Rate, F-Score und true/false Positive/Negative Stand 07.12.2023
Eigene Darstellung



Referent
Prof. Stefan F. Keller

Themengebiet
Software, Verschiedenes