Entwicklung eines Dialogagenten für dialogbasierte Programmierung

Dokumentenart:	Exposé für eine Masterarbeit
Autor:	Mario Schlereth
Matrikel-Nr.:	1450302
Studiengang:	Informationswirtschaft
Betreuer:	Sebastian Weigelt
Datum:	15. September 2016

1 Motivation

Heutzutage verwenden viele Menschen ihr Smartphone um sich Fragen mit Hilfe von Sprachassistenten beantworten zu lassen. Bereits jeder zweite Deutsche hat hierfür auf Systeme wie Siri, Google Now oder Cortana zurückgegriffen[1]. In einer Gesellschaft, in der die Menschen in zunehmender Symbiose mit ihren technischen Geräten leben, ist es naheliegend sich bei Unklarheiten von seinen elektronischen Begleitern helfen zu lassen. Allerdings können die technischen Assistenten viele Aufgaben noch nicht erledigen. Vor allem individuelle für einzelne Nutzer relevante Herausforderungen werden im Kontext des Massenmarktes vernachlässigt. Diese Einschränkung kann durch dialogbasierte Programmierung überwunden werden. Hierbei gibt der Mensch dem System Handlungsanweisungen in Form von gesprochener Sprache. In diesem Zusammenhang kann allerdings auch der Fall auftreten, dass die Maschine bei der Befehlsausführung in Situationen beziehungsweise Zustände gelangt, in denen sie nicht mehr weiter weiß. Dies führt dazu, dass das System dem Benutzer eine Frage stellen muss, um dessen Intention zu erfragen oder zu verifizieren. Dabei kann zwischen den zwei folgenden Fällen unterschieden werden.
Im ersten Fall handelt es sich um Rückfragen zu unverständlichen oder falsch verstandenen Benutzereingaben. Ein Beispiel wäre, wenn eine Person einem Haushaltsroboter die Anweisung gibt: „Put the orange juice into the frigde.“ und das System versteht: „Put the orange juice into the French.“. Da der Roboter mit dieser Information nichts anfangen kann, würde die Software

[1] Statista 2016, Nutzung von Sprachassistenten in Deutschland vgl. https://de.statista.com/infografik/4686/nutzung-von-sprachassistenten-in-deutschland/ abgerufen am 02.09.2016

die Gegenfrage erzeugen: „Where shall I put the orange juice?“. In diesem Kontext entsteht die Frage des Systems somit als direkte Reaktion auf eine Nutzeranweisung.
Der zweite Fall betrachtet Situationen in denen die Software selbstständig Entscheidungen treffen soll, wie zum Beispiel bei Software-Agenten oder im Bereich der künstlichen Intelligenz. Im Gegensatz zu Rückfragen wird, in diesem Zusammenhang, das Bedürfnis des Systems eine Frage zu stellen nicht unmittelbar durch eine menschliche Aktion ausgelöst, vielmehr muss die Software selbst erkennen, wann zusätzliche, nicht selbst genierbare, Informationen erfordlich sind und dann den Dialog initieren. Ein Beispiel für eine solche Situation wäre, wenn ein Haushaltsroboter den Saft aus dem Kühlschrank holen soll und im Kühlschrank dann Orangensaft sowie Apfelsaft stehen. In diesem Fall muss der Roboter fragen, welcher Saft denn gemeint ist.

2 Dialogsysteme

Dialogsysteme bestehen im Wesentlichen aus vier Modulen, Spracherkenner, Sprachsynthetisierer, Dialogmanager und einem Modul zum Sprachverständnis[MRGRD14, S. 3]. Abhängig von der Implementierung können einzelne Module auch zusammengefasst werden, zum Beispiel die Spracherkennung und das Sprachverständnismodul[TD11, S. 171ff.].
Für das Verstehen gesprochener Sprache existieren zwei unterschiedliche Vorgehensweisen. Diese sind der regelbasierte Ansatz sowie der statistische Ansatz.
Bei ersterem wird die Semantik des Gesprochenen anhand wissensbasierter Regeln extrahiert. Hierfür werden kontextfreie Grammatiken definiert und mit Hilfe der darin enthaltenen Beschränkungen wird anschließend die semantische Repräsentation des Gesprochenen konstruiert. Schwierigkeiten dieser Vorgehensweise sind vor allem, dass die Grammatikentwicklung ein fehlerbehafteter Prozess ist und mehrere Iterationen für die Feinabstimmung benötigt. Da die Grammatikerstellung in der Regel von Domänenexperten vorgenommen werden muss, skaliert dieser Ansatz sehr schlecht.[TD11, S. 52ff.] Darüber hinaus ist es notwendig, dass die verwendete Ontologie, für die eingesetzte Domäne, vollständig ist, damit das System erfolgsversprechende Resultate liefert[MRGRD14, S. 10].
Der statistische Ansatz verwendet Methoden des unüberwachten Lernens, um von neuen Daten, zum Beispiel annotierten Sätzen, zu lernen. Bei dieser Vorgehensweise ist ein große Menge annotierter Daten notwendig um sinnvolle Resultate zu erzielen. Dies erschwert die Verwendung statistischer Verfahren in offenen, kaum eingeschränkten Domänen, speziell in der frühen Phase der Systementwicklung. Ein Vorteil dieser Vorgehensweise ist, dass auf vordefinierte Grammatiken verzichtet werden kann. Zudem sind statistische Ansätze sehr robust gegenüber Rauschen und fehlerhaften Eingaben.[TD11,

