

Technische Universität Darmstadt
Fachbereich: Informatik
Sommersemester 2015

Praktikum
Question Answering Systems Project:
Semantic Technologies in IBM Watson

Projektdokumentation

Travel with Watson

Autor:
Marcel Beuth
Martina Kettenbach

[Einleitung](#)

[Projektbeschreibung](#)

[Funktionalitäten](#)

[Back-End](#)

[Datenbank](#)

[Server](#)

[XAMPP](#)

[Front-End](#)

[Schnittstellen zu anderen Systemen](#)

[Watson](#)

[Travel Corpus](#)

[Google API](#)

[Google Places API](#)

[Google Maps Geocoding API:](#)

[OpenStreetMap - Mapsforge](#)

[Graphhopper](#)

[Systemvoraussetzung](#)

[User Guide](#)

[Fazit](#)

[Quellen](#)

Einleitung

Dies Dokumentation wurde im Rahmen des Praktikums "Question Answering Systems Project: Semantic Technologies in IBM Watson" an der TU Darmstadt erstellt. Ziel des Praktikums war es eine Anwendung zu entwickeln, die das von IBM entwickelte Frage-Antwort-System „Watson“ integriert. Wir haben uns für den öffentlichen Travel Corpus entschieden, mit dessen Hilfe wir Reiseinformationen abfragen können, um diese dann in einer benutzerfreundlichen Umgebung, zum Beispiel in der Android App "Travel with Watson", anzuzeigen.

"Travel with Watson" ist ein Reiseführer für Android Smartphones, welcher den Nutzer helfen soll sich in einer unbekannten Stadt zu orientieren. So können Information, wie z.B. Sehenswürdigkeiten, Restaurants oder auch Hotels, abgefragt werden. Desweiteren können eigene Fragen erstellt und gespeichert werden.

Der "Drinking Tour Planer" bietet Ihnen die Möglichkeit sich eine eigene Drinking Tour zu erstellen. Sie können auswählen in welchem Radius sich die gewünschten Lokale befinden sollen und auch um welche Art (Kneipe, Club, usw...) es sich handeln soll.

Über die Navigationfunktionalität können Sie sich bequem von Ihrem aktuellen Standort zu den gefundenen Orten führen lassen oder auch eigen Start- und Endpunkte festlegen, zu denen Sie sich navigieren lassen möchten.

Projektbeschreibung

"Travel with Watson" beinhaltet alle wichtigen Funktionalitäten, die man braucht, um sich in einer nicht bekannten Stadt zurechtzufinden. Darüber hinaus können noch Fragen über Regionen oder auch z.B "How high is Mount Everest?" gestellt werden.

Funktionalitäten

- **Vordefinierte Fragen (Essen, Trinken, Sehenswürdigkeiten, usw...):**

Vordefinierte Fragen beziehen sich auf Städte, wie z.B. "Welche Sehenswürdigkeiten gibt es in Darmstadt". Diese werden bei der Installation der Anwendung mitgeliefert. Es wird dann entweder der aktuelle Standort berücksichtigt oder falls keine GPS Informationen vorhanden sind kann man auch die gewünschte Stadt eingeben. Aus den Informationen die uns Watson bereitstellt extrahieren wir „Locationinformationen" (Name, Adresse, Telefon, usw...).

- **Navigation vom aktuellen Standort zu den gefundenen Locations**

Um die Navigation nutzen zu können müssen erst die entsprechenden Karten von unserem Server heruntergeladen werden. Wenn die erforderliche Karte auf dem Gerät vorhanden ist, kann man sich zu den Reisezielen Navigieren lassen oder auch eigene Routen angeben.

- **Eigene Fragen erstellen und speichern:**

Falls man eine Frage stellen möchte, diese allerdings im Menü nicht vorhanden ist, kann man seine eigene Frage eingeben und diese gegebenenfalls auch speichern. Im Gegensatz zu den festgelegten Kategorien werden die Informationen, die wir von Watson erhalten, nicht weiter verarbeitet.

- **Speech-to-text:**

Fragen können über die Speech-to-text gestellt werden.

- **Text-to-Speech:**

Man kann sich die Antworten mittels Text-to-Speech vorlesen lassen.

- **Eine Drinking Tour erstellen:**

Es besteht die Möglichkeit, dass man sich eine Drinking Tour erstellt. Dafür holen wir uns mittels Watson alle Kategorien, aus denen man dann auswählen was kann, welche man haben möchte, z.B. Clubs und Bars. Anschließend kann man noch bestimmen innerhalb welchem Radius die Lokale voneinander entfernt sein sollen.

