Geospatial Data Mining – Hotel Webcrawler

Raphael Prinz

Geospatial Data Mining, Seminar im WS21, raphael.prinz@online.uni-graz.at)

Abstract

Webcrawler sind Programme, die das WWW systematisch und automatisch nach Informationen durchsuchen (vgl. Kausar et al. 2013, S.31). Im Rahmen des Integrativen Masterseminar (Geospatial Technologies) wurde ein Programm entwickelt, mit dessen Hilfe Hoteldaten der Webseite Booking.com extrahiert werden. Das Programm wurde in der Programmiersprache Python entwickelt und für Hotels im Bereich von Istanbul (Türkei) erprobt. Innerhalb der Entwicklung wurden Daten im Dezember, Jänner und März erhoben. Die Daten wurden visualisiert und Vergleiche hinstlich der Beobachteten Preisstruktur gezogen.

Keywords: WWW, web data mining, information extraction, information retrieval, location-based services

1 EINLEITUNG

Das World Wide Web (WWW) nach dem W3 Protokoll liefert Informationen mithilfe eines verteiltes, nicht-lineares Textsystems, bekannt als Hypertext-Dokumente (vgl. Kausar et al. 2013, S. 31). In den 22 Jahren seit seiner Entstehung hat das WWW den Umgang mit Daten hinsichtlich Zugänglichkeit und Verfügbarkeit signifikant verändert. Stand 2021 nutzen 65,6 % der Weltbevölkerung dieses System und im Bereich der Europäischen Union wird das WWW von 89.4 % der Bevölkerung (397 988 114 Personen) genutzt (Miniwatts Marketing Group). Mit ca. 1.41 Milliarden Webseiten (Kunden) bringt das Internet eine Dichte und Vielzahl an Informationen hervor, die weltweit seinesgleichen sucht. Der Kontext für geografische Forschung hat sich von einer Daten-armen zu einer Daten-reichen Umgebung gewandelt (Miller und Goodchild 2015). Soziale Netzwerke, Citizen-Science und Regierungs Überwachung haben verursachten eine Explosion an Web-Kontent (Li et al. 2012).

1.1 Forschungsfragen

Ziel ist es im Rahmen des Seminares eine Möglichkeit zu finden, um automatisch Hoteldaten aus Preisvergleichs Webseiten wie Trivago und Booking.com zu gewinnen. Im Zusammenhang zum Ziel stellen sich Forschungsfragen:

1) Wie können Daten für Hotels in Form von Koordinaten, und Preisen aus Preisvergleichs Webseiten wie Triavgo und Booking.com gewonnen werden?
2) Wie unterscheiden sich die Daten in Lage, Preis und Verteilung für unterschiedliche Erhebungsperioden für das Testgebiet in Istanbul?

1.2 Ablauf

In Kapitel 1 wird der Leser an das Thema herangeführt und in den nachfolgenden Absätzen mit den Grundlagen des World Wide Webs bekannt gemacht. In Kapitel 2. Methodik wird der von Li et al. (2012) entwickelte Webcrawler vorgestellt, dieser extrahiert die Adressen Feuerwehstationen in den USA. In Methodik wird die Vorgehensweise bei der Progra mmierung des Webcrawlers erläutert und Designsentschlüsse erklärt. In Ergebnisse werden die gewonnen Daten gezeigt und auch kartographisch ausgewertet. Die Diskussion verinhaltlicht die eesentlichen Erkentnisse der vorherigen Kapitel und zeigt stärken und Schwächen des Webcrawlers auf.

1.3 Grundlagen des World Wide Web ‘s

1.3.1 HTML

HyperText Markup Language (HTML) ist der grundlegenste Baustein des Webs. Er beschreibt und definiert den Inhalt einer Webseite zusammen mit dem grundlegenden Layout der Webseite. Neben HTML werden im

Allgemeinen andere Technologien verwendet, um das Erscheinungsbild (CSS) oder die Funktionalität/Verhalten (JavaScript) einer Webseite zu beschreiben. (vgl. Mozilla 2022)

Hypertext bezieht sich auf Links, die Webseiten miteinander verbinden, entweder innerhalb einer einzelnen Webseite oder zwischen mehreren Webseiten. HTM verwendet "Markup", um Texte, Bilder und andere Inhalte für die Anzeige in einem Webbrowser zu kommentieren. HTML-Markup beinhaltet Elemente wie <head>, <title>, <body>, und viele weitere.

Verwendet wurde HTML ursprünglich in CERN zum Austausch von Dokumenten (W3C 1990) , später übernahm das World Wide Web Konsortium (W3C) die Weiterentwicklung des Standards (vgl. W3C 2011).

JavaScript (JS) ist eine leichtgewichtige, interpretirte oder Just in Time (JIT) übersetzte Sprache. Bekannt ist sie hauptsächlich als Skriptsprache für Webseiten geworden, jedoch findet sie auch außerhalb des Browsers Verwendung (vgl. Mozilla 2022). Der Standard für JavaScript ist ECMAScript, 2019 wurde der neueste Standard veröffentlicht (vgl. Ecma International 2019).

Cascading Style Sheets, meistens als CSS abgekürzt, ist eine Beschreibungssprache, die das Erscheinungsbild in einem HTML oder XML Dokument festlegt. CSS beschreibt, wie ein Element am Bildschirm dargestellt wird. (vgl. Mozilla 2022)

1.3.2 HTTP Requests

Hypertext Transfer Protocol (HTTP) wird für die Kommunikation zwischen Webbrowser (Client) und Server zu ermöglichen. Ein Client eröffnet dabei die Kommunikation durch einen Request, anschließend wird auf eine Antwort gewartet.

HTTP definiert eine Reihe von Methoden, um die Kommunikation zwischen Client und Server zu ermöglichen. Insgesamt werden durch HTTP 9 Methoden definiert: 1. GET, 2. HEAD, 3. POST, 4. PUT, 5. DELETE, 6. CONNECT, 7. OPTIONS, 8. TRACE, 9.PATCH vgl. (Fielding und Reschke 2014). Im weiteren Verlauf der Arbeit liegt der Fokus hier auf die GET- und POST-Methoden, welche im Standard der IETF RFC 72231 ((Fielding und Reschke 2014) wie folgt definiert sind:

GET : Transfer a current representation of the target resource

POST: Perform resource-specific processing on the request payload

Die GET Methode verlangt nach dem Transfer einer ausgewählten Ressource, sie ist der primäre Mechanismus von HTTP, um Informationen (Daten) zu erhalten. Die Methode besteht aus einem HEAD und einem BODY, im HEAD sind Informationen zum Statuscode der Anfrage wie 200 für OK und 404 für angefragte Ressource nicht gefunden. Unter anderem wird auch die HTTP Version und Uhrzeit und Datum im HEAD übermittelt. Im Body ist die vom Client gewünschte Ressource enthalten, diese besitzt kein festes Format. Gängige Formate sind zum Beispiel HTML Dokumente welche Webseiten repräsentieren und JSON- oder XML- Dateien für Datenaustausch sowie PNG- und JPG-Bilder. (vgl. Fielding und Reschke 2014)

Post Methoden verlangen, dass eine Ressource prozessiert wird. Dies kann etwa das Absenden eines ausgefüllten Formulars sein oder das Posten einer Nachricht in einem Blog. (Fielding und Reschke 2014)

Häufig werden POST-Requests vom Client initialisiert, auf der Serverseite wird die Anfrage verarbeitet, bei Erfolg wird eine Antwort mit dem Statuscode 201 (Created) gesendet.