S. 55, 88]
Allerdings können die beiden Ansätze auch miteinander verknüpft werden, um die jeweiligen schwächen des Anderen abzumildern. Diese Vorgehensweise wurde bei der Erstellung von Siri angewandt.[MRGRD14, S. 12f.]
In Anbetracht der Existenz diverser Dialogsysteme stellt sich nun die Frage, weshalb in dieser Masterarbeit ein neuer Dialogagent für PARSE entwickelt werden sollte. Ein wesentlicher Grund ist, dass viele Dialogsysteme auf geschriebener Sprache basieren und somit die Probleme, die bei einer verbalen Eingabe entstehen, nicht berücksichtigen. Des Weiteren sind die meisten etablierten Dialogsysteme für gesprochene Sprache proprietäre Lösungen und scheiden deswegen aus. Der dritte entscheidende Aspekt ist, dass die betrachteten frei verfügbaren, sprachbasierten Dialogsysteme alle mindestens ein Ausschlusskriterium besitzen. Sei es eine mangelhafte oder fehlende Dokumentation[2], dass die Benutzung auf vordefinierte Schnittstellen begrenzt ist[3] oder schlicht und ergreifend, dass die versprochene Bedienbarkeit mit Hilfe gesprochener Sprache noch gar nicht implementiert ist[4].

3 Dialogagent für PARSE

Bei PARSE (Programming ARchitecture for Spoken Explanations) handelt es sich um ein agentenbasiertes System, welches die Programmierung mit Hilfe natürlicher Sprache ermöglichen soll. PARSE arbeitet wissensbasiert, um ein tiefer gehendes Sprachverständnis zu erreichen. Darüber hinaus soll Domänenunabhängigkeit durch den einfachen Austausch von Ontologien, welche das jeweilige Domänenwissen enthalten, erzielt werden. Die verarbeiteten verbalen Eingaben sowie der geteilte Datenspeicher werden bei diesem System durch einen Graphen repräsentiert.[WT15]
Um PARSE in die Lage zu versetzen selbstständig Fragen an den Nutzer zu stellen, ist die Erstellung eines Dialogagenten erforderlich. Dieser soll Unklarheiten innerhalb des Systems erkennen und beseitigen können. Eine Unklarheit bedeutet in diesem Kontext ein Zustand in dem das System nicht mehr in der Lage ist, eine vorgegebene Aufgabe ohne zusätzliche, nicht selbst generierbare, Informationen zu erfüllen. Ein derartiger Zustand kann zum Beispiel auftreten, wenn ein Roboter nach einer initialen Anweisung und einer Reihe von Handlungsschritten feststellt, dass ihm zur Ausführung des nächsten Schritts relevante Informationen fehlen. In einer solchen Situation würde das System die Arbeit komplett einstellen. Daher soll die benötigte

[2] OwlSpeak https://sourceforge.net/projects/owlspeak/files/?source=navbar abgerufen am 13.09.2016

[3] Ariadne Spoken Dialogue System http://www.opendialog.org/about_system.html abgerufen am 13.09.2016

[4] Olympus http://wiki.speech.cs.cmu.edu/olympus/index.php/Olympus abgerufen am 13.09.2016, Tutorial 1 http://wiki.speech.cs.cmu.edu/olympus/index.php/Tutorial_1 abgerufen am 13.09.2016

Information durch eine Frage an den Menschen beschafft werden, um eine weitere effektive Arbeitsweise zu gewährleisten.