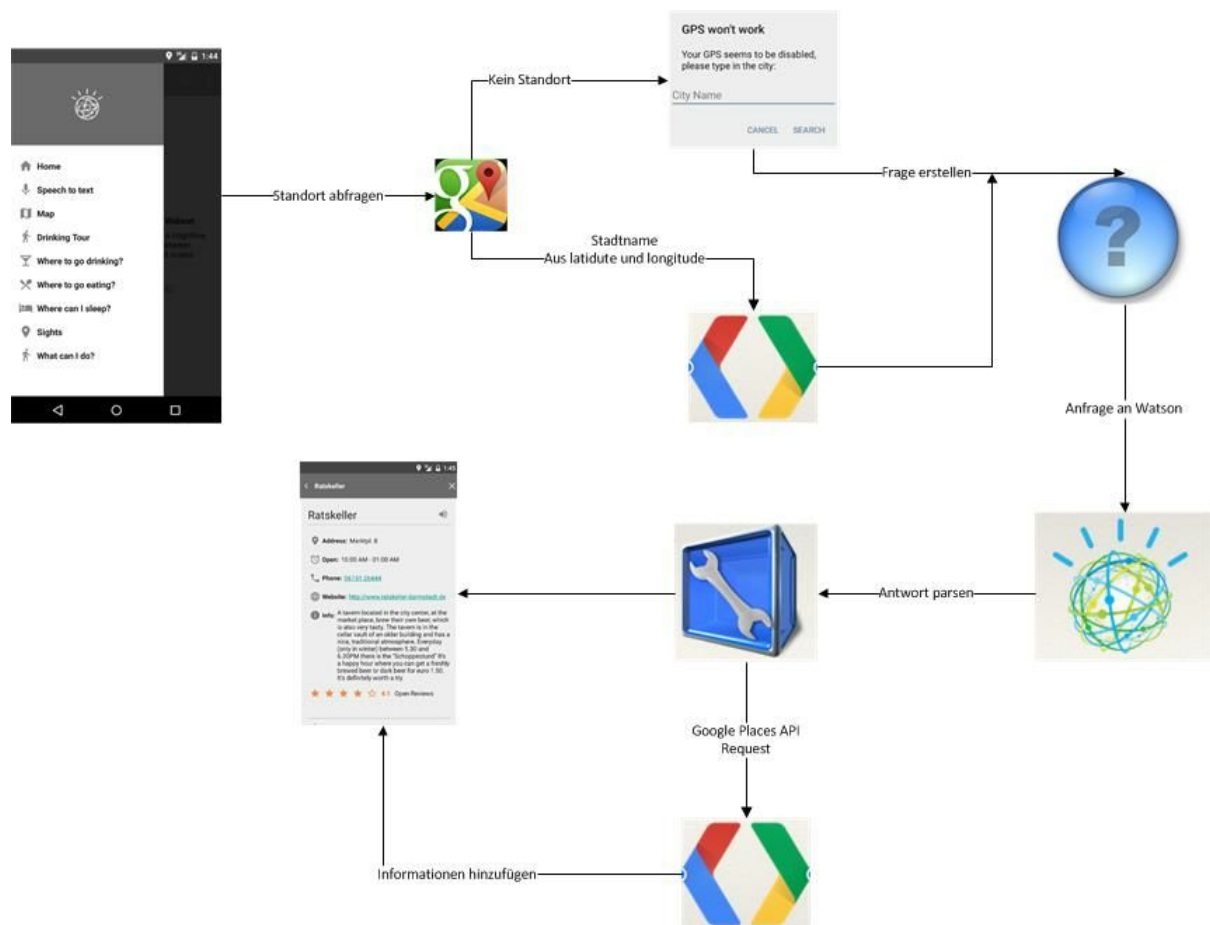
Back-End

Zum Back-End gehört die Verarbeitung der Daten, die wir von Watson erhalten, sowie wie das Vervollständigen der Daten aus weiteren Ressourcen.

- **Parsen der Vordefinierten Fragen:**

Als erstes extrahieren wir uns die gewünschten Daten aus der Watson-Quelle. Die Erkennung, der von uns benötigten Daten, erfolgt über Muster. Es werden z.B. alle Zahlenfolgen die mit „+49“ anfangen als Telefonnummern erkannt.

- **Füllen der leeren Felder durch die Google Places-API**



Datenbank

Zur Anwendung gehört auch eine kleine SQLite-Datenbank, die bei der Installation mitgeliefert wird. Diese besteht aus zwei unabhängigen Tabellen.

- **Questions:** In dieser Tabelle werden die eigen erstellten Fragen, mit den folgenden Feldern gespeichert:
 - QuestionID
 - Titel: Wie die Frage im Menü angezeigt werden soll.
 - Question: Die eigentliche Frage.

- Maps: In dieser Tabelle werden Informationen, über die bereits heruntergeladenen Maps gespeichert.
 - ID
 - Name
 - Size

Server

Mittels XAMPP haben wir uns eine Web-Server erstellt, auf denen die Routing-Maps von Deutschland bereitgestellt werden. Diese Routing-Maps können von unserer Android Anwendung heruntergeladen werden.

XAMPP

Das Programmpaket XAMPP enthält alle notwendigen Programme zur Einrichtung eines Web-Servers.

Front-End

Das Front-End unserer Anwendung ist, wie bereits beschrieben, eine Android Anwendung. Da es natürlich von Vorteil ist, Reisedaten mobil zur Verfügung zu haben um auch Informationen, wie den aktuellen Standort, dynamisch in die Frage mit einzubeziehen oder mittels Navigation sein gewünschtes Ziel einfach zu finden.

Eine genau Beschreibung der Oberfläche befindet sich im Abschnitt User Guide.