1.3.3 Websockets

WebSockets dienen der Zweiwege-Kommunikation zwischen Server und Client (Browser). WebSockets wurden mit dem Ziel entwickelt, Browser-basierten Anwendungen einen Weg zu liefern, mit einem Server zu kommunizieren, ohne auf mehrere HTTP Verbindungen angewiesen zu sein. Das Protocol wurde 201 durch die Organisation Internet Society and the Corporation for National Research initiatives (IETF) standardisiert. Es besteht aus einem Handschlag (opening Handshake) und einem Nachrichtenprotokoll, welches über einen Bytestream nach dem TPC-Protokoll gesendet wird. (Mozilla 2022)

2 STATE OF THE ART

Diese Arbeit beschäftigt sich mit der Extraktion von Daten aus einer Zielwebseite, im Gegensatz dazu arbeiteten Li et al. (2012) an einem Webcrawler, der Webseiten unabhängig von Anbieter untersucht. Im Unterschied zu Google`s Webcrawler steht hier nicht die Indexierung von Webseiten für die Suche im Fokus, sondern die Standorte (Adressen) von Feuerwehrstationen in Los Angeles County (USA) (Li et al. 2012, S.561).

Der von Li et al. (2012, S. 554) umgesetzte Webcrawler stützt sich auf räumliche Ontologien, sowie forward- und backward-screening Algorithmen und Machine Learning.

Unter Vorgabe einer Startwebseite arbeitet sich der Webcrawler automatisch durch die in dieser Webseite angeführten Links, als Start wurden Regierungsseiten getestet. Li et al. unterscheidet hierbei zwischen Seeds (Startwebseiten) und Targets (Webseiten mit Adressen). Dabei zeigte sich, dass die Auswahl der Startwebseiten das Ergebnis erheblich beeinflussen. Die Ontologien, die Li (2012) verwendet, beziehen sich auf die Verwaltungseinheiten der USA. So wurde zum Beispiel das Triplet "County"-> "belongs to" -> "State" oder "Independent City" -> "belongs to" -> "County-Equivalent" verwendet um die Auswertung der Links zu verbessern. (vgl. Li et al. 2012, S. 555-557)

Zielwebseiten beinhalten die Adressen der Feuerwehrstationen, auch hier wurden Ontologien verwendet, um Derivate und Variationen in der Adressbeschreibung zu verarbeiten. Li et al zeigten, dass 90 % aller Adressen eines der Worte "Street", "Road", "Drive", "Way","Highway" oder "Circle" enthalten, nachdem solche Schlüsselwörter ausfindig gemacht wurden, sind die forward- und backword-screening Algorithmen verwendet worden. Dabei wird bei backword-screening der Text vor dem Schlüsselwort und bei forward-screening der Text nach dem Schlüsselwort untersucht. Maßgebend für den Suchradius ist das Aufkommen von Zahlen vor oder nach dem Schlüsselwort, z. B.: neunstellige Postleitzahlen. Um festzustellen, ob es sich bei der Adresse tatsächlich um eine Feuerwehrstation handelt, mithilfe eines Machine Learning Modells (Entscheidungsbaum) der Text vor der Adresse untersucht. Das Modell ist semantisch aufgebaut und kategorisiert Text nach 1. Schlüsselwörter innerhalb des Textes (z. B. Fire station), 2. statistische Parameter (Verhältnisse von Schlüsselworten). 3. Mustern (z. B. "location" gefolgt von einer Zahl), 4. Schlüsselwörter innerhalb des Webseittitels und 5. Schlüsselwortfrequenz.

Li et all konnten 346 Feuerwehrstationen innerhalb des Los Angeles County (Kalifornien, USA) ausfindig machen. Die von dem Authoren per Hand extrahierte Anzahl ist 392, damit besitzt der Webcrawler einen Recall von 88 %. (vgl. Li et al. 2012, S. 556–561)

In Zukunft plant Li et al (2012, S. 562), die Genauigkeit des Webcrawlers weiter zu verbessern und eine Datenbank der Feuerwehrstationen in den USA anzulegen. Dabei wird zur Verifikation der Ergebnisse auf Citizen-Science gesetzt. Lee hofft, eine Plattform bieten zu können, in der Fehl-Klassifizierungen und fehlende Stationen von Freiwilligen eingetragen werden können. (vgl. Li et al 2012, S.562)

3 METHODIK

Als Basis für den Webcrawler wurde die Python Applikation Programming Interface (API) des Selenium WebDriver's (Selenium) verwendet. Hierbei handelt es sich um eine in Java programmierte Anwendung, die darauf ausgerichtet ist, gängige Browser zu automatisieren. Als Browser wurde Mozilla Firefox gewählt, da dieser Browser Quelloffen ist und somit frei verfügbar ist. Ein zusätzlicher Vorteil, dass der Tor Browser auf Firefox basiert, dieser kann die IP-Adresse eines Nutzers verschleiern und wäre durch einige Änderungen im Quellcode nutzbar.

Eine Vielzahl von Programmiersprachen werden für die Ansteuerung durch Selenium unterstützt, die Python Version zeichnet sich durch die Erweiterung “selenium-requests” aus. Diese Erweiterung wird zusätzlich für das Auslesen der ein-/ausgehenden Requests der Webseiten benötigt.

Der Quellcode (HTML, CSS, JavaScript) der Webseiten Trivago und Booking.com sowie die ein-/ausgehenden Requests wurden hinsichtlich geografischer Daten untersucht. Dabei wurde der Fokus darauf gelegt, die HTML, und CSS Tags zur Bedienung der Webseiten gebraucht werden.

Sowohl Trivago als auch Booking.com nutzten Google Maps, um die Standorte der Hotels in Karten darzustellen, der Standort eines Hotels wird hier über Längengrad und Breitengrad definiert. Die Koordinaten selbst sind in einer Datenbank gespeichert und werden nur von der Webseite abgefragt, falls diese benötigt werden. Die Kommunikation zwischen Webseite und Datenbank erfolgt mittels Anfrage, speziell werden GET-Requests von der Webseite zum Server der Datenbank gesendet, die Antwort des Servers enthält die gewünschten Daten zu den Hotels.

Da diese Anfragen nur bei bestimmten Aktionen ausgelöst werden, wurden für beide Webseiten Ereignisschleifen entwickelt. Diese Schleifen brechen die Bedingung der Webseite auf die wesentlichen Mechanismen herunter, um die Anfragen aud die Datenbank auszulösen. Direkt in der Webseite konnten keine Koordinaten festgestellt werden, es ist jedoch möglich, die Adressen aus der Booking.com Webseite zu gewinnen.

