Fachbereich Informatik und Medien

BACHELORARBEIT

Entwicklung und Evaluation eines multimodalen Empfehlungssystems für Lokationen

Vorgelegt von: Benjamin Hoffmann

am: 21.09.2012

zum

Erlangen des akademischen Grades

BACHELOR OF SCIENCE

(B.Sc.)

Betreuer: Prof. Dr.-Ing. Jochen Heinsohn,
Dipl.-Inf. Ingo Boersch,
Dr. Tatjana Scheffler,
Dipl.-Inf. Rafael Schirru

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit zum Thema

Entwicklung und Evaluation eines multimodalen Empfehlungssystems für Lokationen

vollkommen selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Brandenburg an der Havel, den 21.09.2012

Unterschrift

Danksagung

Ich möchte mich ganz herzlich bei Tatjana Scheffler und Rafael Schirru bedanken. Sie haben sich immer viel Zeit genommen, mich motiviert und ohne ihre Anmerkungen und Hinweise wäre diese Arbeit nicht das geworden, was sie ist. Ein großes Dankeschön geht auch an Professor Jochen Heinsohn und Ingo Boersch für ihre großartige Unterstützung.

Weiterhin möchte ich allen Teilnehmern der Evaluierung und dem Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) einen Dank aussprechen.

Zusammenfassung

In den vergangenen Jahren haben multimodale Anwendungen den Schritt von der Forschung in die Praxis gemacht. Ziel der Kombination verschiedener Modalitäten wie Sprache und Touch (Berührung) ist es, eine bessere und natürlichere Interaktion zwischen Benutzer und System zu ermöglichen. In dieser Arbeit wurde ein leichtgewichtiges, multimodales, kartenbasiertes Empfehlungssystem für das Auffinden von Lokationen zur Erfüllung von Aufgaben entwickelt. Existierende Dienste und Komponenten – wie zum Beispiel zur Georeferenzierung oder zur Spracherkennung – wurden erfolgreich zu einem funktionierenden System kombiniert. Der Einsatz von Webtechnologien wie HTML, CSS und JavaScript vereinfacht die Portierung der mobilen Android-App auf andere Plattformen. Eine in JavaScript definierte Grammatik erlaubt verschiedene Varianten bei der Spracheingabe.

Im Anschluss an die Entwicklung wurde eine Evaluation des Systems mit zwölf Versuchsteilnehmern vorgenommen. Diese beinhaltete neben dem Lösen von jeweils sechs Aufgaben die Beantwortung zweier Fragebögen. Es wurde insbesondere untersucht, ob Daten sozialer Netzwerke den Benutzer bei der Auswahl von Lokationen unterstützen können (H1) und ob Sprache die Eingabe erleichtert (H2).

Die Befragung zeigt, dass 11 von 12 Personen der Meinung sind, dass soziale Netzwerke bei der Entscheidungsfindung helfen. Zehn der 12 Probanden bevorzugten die (initiale) Eingabe via Sprache. Nach dem Lösen der Aufgaben verbesserte sich die Bewertung der Nützlichkeit von Spracheingabe bei fünf Personen – nur eine Person änderte ihre Einschätzung zum Negativen. Beide Hypothesen ließen sich somit bestätigen.

Abstract

Over the past few years, multimodal applications have made a leap from research to being applicable in the field. The purpose of combining individual modalities, like speech and touch, is to enable a more natural interaction between user and system. In this thesis, I present a lightweight, multimodal, map-based recommendation system for finding locations to perform user-specified tasks. Existing services and components, e.g. for geocoding or speech recognition, were successfully combined to form a working system. The use of web technologies such as HTML, CSS and JavaScript simplifies the adaptation of the mobile Android app to other platforms. A JavaScript-based grammar allows for different variations of spoken input.

Subsequently, an evaluation of the system with 12 employees of the DFKI was conducted. After solving six different tasks, each test subject was required to complete two questionnaires. In particular, two hypotheses were examined: (H1) information from social networks supports the decision-making process and (H2) the modality speech makes it easier to use the application. The evaluation shows that 11 out of 12 subjects found that social data support their decision making. Speech as an initial input modality is preferred by 10 out of 12 test subjects over touch input. After solving the tasks, five subjects improved their rating of the usefulness of spoken input. Only a single person, who had had a bad experience, lowered his/her score. Both hypotheses have been confirmed.

Inhaltsverzeichnis

1 Einleitung

Die große Verbreitung von Smartphones veranlasst Wissenschaftler auf der ganzen Welt, neue Formen der Interaktion zu erforschen. Eingaben via Touch, Sprache und Gesten können kombiniert werden und ermöglichen eine Benutzeroberfläche, die vielseitige Benutzerwünsche erfüllt.

Fortschritte bei der Spracherkennung und die steigende Leistung mobiler Geräte sind ausschlaggebend für die Verwendung von Sprache in Smartphones. Apples *Siri* – ein intelligenter, sprachgesteuerter Assistent für das iPhone – und Googles *Voice Search* veranschaulichen diesen Trend hin zu neuartigen, alternativen Interaktionskonzepten.

1.1 Motivation

Grundidee des Projektes ist es, Menschen in urbanen Gegenden dabei zu unterstützen, geeignete Lokationen zu finden, um definierte Aufgaben zu erledigen. Smartphone-Besitzer verlassen sich mehr und mehr auf ihre Geräte. Laut einer repräsentativen BITKOM-Umfrage besitzt jeder Dritte (34%) ein Smartphone, „bei unter 30-Jährigen ist es sogar jeder zweite“. [BIT12]

Man stelle sich folgende Situation vor: Florian aus Wien besucht Freunde in Berlin. Auf dem Weg fällt ihm ein, dass es doch nett wäre, das Zusammentreffen in einer schönen Bar zu zelebrieren. Er hat von Bekannten gehört, dass es in Berlin Mitte gute Restaurants geben soll. Da das Tippen auf dem mobilen Gerät mühsam ist, benutzt er das Sprachinterface der Anwendung und findet schnell eine passende Lokation.

Im Anwendungsszenario des Projektes ist der Benutzer unterwegs und möchte etwas unternehmen. Derzeit sind hierfür zwar einige mobile Anwendungen – wie zum Beispiel von Qype oder Google – verfügbar, aber die relevanten Informationen sind an vielen Orten verteilt: So hat jedes soziale Netzwerk seine eigene App mit eigenem Bedienkonzept und bestimmten Informationen.

Zudem ist die Bedienung und Usability oft eingeschränkt - Orte müssen mühsam eingetippt oder über Karten-Interfaces gesucht und ausgewählt werden, der Nutzer folgt typischerweise der Anleitung des Geräts. Innovative Konzepte wie *Siri* bieten zurzeit noch keine Unterstützung für das gewählte Szenario.

1.2 Hypothesen

Im Zuge der Arbeit sollte eine multimodale, mobile Anwendung zur Empfehlung von Lokationen entwickelt werden. Multimodal bedeutet, dass die Eingabe auf verschiedene Arten erfolgen kann. Hervorzuheben sind die Modalitäten Sprache und Touch. Lokationen im Sinne der Anwendung stellen Orte dar, an denen vom Benutzer spezifizierte Aufgaben erledigt werden können. Das System sucht nach *Points of Interest* (POIs, zu deutsch *Orte von Interesse*) in der gewünschten Umgebung zur Erfüllung der angegebenen Aktivität.

Es wurden zwei Hypothesen untersucht:

H1 Für die Erledigung von Aufgaben in urbanen Gebieten (Einkaufen, Freizeitgestaltung, et cetera) können Daten aus sozialen Netzwerken genutzt werden, um den Benutzer bei der Auswahl geeigneter Lokalitäten zu unterstützen.

H2 Die Bedienung einer Smartphone-Applikation zum Finden von Lokationen kann durch die Modalität Sprache erleichtert werden.

Seit 2003 – dem Jahr, in dem die Bezeichnung *Web 2.0* geprägt wurde – lässt sich ein Internet ohne soziale Netzwerke kaum noch vorstellen. *Social-Networking*-Plattformen, wie zum Beispiel *Facebook*[1] (seit Februar 2004), *Yelp*[2] (seit Oktober 2004) oder *Qype*[3] (seit November 2005), werden von vielen Internetnutzern besucht. In dieser Arbeit sollte überprüft werden, ob sich Daten sozialer Netzwerke für die Verwendung in diesem konkreten Anwendungsfall eignen und ob sie den Nutzer bei der Suche nach geeigneten Lokationen unterstützen. (H1)

Außerdem sollte in dieser Arbeit untersucht werden, ob Sprache in dem beschriebenen Anwendungskontext die Benutzung der Anwendung erleichtert. (H2)

1.3 Abgrenzung

Die Arbeit umfasst nicht die Entwicklung einer marktreifen, multimodalen, mobilen Anwendung. Bei dem zu entwickelnden System handelt es sich viel mehr um einen evolutionären

[1] `http://www.facebook.com/`
[2] `http://www.yelp.com/`
[3] `http://www.qype.com/`

Prototypen. Damit sollen die in Abschnitt 1.2 besprochenen Aussagen auf ihre Gültigkeit untersucht werden. Dies bedeutet unter anderem, dass sich die Anwendung auf den Großraum Berlin beschränkt. Weitere Einschränkungen bestehen bei der Auswahl von Kategorien beziehungsweise Orten.

Für die Entwicklung werden verschiedene, existierende Dienste und Komponenten, wie zum Beispiel zur Sprachverarbeitung und Geolokalisierung, verwendet und zu einem funktionierenden Gesamtsystem kombiniert. Nähere Anforderungen werden in Kapitel 2 besprochen.

1.4 Resultat

Es wurde eine Evaluierung des Systems mit 12 deutschsprachigen Mitarbeitern des DFKI durchgeführt. Jedem Versuchsteilnehmer wurden sechs Aufgaben gestellt, wobei die Wahl der Modalität bei den ersten vier vorgegeben und bei den letzten zwei offen gelassen wurde. Nach dem Versuch musste jeder Proband zwei Fragebögen ausfüllen. Außerdem wurde das Experiment für eine spätere Auswertung mit einer Digitalkamera aufgezeichnet.

Elf von 12 Personen sind der Meinung, dass soziale Netzwerke bei der Entscheidungsfindung helfen. Zehn der 12 Probanden bevorzugten die (initiale) Eingabe via Sprache. Die Nützlichkeit der Spracheingabe wurde vor und nach dem Experiment abgefragt. Nach dem Lösen der Aufgaben verbesserte sich die Bewertung dieser Frage bei fünf Personen – nur eine Person änderte ihre Einschätzung zum Negativen. Beide Hypothesen ließen sich somit bestätigen.

1.5 Aufbau der Arbeit

Die Arbeit ist folgendermaßen gegliedert: Im zweiten Kapitel werden die Anforderungen an das System beschrieben, im dritten Kapitel werden verwandte Arbeiten vorgestellt. Das vierte Kapitel beschreibt die Konzeption und die Entwicklung des Systems. Im fünften Kapitel wird die Vorgehensweise der Evaluierung erklärt und die Ergebnisse werden vorgestellt. Das letzte Kapitel bietet eine Zusammenfassung und einen Ausblick für zukünftige Arbeiten.

2 Anforderungen

Im folgenden Abschnitt werden die Anforderungen an die mobile Anwendung näher beschrieben. Dabei wird zwischen funktionalen Anforderungen (Anforderungen zur Zweckerfüllung) und nicht funktionalen Anforderungen (Qualitätsmerkmale) unterschieden.

Die Bachelorarbeit wurde in Zusammenarbeit mit dem Berliner Projektbüro des Deutschen Forschungszentrums für Künstliche Intelligenz (DFKI) im Forschungsbereich Intelligente Benutzerschnittstellen durchgeführt. Die Arbeit fand im Rahmen des Projekts *Voice2Social* statt. Bei diesem Projekt werden Technologien zur multimodalen Interaktion und *Social-Software* kombiniert. Es sollen neuartige und vielfältig angereicherte Möglichkeiten der Interaktion für Nutzer sozialer Medien erforscht werden.[1] Die Integration des in dieser Arbeit zu entwickelnden Systems in das *Voice2Social*-Gesamtsystem soll ermöglicht werden. Deshalb wird versucht, ähnliche Technologien und Ansätze zu verwenden, falls diese für die Anwendung geeignet sind. Dies umfasst zum Beispiel den später vorgestellten Grammatik-Parser, welcher auch in ähnlicher Form in *Voice2Social* eingesetzt wird.

2.1 Funktionale Anforderungen

Die deutschsprachige Anwendung muss in der Lage sein, bei Eingabe einer Aktivität und einer Region passende Empfehlungen innerhalb des festgelegten Gebiets anzuzeigen. Bei den Aktivitäten besteht die Wahl zwischen fünf Kategorien: Essen, Trinken, Einkaufen, Unterhaltung und Nachtleben. Die Suche beschränkt sich auf den Kernbereich der Stadt Berlin. Als Regionen können Namen von Stadtvierteln oder bekannten Orten[2] verwendet werden. Bei vorhandenem GPS-Empfänger sollte das System auch in der Lage sein, im Umkreis der aktuellen Position zu suchen.

Eine weitere Anforderung stellt die Multimodalität dar. Um die zweite Hypothese zu testen, muss die Anwendung die initiale Eingabe sowohl via Sprache als auch via Touch erlauben.

[1] `http://voice2social.dfki.de/`
[2] zum Beispiel *Hauptbahnhof* oder *Alexanderplatz*

Die Spracheingabe kann in ganzen Sätzen erfolgen. Äußerungen, wie zum Beispiel *„Ich möchte essen gehen in Berlin Kreuzberg“*, sollten verstanden werden.

Damit der Einfluss sozialer Netzwerk überprüft werden kann, müssen die Daten von einer solchen Quelle stammen. Falls vorhanden, sollten auch Informationen angezeigt werden – wie zum Beispiel die durchschnittlichen Bewertungen der POIs.

Neben diesen Kernanforderungen gibt es eine Liste weiterer Eigenschaften, welche die Anwendung möglichst erfüllen sollte:

Verschiedene Ansichten: Zusätzlich zur übersichtlichen Darstellung in einer Liste können die gefundenen Treffer auch auf einer Karte angezeigt werden. Der Benutzer ist somit in der Lage, die Umgebung zu betrachten und das Ergebnis besser einzuordnen. Eine Kartenansicht bietet außerdem die Möglichkeit einer geclusterten Darstellung der POIs an. Dies bedeutet, dass Treffer nach örtlicher Nähe in Gruppen eingeteilt werden. Es fällt eventuell einfacher, bestimmte zusammenhängende Gebiete ausfindig zu machen.[3] Welche Ansicht bevorzugt wird, lässt sich während der Evaluierung testen.

Sprachbefehle: Während der Anzeige der Ergebnisse soll es möglich sein, mit der Anwendung via Sprache zu interagieren. So sollte zum Beispiel zwischen den Ansichten hin- und hergewechselt, Suchanfragen geändert oder in einer anderen Region gesucht werden können.

Zusätzliche Informationen: Neben den durchschnittlichen Bewertungen werden zusätzliche Informationen angezeigt, wie zum Beispiel Beschreibungen oder Reviews. Außerdem wäre es wünschenswert, neben den Hauptkategorien auch noch einige Unterkategorien anzubieten.

2.2 Nicht funktionale Anforderungen

Die Anwendung ist für die mobile Benutzung konzipiert und sollte möglichst einfach zu bedienen sein. Es handelt sich um einen evolutionären Prototypen, das heißt, dass dieser zunächst nur Grundfunktionalitäten umsetzt. Ziel ist es, während der Evaluierung die Akzeptanz zu testen. Die App wird für Personen aller Altersgruppen entwickelt, setzt aber voraus, dass sie mit Smartphone-Anwendungen umgehen können. Zusätzlich ist in den meisten Fällen Wissen über Namen von Stadtvierteln oder anderen bekannten Orten erforderlich, um eine Suchanfrage zu starten.

[3] zum Beispiel Einkaufsmeilen

Die Anwendung wird für ein Android-Gerät programmiert, sollte aber leicht auf iOS/iPhone portiert werden können. Performance spielt bei der Entwicklung dieser App eine untergeordnete Rolle, im Fokus steht die Funktionalität.

3 Verwandte Arbeiten

In diesem Kapitel werden für die Entwicklung und Evaluierung relevante Arbeiten präsentiert. In der Literatur gibt es viele Beispiele für Vorschläge von bestimmten Architekturen, zum Beispiel die MATCH-Architektur [JBV+02] oder die Galaxy-II-Architektur [SHL+98]. Aufbauend darauf wurden verschiedene multimodale, mobile Systeme entwickelt. Im Folgenden werde ich das Forschungssystem *SmartWeb* [RBE+05] und *City Browser* [Gru09] vorstellen.

Auch zur Evaluation von multimodalen Interaktionssystemen gibt es verschiedene Ansätze. Es besteht zum Beispiel die Möglichkeit, Tests als Feldexperiment, im Labor oder durch Einsatz bestimmter Dienste – beispielsweise mit Hilfe von *Amazon Mechanical Turk* (AMT)[1] – durchzuführen. In Abschnitt 3.3 stelle ich zwei frühere Arbeiten zu Nutzerstudien an multimodalen Systemen vor, auf welche diese Bachelorarbeit aufbaut.

3.1 SmartWeb

Das Projekt *SmartWeb* wurde von 2004 bis 2007 vom DFKI mit Partnern aus der Industrie und der akademischen Welt durchgeführt. Ziel war es, eine kontextbezogene, mobile und multimodale Benutzerschnittstelle mit Zugriff auf das sogenannte *Semantische Web* [WLH02] zu entwickeln. Das Semantische Web ist ein Konzept zur Weiterentwicklung des Internets mit dem Ziel, Informationen für Maschinen zugänglich zu machen. Im Hauptszenario interagiert der Benutzer mit seinem mobilen Gerät und stellt *Open-Domain*-Fragen[2] im Kontext der Fußballweltmeisterschaft 2006 [RBE+05]. Der Benutzer kann einfache Fakten abfragen, Befehle zur Suche, Erkundung und Inspektion von Informationen geben, sich Statusinformationen anzeigen lassen sowie die laufende Anfrage abbrechen. Im Falle einer Anfrage eines speziellen Dienstes leitet die Anwendung sie an den externen Service weiter.

Zur Eingabe stehen verschiedene Modalitäten zur Verfügung, um eine möglichst natürliche Kommunikation zu gewährleisten: Audio (Sprachein/-ausgabe und sogenannte Earcons[3]),

[1] in Deutschland nicht möglich, `https://www.mturk.com/mturk/welcome`
[2] unter anderem über Spieler, Ereignisse und Wetter
[3] akustische Signale

Grafik und Haptik (Toucheingabe, Tastatur und 3D-Gestik). Die Ausgabe wird in verschiedenen Formaten angeboten, wie zum Beispiel Text, Bild, Audio, Video, Grafik, Internetseite oder -link.

Bei der Gestaltung der Benutzeroberfläche von SmartWeb wurde auf die Einhaltung genereller Benutzerschnittstellenprinzipien geachtet: Es werden Rückmeldungen und Korrekturmöglichkeiten von Benutzereingaben angeboten und Statusinformationen anzeigt.

