Algorithmik – WS 23/24

Prof. Dr. Daniel Gaida

Praktikumsbetreuung: Tim Yago Nordhoff, tim_yago.nordhoff@th-koeln.de

Praktikum 1: Autocomplete

Das Ziel dieses Praktikums ist die Implementierung einer effizienten Autocomplete (Autovervollständigung)-Funktion für die englische Sprache, ähnlich der Funktionen, die Sie beispielsweise von Google-Suchanfragen oder dem Verfassen von Kurznachrichten kennen. Dabei sollen die am häufigsten in der englischen Sprache verwendeten Wörter angezeigt werden, die mit dem vom Benutzer eingegebenen Wortanfang beginnen. Ihnen stehen dafür vier verschiedene Datensätze zur Verfügung (1_gram.csv, 2_gram.csv, 3_gram.csv, 4_gram.csv). Zum Beispiel speichert der Datensatz 1_gram.csv, wie oft jedes englische Wort in den analysierten Texten verwendet wurde. Der Datensatz 2_gram.csv gibt ähnliche Informationen wieder, jedoch werden dabei zwei aufeinanderfolgende Wörter betrachtet, sogenannte 2-Grams (zum Beispiel "you can", "why are").

Autocomplete soll zwei Operationen unterstützen:

- get(word): Die Methode soll möglichst effizient prüfen, ob das übergebene Wort „word“ in dem Datensatz enthalten ist, der der Autocomplete Funktion unterlegt ist. Falls ja, dann soll die Methode das Wort zurückgegeben.
- get_k_possible_suggestions(input_string, k=3): Die Methode soll möglichst effizient für das übergebene Wort(-anfang) „input_string“ die k häufigsten Wörter der englischen Sprache zurück liefern, die mit „input_string“ anfangen. Zusätzlich soll ausgegeben werden wie viele Knoten/Wörter im Datensatz untersucht wurden.

Sie dürfen Ihre Programmiersprache frei wählen (Python, Java, Kotlin, C), wobei unten angegebene Hilfestellungen nur für Python gelten.

Aufgaben:

1) Schauen Sie sich die von uns bereitgestellte Datei „1_gram.csv“ an, in der sehr viele Worte der englischen Sprache stehen und für jedes Wort angegeben wird wie häufig es in den Quellen (Büchern, Webseiten, …), die zur Erstellung des Datensatzes genutzt wurden, vorkommt.
2) Schreiben Sie eine Datenstruktur, die die oben definierte Methode „get()“ in konstanter Laufzeit implementiert.
3) Schreiben Sie eine Datenstruktur, die beide Methoden „get()“ und „get_k_possible_suggestions()“ in logarithmischer Laufzeit implementiert.
4) Testen Sie die Funktionalität und Laufzeit beider Datenstrukturen mit verschiedenen Wörtern/Wortanfängen und Datensätzen (2_gram.csv, 3_gram.csv, 4_gram.csv).
5) Erstellen Sie eine Präsentation mit Ihren Ergebnissen und Erkenntnissen (Präsentationsdauer: 10 Minuten). Zeigen Sie Ihre Resultate für ein paar Wörter.

Hilfestellungen:

1) Als zweite Datenstruktur können Sie einen binären Suchbaum, bspw. AVL-Baum, wählen (eine Python Implementierung liegt in AVLTree.py vor).
2) Die Python-Datei "AutocompleteNgrams.py" könnt ihr als Grundgerüst für den Autocompleter verwenden.
3) Wir empfehlen die Nutzung der Python Distribution „Anaconda“ oder „Miniconda“. Diese können Sie frei herunterladen und installieren. https://docs.conda.io/projects/conda/en/latest/user-guide/install/download.html#anaconda-or-miniconda.
4) Als IDE können Sie bspw. PyCharm Community Edition, Visual Studio Code, … nutzen.