Schnittstellen zu anderen Systemen

Watson

Watson ist ein von IBM entwickeltes Frage-Antwort-System. Ziel des Programms ist eine gestellte Frage semantisch zu verstehen und daraufhin eine präzise Antwort auszugeben. Um dies zu realisieren greift Watson auf einen Domain spezifischen Korpus zurück, der Informationen/Dokumente über das gewünschte Gebiet enthält. Durch hinzufügen neuer Informationen erlernt dieser weiter Sachen (wie der Menschen auch) und kann daraufhin immer genauere und bessere antworten zurückliefern.

Travel Corpus

Da wir für unsere Anwendung Reiseinformationen benötigen, benutzen wir den öffentlichen Travel Corpus. Dieser erhält seine Reiseinformationen aus den folgenden Datenquellen:

- Wikivoyage
- CDC Travel
- TSA.gov

Google API

Google Places API

Die Google Place API enthält Informationen über Places (Adresse, Reviews, ...), so dass wir mittels der API unsere Daten vervollständigen könne.

Zu jeder Location die wir erhalten, wird zuerst eine Anfrage an die Google Places API gestellt:

```
https://maps.googleapis.com/maps/api/place/textsearch/json?query=Ratskeller%20Darmstadt&key=API_KEY
```

Wobei **"Ratskeller%20Darmstadt"** unsere Location ist. Als Anfrage bekommen wir nun folgendes JSON:

```
{
  "html_attributions" : [],
  "results" : [
    {
      "formatted_address" : "Marktpl. 8, 64283 Darmstadt, Deutschland",
      "geometry" : {
        "location" : {
          "lat" : 49.872132,
          "lng" : 8.655927999999999
        }
      },
      "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
      "id" : "f5185a3ec4359790d159f7c3b2102f8e2cefaec8",
      "name" : "Darmstädter Ratskeller Hausbrauerei GmbH",
      "opening_hours" : {
        "open_now" : true,
        "weekday_text" : []
      },
      "photos" : [
        {
          "height" : 560,
          "html_attributions" : [
            "\u003ca href=\\"https://maps.google.com/maps/contrib/108726457715821657628\u003eDarmstädter Ratskeller Hausbrauerei GmbH\u003c/a\u003e"
          ],
          "photo_reference" : "CmRdAAAAAzRuMkj1K40D-cixii3RbydEtkJOQViYyZSJrouvvmG7qoS4Fomse4wbNrC1wZu3n5TLyHKhWOaNNWwW1i7jsLoS-mgHYXKmICAht8hE26Un8Q42DXGigOUvhahCLxLpEhCuPQaVFmJm4vJPQA0sGy0PGhSiVWIE_nS7aHBifbEwfb6t8XGnw",
          "width" : 750
        }
      ],
      "place_id" : "ChIJoRrt42RwvUcRs6sjrxLzhVw",
      "rating" : 4.1,
      "reference" : "CoQBfAAAAFcG3rL1hTKMaZreXmqI39djtzSYe_6UDZymDqLe-BhFPf-IB3-5TPU9K9Dmn5kFJXqPP85DwBteSYOTTohmU5K7hV_xDk2bcHL7aIj9BpdYgi1Uz5g8Txr0fy1EJy6AmADVmgcAwuQFxtzsquoU_wHBGheTsxhqgffB0wDBvAdEhDrbmh95h6plkWG-5oqaRjDGPw_S5ybYsjQPi5QjKtJkUyLdmQ",
      "types" : [ "restaurant", "food", "point_of_interest", "establishment" ]
    }
  ],
}
```

```
}
  "status": "OK"
}
```

Aus diesem JSON Objekt holen wir uns nun die hervorgehobene “place_id”.

Mit dieser können wir nun eine neue Anfrage stellen um mehr Information über die Location zu erhalten:

```
https://maps.googleapis.com/maps/api/place/details/json?placeid=ChIJorRt42RwvUcRs6sjrxLzhVw&language=en&key=API_KEY
```