Die Architektur, wie in Abbildung 3.1 dargestellt, besteht aus drei verarbeitenden Blöcken. Der *PDA-Client* übernimmt die Ein- und Ausgabe unter Verwendung von Java und Actionscript/Flash. Der *Dialog-Server* ist für die Dialogsteuerung verantwortlich, das heißt, dass er unter anderem für die Spracherkennung zuständig ist und Anfragen unter Benutzung des Dialogsystems beantwortet. Dieses unterteilt sich in mehrere Unterkomponenten, wie zum Beispiel SPIN (Sprachinterpretierungsmodul), GEN (Sprachgenerierungsmodul), REAPR (Systemreaktions- und Präsentationsmodul). Die *semantischen Dienste* beantworten entsprechende Anfragen des Dialogsystems, wobei Web-Services zum Einsatz kommen.

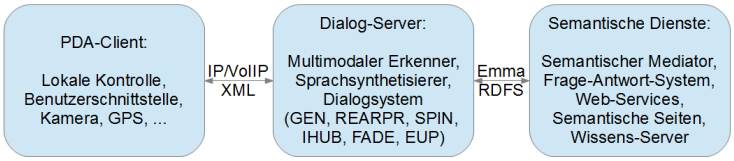

Abbildung 3.1: Generelle Architektur von *SmartWeb* [RBE+05]

Die Kommunikation zwischen den verschiedenen Komponenten innerhalb des Dialog-Servers geschieht mittels SWEMMA[4], einer Erweiterung des EMMA-Standards[5]. Zwischen Client und Dialog-Server wird ein verkürztes XML-Format (beziehungsweise VoIP für Sprachübertragung) verwendet. Der Dialog-Server und die semantischen Dienste verwenden Schnittstellen, die auf RDFS[6] beziehungsweise SWEMMA basieren.

Die vorgestellte Arbeit befasst sich mit der Entwicklung eines multimodalen, mobilen Dialogsystems. Es wird gezeigt, wie ein solches System aufgebaut werden kann. Im Gegensatz zu *SmartWeb* beschränkt sich das im Zuge der Bachelorarbeit zu entwickelnde System auf ein stark eingegrenztes Anwendungsszenario. Teile des Dialogsystems, wie zum Beispiel die Kombination der beiden Modalitäten und das Sprachverstehen, werden anders als bei *SmartWeb* in den Clienten integriert. Beide Systeme greifen auf Webservices zur Informationsbeschaffung zu. Bei der in dieser Bachelorarbeit beschriebenen Anwendung wer-

[4]SmartWeb-EMMA
[5]Extensible MultiModal Annotation markup language, `http://www.w3.org/TR/EMMAreqs/`
[6]Resource Description Framework Schema, `http://www.w3.org/TR/rdf-schema/`

den leichtgewichtige Datenaustauschformate wie JSON[7] (RESTful-Webservices) gegenüber XML (SOAP-Webservices) bevorzugt.

3.2 City Browser

City Browser [Gru09] ist eine englischsprachige, internetbasierte Plattform zum multimodalen Zugriff auf urbane Informationen in den USA. Das Projekt wurde mit Beteiligung des *M.I.T. Computer Science and Artificial Intelligence Laboratory* entwickelt. Es beantwortet neben Suchanfragen nach relevanten Orten (Restaurants, Hotels) in einer bestimmten Region auch Fragen nach Öffnungszeiten und ist auch in der Lage, Richtungsanweisungen zur Navigation zu geben beziehungsweise das Resultat einer Suche auf bestimmte Ergebnisse einzuschränken (Preiskategorie, Bewertung, Art der Küche). In Tabelle 3.1 wird eine beispielhafte Interaktion zwischen dem Benutzer (B) und dem System (S) veranschaulicht (übersetzt aus dem Englischen).

B:	Zeige mir griechische Restaurants in Boston.
S:	Es gibt sechs griechische Restaurants in Boston. [*dargestellt auf Karte*]
B:	Was sind die Öffnungszeiten von *Steves*?
S:	Hier sind die Öffnungszeiten für Steves Restaurant: [*Öffnungszeiten*]
B:	Gibt es irgendwelche italienischen Restaurants in dieser Straße? [*zieht Linie*]
S:	Es gibt 12 italienische Restaurants in dieser Straße. [*dargestellt auf Karte*]
B:	Zeige mir die Internetseite für dieses hier. [*kreist auf Karte ein*]
S:	OK. [*zeigt Internetseite an*]

Tabelle 3.1: Beispiel-Interaktion mit dem *City Browser* [GSW06]

Die Benutzeroberfläche, bestehend aus einer dynamischen Internetseite, ermöglicht die Eingabe via Tastatur, Sprache und Maus[8]. Der überwiegende Teil der Ansicht besteht aus einer Kartendarstellung, wobei die Google Maps API[9] zum Einsatz kommt. Ergebnisse werden auch in einer Listen- beziehungsweise Baumstruktur angezeigt. Zur Realisierung der Spracheingabe wird das WAMI-Toolkit[10] verwendet. Der Client verwendet AJAX-Techniken[11], um das auf dem Web-Server laufende Servlet anzusprechen, welches die Kommunikation mit den Dialogsystemkomponenten steuert. Die Dialogkomponenten (Sprachverarbeitung und -verstehen, Dialogmanagement, Sprachsynthese) werden mittels einer Galaxy-Architektur [SHL+98] (Hub-Architektur) untereinander verbunden. Die Spracherkennung

[7] JavaScript Object Notation, ein kompaktes Datenaustauschformat
[8] zum Beispiel durch Einkreisen bestimmter Regionen
[9] `https://developers.google.com/maps/`
[10] *Web-Accessible Multimodal Applications*, `http://wami.csail.mit.edu/`
[11] Asynchronous Javascript and XML

verwendet das SUMMIT-System [SWHC04], welches dynamische Sprachmodell-Manipulation erlaubt.[12] Informationen über POIs in Ballungsräumen (insbesondere Großstädten) werden automatisiert aus dem Internet zusammengetragen. Diese gesammelten Daten werden bereinigt, für die Verwendung vorbereitet und in eine Datenbank geschrieben. [GSW06]

Im Zuge weiterer Arbeiten wurde der *City Browser* um eine Korrektur der Spracherkennung durch Anzeigen der *N-best Liste* erweitert [GS07]. Eine Masterarbeit beschäftigt sich mit der Optimierung des Desktop-Systems *City Browser* für mobile Endgeräte [Liu10]. Die mobile Anwendung besitzt neben einer Kartenansicht auch eine Listen- und Hilfs-Ansicht. Im Gegensatz zur Browser-Version besitzt die App allerdings keine Korrekturmöglichkeiten bei falscher Erkennung der Spracheingabe [Liu10, Seite 46].

Für die Evaluierung unterschiedlicher Sprachmodelle wurde die Plattform *Amazon Mechanical Turk* eingesetzt, die es erlaubt, eine große Anzahl an Testpersonen mit geringem finanziellen Aufwand zu rekrutieren.[13] In einem Experiment wurden Interaktionsdaten mit einer Live-Version der mobilen Anwendung erhoben und ausgewertet. In einem zweiten Test mittels AMT wurden über 12.000 getippte Beispielanfragen gesammelt, um Trainingsdaten für das Sprachmodell zu erhalten. [Liu10, Seite 41]

Dieses sehr komplexe Projekt spiegelt in Ansätzen das wieder, was auch in dieser Arbeit entwickelt werden soll. Beides sind kartenbasierte Systeme zur Empfehlung von Lokationen. Neu an der vorliegenden Bachelorarbeit ist, dass direkt für ein mobiles System entwickelt wird und Teile der Dialogverarbeitung in den Clienten integriert werden. Dem Nutzer stehen verschiedene Kartenansichten zur Verfügung, beispielsweise eine Ansicht mit gruppierten POIs.

3.3 Evaluation multimodaler Systeme - Forschungsfeld Usability

3.3.1 Mobiles System zur Bildannotation

In einem Forschungsprojekt des spanischen Unternehmens *Telefónica* wurden Sprache und Touch als Eingabemodalitäten zum Annotieren von Bildern verglichen [CAOO09]. Es wurden folgende Hypothesen untersucht:

H1 Sprache wird gegenüber Text als Annotationsmechanismus auf mobilen Geräten bevorzugt (objektives Maß).

[12] Das bedeutet, dass zum Beispiel dem Nutzer ermöglicht wird, neue Landmarken einzuführen, welche in nachfolgenden Interaktionen referenziert werden können.

[13] Die Tester müssen nur auf die Plattform von AMT zugreifen, das heißt, dass eine persönliche Anwesenheit nicht notwendig ist.

H1-bis Sprachvermerke werden vom Benutzer bevorzugt, auch wenn für die Aufgabe mehr Zeit aufgebracht werden muss (subjektives Maß).

H2 Je länger der Vermerk, desto größer ist der Vorteil von Sprachannotationen gegenüber Textannotationen beim Taggen von Bildern auf mobilen Geräten.

H3 Das Abrufen von Bildern auf Handys mit Sprachanmerkungen ist nicht schneller als mit Text (objektives Maß).

Der Prototyp der Anwendung ermöglichte das Taggen der Fotos nach ihrer Aufnahme unter Verwendung von Sprach- beziehungsweise Textanmerkungen. Später wird dem Benutzer das Wiederauffinden der Fotos durch Spezifizierung der entsprechenden Annotation (entweder via Text oder Sprache) ermöglicht. Sprachanmerkungen wurden nicht in Text umgewandelt, sondern direkt gespeichert und beim Abrufen mit der Eingabe verglichen (aus Gründen der Performance und Genauigkeit).

Getestet wurde die Anwendung durch ein 31-tägiges Feldexperiment mit 20 Personen mit anschließendem Kontrollexperiment. Das Experiment umfasste vier Aufgaben zum Abrufen der Tags, wobei insbesondere die benötigte Zeit gemessen wurde. Die Teilnehmer der Studie wurden gleichmäßig auf drei verschiedene Gruppen aufgeteilt. Allen Probanden einer Gruppe wurde die zu benutzende Modalität vorgegeben (Touch, Sprache, Wahl zwischen Touch und Sprache). Ihnen wurde die Anwendung erklärt und sie sollten in dem einen Monat die Applikation in der Praxis testen.

Die Auswertung der Feldstudie ergab, dass Sprache als Annotationsmechanismus nicht bevorzugt wird. Die Hypothesen *H1* und *H1-bis* wurden dadurch widerlegt. Die zweite Aussage (H2), die besagt, dass Sprache vorteilhafter werde, je länger der Vermerk sei, wurde durch Analyse der Zeiten für korrekt befunden. Das Kontrollexperiment im Anschluss an die Feldstudie ergab, dass das Abrufen der Bilder (wie vermutet, siehe *H3*) nicht schneller vonstattengeht.

Die Experimentteilnehmer wiesen anhand von Bemerkungen darauf hin, dass sie Text gegenüber Sprache bevorzugten, weil sie es unangenehm fanden, in der Öffentlichkeit mit ihrem Mobiltelefon zu reden. Außerdem treten Ungenauigkeiten beim Abrufen der Bilder auf – wahrscheinlich wegen der im Freien unvermeidbaren Hintergrundgeräusche. Der Benutzer konnte nie überprüfen, ob die Sprachanmerkung richtig aufgenommen wurde.

Die Forscher kommen zu dem Schluss, dass gewisse Richtlinien beim Entwurf multimodaler, mobiler Bildannotierungsanwendungen eingehalten werden sollten, um den Nutzen zu maximieren. Zum Beispiel sollte dem Nutzer die Wahl zwischen verschiedenen Modalitäten gelassen werden. Ein Zusammenspiel der Modalitäten wäre wünschenswert. Es wird von den

Entwicklern vermutet, dass die Verwendung einer Speech-to-Text-Technik zu schlechteren Ergebnissen geführt hätte als die eingesetzte Technik.

Die vorgestellte Arbeit [CAOO09] beschreibt, wie ein multimodales, mobiles System getestet wurde. Es ähnelt in groben Zügen dem Aufbau dieser Bachelorarbeit und befasst sich mit einer ähnlichen Thematik. In dem vorgestellten Projekt wird unter anderem getestet, ob Sprache gegenüber Text bevorzugt wird. Allerdings geschieht dies innerhalb eines anderen Anwendungskontextes und lässt sich nicht zwingend auf die Untersuchungen in dieser Arbeit übertragen.

3.3.2 Multimodales Raummanagement- und Informationssystem

Neben den drei ausführlich vorgestellten Arbeiten gibt es natürlich eine Vielzahl anderer interessanter Arbeiten. Weiss et al. [WMWK11] untersuchten den Einfluss verschiedener Modalitäten auf die Gesamtqualität der multimodalen Interaktion. Sie stellten dabei fest, dass der Einfluss jeder Modalität auf die Gesamtqualität stark von dem Szenario und der Stärke der Interaktivität abhängt. Es gelang mit den präsentierten Modellen, die Gesamtqualität des Systems auf Basis der Bewertungen der einzelnen Modalitäten annäherungsweise hervorzusagen. [WMWK11] Eines der betrachteten Experimente befasste sich mit der Evaluierung eines multimodalen Raummanagement- und Informationssystems, wobei die Eingabe via Sprache, Touch oder einer Kombination der beiden Modalitäten erfolgt. Getestet wurde dies mittels 36 deutschsprachiger Personen, welche sechs Aufgaben durchführen mussten.[14] Das Experiment wurde in drei Blöcke aufgeteilt: Aufgaben wurden erst mittels einer vorgegebenen Modalität durchgeführt, dann mit der jeweiligen anderen Modalität und als drittes durften sich die Probanden die zu benutzende Modalität auswählen. Zwischen jedem Block wurden die Experimententeilnehmer aufgefordert, das System zu bewerten[15]. Die Reihenfolge des ersten und zweiten Blockes wurde jeweils variiert. [WES+09]

Die Evaluierung des hier zu entwickelnden, mobilen, multimodalen Empfehlungssystems folgt in groben Zügen dem vorgestellten Versuchsaufbau. Nähere Informationen zu den Unterschieden und der genauen Durchführung werden in Kapitel 5 gegeben.

[14] Navigation, Suchen, Anzeigen/Buchen von Räumen, Anzeigen von Veranstaltungen und Suchen von Mitarbeitern

[15] dabei kam der AttrakDiff-Fragebogen zum Einsatz

3.4 Zusammenfassung

Es wurden verschiedene, umfangreiche multimodale Systeme vorgestellt. Im Gegensatz zu diesen Systemen beschränkt sich das im Rahmen dieser Arbeit entwickelte, deutschsprachige System auf einen wesentlich kleineren Dialogumfang. Es ist leichtgewichtiger und basiert auf Webtechnologien (ähnlich wie der *City Browser*) mit dem Unterschied, dass Teile des Dialogs innerhalb des Clienten ausgewertet werden. Die in dieser Bachelorarbeit zu untersuchenden Forschungsfragen (siehe Abschnitt 1.2) wurden im Kontext der vorgestellten Projekte in dieser Art nicht untersucht. Bei dem *City Browser* gab es nach bestem Wissen keine Untersuchungen zum Vergleich von Sprache mit Text-/Toucheingaben, wie sie in der vorliegenden Arbeit erfolgten. Auch der Einfluss von sozialen Netzwerken bei der POI-Auswahl wurde nicht betrachtet.

In der Literatur gibt es neben dem *City Browser* auch noch andere Beispiele für kartenbasierte Empfehlungssysteme, wie zum Beispiel *SpeakIt* [JE10] oder MATCH (*Multimodal Access To City Help*) [JBV+02]. MATCH stellt ein multimodales, englischsprachiges, kartenbasiertes Lokations-Empfehlungssystem für PDAs in New York dar. Besonders hervorgehoben wird die Eingabemöglichkeit via Stift und Sprache. Ähnlich wie beim *City Browser* können zur Eingrenzung der Treffermenge bestimmte Regionen „eingekreist“ werden. Ein Experiment im Labor mit fünf Personen ergab, dass trotz einer unzuverlässigen Spracherkennung die Lösung der Aufgaben zuverlässig gelingt (85%) und die Spracheingabe gegenüber der handschriftlichen Eingabe bevorzugt wird (siehe Tabelle 3.2).

	Anzahl	in Prozent
Nur Sprache	171	51%
Nur Stift	93	28%
Multimodal	66	19%
GUI-Aktionen	8	2%

Tabelle 3.2: MATCH: Übersicht über die Verteilung der Eingabemodalitäten [JBV+02]

Das Beispiel der Evaluation eines Bildannotierungssystems untersucht eine ähnliche Forschungsfrage. Es wird überprüft, ob Sprache der Modalität Text vorgezogen wird. Dies geschieht aber in einem komplett anderen Anwendungsbereich unter Verwendung anderer Technologien.[16] Die beschriebene Vorgehensweise bei der Evaluierung des multimodalen Raummanagement- und Informationssystems bietet eine Grundlage für die Untersuchung der Usability und die Beantwortung der zu untersuchenden Hypothesen.

[16] Es wird kein Speech-to-Text verwendet.

4 Entwicklung einer multimodalen Anwendung zur Empfehlung von Lokationen

In diesem Kapitel wird die Konzeption und der Aufbau der entwickelten Anwendung besprochen. Zunächst werden verschiedene Möglichkeiten zum Aufbau eines solchen Systems dargelegt und Parallelen zu den verwandten Arbeiten gezogen. Es folgt eine ausführliche Beschreibung des entwickelten Systems.

4.1 Konzeption

Ein Einblick in vergleichbare Arbeiten hat gezeigt, dass es vielseitige Möglichkeiten gibt, multimodale Systeme zu entwickeln. In den folgenden Abschnitten werden verschiedene Varianten zur Entwicklung der in Kapitel 1 und 2 beschriebenen Anwendung dargelegt.

Architektur

Es gibt unterschiedliche Herangehensweisen, eine mobile Anwendung zu entwickeln. Eine Möglichkeit wäre, die komplette Anwendung auf dem Smartphone laufen zu lassen. Ein solcher Ansatz wird beim im Abschnitt 3.3 auf Seite 10 besprochenen mobilen System [CAOO09] verwendet. Es arbeitet unabhängig von externen Diensten und kann somit auch ohne Internetverbindung verwendet werden. In unserem Fall müssten sich alle Daten auf dem Smartphone befinden. Außerdem benötigt beispielsweise die Spracherkennung und das Clustern extrem viel Rechenleistung. Dieser Ansatz ist demnach ungeeignet für die Entwicklung der zu implementierenden Anwendung.

Bei der Desktop-Variante des *City Browsers* [Gru09] wird eine Client-Server-Architektur eingesetzt. Eine Internetseite repräsentiert die Benutzeroberfläche. Sprache, getippte Eingaben und Gesten werden an den Server geschickt, wo die Dialogverarbeitung stattfindet.

Es gibt auch die Möglichkeit, beide Ansätze zu kombinieren. Insbesondere bei kleinerem Dialogumfang bietet es sich an, Teile des Dialogs[1] innerhalb des Clienten zu verarbeiten. Es wird mehr Rechenleistung benötigt, dafür wird die Kommunikation zwischen Client und Server verringert.