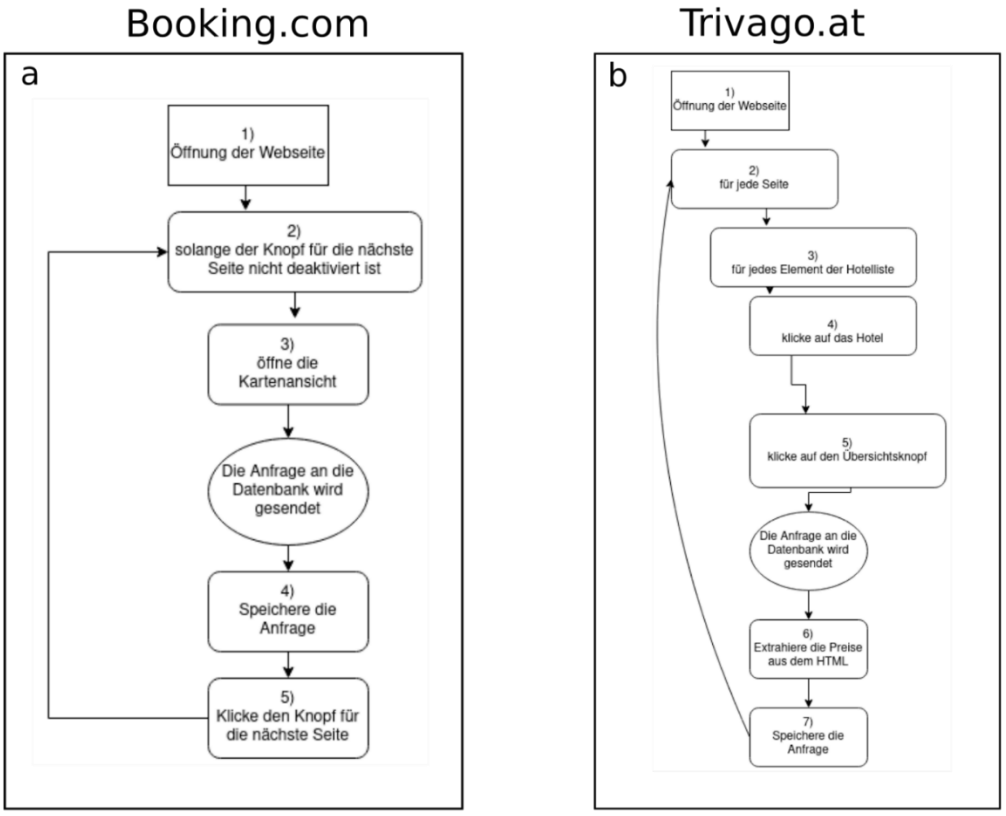

Figure 1: Ereignisschleifen zum Download der Hotaldaten für Booking.com und Triavago (Quelle: eigene Darstellung 2022)

```
{ "b_cities": [{ "b_latitude": 47.89899, "b_basic_type": "city", "b_photo":
"https://r-
xx.bstatic.com/xdata/images/city/360x240/656382.jpg?k=de421f3677e8b225e58fbcb2604827c
e2175f57a8989e3bc4d8b96119f6d3664&o=", "b_name": "Eger", "b_id": -852573,
"b_marker_photo": "https://r-
xx.bstatic.com/xdata/images/city/360x240/656382.jpg?k=de421f3677e8b225e58fbcb2604827c
e2175f57a8989e3bc4d8b96119f6d3664&o=", "b_longitude": 20.37437, "b_marker_type":
"city" }], "compass_pmm_resp": 1, "b_hotels": [{ "b_charges_excluded_charges_detail":
null, "b_review_word": "Sehr gut", "b_hotel_was_sold": 0,
"b_charges_hotel_sum_excluded_raw": 0,
```


Code 1: Ausschnitt der Rohdaten der Booking.com Datenbankanfrage (Quelle: eigene Darstellung 2022)

Die Anfragen selbst unterscheiden sich hinsichtlich der Daten und der Anzahl der Hotels. Für Trivago musste für jedes Hotel eine eigene Anfrage ausgelöst werden, während bei nur eine Anfrage pro Seite ausreichte.

Um die Anfragen auszulösen, musste eine Reihe von Elementen angeklickt werden, es zeigte sich, dass die HTML-Tags der Elemente regelmäßig geändert werden, die allgemeine Struktur der Ereignisschleifen blieb jedoch konstant.

Die Lokalisierung der Knöpfe erfolgte hauptsächlich über CSS-Auswahl. In a 2) der Ereignisschleife für Booking.com wird die Schleife so lange weitergeführt, bis der Knopf für die nächste Seite deaktiviert ist. In a 3) wird die Kartenansicht geöffnet, im Fall von Booking.com wird hier die Anfrage ausgelöst. Die Daten sämtlicher Hotels einer Seite sind in der Antwort der Anfrage enthalten. a 4) Die Webseite lädt über 100 verschiedene Anfragen, die Anfrage mit den gewünschten Informationen muss herausgefiltert werden. Die Kartenansicht wird geschlossen und der Knopf für die nächste Seite angeklickt. Dies führt die Ereignisschleife von a 2) fort, solange der Knopf aktiviert ist.

Für Trivago, in b 2) wird durch die Liste an Seitenknöpfen der Trivago-Webseite iteriert, die Auswahl erfolgt ebenfalls über CSS. Für jedes Hotel muss der Übersichtsknopf geklickt werden, damit die Identifikationsnummer des Hotels gelesen werden kann.

Es zeigte sich bei Trivago als stabiler, die Anfrage auf die Datenbank direkt auszuführen. Die erste und zweite Zeile in 6) zeigen die API der Trivago Datenbank, mithilfe der Hotelidentifikationsnummer kann diese abgefragt werden. In dieser Antwort sind keine Preise enthalten, deshalb werden die Preise aus dem HTML der Webseite bezogen.

Was Trivago betrifft, so konnten an die 200 Hotels für die Analysen extrahiert werden, es zeigte sich, dass ein Großteil dieser Hotels auch in Booking.com vorkommt, deshalb wurde der Hauptfokus auf Booking.com gelegt.

3.1 AutoCrawler - ein flexibler Booking.com Webscraper

Webseiten besitzen Schutzmechanismen, um sich vor automatisierten Anwendungen wie WebCrawler und Bots zu schützen. Diese Schutzmaßnahmen können sich regelmäßige HTML-Tags, aber auch Zeitmessungen der Nutzeraktivitäten sein. Um Zeitmessungen entgegenzuwirken, gibt es zwei Maßnahmen: 1) Längere Abstände zwischen Aktionen, zu viele Klicks pro Sekunde steigern die Chance, dass die Anwendung als Bot erkannt wird. 2) Einbau von Zufälligkeit, eine per Zufall gewählte Wartezeit zwischen Aktionen zeigte, während der Entwicklung großen Erfolg.

Zwischen Aktionen wurde eine zufällige Wartezeit von 7 bis 22 Sekunden einprogrammiert.

```
time.sleep(np.random.choice([x for x in range(7, 22)]))
```

Code 2: Zufaellige Zeitverzögerung von 7 bis 22 Sekunden.

Solche Codezeilen wurden oft für die automatische Bedienung verwendet (Quelle: eigene Darstellung 20222)

Die Ereignisschleifen für Booking.com und Trivago zeigten sich als empfindlich für Änderungen der HTML-Tags, die Änderung eines Tags führt das Programm zum Absturz, da benötigte Elemente nicht mehr angeklickt werden können. Deshalb wurde versucht, eine flexiblere Lösung zu entwickeln, in der Tags ausgetauscht werden können, die Ereignisschleife selbst wird jedoch als statisch betrachtet.

Um eine möglichst einheitliche Bedingung zu gewährleisten, wurde das Programm als Webseite geschrieben, diese wird von einem lokalen Python-Server gehostet. Mit Start des Programmes wird eine neue Firefox Instanz geöffnet, in dieser wird mithilfe von Selenium automatisch die Webseite für die Bedienung und die Webseite für Booking.com gestartet.