Damit auftretende Unklarheiten mit Hilfe von Fragen an den Menschen beseitigt werden können, muss der Dialogagent die folgenden drei Anforderungen erfüllen:

1. Der Agent muss erkennen, dass der aktuelle Zustand des Graphen eine Situation darstellt, die zusätzliche Informationen nötig macht. Eine Möglichkeit hierfür ist die permanente Überwachung des Graphen um Muster zu erkennen, die eine dauerhafte Unklarheit aufwerfen. Alternativ kann der Agent auch durch andere Agenten aufgerufen werden, wenn diese einen Informationsbedarf festgestellt haben.
   Eine besondere Herausforderung ist in diesem Kontext die Generierung einer sinnvollen Frage, welche den Menschen in die Lage versetzt die gewünschten Informationen bereitzustellen. Die Schwierigkeit besteht hierbei darin, zu ermitteln, weshalb sich der Graph in einem statischen Zustand befindet und dieses Wissen in die Frage zu integrieren. Ursachen für den Stillstand des Graphen sind unter anderem Mehrdeutigkeiten oder zu niedrige Konfidenzen. Es können in diesem Zusammenhang drei Arten von Fragestellungen unterschieden werden. Die einfachste Form sind ja/nein-Fragen. Als zweite Möglichkeit bieten sich Fragen an, die dem Menschen Alternativen vorschlagen, zum Beispiel bei einem Haushaltsroboter: „Soll ich den Fußboden kehren, staubsaugen oder wischen?“. Die dritte Kategorie sind offene Fragen, zum Beispiel: „Wie mache ich den Staubsauger an?“
   In diesem Kontext ist abschließend auch noch die Verbalisierung der Frage erforderlich. Hierfür greift der Agent auf vorgefertigte Sprachsynthetisierer, wie zum Beispiel FreeTTS[WLK02] oder MaryTTS[ST03] zurück.

2. Das Verstehen der Antwort des Menschen stellt die zweite wesentliche Herausforderung des Agenten dar. Diese Aufgabe lässt sich in zwei Teilaspekte untergliedern.
   Zunächst müssen die generellen Probleme des Verstehens gesprochener Sprache durch Maschinen gemeistert werden. Im Besonderen sind dies die Verletzung grammatikalischer Regeln, die nicht fließende, das heißt mit nicht konstanter Geschwindigkeit ablaufende Aussprache, Wiederholungen, Korrekturen des Gesprochenen sowie der Neubeginn mancher Wörter, zum Beispiel durch stottern. Darüber hinaus sind in diesem Zusammenhang auch Fehler der automatischen Spracherkenner, welche die Bedeutung der Antwort verändern, zu berücksichtigen.[TD11, S. 46, 148]
   Im zweiten Schritt geht es darum, den Sinngehalt der Antwort zu extrahieren, das heißt beantwortet die Aussage auch die gestellte Frage.

Ist dies nicht der Fall, ist die Generierung einer weiteren Frage erforderlich. Dieser Prozess kann zu einer Endlosschleife aus Fragestellungen und ungenügenden Antworten führen. Daher muss ein Abbruchkriterium definiert werden, falls die Frage beziehungsweise verschiedene Fragevariationen nach einer gewissen Zeit oder Anzahl an Durchläufen nicht die gewünschte Information liefern.

3. Abschließend muss die gewonnene Information noch in den Graphen integriert werden, um die Unklarheit zu beseitigen. Hierbei sind verschiedene Vorgehensweisen möglich.
   Die Information wird dem System als neue Eingabe übergeben was zur Generierung eines neuen Graphen führt. Dies birgt allerdings den Nachteil, dass durch Wechselwirkungen mit anderen Agenten sowie durch Seiteneffekte bei der Eingabe-Integration die Unklarheit nicht oder nicht korrekt beseitigt wird.
   Alternativ kann der Ausgangsgraph aufgrund der Antwort so verändert werden, dass die Unklarheit beseitigt ist. Der Vorteil dieser Vorgehensweise ist die korrekte Integration der neuen Information in das System. Eine valide Antwort kann in diesem Zusammenhang die Beseitigung der Unklarheit sein oder aber die Erkenntnis, dass die Unklarheit nicht durch Interaktion mit einem Menschen gelöst werden kann.