Die Informationen zur Generierung der Empfehlungen stammen aus sozialen Netzwerken. Nutzer können Restaurants bewerten und Reviews schreiben, die wiederum entweder direkt über die jeweilige Seite oder über eine Schnittstelle abgerufen werden können. Zwei Vorgehensweisen stehen bereit: Die Daten der sozialen Netzwerke werden erst bei Anfrage des Nutzers durch Verwendung der jeweiligen Schnittstelle abgerufen. Die so gesammelten Daten sind aktuell, aber es besteht das Risiko, dass Probleme auftreten (die Seite ist nicht verfügbar, Schnittstellen haben sich geändert). Eine andere Variante ist, Informationen zentral in einer Datenbank zu halten. Dies beschleunigt den Datenabruf und hat den Vorteil, dass Auskünfte mehrerer Dienste zentral zur Verfügung stehen. Beim *City Browser* wird die zweite Variante eingesetzt. [GSW06]

Programmierung

Für die Realisierung der Anwendung stehen zwei Optionen zur Auswahl. Zum einen besteht die Möglichkeit, die Anwendung für Android mittels des *Android Software Development Kits*[2] komplett in Java umzusetzen. Dies würde eine spätere Portierung auf andere mobile Plattformen erheblich erschweren.

Eine Alternative stellt die Benutzung eines Frameworks dar, welches es erlaubt, Anwendungen mittels JavaScript, HTML und CSS zu realisieren. *Adobe PhoneGap*[3] unterstützt sieben verschiedene Plattformen. Es entstehen hybride Applikationen, da es sich um internetbasierte Ansichten handelt, die Anwendungen aber – anders als normale Internetseiten – als Apps verteilt werden und auf native APIs zugreifen können.

4.2 Architektur

Das Empfehlungssystem wurde mittels einer Client-Server-Architektur realisiert. Der Server ist verantwortlich für die Generierung der Empfehlungen, wohingegen der Client die

[1] Sprachverstehen/Parser, Kombination der Modalitäten
[2] `http://developer.android.com/sdk/index.html`
[3] `http://phonegap.com/`

Eingabe der benötigten Parameter und die Darstellung der Ergebnisse übernimmt. Das System lässt sich in unterschiedliche Module gliedern: den Spracherkenner, den Parser, die Georeferenzierungskomponente, den Lokationsfinder, die Clustering-Komponente und das Präsentationsmodul. Ein Schema der Architektur ist grafisch in Abbildung 4.1 dargestellt. In den folgenden Abschnitten werden die einzelnen Teile gesondert beschrieben.

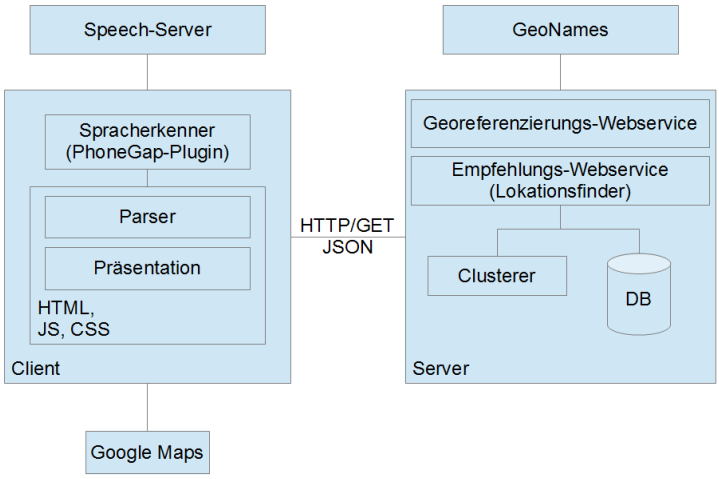

Abbildung 4.1: Komponenten der Anwendung

Ablauf

In Abbildung 4.2 wird gezeigt, wie die initiale Anfrage via Spracheingabe im Groben abläuft. Nachdem der Benutzer die Spracheingabe abgeschlossen hat, liefert der Spracherkenner die erkannte Transkription zurück. Der auf einer Grammatik basierende Parser erkennt die zu unternehmende Aktion und liefert die Aktivität und den Regionsnamen. Falls der Text nicht geparst werden kann, wird eine Fehlermeldung ausgegeben.

Es gibt zwei verschiedene Arten von initialen Suchanfragen: Entweder wird der Ortsname angegeben oder es soll in der Umgebung gesucht werden. Eine Suche in der Nähe des aktuellen Aufenthaltsortes ist nur bei funktionierender GPS-Lokalisierung möglich. Dafür muss die aktuelle GPS-Position ermittelt werden. Anderenfalls muss die Position der genannten Region durch Aufruf des Georeferenzierungs-Webservice herausgefunden werden. In beiden Fällen wird der Lokationsfinder (Empfehlungs-Webservice) mit dem Breiten-/Längengrad und der vom Benutzer eingegebenen Aktivität vom Clienten aufgerufen. Dieser greift direkt auf die Datenbank zu und ist für die Gruppierung der Ergebnisse zuständig. Als Resultat liefert der Webservice eine Liste von Clustern. Der Client bereitet die Daten auf[4] und

[4]Zum Beispiel werden alle POIs in eine Liste aggregiert und für die Listenausgabe sortiert.

übernimmt das Anzeigen der Informationen auf der Karte beziehungsweise in der Liste.

Die Eingabe via Touch funktioniert bis auf den Wegfall der Sprachverarbeitung identisch.

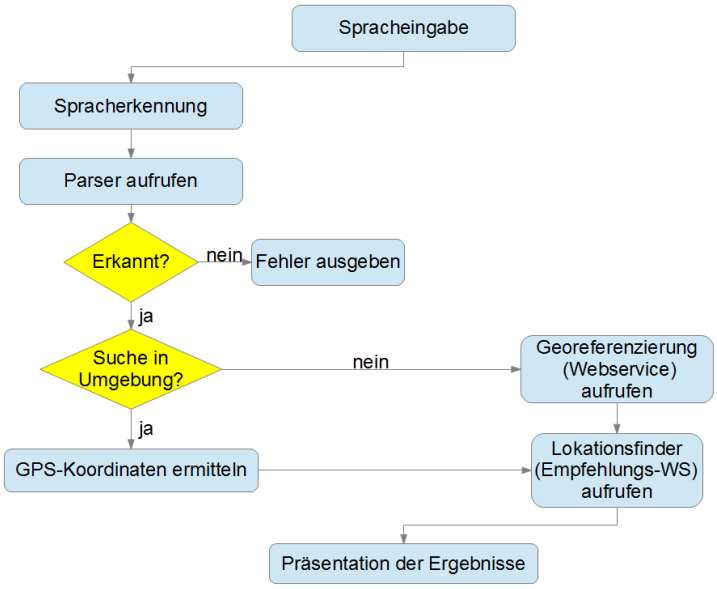

Abbildung 4.2: Ablaufskizze der initialen Suchanfrage bei Spracheingabe

Server

Der Server stellt zwei in Java programmierte RESTful-Webservices zur Verfügung. Ein Webservice ist für die Georeferenzierung[5] zuständig, der andere Webservice übernimmt die Generierung der Empfehlungen und das Clustern der gefundenen POIs. Die Webservices laufen in einem Apache Tomcat-Server[6] und benötigen Zugriff zum Internet für die Georeferenzierung beziehungsweise Zugriff auf die MySQL-Datenbank mit den Informationen zu den POIs. Sie werden beide über HTTP-GET-Anfragen auf den entsprechenden Ressourcennamen aufgerufen und liefern Ergebnisse im JSON-Format zurück. In Tabelle 4.1 auf der nächsten Seite werden beide Webservices dargestellt. Eine detailliertere Beschreibung befindet sich in den Abschnitten 4.5, 4.6 beziehungsweise 4.7.

[5] Umwandlung von Ortsnamen zu den entsprechenden geografischen Koordinaten
[6] `http://tomcat.apache.org/`

	Georeferenzierung-Webservice	Empfehlungs-Webservice
Ressource	parseLocation/	getClusteredRecommendation/
Query-Parameter	q - Name der Lokation *Parameter für die Begrenzung der Suche:* latLow - kleinster Breitengrad latHigh - größter Breitengrad lngLow - kleinster Längengrad lngHigh - größter Längengrad	category - Name der Aktivität lat - Breitengrad des Mittelpunktes lon - Längengrad des Mittelpunktes radius - Radius der Suchregion
Rückgabe-wert	Liste von möglichen Orten jeweils mit *name*, *longitude* und *latitude* Attribut	Liste von Clustern (Clusterattribute: *hull*, *info*, *member*, *avgRating*)
Rückgabe-format	JSON	JSON

Tabelle 4.1: Übersicht der Webservices

Client

Der Client wurde unter Verwendung des Frameworks *PhoneGap* umgesetzt. Dies vereinfacht eine Portierung auf andere Systemumgebungen, da für den Aufbau der Oberfläche Webtechnologien, wie zum Beispiel HTML, JavaScript und CSS, verwendet werden. Für die Entwicklung stand ein Samsung Galaxy Note zur Verfügung, auf welchem Android 2.2 läuft.[7]

Um die Entwicklung zu erleichtern, wurde auf die JavaScript-Bibliothek jQuery[8] zurückgegriffen. JQuery UI[9] und jQuery Mobile[10] wurden eingesetzt, um die Oberfläche interaktiv zu gestalten. Die Kartendarstellung verwendet die API von Google Maps[11]. Der Client ist für die Steuerung der Eingabe zuständig, kommuniziert via AJAX[12] mit den Webservices, bereitet die Ergebnisse auf und stellt sie anschließend dar.

4.3 Spracherkenner

Die Eingabe via Sprache wird durch die Verwendung des Spracherkenners von *Nuance* ermöglicht. *Nuance* ist einer der führenden Anbieter von Sprachlösungen und wurde beispielsweise bei Apples *Siri* eingesetzt. Angebunden wird der Spracherkenner mittels des *Dragon Mobile*

[7] Ein zweites Testgerät (Samsung Galaxy Note) verwendet Android 4.0.4 als Betriebssystem.
[8] Version 1.7.1, `http://jquery.com/`
[9] Version 1.8.23, `http://jqueryui.com/`
[10] Version 1.1.1, `http://jquerymobile.com/`
[11] Version 3, `https://developers.google.com/maps/documentation/javascript/`
[12] Asynchronous JavaScript and XML

SDKs[13]. Das SDK übernimmt die Aufnahme des Gesprochenen bis zur Rückgabe einer N-best-Liste von Treffern[14]. Ein vom DFKI bereitgestelltes PhoneGap-Plugin ermöglicht den Aufruf des Spracherkenners in JavaScript. Aktuell wird nur das beste Ergebnis zurückgeliefert. In Abbildung 4.3 wird die Architektur des Speech-Kits dargestellt. Die Erkennung ist gekapselt und läuft innerhalb des SDKs in mehreren Schritten ab:

1. Das PhoneGap-Plugin wird aufgerufen und eine Instanz des Speech-Kits wird initialisiert.
2. Ein Ton[15] zeigt den Beginn der Aufnahme an. Das Ende der Aufnahme wird durch das Auftreten einer längeren Pause im Sprachfluss automatisch erkannt.
3. Die Aufnahme wird kodiert an einen Server gesendet, dort in Text umgewandelt und zurück an die Anwendung geleitet. Teilweise stehen mehrere mögliche Texte zur Auswahl, wobei jeweils ein Wert die Konfidenz des Textes anzeigt.
4. Das beste Ergebnis wird der App als einfacher String zurückgegeben.

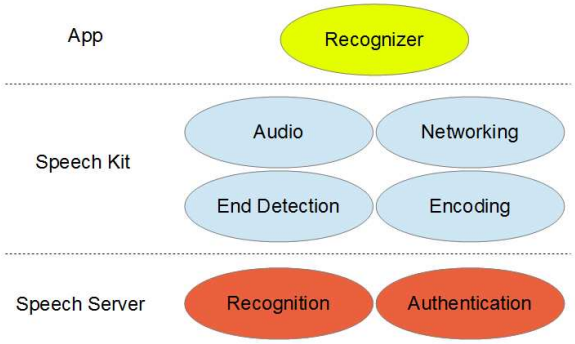

Abbildung 4.3: Architektur des Speech-Kits [Nua10]

4.4 Parser

Ein Parser wird benötigt, um dem aus der Spracherkennung gewonnenen Text eine Bedeutung abzugewinnen. Neben der initialen Eingabe wird Sprache auch zur Steuerung der Benutzeroberfläche eingesetzt.

Zur Umsetzung des Parsers wird der JS/CC-Grammatikparser[16] verwendet. Dem Parser wird eine kontextfreie Grammatik übergeben. Die gesamte Grammatik wird als ein einziges

[13] `http://dragonmobile.nuancemobiledeveloper.com/public/index.php?task=home`
[14] Konfidenzwert und das Gesprochene als Text
[15] Dieser wird innerhalb des PhoneGap-Plugins, beim Aufrufen des Speech-Kits festgelegt.
[16] `http://jscc.jmksf.com/`

JavaScript-Objekt definiert. Diese Form der Grammatik wird mittels einer im DFKI entwickelten Komponente in eine für den JS/CC-Parser verständliche Form konvertiert. Intern wird vom JS/CC aus dieser Grammatik JavaScript-Code zur Auswertung des Ausdrucks generiert. Die Grammatik wird, da sie in der Anwendung statisch ist, nur einmal verwendet, um den JavaScript-Code zu erzeugen. Die Auswertung des Textes geschieht komplett in JavaScript durch Ausführung des generierten Codes.

Durch die Konvertierung der Grammatik und die Generierung des JavaScript-Codes beim Starten verzögert sich das Laden. Bei dem Entwicklungsgerät (Samsung Galaxy Note, Android 4.0.4) dauert dieser Prozess circa 20 Sekunden. Es ist vorstellbar, die Grammatik in konvertierter Form zu laden beziehungsweise den generierten Code zu speichern, um den Start zu beschleunigen.

Suche von Lokationen in einer bestimmten Region

Ziel des Parsers ist es, die durchzuführende Aktivität und die Region zu extrahieren.

Die von mir definierte Grammatik beschränkt sich nicht auf eine Formulierung, sondern ermöglicht das Verwenden unterschiedlicher Sätze je Aufgabe. Sie ist nicht sehr restriktiv, das heißt, es werden auch Wendungen zugelassen, die keine korrekten Sätze in der deutschen Sprache darstellen. Vier grobe Gruppen von Phrasen werden erkannt:

- *„Kategorie* in *Lokation“*
- „Wo kann [ich] *Aktivität* in *Lokation“* beziehungsweise „Wo kann [ich] in *Lokation Aktivität“*
- „Ich möchte *Aktivität* in *Lokation*“ beziehungsweise „Ich möchte in *Lokation Aktivität*“
- „Zeige *Aktivität* in *Lokation*“ beziehungsweise „Zeige in *Lokation Aktivität*“

Aktivitäten und Lokationen wurden in der Grammatik des Prototyps fest vorgegeben. Aktivitäten sind entweder Verben („essen“, „trinken“, „einkaufen“) oder Substantive („Restaurants“, „Bars“, „Shops“). Für Lokationen werden vordefinierte Orte in Berlin erkannt (insbesondere Stadtviertel), aber auch beliebige Straßennamen[17] und Plätze. Außerdem erkennt die Grammatik auch Phrasen, welche anzeigen, dass in der Nähe des Benutzers gesucht werden soll (zum Beispiel: „in der Umgebung“, „in der Nähe“, „hier“).

Um die Anzahl von Varianten zu erhöhen, werden sogenannte Stoppwörter definiert, die vor Benutzung des Parsers herausgefiltert werden. Die Stoppwortliste beinhaltet unter anderem Artikel, Personalpronomen, Possessivpronomen und Wörter wie „bitte“ und „mal“.

[17] auf „Straße“ endend

Synonyme für Wörter, welche Teil der Grammatik sind, werden ebenfalls eingesetzt, um die Auswahl an möglichen Sätzen zu vergrößern. Tabelle 4.2 zeigt Beispiele von Eingaben, die die Grammatik akzeptiert.

Beispielphrasen
„Zeige mir bitte mal Restaurants in Berlin Kreuzberg“
„Ich möchte am Hauptbahnhof essen gehen“
„Wo kann ich Kaffee trinken am Alexanderplatz“
„Suche mir bitte in der Warschauer Straße Pizzerien“

Tabelle 4.2: Durch die Grammatik erkannte Sätze

Sprachbefehle

In der Kartenansicht stehen Befehle zur Steuerung der Benutzeroberfläche zur Verfügung. Sie sind kurz gehalten und strikter in ihrer Anwendung als die initiale Suchanfrage. Es gibt Befehle zum Wechseln der Ansicht, zum Wechseln der Lokation, zum Anzeigen von Informationen und zum Ändern der Aktivität. Diese sind in Tabelle 4.3 aufgeführt.

Befehl in Kartenansicht	Beispielphrasen
Ansicht wechseln	„Zeige Listenansicht“, „Liste anzeigen“, „Zeige Cluster“, „Cluster ausschalten“, „Zeige Empfehlungen“
Lokation wechseln	„Suche hier“, „Suche in der Umgebung“
POI-Liste eines Clusters anzeigen	„Zeige Cluster fünf“, „Gruppe 13“
Aktivität wechseln	„Suche Cafés“, „Zeige Restaurants“

Tabelle 4.3: Befehlsliste

Der Parser liefert ein Objekt mit der Befehlsart und zusätzlichen Informationen zurück (je nach Befehl, zum Beispiel Name der Aktivität und Lokation bei einer Suchanfrage). Für jeden Befehl wird eine aufzurufende Funktion definiert. Falls der geparste Befehl in der aktuellen Ansicht zulässig ist, wird die passende Funktion ausgeführt.

In Abbildung 4.4 wird der Parserablauf schemenhaft dargestellt. Der von der Spracherkennung zurückgelieferte Text („Zeige Restaurants in Kreuzberg“) wird dem Parser mittels eines Funktionsaufrufes übergeben. Falls die Eingabe mit einem der in der Grammatik definierten Sätze („*phrases*“, siehe Abbildung 4.5) übereinstimmt, wird das unter

„semantic“ definierte Objekt zurückgegeben. Die darin enthaltenen Platzhalter (*„_$category[0]['semantic']“* und *„_$location[0]['semantic']“*) werden vorher durch die entsprechenden Textteile (*„restaurants“* und *„kreuzberg“*) ersetzt . Dieses Objekt wird durch eine Funktion ausgewertet und die richtige Aktion wird ausgeführt.