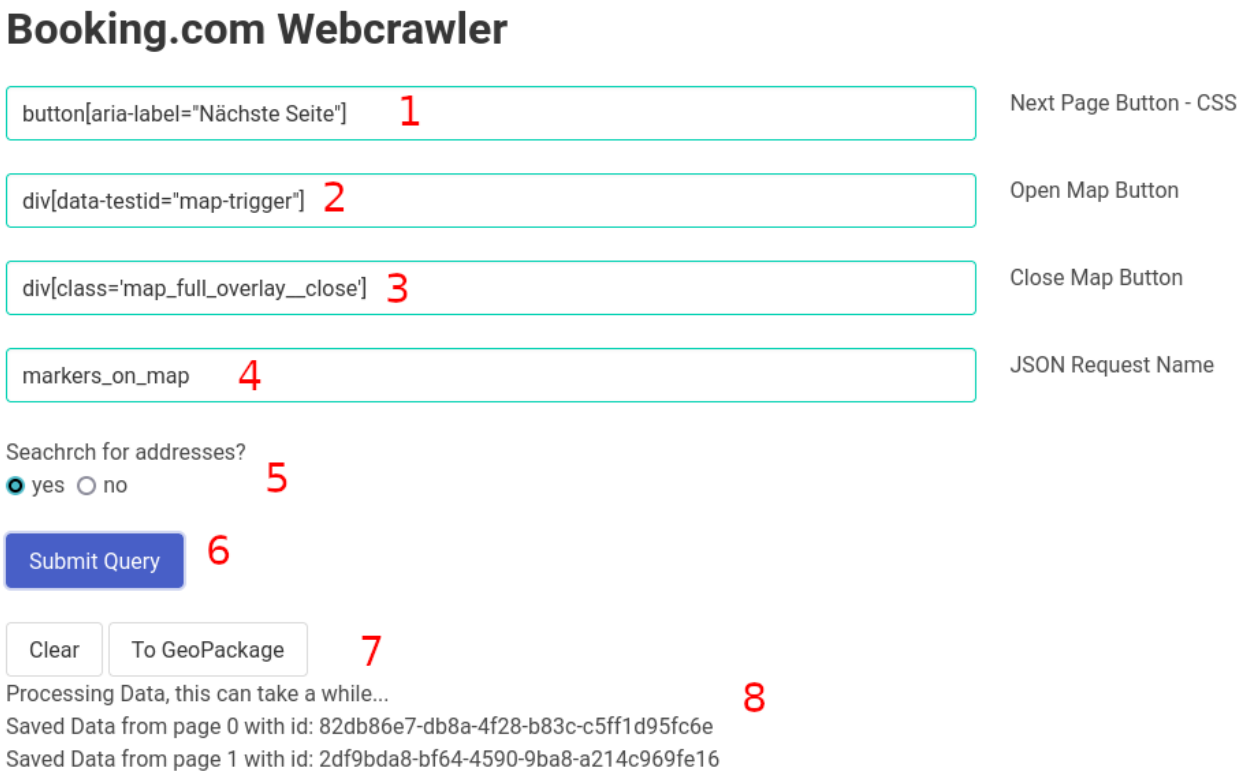

Figure 2: Bedienung des automatisierten Webcrawler (Quelle: eigene Darstellung 2022)

Der Server wurde mithilfe von der Python-Bibliotheken Flask und SocketIO programmiert. Die Webseite beinhaltet 4 Formulare, in denen Default-Tags für die Durchführung der Ereignisschleife zu sehen sind.

In 1) Ist der HTML-Tag für den Knopf anzugeben, der die nächste Seite öffnet, die Auswahl muss in CSS-Notation eingegeben werden. In 2) wird der Knopf angegeben, mit dem man die Karte öffnet und in 3) jener, der die Karte wieder schließt.

In 4 wird der Name des JSON Objektes mit den Hoteldaten gefordert. In 5) kann der Nutzer auswählen, ob er auch nach Adressen der Hotels suchen will. Die Suche der Adressen dauert ca. 3 Minuten pro Seite, ohne Adelsuche beträgt die Laufzeit ca. 10 Sekunden pro Seite.

Mit einem Klick auf 6. wird die Ereignisschleife gestartet und die ausgefüllten Formulare mittels POST-Request an den Server übergeben. Vom Nutzer wird erwartet, dass er zuvor in Booking.com einen Standort ausgesucht hat und eventuelle auftauchende Pop-up Werbungen deaktiviert hat. Die Webseite sollte entsprechend Abbildung 2 aussehen.

Mit Start des Vorganges durch 6) ist es auch möglich, 7) “Clear” und “To GeoPackage” zu betätigen, mit “Clear” werden sämtliche Daten (JSON-Datein) im Verzeichnis “bookingData” gelöscht, “To GeoPackage”

erstellt basierend auf den gesammelten Daten eine neue GeoPackage Datei. In 8 werden die Logs des Webcrawlers ausgegeben und Fehlermeldungen angezeigt.

Die Anzahl der gesammelten Seiten wird mittels WebSocket in Echtzeit vom Server in die Kontroll-Webseite eingeschrieben. Durch einen Klick auf "To Geopackage" wird gleichzeitig ein Download für das GeoPackage File ausgelöst. Durch die Anwendung zufälliger Zeitintervalle kann es bei der Ausführung des Programmes maximal 22 Sekunden dauern bis ein neues Ereignis ausgeführt wird. Bei längeren Intervallen kann es sich um einen Fehler handeln. Falls ein Fehler auftritt, wird dies ebenfalls in die Webseite geschrieben.