Mit dieser Anfrage bekommen wir folgendes JSON:

```
"html_attributions" : [],
"result" : {
  "address_components" : [
    {
      "long_name" : "8",
      "short_name" : "8",
      "types" : [ "street_number" ]
    },
    {
      "long_name" : "Marktplatz",
      "short_name" : "Marktpl.",
      "types" : [ "route" ]
    },
    {
      "long_name" : "Darmstadt",
      "short_name" : "Darmstadt",
      "types" : [ "locality", "political" ]
    },
    {
      "long_name" : "Germany",
      "short_name" : "DE",
      "types" : [ "country", "political" ]
    },
    {
      "long_name" : "64283",
      "short_name" : "64283",
      "types" : [ "postal_code" ]
    }
  ],
  "adr_address" : "\u003cspan class=\u0026quot;street-address\u0026quot;\u003eMarktpl. 8\u003cspan class=\u0026quot;postal-code\u0026quot;\u003eD-64283\u003cspan class=\u0026quot;locality\u0026quot;\u003eDarmstadt\u003cspan class=\u0026quot;country-name\u0026quot;\u003eGermany\u003cspan class=\u0026quot;formatted_address\u0026quot; : \u0026quot;Marktpl. 8, 64283 Darmstadt, Germany\u0026quot;,
  "formatted_phone_number" : \u0026quot;06151 26444\u0026quot;,
  "geometry" : {
    "location" : {
      "lat" : 49.872132,
      "lng" : 8.655927999999999
    }
  },
  "icon" : "https://maps.gstatic.com/mapfiles/place_api/icons/restaurant-71.png",
  "id" : "f5185a3ec4359790d159f7c3b2102f2b2cefaec8",
  "international_phone_number" : "+49 6151 26444",
  "name" : "Darmstädter Ratskeller Hausbrauerei GmbH",
  "opening_hours" : {
    "open_now" : true,
    "periods" : [
      ...
    ],
    "weekday_text" : [
      "Monday: 10:00 am – 1:00 am",
      "Tuesday: 10:00 am – 1:00 am",
      "Wednesday: 10:00 am – 1:00 am",
      "Thursday: 10:00 am – 1:00 am",
      "Friday: 10:00 am – 1:00 am",
      "Saturday: 10:00 am – 1:00 am",
      "Sunday: 10:00 am – 1:00 am"
    ]
  }
}
```



```

    ],
    "photos" : [
      ...
    ],
    "place_id" : "ChIJorRt42RwvUcRs6sjrxLzhVw",
    "rating" : 4.1,
    "reference" :
    "CoQBF4AAAPF4xfbohu20kNrYJQUtCE_h1BQgxu9gZjhPBN80L5dzEHW7gBUDxDw5vLXwxh_oQwaW1dpnDLQPGUagoW95Hk0EkrcuA8iu
    sJ0Ggwwo05HuCWedipYqUh_xoEcch3LG2gJTCjlravSU_yP4w2ZXyKd1vFyFis5MdOTjiQpMP0ADEhAYI-geIEKT8cRHVTQ_-i-xQGhSr7DvDk
    D0oJO8yyIvT3RYnkrItaw",
    "reviews" : [
      {
        "aspects" : [
          {
            "rating" : 2,
            "type" : "overall"
          }
        ],
        "author_name" : "Mikhail Okun",
        "author_url" : "https://plus.google.com/105013103240544593019",
        "language" : "en",
        "rating" : 4,
        "text" : "A lot of tables, fast service. Not a big food selection, nothing for gourmets. The beer is excellent!",
        "time" : 1436890400
      },
      {
        "aspects" : [
          {
            "rating" : 2,
            "type" : "overall"
          }
        ],
        "author_name" : "Sagar Nagalkar",
        "author_url" : "https://plus.google.com/111324913103990345008",
        "language" : "en",
        "rating" : 4,
        "text" : "Wonderful place...one could spend a nice time during summer ",
        "time" : 1415657392
      },
      ...
    ],
    "scope" : "GOOGLE",
    "types" : [ "restaurant", "food", "point_of_interest", "establishment" ],
    "url" : "https://plus.google.com/108726457715821657628/about?hl=en",
    "user_ratings_total" : 63,
    "utc_offset" : 120,
    "vicinity" : "Marktplatz 8, Darmstadt",
    "website" : "http://www.ratskeller-darmstadt.de/"
  },
  "status" : "OK"
}

```

Aus diesem JSON Objekt extrahieren wir alle hervorgehobenen Information(Reviews, Öffnungszeiten, Adresse, Telefonnummer).

Google Maps Geocoding API:

Die Google Maps Geocoding API bietet uns die Möglichkeit, eine Adresse in Longitude und Latitude umzuwandeln und umgekehrt. Diese verwenden wir um aus dem aktuellen Standort den Stadtname zu erhalten, so dass wird diesen in unsere Frage mit einbauen könne.

Adresse zu Longitude/Latitude:

```
https://maps.googleapis.com/maps/api/geocode/json?address=Marktpl. %208, %2064283 %20Darmstadt&key=API_KEY
```

Longitude/Latitude zu Adresse:

<https://maps.googleapis.com/maps/api/geocode/json?latlng=49.872132,8.655927999999999&key=AlzaSyBFDjNfBXoZw31PZ3MnYcdvqelu8s5GBFU>

OpenStreetMap - Mapsforge

Wir haben uns dafür entschieden OpenStreetMap-Mapsforge zu verwenden, da es ein freies Projekt, das für jeden frei nutzbare Geodaten sammelt (Open Data). Im Gegensatz zu den Google-Karten, die kostenlos, aber nicht frei sind. Der Vorteil einer Freien Software ist, dass damit keine Bedingungen an die Benutzung geknüpft ist. So muss man sich keine Gedanken machen, ob man die Karte z.B. ausdrucken und weitergeben kann. Dies ist oftmals schon nicht erlaubt.

Desweiteren bietet OpenStreetMap die zugrundeliegenden Geodaten zur Verfügung, welches einem einen größeren Spielraum zur Verfügung stellt. Dadurch ist man z.B. in der Lage seine eigenen Routing-Algorithmen auszuprobieren.

Graphhopper

Um das Routing in unserer Anwendung zu ermöglichen verwenden wir Graphhopper. Graphhopper ist ein schnelles, flexibles Open Source Projekt, welches mithilfe von OpenStreetMaps Karten Routing Daten erstellt. Mit diesen Routing Daten ist es möglich für Auto, Fahrrad und Fußgänger Routenplanungen zu erstellen.