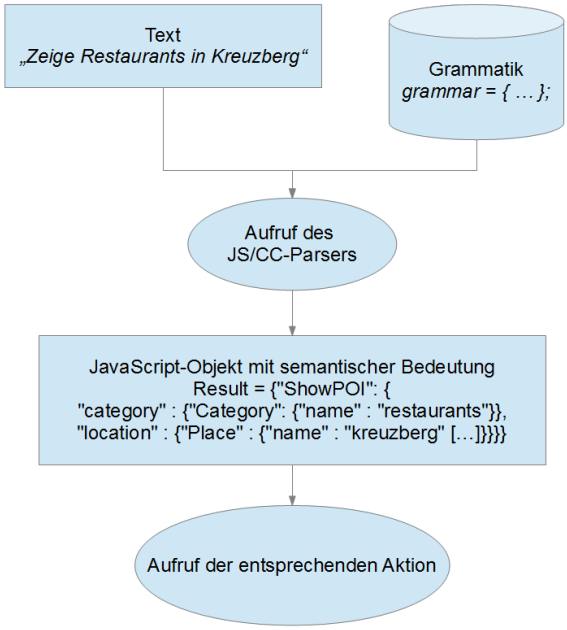

Abbildung 4.4: Parserablauf

```
"SHOW_POIS" : {
            "phrases" : [ "VERB CATEGORY PREPOSITION LOCATION" /*[...]*/ ],
            "semantic" : {
                "ShowPOI" : {
                    "category" : "_$category[0]['semantic']",
                    "location" : "_$location[0]['semantic']",
                }
            }
        }
```

Abbildung 4.5: Verkürzter Auszug aus der Grammatik

4.5 Georeferenzierung

In diesem Schritt wird eine Zuordnung des Lokationsnamen, der entweder aus dem gesprochenen Satz extrahiert oder via Touch eingegeben wurde, zu seinen geografischen Koordinaten (Breiten- und Längengrad) vorgenommen. Es gibt eine Vielzahl von Geocoding-Services,

wie zum Beispiel Googles Geocoding-API[18], OpenStreetMaps Nominatim[19] oder die Webservices von GeoNames[20].

Im Falle der hier vorgestellten Anwendung geschieht die Georeferenzierung durch den implementierten Georeferenzierungs-Webservice (siehe Tabelle 4.1), welcher auf den Service von GeoNames.org zurückgreift. Nach dem Starten der Suche wird zunächst eine Anfrage mit dem Namen des Ortes und einem groben Gebiet (Bounding-Box, die Berlin abdeckt) gestellt, um die genauen Koordinaten zu ermitteln. Es folgt eine auf Deutschland beschränkte Anfrage an den Geocoding-Service von GeoNames unter Verwendung der Java-Bibliothek[21]. Nach der Suche werden Ergebnisse außerhalb der angegebenen Bounding-Box entfernt und die verbleibenden Lokationen zurückgegeben.[22]

Falls vom Nutzer die nähere Umgebung als Lokation gewählt wurde, fragt die Anwendung die aktuellen GPS-Koordinaten ab und verwendet diese als Ausgangspunkt für die Suche.

4.6 Lokationsfinder

Informationen (Namen, Bewertungen, Reviews, Beschreibung) über POIs werden im Internet von vielen sozialen Netzwerken, wie zum Beispiel *Yelp* oder *Google Places*[23], angeboten und stehen durch Webservices zum Abruf bereit. Die Daten, die in der Anwendung eingesetzt werden, stammen aus *Qype*. Zur Bereitstellung der POI-Daten wurde der Empfehlungs-Webservice (siehe Tabelle 4.1) implementiert, welcher Anfragen mit der Aktivität, dem Längen- und Breitengrad des Ortes und dem Radius als Parameter bearbeitet. Aus dem Projekt *Voice2Social* liegt eine MySQL-Datenbank mit POIs der Region Berlin vor. Insgesamt befinden sich 144.675 Einträge in der Datenbank, es gibt 7.087 unterschiedliche Kategorien beziehungsweise Kategorie-Kombinationen.

Ein POI besitzt in der Datenbank folgende Attribute, wobei nur die ersten drei hier aufgeführten Eigenschaften vorhanden sein müssen:

- eine ID
- eine Geo-Position (Breiten-/Längengrad)
- ein Titel

[18] `https://developers.google.com/maps/documentation/geocoding/`
[19] `http://nominatim.openstreetmap.org/`
[20] `http://www.geonames.org/export/ws-overview.html`
[21] `http://www.geonames.org/source-code/`
[22] Im Clienten wird mit dem ersten (besten) Treffer weitergearbeitet.
[23] `http://www.google.com/places/`

- eine Beschreibung
- ein Link zu einem (Miniatur-)Bild
- eine oder mehrere Kategorien
- eine durchschnittliche Bewertung (1-5 Sterne)
- mehrere Reviews (Text und Zeitstempel)

In Tabelle 4.4 stellt eine Übersicht über die Aufteilung der Bewertungen dar. Ein Großteil der POIs (78,85%) sind unbewertet, ungefähr die Hälfte der bewerteten POIs (47,38%) besitzen die beste Bewertung.[24]

Durchschnittliche Bewertung	5	4	3	2	1	unbewertet
POI-Anzahl	14501	8803	4347	1369	1583	114072

Tabelle 4.4: Anzahl an bewerteten POIs in der Datenbank

Ablauf

Es muss eine Zuordnung des Begriffes („essen“, „Restaurant“) zu der beziehungsweise den passenden Kategorie(n) (beispielsweise „Food&Drink/“, „Restaurant/“, „Restaurants/“) vorgenommen werden. Tabelle 4.5 zeigt einige relevante Kategorien mit ihrer Häufigkeit. Durch Paare von regulären Ausdrücken werden einer Menge von Benutzereingaben (Aktivitäten) die jeweiligen Kategorien zugeordnet. Einige Paare befindet sich in Tabelle 4.6.

Anzahl POIs	Kategorie	Anzahl POIs	Kategorie
2724	Fashion	1299	Cafés
2532	Food&Drink	941	Eating&Drinking
2249	Restaurant	936	Italian&Pizza
1976	Shopping	838	Restaurants
1789	Hotels	763	Arts&Entertainment
1310	Bakeries	762	Perfume&Cosmetics

Tabelle 4.5: Häufigkeiten ausgewählter Kategorien in der Datenbank

Eingegebene Aktivität	Ausgabe der Datenbank-Kategorien
pizza(s)?.*|pizzeria|pizzerien	Pizza/|Italian&Pizza/
caf[ée](s)?.*|kaffee(s)?.*	Cafés/|Cafés&CoffeeShops/|CoffeeShops/|Coffee&TeaShops/

Tabelle 4.6: Paare regulärer Ausdrücke für die Zuordnung der Aktivitäten zu den DB-Kategorien

[24] Die schlechteste Bewertung ist 1, die beste Bewertung ist 5.

POIs, die der passenden Kategorie angehören und sich im Umkreis des gewählten Ortes befinden, werden aus der Datenbank abgerufen. Da sich die POIs in einer etwas größeren, ungefähr rechteckigen Region befinden, werden alle Orte, welche mehr als eine bestimmte Distanz[25] vom übergebenen Mittelpunkt entfernt sind, herausgefiltert.

4.7 Clusterer

Alle gefundenen POIs werden nach örtlicher Nähe gruppiert. Durch den Schritt in Richtung Schematisierung soll es den Nutzern ermöglicht werden, sich einen Überblick über Strukturen zu verschaffen. [SS11] Zum Einsatz kommt ein modifizierter K-Means-Algorithmus, welcher von Bertram Sändig im Rahmen seiner Bachelorarbeit für das DFKI entwickelt wurde. [Ber12] Um für die Qype-Datensätze anwendbar zu sein, musste die Implementierung geringfügig geändert werden.

Als Eingabe für das Clustern erwartet der Algorithmus neben den POIs eine Distanzschwelle und die angestrebte Größe eines Clusters. Orte, deren Entfernung zueinander 200m überschreitet, werden bei der Initialisierung nicht miteinander verbunden.[26] Die angestrebte Größe beträgt 30.000 m². Dies entspricht in etwa der Fläche eines Quadrats mit einer Seitenlänge von 173m.

4.8 Präsentation

Die Benutzeroberfläche setzt sich aus drei verschiedenen, bildschirmfüllenden Seiten zusammen: der Eingabemaske, der Kartenansicht und der Listenansicht. Diese werden nachfolgend detailliert beschrieben.

4.8.1 Eingabemaske

Beim Start der Anwendung wird zunächst die Eingabemaske (siehe Abbildung 4.6) gezeigt. Es steht je ein Eingabefeld für die Aktivität und die Region zur Verfügung. Eine Liste erleichtert die Auswahl einer Aktivität. Falls die Region nicht ausgefüllt wird, fragt das System den Benutzer, ob in der Umgebung gesucht werden soll. Eine Bejahung hat zur

[25] In der Anwendung beträgt sie für alle Anfragen 500m.

[26] Nach dem Erstellen des Graphen mit allen POIs werden alle Knoten paarweise miteinander verbunden, solange die Entfernung unter dieser Schwelle liegt. Im Anschluss werden die Flächen der Cluster für alle verbundenen Teilgraphen berechnet und zerteilt (Anwendung des K-Means-Algorithmus), um die angestrebte Größe ungefähr zu erreichen.

Folge, dass die aktuelle GPS-Position ermittelt und als Mittelpunkt für die Suchanfrage verwendet wird. Am unteren Rand befindet sich ein Button zum Starten der Spracheingabe. Ein Ton signalisiert den Anfang der Aufnahme.

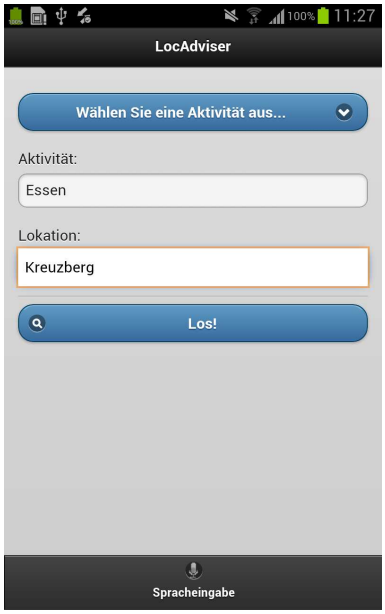

Abbildung 4.6: Eingabemaske

4.8.2 Kartenansicht

Ergebnisse werden nach erfolgreicher Suche in der Kartenansicht (siehe Abbildung 4.7) dargestellt. Die Karte bedeckt den gesamten Bildschirm bis auf ein Panel am unteren Bildschirmrand. Im Panel befindet sich ein Sprachbutton, der neben den Befehlen[27] auch das Starten einer neuen Anfrage ermöglicht. Die Betätigung der physischen Menütaste des Android-Gerätes lässt im Panel eine Auswahl an Buttons zur Steuerung der Anwendung erscheinen.

Das Optionsmenü besitzt in der oberen Reihe drei Buttons zum Wechseln der jeweiligen Ansicht. Die untere Reihe beherbergt (von links nach rechts) einen Button zum Anzeigen der Eingabemaske, der Listenansicht aller POIs in der Region und einen Button zur erneuten Suche (bei gleichbleibender Aktivität), wobei als Ort das Zentrum der Karte angenommen wird.

In jeder Form der Darstellung symbolisiert die weiße Flagge den Mittelpunkt des Suchkreises. Diese kann durch *Drag-and-Drop* an eine andere Position verschoben werden, um bei gleichbleibender Aktivität in der Umgebung der neuen Position zu suchen. POIs werden au-

[27] siehe Seite 21

ßer in der Empfehlungsansicht als kleine Kreise dargestellt, wobei die Farbe Auskunft über die durchschnittliche Bewertung gibt.[28] Durch Anklicken des jeweiligen POIs lässt sich ein Informationsfenster öffnen. Die dargestellten Informationen enthalten neben dem Titel, der durchschnittlichen Bewertung und der Anzahl der Reviews auch ein Miniaturbild und eine Beschreibung[29] der Lokalität.

Insgesamt gibt es drei verschiedene Kartenansichten:

Clusteransicht Cluster werden als graues Polygon darstellt. Zusätzlich werden Cluster mit einem nummerierten Marker versehen, welcher beim Anklicken eine Listenansicht aller POIs innerhalb dieses Clusters öffnet.

Normale Ansicht Cluster sind in dieser Ansicht nicht sichtbar. Nur POIs werden angezeigt.

Empfehlungsansicht Die fünf besten POIs werden durch Sterne markiert und alle anderen ausgeblendet. Sortiert wird nach durchschnittlicher Bewertung beziehungsweise bei gleicher Bewertung nach Reviewanzahl.

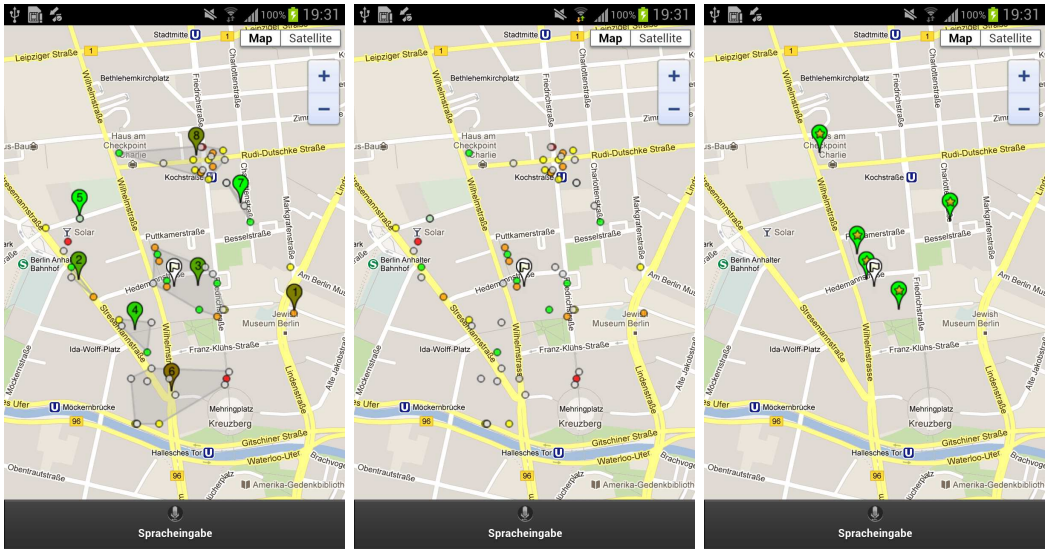

Abbildung 4.7: Clusteransicht, Normale Ansicht, Empfehlungsansicht (v. l. n. r.)

4.8.3 Listenansicht

Die Listenansicht zeigt alle POIs an, absteigend sortiert nach durchschnittlicher Bewertung (und Reviewanzahl als zweites Sortierkriterium bei gleicher Bewertung, siehe Abbildung 4.8). Angezeigt werden in dieser Ansicht lediglich der Name, die Bewertung und die Anzahl

[28] Grün steht für die beste Bewertung, rot für die schlechteste und grau für unbewertete Lokalitäten.
[29] Falls keine Beschreibung vorhanden ist, wird ein Review verwendet.

der Reviews aller POIs der gesamten Region beziehungsweise der POIs innerhalb des ausgewählten Clusters. Durch Klicken auf den entsprechenden Eintrag wird in die Kartenansicht gewechselt und das Informationsfenster geöffnet.

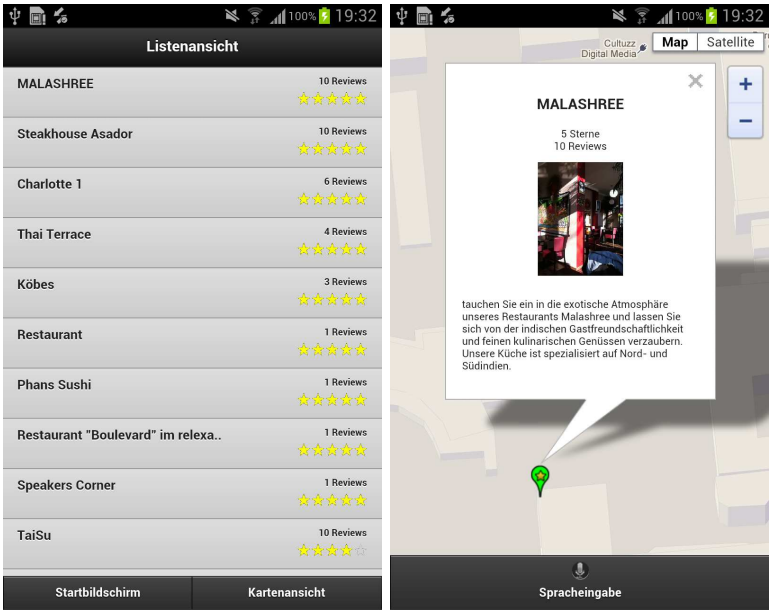

Abbildung 4.8: Listenansicht und Informationsfenster

5 Evaluation

Eine Benutzerstudie der multimodalen Anwendung wurde durchgeführt, um die in Abschnitt 1.2 auf Seite 2 vorgestellten Hypothesen zu testen und die Anwendung auf Usabilty zu überprüfen. Die erste Aussage (H1) ist, dass Daten aus sozialen Netzwerken dem Nutzer bei einer Auswahl einer passenden Lokalität unterstützen. Die zweite Hypothese sagt aus, dass Sprache die Bedienung einer Smartphone-Applikation zum Finden von Lokalitäten erleichtert.

Des Weiteren sollen Schwachstellen – zum Beispiel in der Grammatik – aufgedeckt werden. Der Test fand in einer kontrollierten Umgebung statt. Es war somit möglich, die Interaktionen zwischen dem Benutzer und der mobilen Anwendung mit einer Videokamera aufzuzeichnen.

5.1 Vorgehensweise

Für die Evaluation des in Kapitel 4 beschriebenen Prototyps wurden zwölf deutschsprachige Teilnehmer rekrutiert. Sie hatten keine Erfahrung mit dem System und ihnen wurde nur mitgeteilt, dass es sich um den Test einer Anwendung handele. Alle Teilnehmer benutzten das gleiche Gerät, ein Samsung Galaxy Note mit Android 4 als Betriebssystem. Die Durchführung des Tests beanspruchte zwischen 20 und 40 Minuten. Als Grundlage des Experiments dient die in Abschnitt 3.3.2 auf Seite 12 besprochene Arbeit zur Evaluierung multimodaler Systeme.

Vor dem Experiment füllte jeder Versuchsteilnehmer einen Fragebogen aus. Abgefragt wurden Informationen zur Person und ihre Einstellung zur Spracheingabe. Anschließend wurde jeder Testperson die Anwendung kurz vorgeführt. Um einer Beeinflussung entgegenzuwirken, wurde als Interaktionsbeispiel die Aktivität *Sport treiben* in *Berlin Mitte* gewählt. Alle Ansichten und Funktionalitäten wurden kurz vorgestellt. Die Existenz der Sprachbefehle wurde angemerkt, wobei der genaue Wortlaut jedes Befehls jedoch nicht offenbart wurde. Dieses Vorgehen wurde mit der Absicht gewählt, die Testperson zum Experimentieren anzuregen, um Schwächen in der Grammatik zu entdecken.

5.1.1 Aufgaben

Insgesamt wurden jedem Proband sechs Aufgaben gestellt. Für die ersten vier wurde die zu benutzende Modalität bei der initialen Eingabe vorgeschrieben. Bei den letzten beiden war die Wahl der Modalität freigestellt. Bei fünf der sechs Aufgaben wurde ein Paar aus durchzuführender Aktivität und Region vorgegeben. Die letzte Aufgabe wurde offener gestaltet, um zu schauen, wie die Versuchsteilnehmer reagieren und was sie von solch einem System erwarten. Eine Auflistung der genauen Aufgabeformulierungen steht in Tabelle 5.1.

Eine Aufgabe wurde als abgeschlossen gewertet, wenn ein POI ausgewählt wurde. Der Name der Lokalität musste notiert werden. Zusätzlich wurde nach jeder Teilaufgabe die Regionsübereinstimmung und die Relevanz der Empfehlung abgefragt.