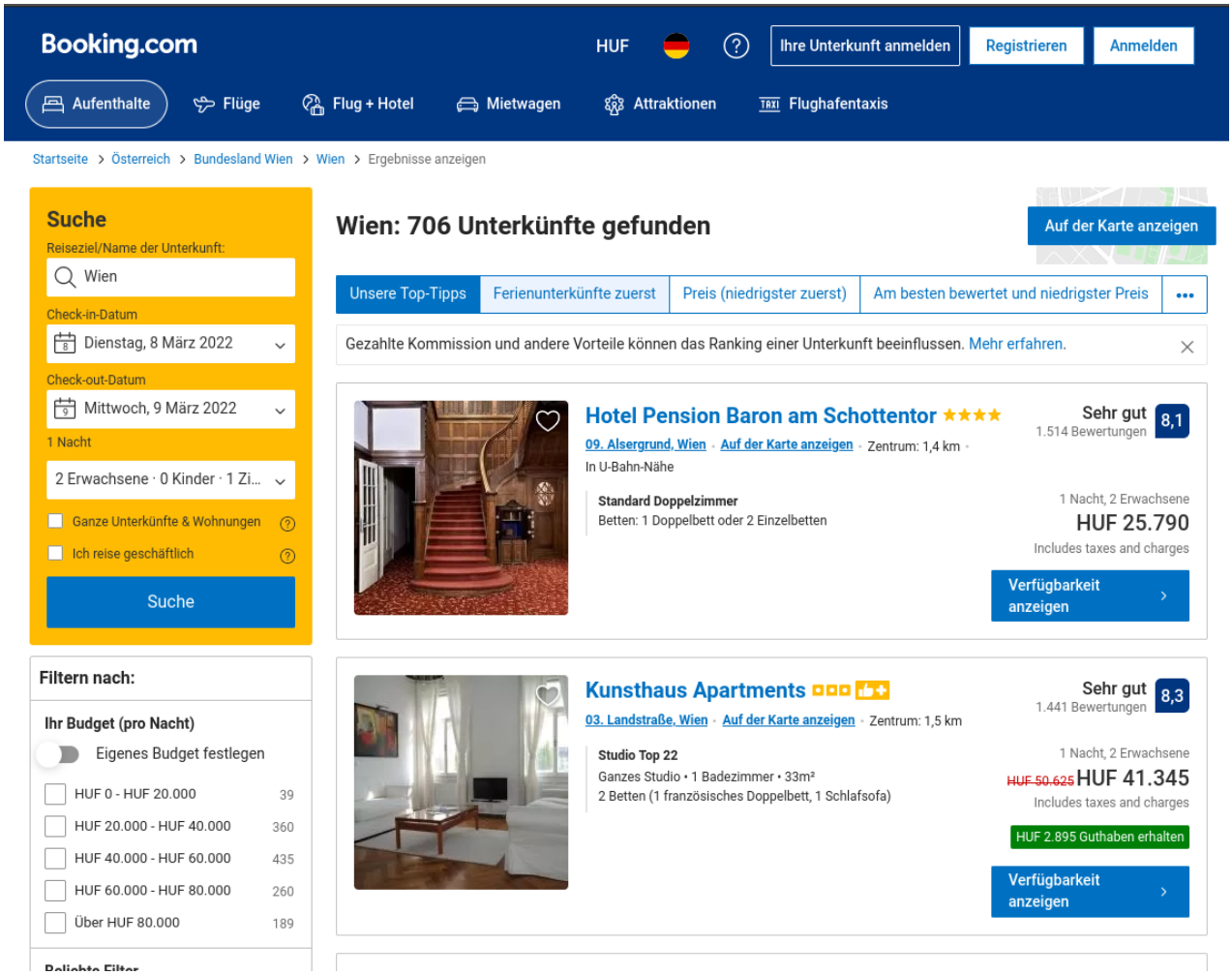

Figure 3: Booking.com Webseite mit Datum, Hotels und Kartenanzeige-Knopf (Quelle: eigene Darstellung 2022)

Serverseitig wird mithilfe des Flask-Modules, das Routing des Servers erledigt. Dazu werden Funktionen "@app.route()" Dekorator versehen. Wenn ein Webbrowser diese Seite (http://127.0.0.1:5000/start) aufruft, werden die Formulardaten von dem Browser (Client) an den Server als JSON-Datei gesendet. Dies ist mit dem zuvor beschriebenen Datenbankzugriff von Booking.com und Trivago zu vergleichen. Die Formulardaten können über "request.form" bezogen und in Variablen gespeichert werden.

Im zugrundeliegenden HTML Dokument wurde die Route über action="/start" definiert, während name="nxt_pg_btn" den Namen für den Zugriff in Python kennzeichnet.

Bei <input type="submit" handelt es sich um den Submit-Knopf durch diesen wird die Python funktion aktiviert.

```
<form action="/start" method="post" target=frame class="myform">
<div
  class="column is-two-fifths"><input name="nxt_pg_btn" class="input
  value='button[aria-label="Nächste Seite"]' type="text" >
</div>
<input type="submit" class="button is-link" id="submit_btn" />
</form>
```

Code 3: HTML-Formular zur übergabe der Webcrawler-Einstellungen (Quelle: eigene Darstellung 2022)

Nach dem Klick wird die Funktion start_collecting() ausgeführt, diese Startet einen neuen parallellen Thread (Prozess) der neben dem Hauptprogramm läuft. Dadurch ist gewährleistet, dass Fehler beim Sammeln nicht sofort zu einem Absturz der Bedienungs-Webseite führen.

```
@app.route("/start", methods=["POST"])
def start_collecting():
    …
    threading.Thread(target=booking_collect, args=(
        nxt_pg_btn, map_btn, close_map_btn, json_name)).start()

    return "ok"
```

Code 4: Startfunktion des Webcrawlers, durch Klick auf "Submit Query" läuft diese Funktion so lange, bis alle Seiten der Booking.com Webseite erfasst wurden (Quelle: eigene Darstellung 2022)

Die Kommunikation zwischen Browser (Client) und Server (Python-Programm) läuft nicht nur über Formulare, sondern auch über WebSockets, dies ermöglicht weitere Kommunikation zwischen Client und Server. Diese Technologie wird angewendet, da POST-Requests ein Aktualisieren der Webseite benötigen, da der Client auf eine Antwort des Servers wartet. Durch Websockets muss die Webseite nicht laufend aktualisiert werden, um Echtzeit-Kommunikation zwischen beiden Seiten zu ermöglichen.

Um zusätzlich die Adressen der Hotels ausfindig zu machen, muss die Beschreibungswebseite jedes Hotels geöffnet werden. Dies geschieht im Hintergrund zum eigentlichen Programmlauffaden, allerdings vergrößert sich dadurch die Laufzeit pro Seite von ca. 10–15 Sekunden auf ca. 1 - 2 Minuten. Im Testfall für die Stadt Istanbul mussten ca. über 1600 GET-Requests pro Durchlauf (ca. 40 Hotels pro Seite und 40 Seiten) getätigt werden. Um die Programmlaufzeit zu optimieren, wurde hier asynchrone Programmierung verwendet. Alle 40 Requests für eine Seite wurden gleichzeitig gestartet und die Antworten der Booking.com Webseite gleichzeitig bearbeitet, bis die Programmlauffäden wieder zusammengeführt wurden.

Wird in der Webseite der Knopf für die Speicherung von als Geopackage gewählt, so wird die Funktion files2gpkg() ausgeführt, diese ruft eine weitere Funktion booking2gpkg() auf, welche die eigentliche Konvertierung der JSON Files in Geopackages übernimmt.

```
@socketio.on("gpkg")
def files2gpkg(_):
    try:
        booking2gpkg()
        clear_folder("")
        socketio.emit('download_ready', {'data': "download_ready"})
    except Exception as e:
        …
```

Code 5: Python-Funktion, die durch das Signal "gpkg" vom Browser des Nutzers ausgelöst wird (Quelle: eigene Darstellung 2022)

Für die Konvertierung werden die gewünschten Koordinaten aus JSON gelesen, als Zwischenergebnis in einen Geopandas-Dataframe konvertiert, der wiederum als GeoPackage gespeichert wird. Durch try & except können Fehler, die beim Speichern passieren, aufgefangen werden.

Im JavaScript Code der Webseite findet sich das Gegenstück zur WebSocket Python Funktion, in dieser wird eine Nachricht mit dem Inhalt “gpkg” an den Server gesendet. Aufgrund des Inhalts wird die Funktion in Python aktiviert.

```
$('#gpkg').on("click", function() {
… {
        $('#log').text("").text("Processing Data…);
        socket.emit("gpkg", { data: 'gpkg' });
        $('#clear').removeClass("is-link");
        $('#gpkg').removeClass("is-link");
    }
 });
```

Code 6: JavaScript funktion zum Senden der Nachtricht "gpkg" via WebSocket (Quelle: eigene Darstellung 2022)

WebSockets wurden bevorzugt benutzt, um laufende Prozesse durch die Webseite steuern zu können, hingegen wurden Requests für das Routing und den Download-Mechanismus des GeoPackage's benutzt.