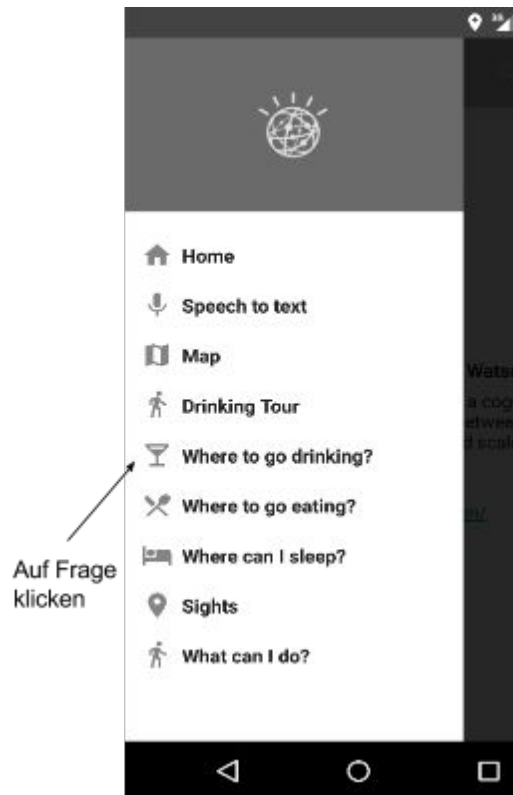
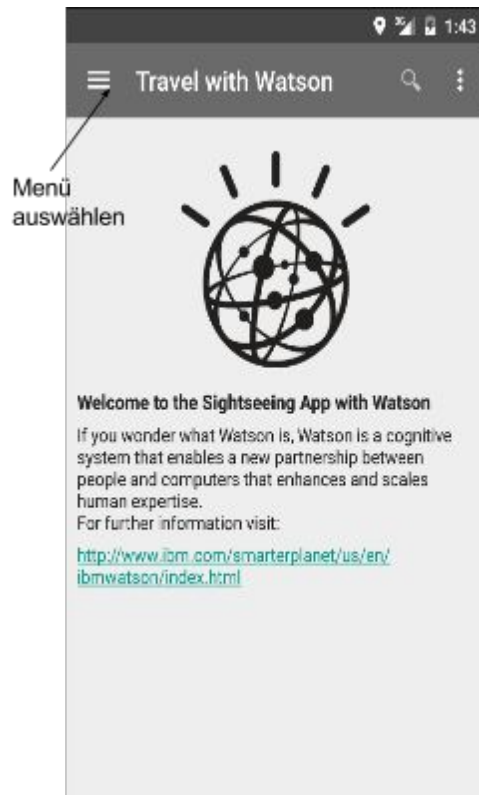
Systemvoraussetzung

Voll funktionsfähig und getestete Funktionalität der Anwendung ab Android API Version 21 oder höher.