	Aktivität	Ort
Aufgabe 1	Essen	Berlin Friedrichshain
Aufgabe 2	Café	Berlin Hauptbahnhof
Aufgabe 3	Einkaufen	Berlin Alexanderplatz
Aufgabe 4	Sushi	Berlin Mitte
Aufgabe 5	Pizzeria	Berlin Tempelhof
Aufgabe 6	Am Abend etwas unternehmen	<*offen*>

Tabelle 5.1: Aufgabenübersicht

Die zwölf Teilnehmer wurden in vier Gruppen à drei Teilnehmer eingeordnet. Die Reihenfolge der ersten vier Aufgaben sowie die zu verwendende Modalität wurden variiert – eine genaue Aufschlüsselung der Reihenfolge und Zuordnung der Modalität veranschaulicht Tabelle 5.2. Da den Versuchspersonen keine Zeit zur Eingewöhnung gegeben wurde, sollte das Mischen der Problemstellungen Auswirkungen eines Lerneffekts entgegenwirken.

	Gruppe A	Gruppe B	Gruppe C	Gruppe D
Aufgabe 1	1. - Touch	3. - Sprache	1. - Sprache	3. - Touch
Aufgabe 2	2. - Touch	4. - Sprache	2. - Sprache	4. - Touch
Aufgabe 3	3. - Sprache	1. - Touch	3. - Touch	1. - Sprache
Aufgabe 4	4. - Sprache	2. - Touch	4. - Touch	2. - Sprache

Tabelle 5.2: Übersicht über die Aufgabenverteilung der einzelnen Gruppen - in jeder Zelle wird die Reihenfolge der Aufgabe gefolgt von der Modalität angegeben

Während der Lösung der Aufgaben wurde der Bildschirm des Geräts inklusive Audio aufgezeichnet. Die Aufnahme erfolgte mit einer Digitalkamera, welche mittels eines auf dem Tisch stehenden Stativs auf das Gerät gerichtet wurde.

5.1.2 Fragebogen

Nach dem Lösen der sechs Problemstellungen wurden den Studienteilnehmern Fragen zur Anwendung gestellt. Neben einem eigenen Fragenkatalog wurde den Probanden auch der USE-Umfragebogen[1] zur Beantwortung vorgelegt.

Die Auswahl des Fragebogens wurde durch den Artikel *Evaluating Multimodal Systems*[2] beeinflusst. In [KWWM10] wurden verschiedene Fragebögen, wie zum Beispiel *AttrakDiff*, *SUS* oder *USE*, auf ihre Nützlichkeit zur Bewertung von Qualitätsaspekten multimodaler Systeme überprüft. Die Untersuchung zeigt unter anderem, dass der USE-Fragebogen für die Evaluierung eines multimodalen Systems geeignet ist und die Usability, die Effizienz, die Effektivität und die Intuitivität misst.

5.2 Auswertung

Bei den zwölf Teilnehmern (9 männlich, 3 weiblich) handelt es sich um Mitarbeiter des Deutschen Forschungszentrums für Künstliche Intelligenz. Der Median des Alters liegt bei 33,5 Jahren (Durchschnitt: 36,25; Minimum: 26; Maximum: 53). Ein Großteil von ihnen arbeitet als Forscher im Bereich Informatik beziehungsweise Linguistik. Sieben Personen (58,34%) besitzen ein Smartphone, wobei drei Personen als Betriebssystem iOS und zwei Android verwenden. Im Vorab-Fragebogen wurde nach der Erfahrung und Nützlichkeit von Spracheingaben gefragt. Ungefähr die Hälfte (7 Probanden) gab an, dass sie Erfahrung mit Sprache als Eingabemodalität hatten. Die Möglichkeit der Spracheingabe wurde von der Mehrzahl der Versuchsteilnehmer im Vorhinein für nützlich befunden: Auf einer Likert-Skala von eins (die Nützlichkeit der Sprache trifft zu) bis sieben (trifft nicht zu) beantwortete die Hälfte der Probanden die Frage mit eins, vier Personen beantworteten die Frage eher neutral (3) und nur eine Person stimmte der Aussage nicht zu (6). In Abbildung auf der nächsten Seite werden die Altersverteilung und der Smartphonebesitz grafisch veranschaulicht.

[1] siehe Abschnitt 5.2.2 auf Seite 38, `http://www.stcsig.org/usability/newsletter/0110_measuring_with_use.html`, abgefragt am 10.09.2012

[2] zu deutsch *Evaluierung multimodaler Systeme*

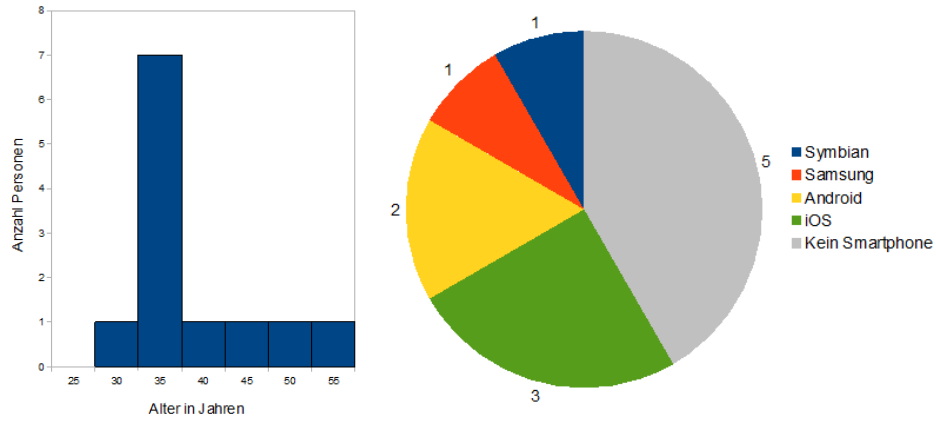

Abbildung 5.1: Statistik über Versuchsteilnehmer (links: Altersverteilung, rechts: Smartphonebesitz)

5.2.1 Gemessene Variablen

Die Videoaufzeichnung der Aufgabendurchführung ermöglichte die Messung der genauen Zeiten. Die Empfehlungen für interaktionsbeschreibende Parameter multimodaler Systeme der *International Telecommunication Union* (ITU) wurden berücksichtigt. [ITU11] Neben der Dialogdauer wurden auch die Dauer der Benutzereingabe[3] und die Systemantwortszeit gemessen. Ein Dialog umfasst in diesem Zusammenhang alle Interaktionen von der Eingabe bis zur Auswahl eines spezifischen POIs. Die Zeitmessung der Benutzereingabe startet mit dem Sprachbeginn beziehungsweise der Berührung des Touchdisplays und endet mit dem Abschicken der Anfrage beziehungsweise dem Ende des Sprechens. Bei der Auswertung wurde nur die initiale Eingabe betrachtet. Falls eine Eingabe nicht erfolgreich war, wurde die Zeitmessung nicht gestoppt, sondern lief weiter, bis sie korrekt erfolgte.

5.2.2 Ergebnisse

In der Abbildung 5.2 sind die durchschnittlichen Zeiten für die Aufgaben vermerkt. Auffallend ist, dass die Versuchsperson im Durchschnitt für die dritte Aufgabe (*Einkaufen am Berliner Alexanderplatz*) circa 50 Prozent mehr Zeit benötigten. Dies lässt sich durch die große Anzahl an Treffern (um die 200) erklären. Die letzte Aufgabe dauerte länger, insbesondere die Eingabe des Benutzers verdoppelte sich im Vergleich zur Durchschnittszeit. Durch die offene Gestaltung waren mehrere Anläufe zur Lösung des Problems erforderlich, weil die Anwendung einige Eingaben nicht unterstützte.

[3] User Turn Duration (UTD)

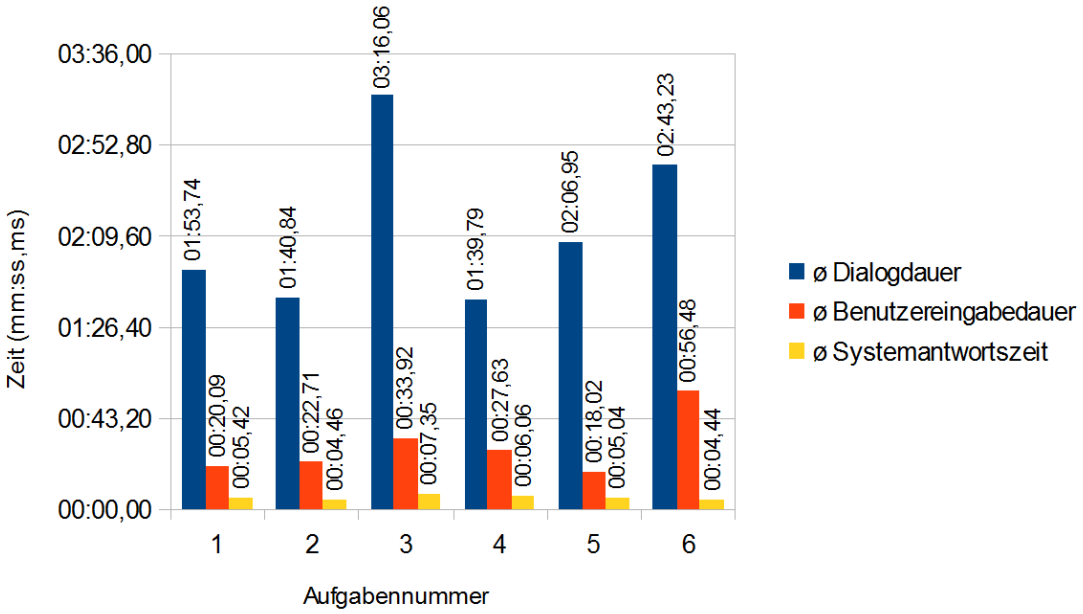

Abbildung 5.2: Gemessene, durchschnittliche Zeiten der einzelnen Aufgaben (1-6)

Im Anschluss an das kontrollierte Experiment beantwortete jede Testperson zwölf Fragen zur Benutzeroberfläche und den Modalitäten. Zusätzlich wurde den Versuchsteilnehmern ein standardisierter Fragebogen zur Evaluierung der Anwendung vorgelegt. Von den insgesamt 72 durchgeführten Aufgaben führten 69 zum Erfolg (95,83%), das heißt, dass sie mit einem für die Testperson positiven Ergebnis abgeschlossen wurden. In den übrigen drei Fällen kam es zu Problemen, weil kein passender POI gefunden wurde beziehungsweise das System die Anfrage nicht lösen konnte. In letzterem Fall wurde die wörtliche Eingabe von *am Abend etwas unternehmen* nicht verstanden. Für eine erfolgreiche Anfrage benötigten die Probanden im Durchschnitt ungefähr 1,5 Versuche. Eine genaue Auflistung der Befragungsergebnisse samt grafischer Darstellung der Resultate des ersten Fragebogens befindet sich im Anhang auf Seite 54.

Ansichten

Die Befragung ergab, dass 10 der 12 Personen (83,34%) die Kartenansicht bevorzugten, die restlichen zwei betrachteten die Aussage neutral. Einige Probanden merkten an, dass sich die Karten- und Listenansicht gegenseitig ergänzen. Der Durchschnitt[4] bei der Frage, ob die Listenansicht beim Auffinden von Lokationen hilft, liegt bei 3,5 (Median: 3; Minimum: 1;

[4]auf einer Likert-Skala (1 - trifft zu, 7 - trifft nicht zu)

Maximum: 7; Standard-Abweichung: 1,68) und verhält sich somit eher neutral, mit auseinandergehenden Meinungen. Dies spiegelt sich auch im Verhalten der Testpersonen wieder. Es gab Fälle, in denen die Versuchsteilnehmer die Listenansicht nicht benutzten und es bevorzugten, einzelne POIs auf der Karte anzuklicken. Sie begründeten ihr Verhalten mit einer besseren Orientierung durch die Verwendung der Karte. Die Clusteransicht wird von vier Personen als hilfreich angesehen (Werte von 1 und 2 auf der Likert-Skala von 1 bis 7), zwei Personen hielten diese Ansicht für wenig hilfreich (Werte von 6 und 7), der Rest antwortete mit Werten zwischen drei und vier. Die Abbildung auf dieser Seite stellt das Ergebnis grafisch dar. Die Standardabweichung war mit 1,91 bei dieser Frage am größten.

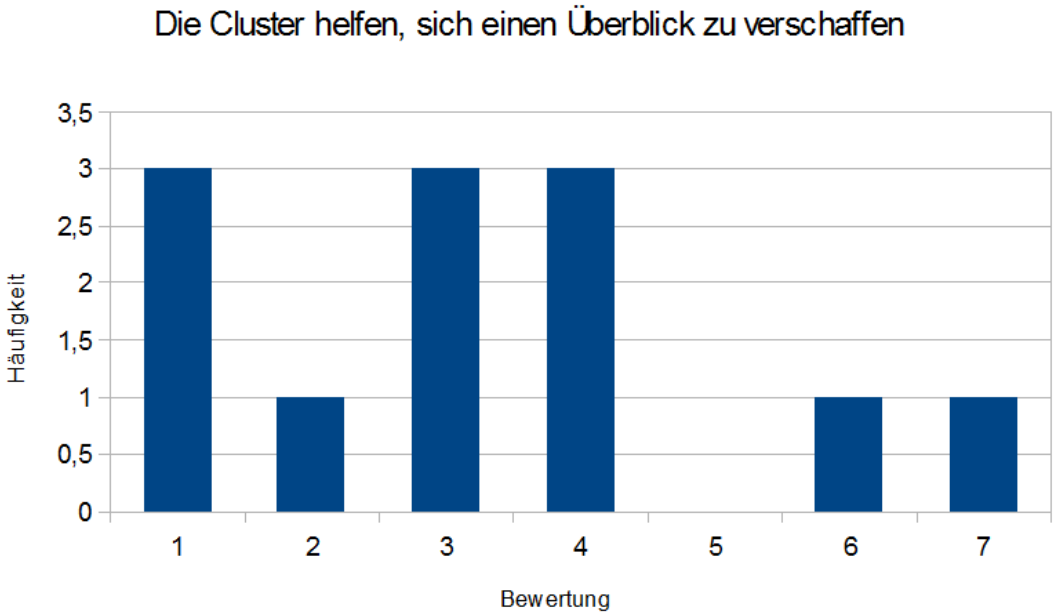

Abbildung 5.3: Bewertung der Clusteransicht

Eingabemodalitäten

Ein Großteil der Versuchsteilnehmer war von der Sprache als Eingabemodalität angetan. So bewerteten 92% der Probanden (11 von 12 Personen) die Nützlichkeit der Sprache mit 1 (neunmal) beziehungsweise 2 (zweimal). Nur eine Person kreuzte eine 5 an. Dies hängt damit zusammen, dass die Spracherkennung bei dieser mit Spracheingabe unerfahrenen Person (Nummer 9) sehr schlecht funktionierte. Die vierte Aufgabe[5] wurde von ihm/ihr erst nach dem fünften, erfolglosen Versuch gelöst, obwohl die Grammatik alle Formulierungen verstanden hätte. Die Spracherkennung von Nuance lieferte immer falsche Transkripte.

[5] *Sushi essen in Berlin Mitte* via Sprache

Die Abbildung 5.4 veranschaulicht sehr gut die unterschiedliche Bewertung der Nützlichkeit einer Spracheingabe vor und nach dem Lösen der Aufgaben. Bei der Hälfte der Versuchsteilnehmer blieb die Einschätzung konstant sehr gut, fünf Personen bewerteten die Nützlichkeit nach dem Experiment besser und nur eine Person korrigierte ihre Einschätzung zum Schlechteren. Diese Person (Nummer 9) ist diejenige, die Probleme mit der Spracherkennung hatte. Der Proband mit der Nummer 12 änderte sogar komplett seine Meinung – von einer sehr negativen Einstellung zu einer sehr positiven.

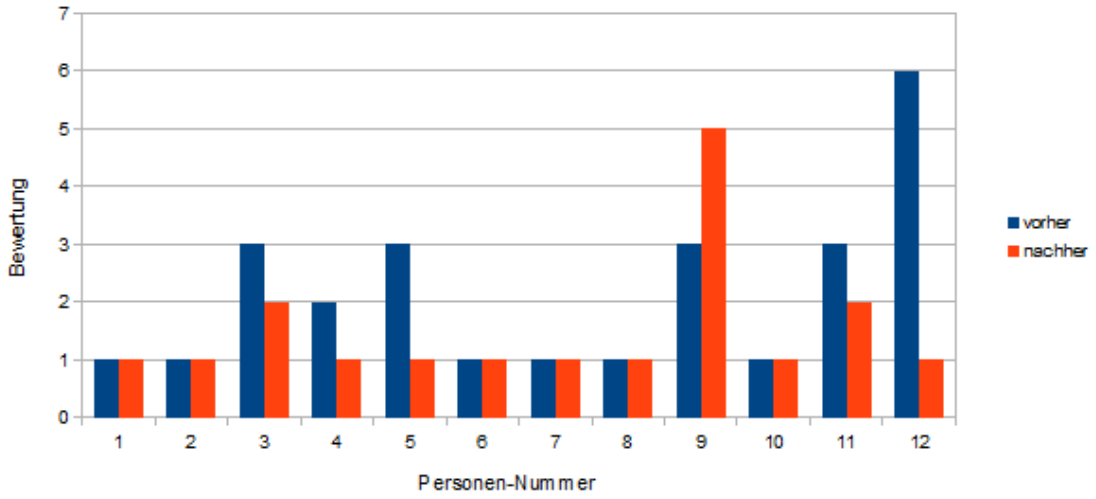

Abbildung 5.4: Bewertung der Nützlichkeit von Spracheingabe (1 - stimme zu, 7 - stimme nicht zu) vor und nach der Durchführung des Laborexperiments

Ähnliche Antworten wie die vorherige erzielte auch die Frage nach der Erleichterung der Bedienung durch die Modalität Sprache. Zehn Personen stimmten der Aussage zu (Werte 1 und 2) und nur zwei Personen waren nicht überzeugt (Wert 4 und 7). Fast alle Personen stimmten überein, dass es Situationen gibt, in denen sie Sprache bevorzugen würden (Durchschnitt: 1,5; Median: 1; Maximum: 4).

Die Zuverlässigkeit der Spracheingabe wird mittelmäßig bewertet und die Ergebnisse variieren stark. Der Median liegt bei 3 (Mittelwert: 2,92; Minimum: 1; Maximum: 7; Standardabweichung: 1,88). Zehn sehr guten bis mittelmäßigen Werten (1 bis 3) stehen zwei schlechte Bewertungen (6 und 7) gegenüber. Die Intuitivität der Spracheingabe wird von allen Teilnehmern positiv bewertet (Mittelwert: 1,75; Median: 1,5; Minimum: 1; Maximum: 3).

Es wurde außerdem gefragt, ob die Eingabe via Touch als zu langsam empfunden wird. Der durchschnittliche Wert betrug 2,67 (Median: 3; Minimum: 1; Maximum: 5), was bedeutet, dass die Studienteilnehmer der Aussage eher neutral gegenüber standen.