Die Ordnerstruktur basiert auf den Empfehlungen der Flask-Entwickler (Flask). Im Ordner "autoCrawler" befinden sich die Unterordner "app", "venv" und die Datei "dependencies.txt"

Im "app"-Ordner ist die Hauptanwendung "autoCrawler.py" zu finden, in dieser ist der Server programmiert. Dieser Server hostet die Webseite, deren Vorlage ist im Ordner "templates" als "index.html" zu finden. Für die Dynamik der Webseite wird Javascript benötigt, dieses wird vom Browser (Firefox) ausgeführt. Der JavaScript-Quellcode befindet sich in "static/frontend.js". Durch die Datei "geckodriver" kann Firefox angesteuert werden, Selenium benötigt diese Datei, deshalb befindet sie sich in "app". In der Datei "flatten.py befindet sich der Code zur Konvertierung der gesammelten Anfragen in das GeoPackage Format.

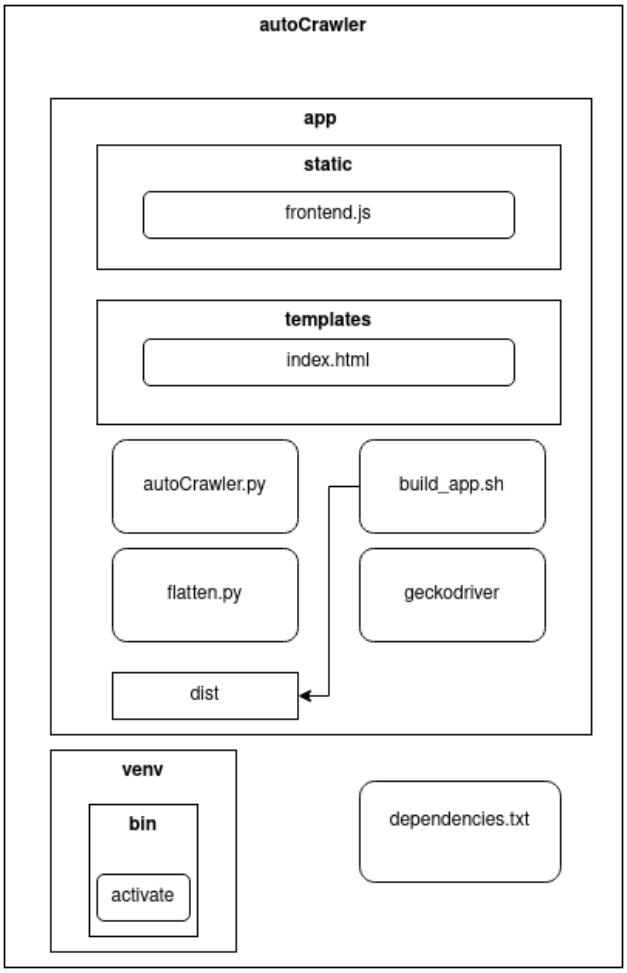

Figure 4: Dateiaufbau des Webcrawlers (Quelle: eigene Darstellung 2022)

Zur Ausführung des Programmes können zwei Varianten verwendet werden, entweder über Aktivierung des Python-Interpreters oder über die zu Maschinencode umgewandelte .exe Datei. Beide Varianten besitzen Vor- und Nachteile.

Zur Ausführung über den Pythoninterpreter wird eine virtuelle Umgebung in Form des Ordners "venv" zur Verfügung gestellt, dieses funktioniert nur auf Linux - Ubuntu, für eine Windows-Variante muss eine neue Umgebung erstellt werden, die Datei "dependencies.txt" enthält eine Liste der dafür notwendigen Module. Die Ausführung über den Pythoninterpreter ist für einen Endnutzer im Vergleich zum Maschinencode recht aufwendig, jedoch sind Änderungen sofort wirksam und lassen somit eine schnellere Entwicklung zu. Bei einer Fortsetzung dieser Arbeit kann nur diese Variante verwendet werden, da für die Kompilierung die virtuelle Umgebung gebraucht wird.

Die ".exe"-Version ist durch einen einfachen Doppel-Click ausführbar. Dies ist allerdings ein geschlossenes Programm, bei Fehlern oder Änderungen der Ereignisschleife muss eine neue Version erstellt werden. Im Ordner "app/build_app.sh" ist ein für Linux geeignetes Shell-Skript zu finden, welches die Kompilierung automatisch durchführt, zum derzeitigen Stand der Arbeit wird noch an einer Windows-Version gearbeitet.

Während der Auswertung der Hotels fiel auf, zeigte sich vorwiegend die Verknüpfung der Datensätze aus unterschiedlichen Perioden als schwierig. Nachträglich wurde deshalb die eindeutige Identifikationsnummer jedes Hotels aus den Booking.com Rohdaten in die GeoPackage Attributtabelle mitaufgenommen. Dies wurde nicht für den Datensatz aus Jänner und Dezember gemacht.

4 UNTERSUCHUNGSGEBIET UND DATENGRUNDLAGEN

Als Untersuchungsgebiet wurde die Stadt Istanbul mit einem Fokus auf die Altstadt ausgewählt. Für Booking.com und Trivago wurde versucht, möglich alle Hotels zu erheben, die sich weniger als 8 km von der Altstadt befinden. Für einen Vergleich der erhobenen Daten wurden Hotels aus der OpenStreetMap verwendet. Datenerhebungen wurden im Dezember 2021, Jänner 2022 und im März 2022 durchgeführt.

Im Dezember wurden Hotels für zwei Personen vom 31.12.21 bis 02.01.22. Im Jänner für zwei Personen vom 29.01.22 bis 31.01.22 und im März für 01.04.22 bis 03.04.22.

5 ERGEBNISSE

Im Ergebnisteil der Seminararbeit berichten Sie über die Ergebnisse Ihrer Studie basierend auf den Informationen, die als Ergebnis der von Ihnen angewendeten Methodik [oder Methodiken] gesammelt wurden. Der Ergebnisteil sollte einfach die Ergebnisse ohne Voreingenommenheit oder Interpretation darstellen und in einer logischen Reihenfolge angeordnet sein.

Erhebungen konnten mit dem automatisierten Tool für Booking.com durchgeführt werden. Für die Abfrage im März, mit Buchungsdatum April konnten 1013 Hotels extrahiert werden. Im Jänner 1040 und im Dezember 845. Die Anzahl der Hotels, die in allen Perioden auftauchten, betrug 1013. Werden alle Perioden zusammengefasst, so konnten 1635 Hotels gewonnen werden, davon betrug die Anzahl der Hotels, zu denen kein Preis gefunden wurde, 63. Insgesamt konnte eine Steigung des mittleren Preises beobachtet werden. Der mittlere Preis im Dezember beträgt 75 €, im Jänner 147 € und im April 193 €.

Die Anzahl der Hotels ist in der Altstadt deutlich höher als in anderen Gebieten, am meisten Hotels konnten aus den Bezirken Canurtaran (60 Hotels) und Hocapase (65 Hotels) gewonnen werden. Insgesamt sind auf der europäischen Seite mehr Hotels als auf der asiatischen Seite zu finden.

