

PlagiatChecker

Vorlesungsbezug:

- Regex
- Funktionale Programmierung
- GUI
- Python

Python-Module:

- GoogleSearch
- Wikipedia
- QT (für Gui)
- Regex
- BeautifulSoup

Zu installierende Module:

- google
- wikipedia
- pyside6

Moderne Programmierkonzepte
WWI 21 SEB
Luca Chmielarski
Nora Klemp
Marcel Bulling

Grund-Idee des Projektes:

Unsere Idee war es, ein Tool zu schaffen, mit dem man seine Texte prüfen kann. Den öfters mal kann es passieren, dass man beim Schreiben von wissenschaftlichen Texten unterbewusst seine Quellen zitiert oder kopiert. Um solche Fehler frühzeitig zu erkennen, haben wir unseren Plagiat-Checker entwickelt. Umgesetzt ist das ganze Projekt in Python.

Programm technischer Ablauf:

Über ein Textfeld lässt sich der zu prüfende Text in unser Programm einfügen. Mit Klick auf den Button „Check“ wird die Überprüfung gestartet. Nun wird im Hintergrund der Text durch eine Regex Anweisung in einzelne Sätze zerlegt und diese dann in einer Liste gespeichert. Im Folgenden werden diese Sätze dann alle einzeln in einer Suchmaschine gesucht und überprüft, ob die Sätze eins zu eins, Inhalt der gefundenen Webseite ist. Sollte dies der Fall sein, wird der Satz in auf dem Frontend angezeigt. Sätze, die nicht als Plagiat erkannt werden, werden nicht in der Ergebnistabelle angezeigt.

Suchanfragen ins Internet:

Als Suchmaschine, mit der wir unsere Sätze im Netz prüfen lassen, haben wir uns für eine API von Google entschieden. Die Sätze werden einzeln an die API übergeben und es wird nach einer genauen Übereinstimmung gesucht, sollte ein Satz gefunden werden, so wird dieser in als Plagiat angezeigt. Zum Austausch zwischen unserem Programm und die Google-Suche werden http-requests benutzt. Leider sind wir bei der Entwicklung hier auf ein größeres Problem gestoßen! Wenn man dementsprechend längere Texte in unserem Programm überprüfen lässt, so hat man auch sehr viele Anfragen zu senden. Die Google API ist in ihrer kostenlosen Version stark begrenzt. Es lassen sich nur eine bestimmte Anzahl an Anfragen am Tag senden, was bei einer größeren Benutzerzahl sehr schnell zu Problemen führen kann. Um eine unbegrenzte Anzahl an Anfragen senden zu können, wäre eine kostenpflichtige Version der Google API nötig gewesen. Für dieses Projekt haben wir uns jedoch gegen eine kostenpflichtige Version entschieden und stattdessen einer zweiten API bedient. Sobald die Anfragen auf an Google aufgebraucht sind, wechseln wir auf die Wikipedia API. Zwar lassen sich hiermit nicht alle Plagiate finden. Jedoch ist die Anzahl an Anfragen unbegrenzt und das meiste lässt sich trotzdem finden.

Vorlesungsbezug:

Bei unserem Projekt wollten wir uns nicht nur auf ein exaktes Thema der Vorlesung beziehen, sondern einige unterschiedliche Bereiche abdecken. Daher haben wir uns dazu entschieden eine Anwendung zu entwickeln, welche sowohl aus einem Back-End als auch aus einem Front-End besteht und beispielsweise nicht in der Konsole ausgeführt wird. Das Projekt wurde in Python umgesetzt und die GUI mittels „QT“ erstellt. Der Großteil des Programms ist funktional programmiert, es gibt aber auch einen Teil, welcher objektorientiert ist. Für die Aufteilung des eingegebenen Textes nutzen wir Regex-Anweisungen, welche den Text an den jeweiligen Punkten auftrennt und somit separiert.