Zum Abschluss wurde gefragt, welche Modalität die Versuchsteilnehmer bevorzugten. In der Abbildung 5.5 wird die Verteilung grafisch veranschaulicht. Auf einer Skala von 1 (Sprache) bis 7 (Touch) betrug der Median 2 (Mittelwert: 2,33; Minimum: 1; Maximum: 7), was andeutet, dass Sprache als Eingabemodalität favorisiert wird. Die einzige Person, die eine 7 ankreuzte, war Nummer 9. Die Auswertung des Fragenkatalogs bestätigt somit die zweite Hypothese, dass Spracheingabe die Bedienung der Anwendung erleichtert.

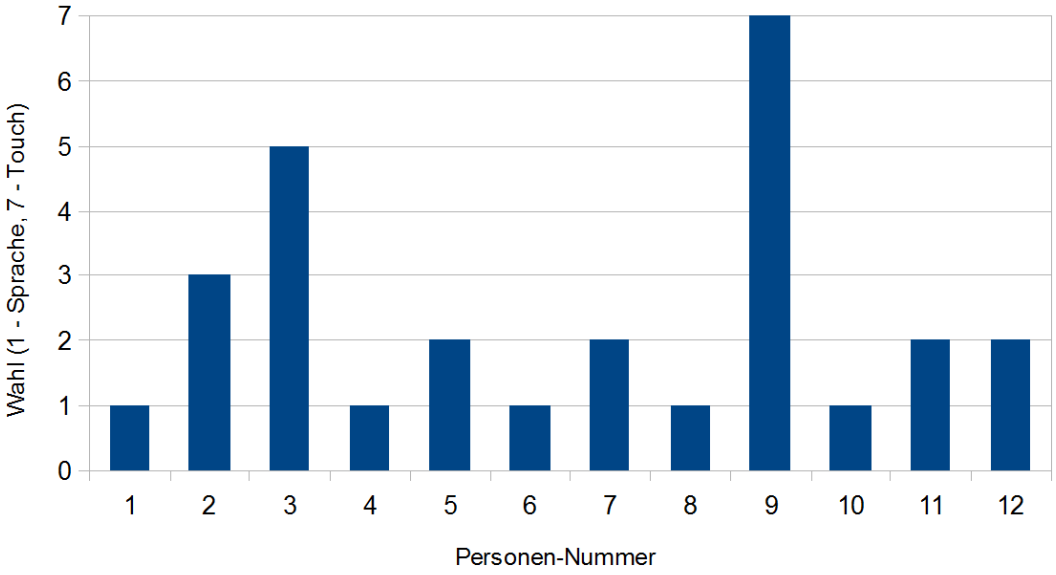

Abbildung 5.5: Favorisierte Eingabemodalität (1 - Sprache, 7 - Touch)

Bei der fünften und sechsten Aufgabe entschied sich die Mehrheit für die Modalität Sprache: 16 der 23 Aufgaben[6] (69,5%) wurden mittels Spracheingabe gelöst, 4 mittels Toucheingabe und 3 via Kombination aus Touch- und Spracheingabe (siehe Abbildung 5.6).

Die Videoauswertung ermöglichte eine objektive Analyse der beiden Modalitäten. Bei der Auswertung der Zeiten wurden jeweils die ersten vier Aufgaben betrachtet. In Abbildung 5.7 werden die durchschnittlichen Zeiten eines Zuges[7] grafisch dargestellt. Die durchschnittliche Zeit einer Spracheingabe mit entsprechender Antwort[8] dauert 18,51 Sekunden (Median: 13 Sekunden; Minimum: 9,11 Sekunden; Maximum: 56,78 Sekunden). Die durchschnittliche Dauer bei der Bedienung mittels Touchs beträgt 45,30 Sekunden (Median: 43 Sekunden; Minimum: 24,3 Sekunden; Maximum: 1:32,19). Diese Ergebnisse zeigen, dass die Eingabe

[6] 23 anstatt 24 Aufgaben, weil eine nicht erfolgreich durchgeführt wurde.
[7] Dauer der initialen Eingabe; die Summe aus Benutzereingabezeit und Systemantwortzeit
[8] Die Antwort des Systems nach einer Spracheingabe dauert länger, da die Sprache in Text umgewandelt und interpretiert werden muss.

via Sprache im Durchschnitt mehr als doppelt so schnell ist. Angemerkt sei hier, dass das Touchinterface verbesserungswürdig ist. Das Hinzufügen einer Autovervollständigung beziehungsweise die Möglichkeit der Regionsauswahl per Touch würde die Eingabedauer der Modalität Touch deutlich verringern.

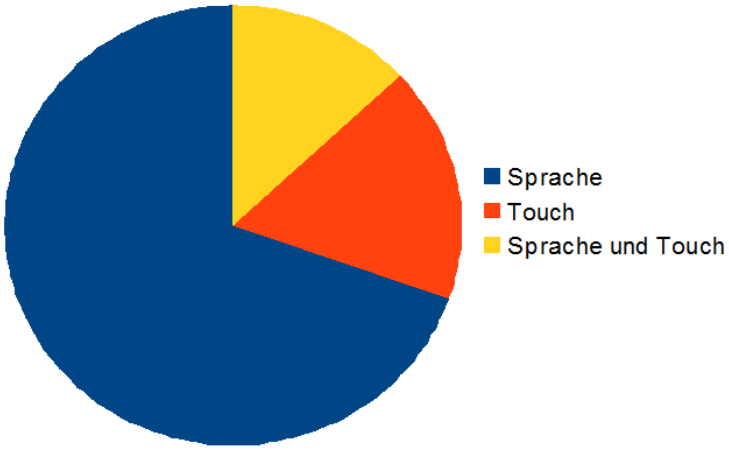

Abbildung 5.6: Verteilung der Eingabemodalitäten bei der 5. und 6. Aufgabe

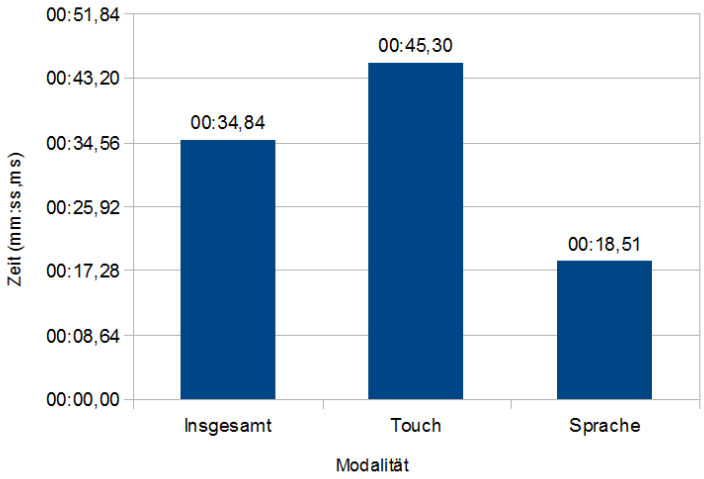

Abbildung 5.7: Durchschnittliche Dauer eines Zuges (Benutzereingabezeit und Systemantwortzeit) bei der initialen Eingabe

Web-Services

Während des Experiments wurden die Bearbeitungszeiten der beiden Webservices geloggt. Im Durchschnitt dauerte die Georeferenzierung 0,18 Sekunden (Median: 0,16s; Minimum:

0,04s; Maximum: 0,36s) und die Bereitstellung der Ergebnisse inklusive der Gruppierung akzeptable 0,19 Sekunden (Median: 0,1s; Minimum: 0,01s; Maximum: 1,07s). Über alle Anfragen liegt der Median der POI-Trefferanzahl bei 11 (Durchschnitt: 57; Minimum: 1; Maximum: 224) und die Median der Clusteranzahl bei vier (Durchschnitt: 6,04; Minimum: 1; Maximum: 21).

USE-Fragebogen

Der USE-Fragebogen ist laut einer Studie von Kühnel et al.[KWWM10] für die Evaluation multimodaler Systeme geeignet. Anders als in der Studie[9], wurde *ein* Fragebogen für beide Modalitäten verwendet.

Insgesamt umfasst der Fragenkatalog 30 Fragen und deckt vier Gebiete ab: Nützlichkeit, Einfachheit der Benutzung, Einfachheit des Lernens und Zufriedenheit[10]. Alle Fragen werden wie bei den vorherigen auf einer Likert-Skala von 1 (trifft zu) bis 7 (trifft nicht zu) bewertet. Eine genaue Aufschlüsselung aller Ergebnisse befindet sich im Anhang auf Seite 55. Im Folgenden werden interessante Sachverhalte veranschaulicht.

Besonders gut schnitt die Frage nach der Nützlichkeit ab (Frage 3). Der Durchschnitt beträgt 1,67 (Median: 1,5; Minimum: 1; Maximum: 3). Die gesamte Gruppe erreicht aber nur einen Durchschnitt von 2,76 und ist von den vier Fragegruppen somit die schlechteste. Die Gruppe *Einfachheit der Benutzung* erreicht den zweitbesten Durchschnitt mit 2,29. Insbesondere der Aussage, dass für die Nutzung der Anwendung keine geschriebene Instruktionen erforderlich sind, stimmen die Testpersonen zu. Verbesserungsbedarf sehen sie beim Korrigieren von Fehlern und der Flexibilität.

Am besten Schnitt die Fragegruppe *Einfachheit des Erlernens* ab. Die Anwendung ist einfach zu benutzen (Durchschnitt: 1,25; Maximum: 3). Der Durchschnitt der Zufriedenheit-Fragegruppe erreicht nur Rang 3 von 4, was unter anderem daran liegt, dass sie die Anwendung nicht unbedingt haben müssen[11] (Durchschnitt: 3,75). Die Befragten sind insgesamt zufrieden, würden die Anwendung Freunden empfehlen und es macht ihnen Spaß sie zu benutzen (Durchschnitt jeweils 1,83).

Insgesamt lässt sich schlussfolgern, dass die Anwendung einfach zu erlernen ist, es jedoch noch Verbesserungspotential bei der Behandlung von Fehlern gibt.

[9] jeweils ein Fragebogen für eine Modalität
[10] Englisch: Usefulness, Ease of Use, Ease of Learning, Satisfaction
[11] Englisch: „I feel I need to have it.“

Relevanz der Empfehlungen

Um die erste Hypothese zu überprüfen, wurden die Probanden gefragt, ob die Empfehlungen aus sozialen Netzwerken bei der Entscheidungsfindung helfen. Die Antworten sind in Abbildung 5.8 grafisch dargestellt. Es ist ersichtlich, dass 11 der 12 Befragten aussagten, dass sie dieser Aussage eher zustimmen (Durchschnitt: 2,25; Median: 2; Standardabweichung: 1,6). Nur eine Person antwortete (Proband Nummer 11), dass Empfehlungen nicht hilfreich sind. Bei Nachfrage bemerkte die Person, dass die Informationen – insbesondere die durchschnittliche Bewertung – wohl dem Empfinden der Mehrheit entspricht, er/sie jedoch aus Erfahrung weiß, dass der Geschmack der Mehrheit nicht seinem/ihrem eigenen entspricht.

Nach jeder Aufgabe bewerteten die Probanden die Relevanz der Treffer mit Werten zwischen 1 (gut) und 3 (schlecht). Der Durchschnitt betrug ungefähr 1,3. Es gab auch mehrere Fälle, in denen die Treffer nicht alle für relevant empfunden wurden. Dies war besonders bei der dritten Aufgabe (*Einkaufen am Alexanderplatz*) der Fall und ist durch Aufspaltung beziehungsweise Einschränkung der Kategorien lösbar.

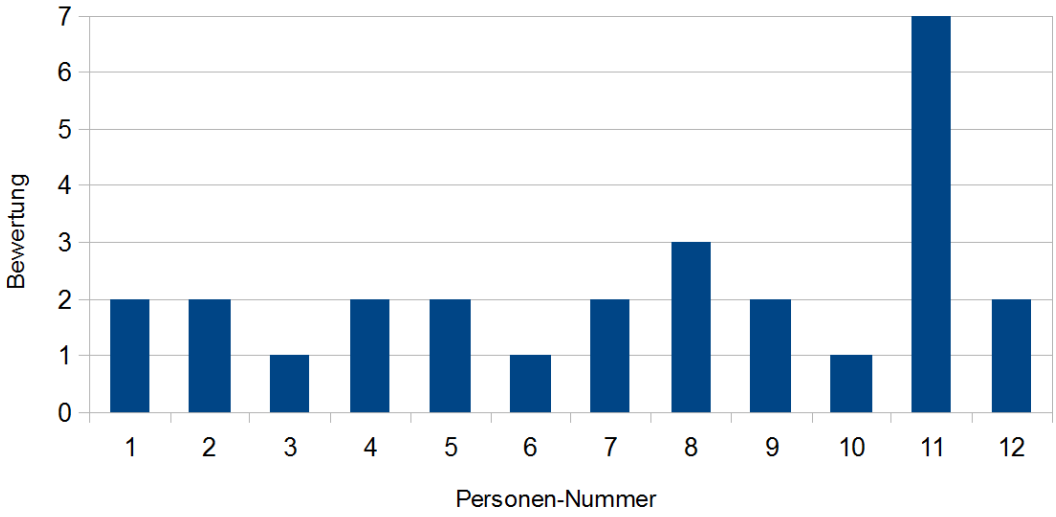

Abbildung 5.8: Bewertung von Informationen aus sozialen Netzwerken

Insgesamt wurden durch die Probanden 69 POIs ausgewählt. Drei von ihnen besaßen keine durchschnittliche Bewertung. Der Durchschnitt[12] aller Bewertungen der ausgewählten POIs beträgt 4,18. Die Befragung und Auswertung der ausgewählten Treffer lassen die Schlussfolgerung zu, dass Informationen aus sozialen Netzwerken die Auswahl geeigneter POIs

[12]keine Bewertung wird als eine Bewertung von 0 angesehen

unterstützen. Ob jedoch die durchschnittliche Bewertung gerechtfertigt ist, also mit dem jeweiligen subjektiven Empfinden der Person übereinstimmt, lässt diese Untersuchung offen. Dies würde erfordern, dass die ausgewählten POIs wirklich besucht werden würden.

Spracherkennung und Sprachverstehen

Während des Laborexperiments wurden Spracheingaben getätigt und Sprachbefehle benutzt. Falls das System korrekt reagierte, wurde die Eingabe als verstanden markiert. Von den 128 Äußerungen wurden 91 richtig verstanden (71,09%). Von diesen 91 richtig verstandenen Eingaben reagierte das System in 73 Fällen (80,22%) korrekt, das heißt, dass der Grammatik-Parser den Text richtig interpretierte. Eine Mehrzahl der richtig verstandenen, aber nicht durch die Grammatik abgedeckten Äußerungen hatten folgendes Muster: *„<Aktivität> <Lokation>“* (zum Beispiel: *Nachtleben Warschauer Straße*). Von der Grammatik erkannt wurden nur Texte der Art *„<Aktivität> <Präposition> <Lokation>“*. Außerdem hatte der Parser Probleme mit der Satzstruktur *„[...] <Aktivität> <Präposition> <Umgebung> <Artikel/Präposition> <Lokation>“* (zum Beispiel: *Bitte ein Café in der Nähe des Hauptbahnhofs*).

Während der Entwicklung fiel mir auf, dass bestimmte Formulierungen helfen, die Zuverlässigkeit der Spracherkennung zu verbessern. So führt zum Beispiel das Einfügen von Wörtern wie *bitte* oder die Verwendung von Synonymen, beispielsweise *suche [...]* anstelle von *zeige [...]* zu besseren Ergebnissen. Dies lässt vermuten, dass durch Training, Übung und Erfahrung die Zuverlässigkeit der Spracheingabe verbessert werden kann.

Lokalisierungsservice

Im System wird der Georeferenzierungsservice von GeoNames benutzt[13]. Dabei wird immer der erste Eintrag[14] der zurückgegebenen Ergebnisliste verwendet. Die Benutzer bewerteten die subjektive Übereinstimmung der Region mit ihrer Vorstellung mit 1,26 (Skala von 1 - gut bis 3 - schlecht). Einige interessante Fälle werden im Folgenden aufgelistet:

- Die Eingabe von *Berlin Alexanderplatz* führt zu *NH Berlin Alexanderplatz,* einem Hotel in einiger Entfernung zum Alexanderplatz
- Bei der Eingabe von *Kreuzberg* erkennt der Webservice als besten Treffer den Stadtteil *Friedrichshain-Kreuzberg.* Dies entspricht auf der Karte der Grenze zwischen Kreuzberg und Friedrichshain – *Berlin Kreuzberg* führt hingegen zur richtigen Lokalisierung.

[13] siehe Abschnitt 4.5
[14] laut GeoNames der Eintrag mit der besten Übereinstimmung

Ein einziges Mal kam es vor, dass der Ort vom Parser richtig extrahiert wurde, dieser jedoch nicht durch GeoNames gefunden werden konnte. Dabei handelte es sich um den *Nollendorfplatz*. Da GeoNames nur über beschränktes Wissen über Plätze und Straßen verfügt, bietet es sich an, einen weiteren Service, wie zum Beispiel Nominatim[15], zu verwenden.

Diskussion

Die Evaluierung des Systems ergab, dass die Spracheingabe von der Mehrheit bevorzugt wird. Andere Arbeiten bestätigen, dass Sprache eine passende Modalität zur Bedienung mobiler Geräte darstellt [TMH+09, JBV+02]. Es stellte sich heraus, das Sprache sowohl bei subjektiver als auch bei objektiver Betrachtung von den meisten Probanden favorisiert wird. Im Allgemeinen funktionierte die Spracherkennung sehr gut, jedoch benötigten einige Personen für die Aufgabenlösung mittels Spracheingabe mehr als vier Versuche.

Das Experiment wurde in einer kontrollierten Umgebung unter idealen Bedingungen durchgeführt. Einige Versuchsteilnehmer bemerkten, dass sie sich Situationen vorstellen könnten, in denen sie Touch der Eingabe via Sprache vorziehen würden. Als Beispiele wurden Situationen mit lauter Geräuschkulisse beziehungsweise Orte mit vielen Leuten genannt. Eine Testperson sagte, dass es ihm/ihr unangenehm wäre, im Umkreis anderer Personen mit dem Smartphone zu reden. Eine andere Person wiederum hätte kein Problem damit. Feldexperimente könnten helfen, solche Fragestellungen zu untersuchen.

Viele der Teilnehmer stammten aus Berlin. Es kam deshalb vor, dass sie einige der empfohlenen POIs bereits kannten. Im Gegensatz dazu gab es auch Situationen, bei denen erwartete POIs nicht in der Ergebnisliste enthalten waren. Trotzdem gelang es den Probanden, in 69 der 72 Fällen passende Treffer zu finden.

5.3 Zusammenfassung

In diesem Kapitel wurden der Evaluationsprozess und die Evaluationsergebnisse des multimodalen, mobilen Empfehlungssystems beschrieben. Zwölf Personen lösten in einer kontrollierten Umgebung sechs Aufgaben mit Hilfe des Prototyps, wobei unterschiedliche Eingabemodalitäten verwendet wurden. Zusätzlich zu der Analyse der Videoaufzeichnung wurden die Probanden befragt. Durch Auswertung des Fragebogens konnten die zu untersuchenden Hypothesen bestätigt werden.