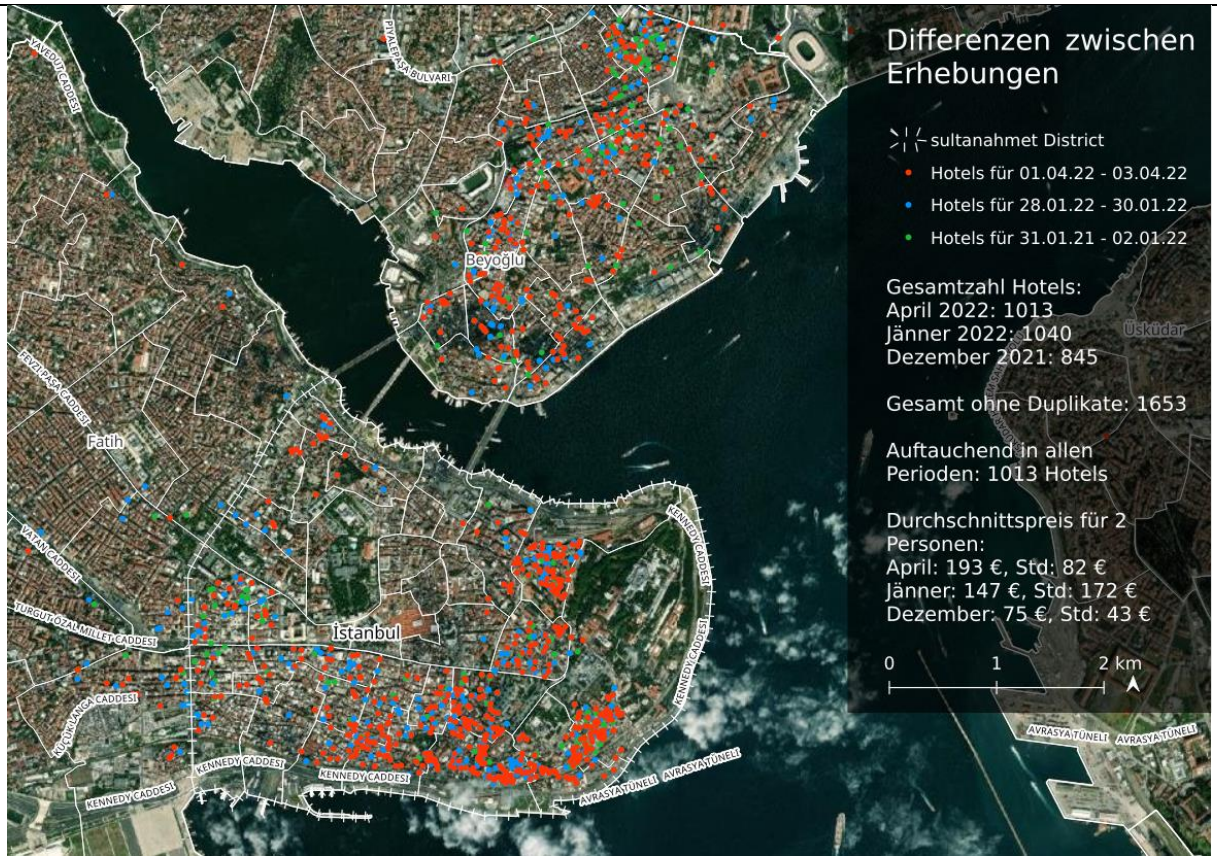

Figure 5: Durch Booking.com erhobene Hoteldaten (Quelle: eigene Darstellung 2020)

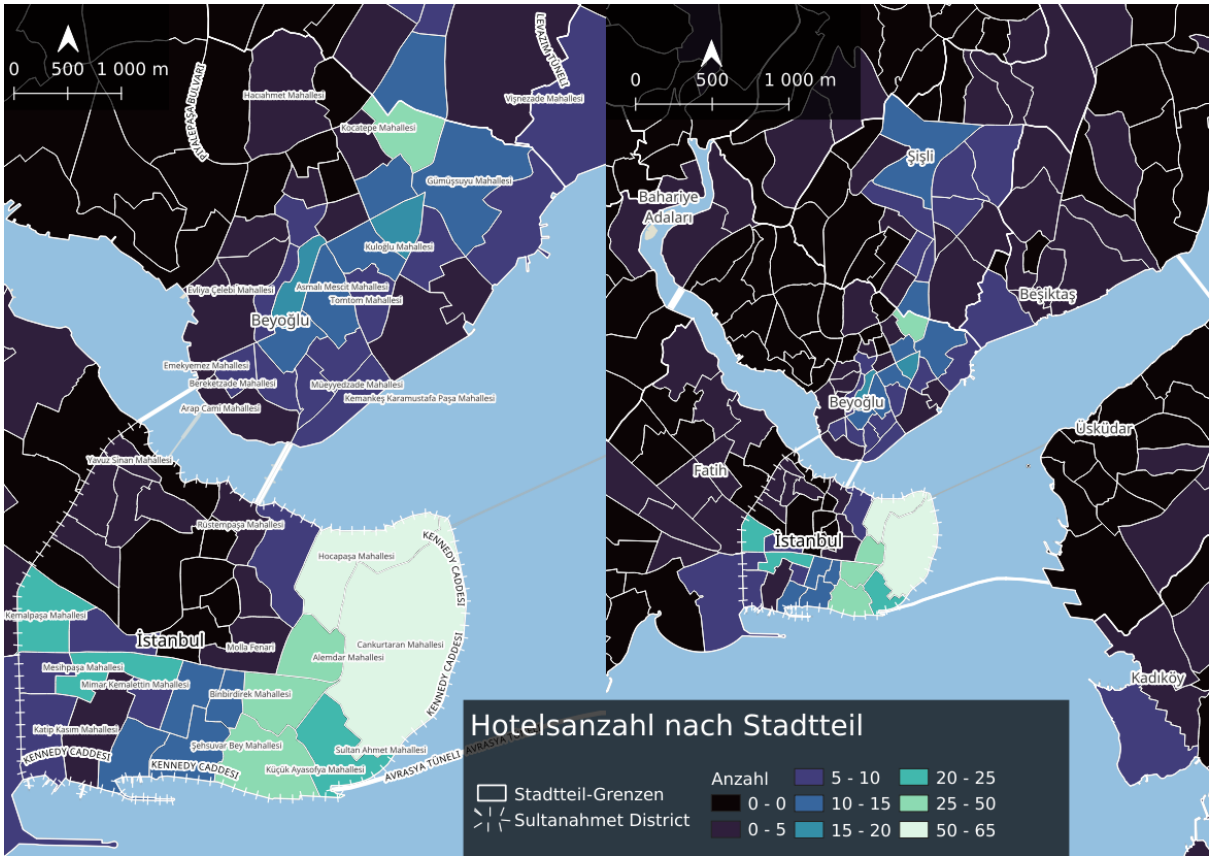

Figure 6: Anzahl der erhobenen Hotels nach Stadtteil (Quelle: eigene Darstellung 2022)