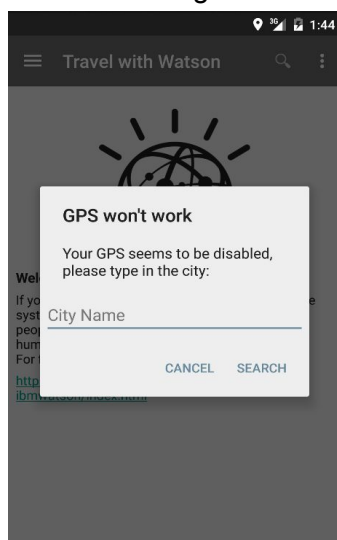
User Guide

Vordefinierte Frage stellen

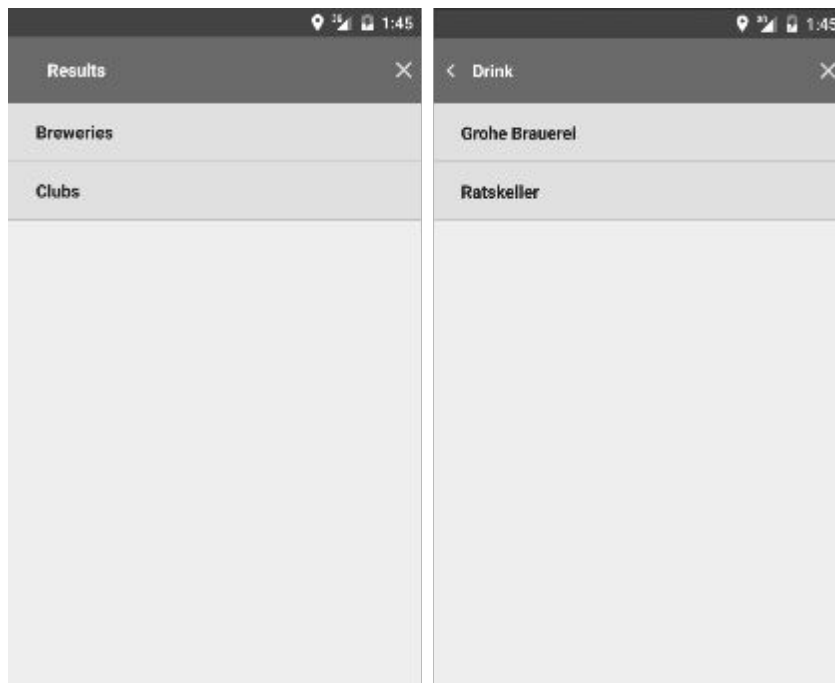
Klicke auf Menü und dann auf Frage



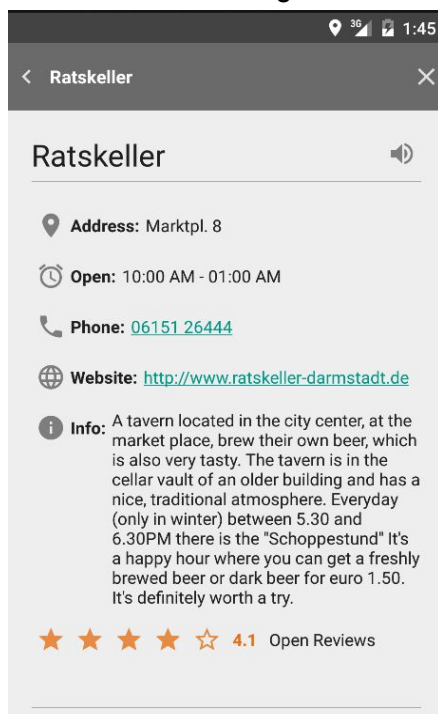
Falls kein GPS vorhanden ist erscheint der folgende Dialog, in dem der gewünschte Stadtname eingeben werden kann.



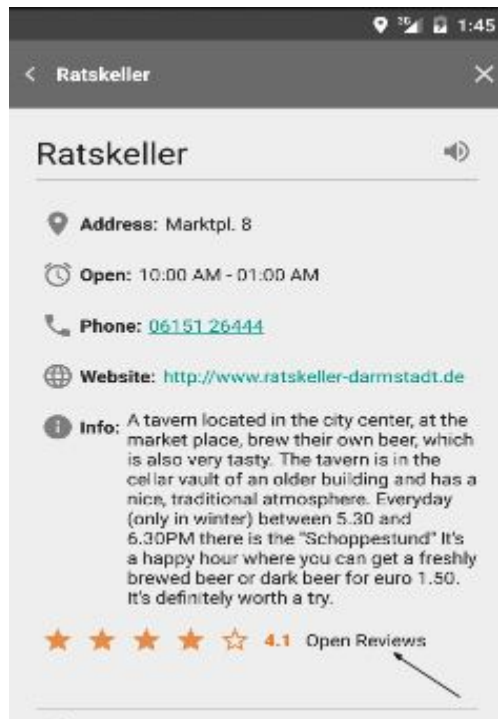
Anschließend erscheinen die gefundenen Kategorien. Nachdem die gewünschte Kategorie (in dem Fall "Breweries") ausgewählt wurde, erscheint eine Liste mit den gefundenen Locations



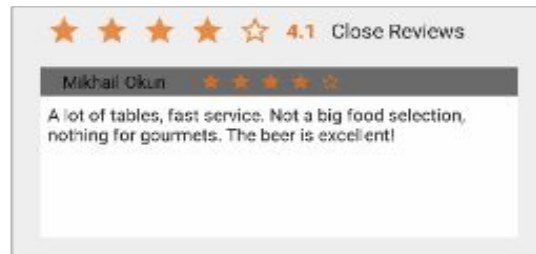
Auswählen der gewünschten Location. Nachdem eine Location ausgewählt worden ist, erscheinen dann die genaueren Details.



Um sich die Reviews anzuzeigen klicke auf "Open Reviews"