[15] `http://nominatim.openstreetmap.org/`

6 Fazit

Multimodale, mobile Anwendungen gewinnen durch die zunehmende Verbreitung von Smartphones immer mehr an Bedeutung. Das Angebot verschiedener Modalitäten ermöglicht dem Nutzer, die für die jeweilige Situation geeignetste Modalität auszuwählen. Außerdem können Modalitäten kombiniert werden, um eine natürlichere Interaktion zu erlauben.

In der Arbeit wurde die prototypische Entwicklung und Evaluation eines multimodalen Empfehlungssystems für Lokationen beschrieben. Leichte Modifizierungen vorhandener Komponenten – wie Spracherkennung und Grammatikparser – wurden erfolgreich mit Diensten aus dem Internet zu einer multimodalen Anwendung kombiniert. Eine eigene Grammatik zum Parsen von Sprachbefehlen wurde entwickelt. Der Einsatz von Webtechnologien und die zielgerichtete Verknüpfung vorhandener Module ermöglichten die zügige Entwicklung der mobilen Android-App. Das System lässt sich sowohl mittels Touch-Interface als auch per Sprache bedienen. Alle zu erreichenden Anforderungen wurden erfüllt.

Das System wurde mit 12 Versuchspersonen evaluiert. Die Ergebnisse fielen größtenteils positiv aus. Während der Arbeit wurden zwei Hypothesen untersucht. H1 besagt, dass Daten aus sozialen Netzwerken Benutzer bei der Entscheidungsfindung unterstützen. Über 90% (11 von 12) der befragten Personen stimmten der Aussage zu, dass soziale Netzwerke bei der Entscheidungsfindung helfen.

Eine Befragung und die Videoauswertung des Experimentes zeigten, dass ein Großteil der Probanden die Sprachimplementierung gegenüber der angebotenen Toucheingabe vorzogen. Zehn der 12 Probanden bevorzugten die initiale Eingabe via Sprache. Nach dem Lösen der Aufgaben verbesserte sich die Bewertung der Nützlichkeit von Spracheingabe bei fünf Personen – nur eine Person änderte ihre Einschätzung zum Negativen. Bei den Aufgaben, wo sie die Modalität frei wählen durften, entschied sich die Mehrheit für die Modalität Sprache. Bei 16 der 23 Aufgaben (69,5%) wurde das Sprach-Interface verwendet, 4 Aufgaben wurden mittels Toucheingabe und 3 via Kombination aus Touch- und Spracheingabe gelöst. Die zweite Hypothese, dass Sprache die Bedienung eines multimodalen, lokationsbasierten Empfehlungssystems erleichtert, wurde bestätigt.

6.1 Ausblick

Die vorliegende Arbeit eröffnet einige Möglichkeiten für zukünftige Erweiterungen. Zur Zeit besitzt das System nur Zugriff auf Lokationen in Berlin. Daten anderer Städte müssten der Datenbank hinzugefügt werden. Andere Quellen könnten weitere Informationen zu Öffnungszeiten oder Preisen liefern. Zusätzlich wäre eine Erweiterung der Grammatik erforderlich, um auch Ortsnamen in unbekannten Regionen besser zu erkennen. In gleicher Weise könnten bisher nicht verstandene Formulierungen der Grammatik hinzugefügt werden.

Weiterhin wäre es interessant, die beiden Modalitäten besser miteinander zu verknüpfen, beziehungsweise andere Modalitäten einzubinden. Gesten innerhalb der Kartenansicht könnten die Interaktion natürlicher gestalten (zum Beispiel durch Einkreisen bestimmter Gebiete). Sprachausgabe oder andere Formen der Bedienung, wie zum Beispiel *Sliding Maps* [MHJ04], könnten integriert und getestet werden. Verbesserungen des Touchinterfaces mittels Autovervollständigung und bessere Auswahlmöglichkeiten wurden von einigen Probanden gewünscht.

Des Weiteren könnte eine flexiblere Gestaltung der Suchregion dem Benutzer helfen, sich einen Überblick über größere Regionen zu verschaffen. Durch eine bessere Granularität der Kategorien und Optimierungen der Darstellung könnte die Übersichtlichkeit bei einer sehr großen Anzahl von Treffern noch erhöht werden.

Weitere Evaluationen könnten helfen, Schwächen – besonders bei der Verwendung in der Öffentlichkeit – aufzudecken und zu testen, ob Sprache auch im Freien bevorzugt wird.

6.2 Schlusswort

Die vorliegende Arbeit zeigt, dass Sprache und Touch als Eingabeformen gut zusammenarbeiten und auch von den Nutzern akzeptiert werden. Dieses Resultat bestätigt Ergebnisse anderer Arbeiten. In der mobilen Welt zeigen populäre Apps wie Apples *Siri*, dass Interesse an sprachgesteuerten Systemen vorhanden ist. Die Zukunft wird zeigen, ob bessere Technologien und Weiterentwicklungen dazu führen, dass sich multimodale Anwendungen im mobilen Bereich durchsetzen.

A Anhang

A.1 Grammatik

```
var grammar = {
    "stop_word" : [ "bitte", "doch", "gerne", "der", "die", "das", "dem",
        "den", "des", "ein", "einen", "eines", "einer", "mein", "meine",
        "meines", "meiner", "diese", "dieser", "diesem", "diesen",
        "dieses", "mir", "mal", "jetzt", "etwas", "lass", "berlin",
        "berliner", "gehen", "besuchen", "betreten", "machen", "erleben",
        "werden", "ich", "du", "er", "sie", "es", "wir", "sie", "man",
        "gute", "gut", "guten" ],

    "tokens" : {
        "V_SHOW" : [ "zeig", "zeige", "zeigen", "anzeigen", "finden", "finde",
                "find", "scann", "scanne" ],
        "V_WISH" : [ "m___oe___cht(e)?(n)?", "moecht(e)?(n)?",
                "w___ue___rd(e)?(n)?", "wuerd(e)?(n)?", "will", "wollen",
                "wollt", "suche", "such", "suchen" ],
        "V_CAN" : [ "sind", "ist", "gibt", "gibts", "kann", "k___oe___nnen",
                "darf", "d___ue___rfen" ],
        "WHERE" : [ "wo", "wie" ],

        // for commands

        "CLUSTER_VIEW" : [ "cluster", "clusteransicht", "gruppe(n)?",
                "gruppen(␣)?ansicht" ],
        "NORMAL_VIEW" : [ "normal(e)?(n)?(r)?(␣)?ansicht",
                "standard(␣)?ansicht" ],
        "NUMBER" : [ "eins", "zwei", "drei", "vier", "f___ue___nf", "sechs",
                "sieben", "acht", "neun", "zehn", "elf", "zw___oe___lf",
                "[1-9][0-9]" ],
        "ON_OFF" : [ "einschalten", "ausschalten" ],
        "LIST" : [ "list(e)?", "listenansicht" ],
        "RECOMMENDATION" : [ "empfehlung(en)?", "beste(n)?",
                "beste(n)?␣treffer", "beste(n)?␣ergebnis(se)?" ],

        // categories
        "CATEGORY_VERB" : [ "essen", "speisen", "fr___ue___hst___ue___cken",
                "dinieren", "abendessen", "abend␣essen", "mittagessen",
                "mittag␣essen", "brunchen", "cafe␣trinken", "kaffee␣trinken",
```

```
            "trinken", "pizza(s)?␣essen", "sushi␣essen", "partien",
            "feiern", "musik␣h__oe__r(e)?n", "tanzen", "shoppen",
            "einkaufen", "ausgehen", "sport␣treiben", "unternehmen",
            "unterhalten" ],
    "CATEGORY_NOUN" : [ "sushi[a-z]*", "sushi␣restaurant(s)?",
            "sushi␣bar(s)?", "bar(e)?(s)?", "kneipe(n)?", "cafe(s)?",
            "kaffee(s)?", "restaurant(s)?", "pizzeri[ae](n)?", "pizza(s)?",
            "fastfood", "fast␣food", "fastfood␣restaurants", "d__oe__ner",
            "imbiss", "nacht[ck]lub(s)?", "nachtleben", "dis[ck]o(s)?",
            "party(s)?", "shops", "gesch__ae__ft(e)?", "laden",
            "l__ae__den", "laeden", "einkauf[a-z]*", "unterhaltung",
            "sport", "unterhaltungsm__oe__glichkeit(en)?", "kino(s)?",
            "theater(s)?", "kultur[a-z]*", "kunst[a-z]*" ],

    // locations
    "PREPOSITION" : [ "an", "um", "am", "im", "in", "auf", "f__ue__r",
            "nach", "ins" ],
    "NEAR_TO_ME" : [ "n__ae__he", "naehe", "umfeld", "umgebung",
            "nachbarschaft", "umkreis", "hier" ],
    "STREET" : [ "[a-z]+␣str", "[a-z]+␣stra__ss__e", "[a-z]+␣str.",
            "[a-z]+str", "[a-z]+stra__ss__e", "[a-z]+str.", "[a-z]+damm" ],
    "PLACE" : [ "[a-z]+platz", "([a-z]+-)+platz" ],
    "LOCATION_NAME" : [ "friedrichshain", "hellersdorf",
            "hohensch__oe__nhausen", "k__oe__penick", "mitte", "pankow",
            "prenzlauer␣berg", "treptow", "wei__ss__ensee",
            "charlottenburg", "kreuzberg", "neuk__oe__lln",
            "reinickendorf", "sch__oe__neberg", "spandau", "steglitz",
            "tempelhof", "tiergarten", "wedding", "wilmersdorf",
            "zehlendorf", "mitte", "friedrichshain␣kreuzberg",
            "charlottenburg␣wilmersdorf", "steglitz␣zehlendorf",
            "tempelhof␣sch__oe__neberg", "treptow␣k__oe__penick",
            "marzahn␣hellersdorf", "lichtenberg", "moabit",
            "zoo", "hauptbahnhof", "wedding", "prenzlauer␣berg" ]
},
"utterances" : {
    "SHOW_LIST" : {
        "phrases" : [ "V_SHOW␣LIST", "LIST␣V_SHOW", "LIST" ],
        "semantic" : {
            "ShowList" : {}
        }
    },
    "SHOW_CLUSTER" : {
        "phrases" : [ "V_SHOW␣CLUSTER_VIEW␣NUMBER", "CLUSTER_VIEW␣NUMBER",
                "CLUSTER_VIEW␣NUMBER␣V_SHOW" ],
        "semantic" : {
            "ShowCluster" : {
                "cluster" : "_$number[0]"
            }
        }
    },
```

```
"SHOW_BEST" : {
    "phrases" : [ "V_SHOW RECOMMENDATION", "RECOMMENDATION V_SHOW",
            "RECOMMENDATION" ],
    "semantic" : {
        "ShowBest" : {}
    }
},
"SEARCH_HERE" : {
    "phrases" : [ "VERB NEAR_TO_ME", "VERB PREPOSITION NEAR_TO_ME" ],
    "semantic" : {
        "ShowHere" : {}
    }
},
"CLUSTER_TOGGLE" : {
    "phrases" : [ "CLUSTER_VIEW ON_OFF", "V_SHOW CLUSTER_VIEW",
            "CLUSTER_VIEW V_SHOW", "CLUSTER_VIEW" ],
    "semantic" : {
        "ClusterToggle" : {
            "mode" : "_$on_off[0]"
        }
    }
},
"SHOW_NORMAL_VIEW" : {
    "phrases" : [ "NORMAL_VIEW", "V_SHOW NORMAL_VIEW",
            "NORMAL_VIEW V_SHOW" ],
    "semantic" : {
        "ShowNormalView" : {}
    }
},
"CATEGORY" : {
    "phrases" : [ "CATEGORY_VERB", "CATEGORY_NOUN" ],
    "semantic" : {
        "Category" : {
            "name" : "_$phrase"
        }
    }
},

"LOCATION" : {
    "phrases" : [ "LOCATION_NAME", "STREET", "PLACE", "NEAR_TO_ME" ],
    "semantic" : {
        "Place" : {
            "name" : "_$phrase",
            "nearby" : "_$near_to_me[0]"
        }
    }
},

"VERB" : {
    "phrases" : [ "V_SHOW", "V_WISH", "V_CAN" ],
```

```
            "semantic" : {}
        },

        "SHOW_POIS" : {
            "phrases" : [ "VERB␣CATEGORY␣PREPOSITION␣LOCATION",
                    "CATEGORY␣PREPOSITION␣LOCATION",
                    "VERB␣PREPOSITION␣LOCATION␣CATEGORY",
                    "WHERE␣PREPOSITION␣LOCATION␣VERB␣CATEGORY",
                    "WHERE␣VERB␣CATEGORY␣PREPOSITION␣LOCATION",
                    "WHERE␣VERB␣PREPOSITION␣LOCATION␣CATEGORY" ],
            "semantic" : {
                "ShowPOI" : {
                    "category" : "_$category[0]['semantic']",
                    "location" : "_$location[0]['semantic']",
                }
            }
        },
        "CHANGE_CATEGORY" : {
            "phrases" : [ "VERB␣CATEGORY" ],
            "semantic" : {
                "ChangeCategory" : {
                    "category" : "_$category[0]['semantic']",
                }
            }
        }
    }
};
```

A.2 Evaluationsbogen (Gruppe A)

Durchführung

	Alter:	
	Geschlecht:	M ○ ○ W
	Ausbildungsabschluss:	
	Beruf:	
a)	Besitzen Sie ein Smartphone (Wenn ja: welches OS?):	
b)	Erfahrung mit Spracheingabe:	Keine ○ ○ ○ ○ ○ Viel
c)	Ich denke, dass die Eingabe via Sprache nützlich sein kann:	Zutreffend ○ ○ ○ ○ ○ ○ ○ Nicht zutreffend

Im Nachfolgenden werden Ihnen für jede Eingabemodalität (Touch, Sprache, freie Wahl) jeweils zwei Aufgaben gestellt. Stellen Sie sich vor, dass Sie sich wünschen, die vorgegebene Aktivität in der vorgegebenen Gegend zu unternehmen. Bitte notieren Sie für jede Aufgabe Ihre Wahl (den Namen des POIs) in dem vorgesehenen Feld. Achten Sie bitte darauf, dass sich die gefundenen Lokalitäten in der gewünschten Umgebung befindet. Bewerten Sie die Ortsübereinstimmung und die Nützlichkeit der Empfehlung durch Ankreuzen der passenden Wahl.

I. Touch

Für die zwei folgenden Aufgaben benutzen Sie bitte entweder die Auswahlliste oder die Eingabe via Touchscreen-Tastatur.

1. Essen – Berlin Friedrichshain

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

2. Café – Berlin Hauptbahnhof

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

II. Sprache

Für diese Aufgabe verwenden Sie bitte das Sprachinterface der Anwendung. Bevor die Anwendung Gesprochenes akzeptiert, muss der „Spracheingabe“-Button (unterer Bildschirmrand) gedrückt werden. Ein Ton signalisiert die Bereitschaft zur Spracheingabe.

1. Einkaufen – Berlin Alexanderplatz

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

2. Sushi – Berlin Mitte

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

III. Multimodal

Bei den nächsten zwei Aufgaben steht es Ihnen frei, welche Modalität Sie zur Eingabe verwenden.

1. Pizzeria – Tempelhof

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

2. Am Abend etwas unternehmen

Ausgewählter POI:	
Ortsübereinstimmung:	Gut ○ ○ ○ Schlecht
Nützlichkeit der Empfehlung:	Gut ○ ○ ○ Schlecht

,

Fragebogen

		zutreffend - nicht zutreffend
1.	Ich bevorzuge die Kartenansicht.	○ ○ ○ ○ ○ ○ ○
2.	Die Listenansicht erleichtert das Auffinden von Lokationen.	○ ○ ○ ○ ○ ○ ○
3.	Die Cluster helfen, sich einen Überblick zu verschaffen.	○ ○ ○ ○ ○ ○ ○
4.	Die Spracheingabe ist nützlich.	○ ○ ○ ○ ○ ○ ○
5.	Die Modalität Sprache erleichtert die Bedienung.	○ ○ ○ ○ ○ ○ ○
6.	Es gibt Situationen, in denen ich eine Spracheingabe bevorzugen würde.	○ ○ ○ ○ ○ ○ ○
7.	Die Spracheingabe funktioniert zuverlässig.	○ ○ ○ ○ ○ ○ ○
8.	Die Spracheingabe ist intuitiv.	○ ○ ○ ○ ○ ○ ○
9.	Die Eingabe via Touchscreen dauert mir zu lange.	○ ○ ○ ○ ○ ○ ○
10.	Die Informationen aus sozialen Netzwerken helfen mir bei der Entscheidungsfindung.	○ ○ ○ ○ ○ ○ ○
11.	Ich würde die Anwendung auch in der Zukunft verwenden.	○ ○ ○ ○ ○ ○ ○
		Sprache Touch
12.	Welche Modalität bevorzugen Sie?	○ ○ ○ ○ ○ ○ ○

Was war positiv:

Was kann verbessert werden:

Sonstige Anmerkungen:

USE-Fragebogen

Usefulness

		strongly agree - strongly disagree
1.	It helps me be more effective.	○ ○ ○ ○ ○ ○ ○
2.	It helps me be more productive.	○ ○ ○ ○ ○ ○ ○
3.	It is useful.	○ ○ ○ ○ ○ ○ ○
4.	It gives me more control over the activities in my life.	○ ○ ○ ○ ○ ○ ○
5.	It makes the things I want to accomplish easier to get done.	○ ○ ○ ○ ○ ○ ○
6.	It saves me time when I use it.	○ ○ ○ ○ ○ ○ ○
7.	It meets my needs.	○ ○ ○ ○ ○ ○ ○
8.	It does everything I would expect it to do.	○ ○ ○ ○ ○ ○ ○

Ease of Use

		strongly agree - strongly disagree
9.	It is easy to use.	○ ○ ○ ○ ○ ○ ○
10.	It is simple to use.	○ ○ ○ ○ ○ ○ ○
11.	It is user friendly.	○ ○ ○ ○ ○ ○ ○
12.	It requires the fewest steps possible to accomplish what I want to do with it.	○ ○ ○ ○ ○ ○ ○
13.	It is flexible.	○ ○ ○ ○ ○ ○ ○
14.	Using it is effortless.	○ ○ ○ ○ ○ ○ ○
15.	I can use it without written instructions.	○ ○ ○ ○ ○ ○ ○
16.	I don't notice any inconsistencies as I use it.	○ ○ ○ ○ ○ ○ ○
17.	Both occasional and regular users would like it.	○ ○ ○ ○ ○ ○ ○
18.	I can recover from mistakes quickly and easily.	○ ○ ○ ○ ○ ○ ○
19.	I can use it successfully every time.	○ ○ ○ ○ ○ ○ ○

Ease of Learning

		strongly agree - strongly disagree
20.	I learned to use it quickly.	○ ○ ○ ○ ○ ○ ○
21.	I easily remember how to use it.	○ ○ ○ ○ ○ ○ ○
22.	It is easy to learn to use it.	○ ○ ○ ○ ○ ○ ○
23.	I quickly became skilful with it.	○ ○ ○ ○ ○ ○ ○