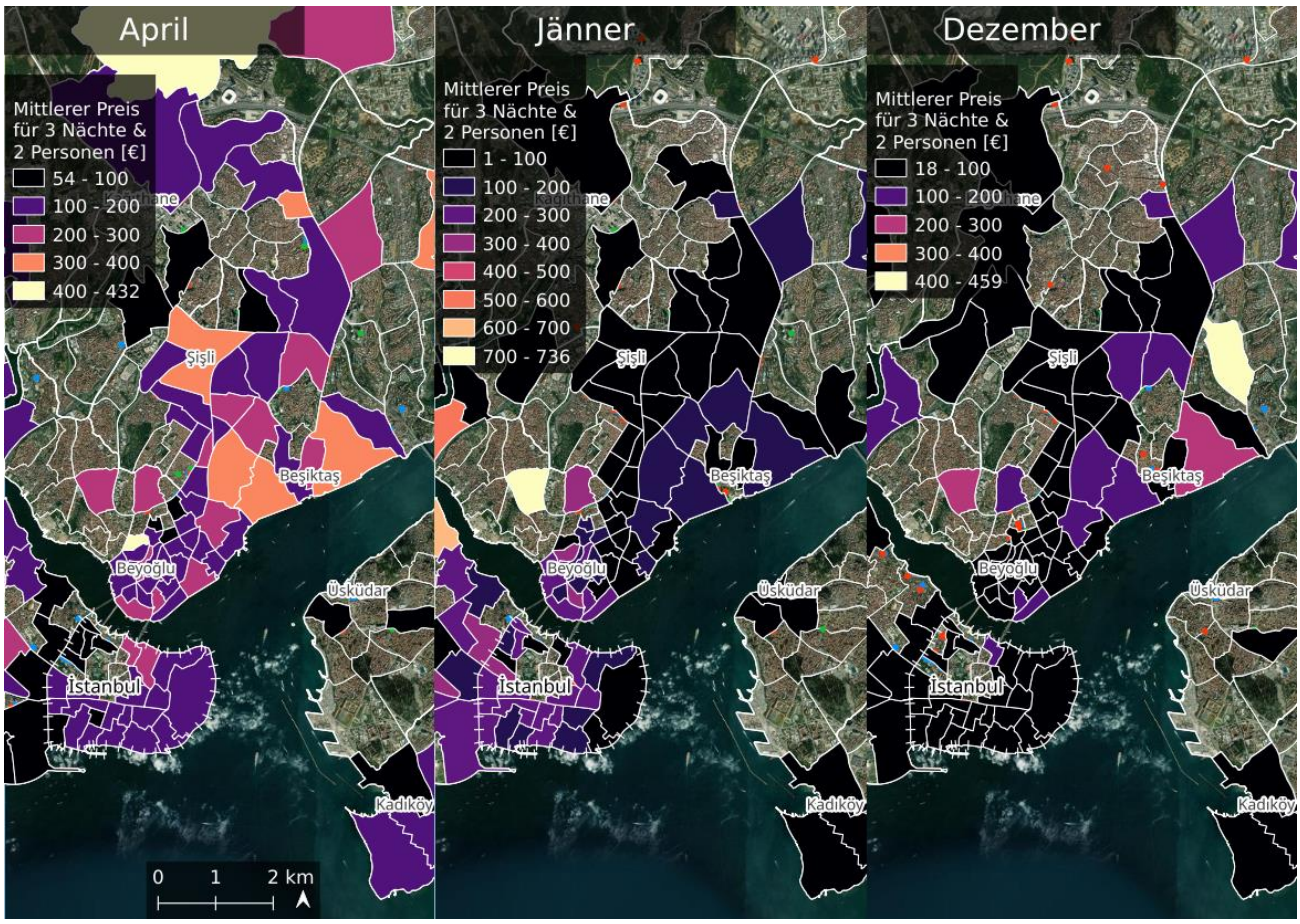

Figure 7: Hotelpreise nach Stadtteil (Quelle: eigene Darstellung 2022)

6 DISKUSSION UND OUTLOOK

Mit "autoCrawler" wird ein leicht zu bedienendes Werkzeug zur Datenerfassung zur Verfügung gestellt, welches es ermöglichen sollte längerfristig Daten von Booking.com zu erheben. Die Ereignisschleifen von Booking.com und Trivago sollten es ermöglichen, den Prozess ohne Abhängigkeit von CSS-Tags durchzuführen.

Die im Rahmen der Entwicklung erhobenen Daten zeigten eine Steigung des Durchschnittspreises von Dezember bis Jänner. Es stellten sich Probleme bei der Verknüpfung der Zeitreihen heraus, von 845 Objekten im Dezember konnten nur etwa 300 über die Position mit den neueren Datensätzen verknüpft werden. Dies könnte darauf hindeuten, dass Bookin.com eine Prozessierung der Koordinaten vornimmt, oder die Koordinaten von Dezember zu März aktualisiert hat. In der Altstadt ist eine Abnahme der erfassten Hotels von Dezember bis Jänner zu verzeichnen, im Süden der Altstadt im Stadtteil Küçükayasofya Mahallesi wurden im April 24 Hotels weniger als im Jänner detektiert. Die Detektion könnte jedoch davon abhängig sein, ob in Booking.com ausgebuchte Hotels nicht angezeigt werden. Generell ist eine Zunahme des Preises in der Altstadt zu beobachten, wobei der Hotelpreis in der Altstadt im Jänner höher war als im April. Dennoch stieg der Durchschnittspreis von 3 Tagen für zwei Personen von 147 € im Jänner zu 193 € im April.

Eine Weiterverfolgung der Arbeit kann sich mit der Verifikation der Hotel Koordinaten über die verfügbaren Adressen stützen. Ein Vergleich zwischen Geocoding und Booking-Koordinaten könnte aufschlussreiche Ergebnisse liefern. Es ist auch denkbar einen flexibleren Webcrawler zu entwickeln, dies könnte auf Basis von Adressen geschehen, ähnlich zu Li et al. (2012). Natürlich können mit dem entwickelten Webcrawler auch regelmäßig Daten eines Gebietes erhoben werden, um weitere Entwicklungen zu verfolgen.

7 LITERATUR

Ecma Internationa (Hg.) https://www.ecma-international.org/, zuletzt geprüft am 14.03.2022.

Flask (Hg.) Welcome to Flask — Flask Documentation (2.0.x). https://flask.palletsprojects.com/en/2.0.x/, zuletzt geprüft am 07.03.2022.

Kausar, M.A., Dhaka, V.S., Singh, S.K. (2013): Web crawler: a review. In: International Journal of Computer Applications 63, 2, .

Kunden, M. (Hg.) WorldWideWebSize.com | The size of the World Wide Web (The Internet). https://www.worldwidewebsize.com/, zuletzt geprüft am 06.03.2022.

Li, W., Goodchild, M.F., Church, R.L., Zhou, B. (2012): Geospatial Data Mining on the Web: Discovering Locations of Emergency Service Facilities. In: Zhou, S.,Zhang, S.,Karypis, G. (Hg.): Advanced Data Mining and Applications. Berlin, Heidelberg: Springer Berlin Heidelberg , S. 552–563.

Miller, H.J., Goodchild, M.F. (2015): Data-driven geography. In: GeoJournal 80, 4, S. 449–461.

Miniwatts Marketing Group (Hg.) World Internet Users Statistics and 2021 World Population Stats. https://www.internetworldstats.com/stats.htm, zuletzt geprüft am 06.03.2022.

Mozilla (Hg.) (2022): MDN Web Docs. https://developer.mozilla.org, zuletzt geprüft am 14.03.2022.

Selenium (Hg.) About Selenium. https://www.selenium.dev/about/, zuletzt geprüft am 06.03.2022.

W3C (Hg.) (2011): HTML5. https://dev.w3.org/html5/spec-LC/, zuletzt geprüft am 10.03.2022.

W3C (Hg.) (1990): The original proposal of the WWW, HTMLized. https://www.w3.org/History/1989/proposal.html, zuletzt geprüft am 10.03.2022.

8 STATEMENT

Sehr gut waren vor allem die schnellen Antworten auf Mails.

9 ABGABE 7 DELIVERIES

Öffentlicher Quellcode: https://github.com/raphi-web/autoCrawler
Daten, Quellcode in Dateiformat: https://drive.google.com/drive/folders/1KmYQugvOwb6xoGsmVs4mox00OLdN-u6u?usp=sharing