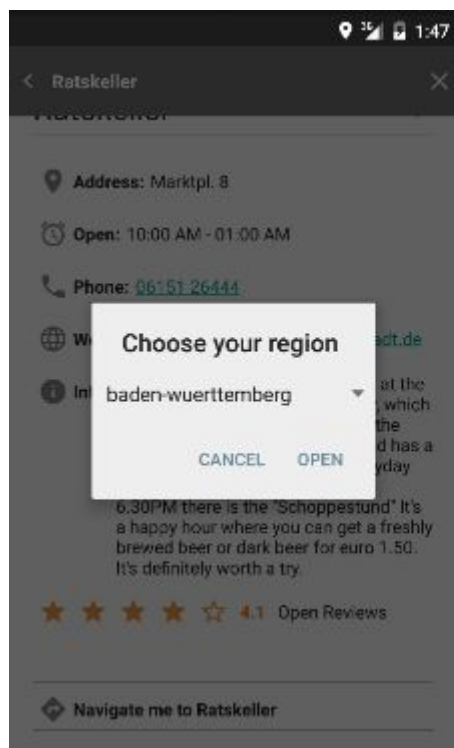


Durch Wischen kann man sich die nächste Review ansehen

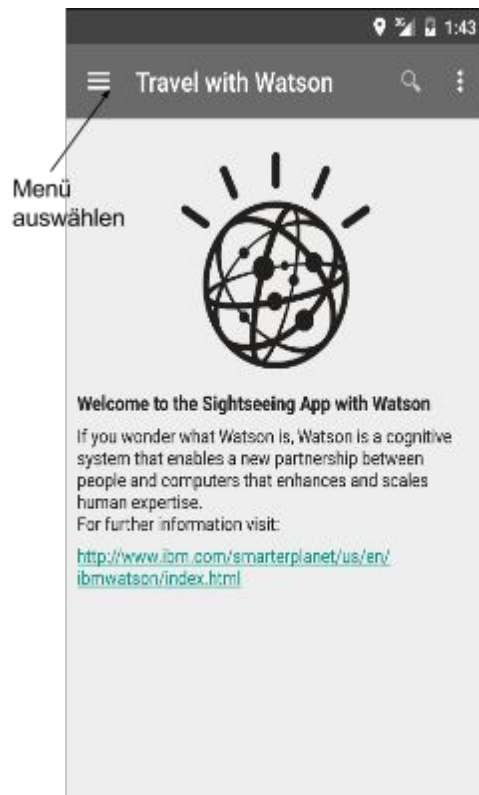


Navigieren zum Ziel:

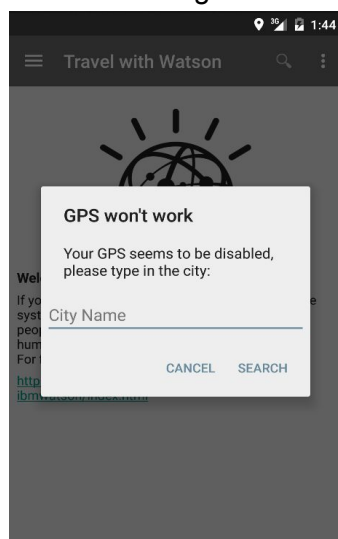
Um sich zum Ziel navigieren zu lassen muss man auf Navigate me to "Ratskeller" und anschließend die Karte auswählen, in der sich die Stadt befindet. Daraufhin erscheint die Karte



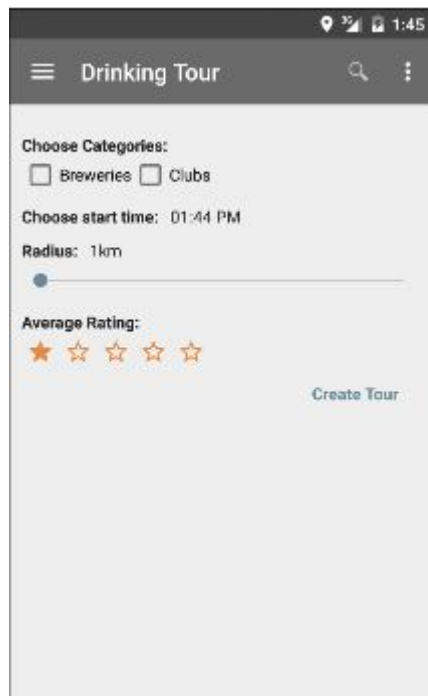
Drinking Tour



Falls kein GPS vorhanden ist erscheint der folgende Dialog, in dem der gewünschte Stadtname eingeben werden kann.

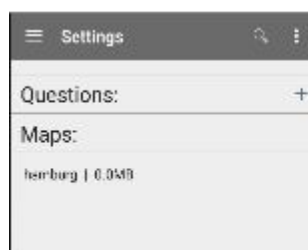


Nachdem Sie die gewünschten Bedingungen festgelegt haben und auf “create Tour” geklickt haben, erscheinen die Details. Um die nächste Location zu bekommen klicken Sie auf “Next Location”.



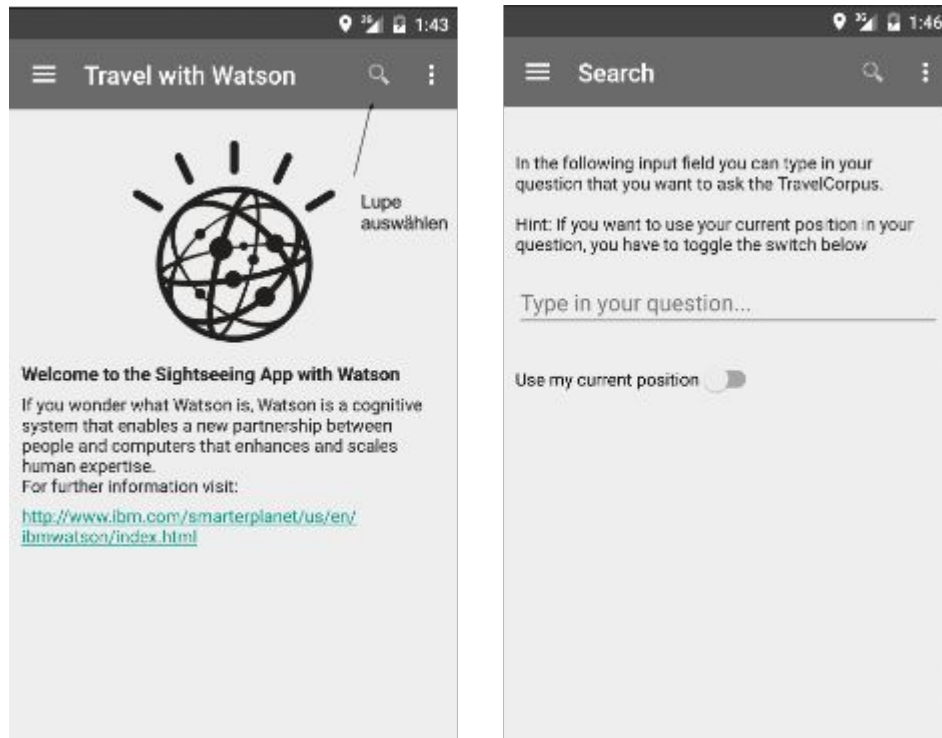
Eigene Fragen erstellen

Um eine eigene Frage zu erstellen, gehen Sie auf “Settings” → “+”. Anschließend geben Sie einen Titel (wie die Frage im Menü angezeigt werden soll) und die eigentliche Frage ein. Nachdem Sie auf “Save” geklickt haben, ist die Frage gespeichert und kann im Menü ausgewählt werden.