Satisfaction

		strongly agree - strongly disagree
24.	I am satisfied with it.	○ ○ ○ ○ ○ ○ ○
25.	I would recommend it to a friend.	○ ○ ○ ○ ○ ○ ○
26.	It is fun to use.	○ ○ ○ ○ ○ ○ ○
27.	It works the way I want it to work.	○ ○ ○ ○ ○ ○ ○
28.	It is wonderful.	○ ○ ○ ○ ○ ○ ○
29.	I feel I need to have it.	○ ○ ○ ○ ○ ○ ○
30.	It is pleasant to use.	○ ○ ○ ○ ○ ○ ○

USE-Fragebogen (deutsch)

Nützlichkeit

1. Es hilft mir, effektiver zu sein.

2. Es hilft mir, produktiver zu sein.

3. Es ist nützlich.

4. Es gibt mir mehr Kontrolle über Aktivitäten in meinem Leben.

5. Es ermöglicht mir die Dinge, die ich erreichen möchte, einfacher zu erledigen.

6. Es spart mir Zeit, wenn ich es benutze.

7. Es erfüllt meine Bedürfnisse.

8. Es macht alles, was ich von ihm erwarte.

Einfach zu benutzen

9. Es ist einfach zu benutzen.

10. Es ist schlicht (nicht komplex).

11. Es ist benutzerfreundlich.

12. Es erfordert die geringste Anzahl an Schritten, um das zu erreichen, was ich machen möchte.

13. Es ist flexibel.

14. Die Anwendung lässt sich mühelos benutzen.

15. Ich kann es ohne schriftliche Anleitung benutzen.

16. Ich bemerke keine Widersprüche beim Benutzen.

17. Es wurde sowohl gelegentlichen als auch häufigen Nutzern gefallen.

18. Ich kann Fehler schnell und einfach korrigieren.

19. Ich kann es jedes Mal erfolgreich benutzen.

Leichtigkeit des Lernens

20. Ich habe schnell gelernt, es zu benutzen.

21. Ich kann mich leicht erinnern, wie es benutzt wird.

22. Es ist einfach zu erlernen.

23. Ich habe schnell gelernt, geschickt damit umzugehen.

Zufriedenheit

24. Ich bin damit zufrieden.

25. Ich würde es Freunden empfehlen.

26. Es macht Spaß zu benutzen.

27. Es funktioniert so, wie ich es möchte.

28. Es ist wunderbar.

29. Ich denke, dass ich diese Anwendung haben muss.

30. Es ist angenehm zu benutzen.

A.3 Ergebnisse des eigenen Fragebogens

Nr.	1	2	3	4	5	6	7	8	9	10	11	12	Ø	$\tilde{x}$	min	max
a)	j	n	j	j	n	j	j	j	n	j	n	n	-	j	-	-
b)	5	4	5	2	2	2	5	5	1	4	5	1	3,42	4	1	5
c)	1	1	3	2	3	1	1	1	3	1	3	6	2,17	1,5	1	6
1.	1	2	1	1	2	4	4	1	2	1	2	1	1,83	1,5	1	4
2.	3	3	3	3	2	2	3	5	1	5	5	7	3,5	3	1	7
3.	4	3	2	1	4	6	1	3	3	7	4	1	3,25	3	1	7
4.	1	1	2	1	1	1	1	1	5	1	2	1	1,5	1	1	5
5.	4	1	2	2	2	1	1	1	7	1	2	2	2,17	2	1	7
6.	1	1	1	2	1	1	1	1	3	1	4	1	1,5	1	1	4
7.	2	1	3	2	3	6	3	1	7	1	3	3	2,92	3	1	7
8.	3	1	1	2	3	1	3	1	2	1	2	1	1,75	1,5	1	3
9.	4	3	4	1	2	3	1	1	3	2	5	3	2,67	3	1	5
10.	2	2	1	2	2	1	2	3	2	1	7	2	2,25	2	1	7
11.	1	1	1	2	4	1	1	1	1	2	4	1	1,67	1	1	4
12.	1	3	5	1	2	1	2	1	7	1	2	2	2,33	2	1	7
Gr.	A	B	C	D	A	B	C	D	A	B	C	D				

A.4 Ergebnisse des USE-Fragebogens

Nr.	1	2	3	4	5	6	7	8	9	10	11	12	Ø	$\tilde{x}$	min	max
1.	1	3	2	1	3	1	2	3	2	2	4	1	2,08	2	1	4
2.	2	3	2	2	3	4	6		3	2	4	3	3,09	3	2	6
3.	1	3	1	1	3	1	2	2	1	2	2	1	1,67	1,5	1	3
4.	6	5	3	3		7	4	4	3	2	7	2	4,18	4	2	7
5.	5	4	3	1	3	2	2	1	3	2	4	1	2,58	2,5	1	5
6.	2	3	2	1	3	4	2	1	2	1	7	3	2,58	2	1	7
7.	4	4	2	2	4	4	3	1	2	2	5	1	2,83	2,5	1	5
8.	1	4	3	3	4	2	3	5	3	3	5	2	3,17	3	1	5
9.	2	2	1	1	3	2	1	1	2	1	2	1	1,58	1,5	1	3
10.	2	2	2	1	3	2	1	1	2	1	3	2	1,83	2	1	3
11.	1	2	1	2	4	1	2	2	1	1	2	2	1,75	2	1	4
12.	2	3	2	2	2	5	2	4	1	2	4	1	2,5	2	1	5
13.	2	2	3	3	4	4	5	1	1	3	4	1	2,75	3	1	5
14.	2	3	3	2	5	2	2	2	3	1	4	3	2,67	2,5	1	5
15.	2	1	1	1	3	1	4	1	1	1	4	1	1,75	1	1	4
16.	1	3	1	2	5	7	1	1	1	1	3	3	2,42	1,5	1	7
17.	2	2	4	1	3	1	2	1	1	3	4	2	2,17	2	1	4
18.	2	4	3	1	4	7	2	1	3	2	3	3	2,92	3	1	7
19.	1	4	2	1	5	2	2	3	4	3	4	3	2,83	3	1	5
20.	2	1	1	1	3	1	1	1	3	1	3	1	1,58	1	1	3
21.	2	1	1	1	2	1	1	1	1	1	4	1	1,42	1	1	4
22.	1	1	1	1	2	1	1	1	1	1	3	1	1,25	1	1	3
23.	3	1	1	3	2	3	1	1	2	1	3	1	1,83	1,5	1	3
24.	1	3	2	1	2	2	1	2	2	2	3	1	1,83	2	1	3
25.	2	3	1	1	4	1	1	1	2	2	3	1	1,83	1,5	1	4
26.	1	2	1	2	2	4	1	3	1	2	2	1	1,83	2	1	4
27.	1	4	2	2	3	3	2	2	1	2	4	3	2,42	2	1	4
28.	4	4	2	4	4	2	3	1	2	3	4	1	2,83	3	1	4
29.	3	4	3	3	6	7	3	2	3	3	7	1	3,75	3	1	7
30.	1	3	2	2	3	4	2	1	2	2	2	1	2,08	2	1	4

A.5 Histogramme - eigener Fragebogen

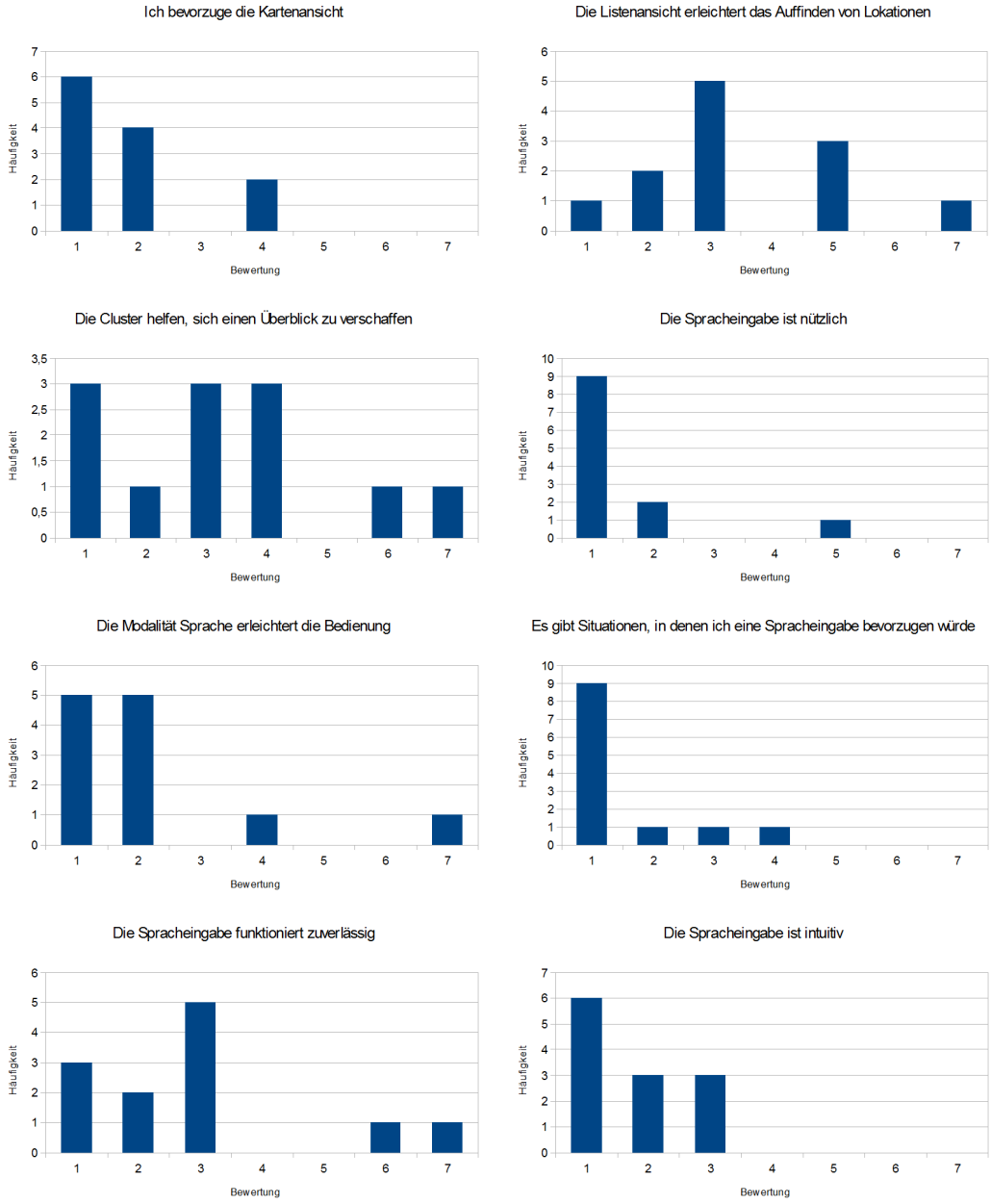

Abbildung A.1: Fragen 1 bis 8

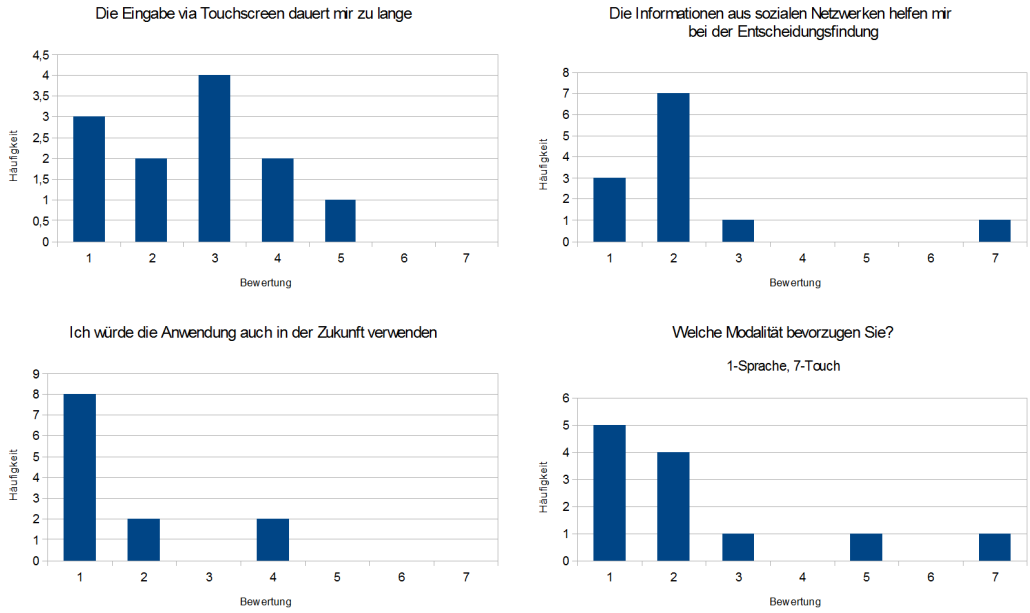

Abbildung A.2: Fragen 9 bis 12

Literaturverzeichnis

[Ber12] Bertram Sändig: *Entwicklung und Evaluierung von Clustering-Verfahren für Points of Interest verschiedener thematischer Kategorien.* 2012

[BIT12] BITKOM: *Presseinformation - Jeder Dritte hat ein Smartphone.* `http://www.bitkom.org/de/presse/8477_71854.aspx`. Version: 2012. – [Online; Stand 11. September 2012]

[CAOO09] Cherubini, Mauro ; Anguera, Xavier ; Oliver, Nuria ; Oliveira, Rodrigo de: Text versus speech: a comparison of tagging input modalities for camera phones. In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services.* New York, NY, USA : ACM, 2009 (MobileHCI '09). – ISBN 978–1–60558–281–8, 1:1–1:10

[Gru09] Gruenstein, Alexander: *Toward Widely-Available and Usable Multimodal Conversational Interfaces*, DSpace@MIT: Massachusetts Institute of Technology, Diss., 2009

[GS07] Gruenstein, Alexander ; Seneff, Stephanie: Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms. In: *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue*, 2007

[GSW06] Gruenstein, Er ; Seneff, Stephanie ; Wang, Chao: Scalable and portable web-based multimodal dialogue interaction with geographical databases. In: *in Proc. of InterSpeech*, 2006

[ITU11] ITU: *SERIES P: TERMINALS AND SUBJECTIVE AND OBJECTIVE ASSESSMENT METHODS, Parameters describing the interaction with multimodal dialogue systems.* `http://www.itu.int/rec/T-REC-P.Sup25-201101-I/en`. Version: 2011

[JBV+02] Johnston, Michael ; Bangalore, Srinivas ; Vasireddy, Gunaranjan ; Stent, Amanda ; Ehlen, Patrick ; Walker, Marilyn ; Whittaker, Steve ; Maloor, Preetam: MATCH: an architecture for multimodal dialogue

systems. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA : Association for Computational Linguistics, 2002 (ACL '02), 376–383

[JE10] Johnston, M. ; Ehlen, P.: Speak4IT: Multimodal interaction in the wild. In: *Spoken Language Technology Workshop (SLT), 2010 IEEE*, 2010, S. 159–160

[KWWM10] Kühnel, Christine ; Westermann, Tilo ; Weiss, Benjamin ; Möller, Sebastian: Evaluating multimodal systems: a comparison of established questionnaires and interaction parameters. In: *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. New York, NY, USA : ACM, 2010 (NordiCHI '10). – ISBN 978–1–60558–934–3, 286–294

[Liu10] Liu, Sean (Sean Y.: *Multimodal speech interfaces for map-based applications*, DSpace@MIT: Massachusetts Institute of Technology, Diplomarbeit, 2010

[MHJ04] Merdes, Matthias ; Häussler, Jochen ; Jöst, Matthias: 'SlidingMap': introducing and evaluating a new modality for map interaction. In: *Proceedings of the 6th international conference on Multimodal interfaces*. New York, NY, USA : ACM, 2004 (ICMI '04). – ISBN 1–58113–995–0, 325–326

[Nua10] Nuance: *Dragon Mobile SDK Reference - Speech Kit Architecture*. `http://dragonmobile.nuancemobiledeveloper.com/public/Help/DragonMobileSDKReference_iOS/SpeechKit_Guide/Basics.html`. Version: 2010. – [Online; Stand 2. September 2012]

[RBE+05] Reithinger, Norbert ; Bergweiler, Simon ; Engel, Ralf ; Herzog, Gerd ; Pfleger, Norbert ; Romanelli, Massimo ; Sonntag, Daniel: A look under the hood: design and development of the first SmartWeb system demonstrator. In: *Proceedings of the 7th international conference on Multimodal interfaces*. New York, NY, USA : ACM, 2005 (ICMI '05). – ISBN 1–59593–028–0, 159–166

[SHL+98] Seneff, Stephanie ; Hurley, Ed ; Lau, Raymond ; Pao, Christine ; Schmid, Philipp ; Zue, Victor: Galaxy-II: A Reference Architecture For Conversational System Development. In: *in Proc. ICSLP*, 1998, S. 931–934

[SS11] Seifert, Inessa ; Sändig, Bertram: Clustering and Regionalization for Mobile Applicationsn. In: *Proceedings of Workshop: Visibility in Information Spaces and in Geographic Environments at KI 2011*, 2011

[SWHC04] Seneff, Stephanie ; Wang, Chao ; Hetherington, I. L. ; Chung, Grace: A dynamic vocabulary spoken dialogue interface. In: *INTERSPEECH'04*, 2004

[TMH+09] Turunen, Markku ; Melto, Aleksi ; Hella, Juho ; Heimonen, Tomi ; Hakulinen, Jaakko ; Mäkinen, Erno ; Laivo, Tuuli ; Soronen, Hannu: User expectations and user experience with different modalities in a mobile phone controlled home entertainment system. In: *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services*. New York, NY, USA : ACM, 2009 (MobileHCI '09). – ISBN 978–1–60558–281–8, 31:1–31:4

[WES+09] Wechsung, Ina ; Engelbrecht, Klaus-Peter ; Schaffer, Stefan ; Seebode, Julia ; Metze, Florian ; Möller, Sebastian: Usability-Evaluation multimodaler Schnittstellen: Ist das Ganze die Summe seiner Teile? In: Wandke, Hartmut (Hrsg.) ; Kain, Saskia (Hrsg.) ; Struve, Doreen (Hrsg.): *Mensch & Computer 2009: Grenzenlos frei!?* München : Oldenbourg Verlag, 2009, S. 495–498

[WLH02] Wahlster, Wolfgang ; Lieberman, Henry ; Hendler, James: *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. The MIT Press, 2002. – ISBN 0262062321

[WMWK11] Weiss, Benjamin ; Möller, Sebastian ; Wechsung, Ina ; Kühnel, Christine: Quality of Experiencing Multi-Modal Interaction. Version: 2011. `http://dx.doi.org/10.1007/978-1-4419-7934-6_9`. In: Minker, Wolfgang (Hrsg.) ; Lee, Gary G. (Hrsg.) ; Nakamura, Satoshi (Hrsg.) ; Mariani, Joseph (Hrsg.): *Spoken Dialogue Systems Technology and Design*. Springer New York, 2011. – ISBN 978–1–4419–7934–6, 213-230

Abbildungsverzeichnis

Tabellenverzeichnis