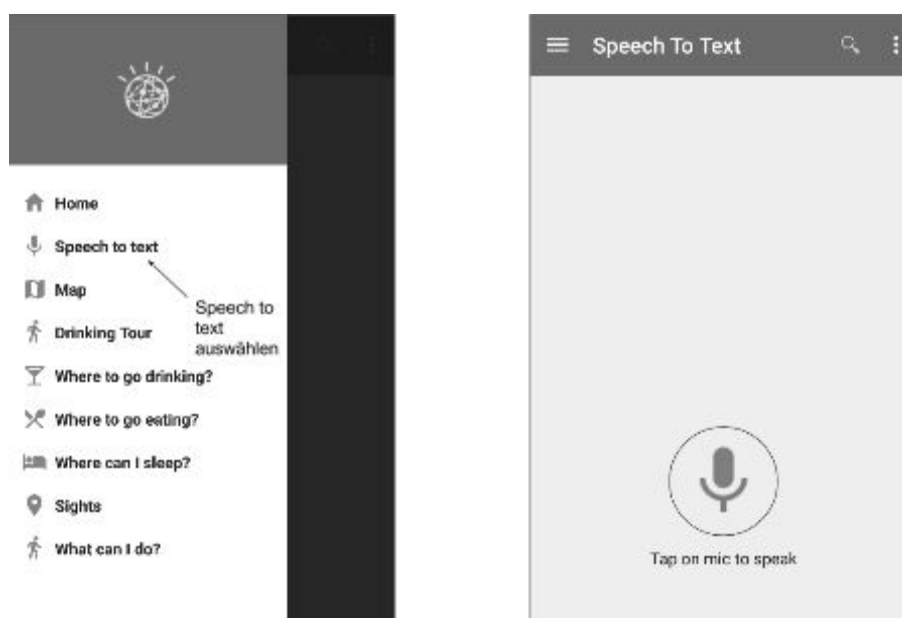
Eigene Frage stellen

Um eine eigene Frage zu stellen klicken Sie auf oben auf die Lupe, danach könne Sie die Frage eingeben



Frage stellen mit Speech-to-Text

Um eine Frage, mittels Speech-to-Text, zu stellen klicken Sie auf "Speech to Text" und anschließend auf "Tap on mic to speak". Danach einfach nur Frage stellen



Fazit

Da wir in beide noch keine Erfahrungen im Bereich NLP und Android Entwicklung gemacht haben, war dies ein interessanter Einstieg in beide Themen.

Wie bereits genannt, haben wir den Öffentlichen Travel Korpus verwendet, somit haben wir Watson nur als Informationsquelle (Blackbox) verwendet. Dies hat uns jedoch trotzdem ein paar Einblicke in die Probleme von NLP gegeben. Da wir aus den von Watson erhaltenen Antworten Informationen extrahieren wollten, haben wir uns ein bisschen mit dem Thema auseinandersetzen müssen. Wie erkennen wir die gewünschten Daten, z.B. Straßennamen oder auch Telefonnummern in einem Text? Unser Lösungsansatz war es, durch Nutzung von Mustern, Daten zu erkennen. Mit diesem Ansatz konnten wir schon einen Großteil der Daten erkennen, jedoch gehen viel Informationen verloren, da sie nicht in das Angegebene Muster passen.

Mit der Erstellung der Android Anwendung haben wir uns Anfangs recht schwer getan, da wir erst einmal den Aufbau einer Android Anwendung und deren Funktionalitäten verstehen mussten. Leider hatten wir auch kein geeignetes Testgerät, sondern nur eins mit einer veralteten SDK Version und den Emulator. Dadurch konnten wir nicht die ganze Funktionalität auf einmal testen, sondern nur ein Teil auf dem Emulator und ein Teil auf dem Testgerät.

Quellen

IBM Watson:

<http://www-05.ibm.com/de/watson/>

Google Places API:

<https://developers.google.com/places/web-service/>

Google Maps Geocoding API:

<https://developers.google.com/maps/documentation/geocoding/intro>

Open StreetMap

<http://www.openstreetmap.de/>

<http://www.openstreetmap.de/faq.html>

<https://github.com/mapsforge/mapsforge>

XAMPP

<https://de.wikipedia.org/wiki/XAMPP>

<https://www.apachefriends.org/de/index.html>

Graphhopper

<https://graphhopper.com/>

<https://github.com/graphhopper/graphhopper/>

<https://github.com/graphhopper/graphhopper/blob/0.5/docs/android/index.md>