4 Zielsetzung

Das Ziel dieser Arbeit ist die Erstellung eines Agenten welcher selbstständig in der Lage ist bei Unklarheiten Fragen aus einem Graphen zu generieren und zu verbalisieren. In diesem Zusammenhang muss der erstellte Agent auch die Antwort verstehen und auf ihren Sinngehalt zur Beseitigung der Unklarheit überprüfen können. Ist Letzteres nicht erfüllt, ist die Generierung weiterer Rückfragen erforderlich, bis die benötigten Information vorliegen oder der Agent die Befragung mangels Erkenntnisgewinn abbricht. Im Falle einer zufriedenstellenden Antwort durch den Menschen, muss die gewonnene Information noch in den Graphen integriert werden, um die Unklarheit abschließend zu beseitigen.

5 Evaluation

Für die Bewertung des Dialogagenten wird zunächst eine intrinsische Evaluation der drei in 3 vorgestellten Anforderungen durchgeführt. Anschließend findet eine extrinsische Evaluation in Form einer Nutzerstudie statt.
Zur Bewertung der Frageextraktion aus einem Graphen können zwei unterschiedliche Ansätze gewählt werden. Da sich die Fragen aus den Unklarheiten ableiten, wird bei der ersten Methode überprüft, ob die im Graph enthal-

tenen Unklarheiten erkannt wurden und auch nur genau die. Hierfür eignen sich die Maße Präzision und Ausbeute sowie deren Kombination das F-Maß. Der zweite Ansatz untersucht, ob die gestellte Frage auch dem tatsächlichen Sinn der Unklarheit entspricht. Dies wird durch einen Vergleich mit Musterlösungsfragen überprüft.
Da das Verstehen der Antwort eines Menschen eine Teilmenge des Verstehens gesprochener Sprache ist, können hierbei die gleichen Evaluationsmetriken wie für das Verständnis gesprochener Sprache verwendet werden. Ein geeignetes Maß ist in diesem Zusammenhang die Konzeptfehlerrate[HLRN08]. Als Beispiel für ein Konzept, kann die ja-Kategorie einer ja/nein-Fragen angesehen werden.
Um die Integration einer Antwort in das System zu evaluieren, wird überprüft, ob die zugehörige Unklarheit im Graphen tatsächlich aufgelöst wurde. Hierfür eignen sich ebenso die Maße Präzision, Ausbeute und F-Maß.
Abschließend wird die Funktionsweise des Dialogagenten mit Hilfe einer Nutzerstudie, Ende-zu-Ende, bewertet. Das Ziel ist herauszufinden, wie effektiv die Interaktion mit einem Menschen das System bei der Aufgabenerfüllung unterstützt hat. Dies kann anhand der Anzahl der gestellten Fragen sowie des Grades der Aufgabenerfüllung, das heißt der Anzahl der vom System durchgeführten Schritte um die Aufgabe zu lösen, bewertet werden.

Literatur

[HLRN08] Hahn, Stefan ; Lehnen, Patrick ; Raymond, Christian ; Ney, Hermann: A Comparison of Various Methods for Concept Tagging for Spoken Language Understanding. In: *Proceedings of LREC* 1 (2008), Nr. 2, S. 2–5

[MRGRD14] Mariani, Joseph ; Rosset, Sophie ; Garnier-Rizet, Martine ; Devillers, Laurence: *Natural Interaction with Robots, Knowbots and Smartphones*. New York, New York, USA : Springer, 2014. – 397 S. http://dx.doi.org/10.1007/978-1-4614-8280-2. http://dx.doi.org/10.1007/978-1-4614-8280-2. – ISBN 978–1–4614–8279–6

[ST03] Schröder, M. ; Trouvain, J.: The German Text-to-Speech Synthesis System MARY: A Tool for Research, Development and Teaching. In: *International Journal of Speech Technology* 6 (2003), S. 365–377. http://dx.doi.org/10.1023/A:1025708916924. – DOI 10.1023/A:1025708916924. – ISSN 1381–2416

[TD11] Tur, Gokhan ; De Mori, Renato: *Spoken Language Understanding : Systems for Extracting Semantic Information from*

Speech. Chichester : John Wiley & Sons, 2011. – 450 S. – ISBN 978–0–470–68824–3

[WLK02] Walker, Willie ; Lamere, Paul ; Kwok, Philip: FreeTTS: a performance case study. (2002)

[WT15] Weigelt, Sebastian ; Tichy, Walter F.: Poster: ProNat: An Agent-Based System Design for Programming in Spoken Natural Language. In: *Proceedings - International Conference on Software Engineering* 2 (2015), S. 819–820. `http://dx.doi.org/10.1109/ICSE.2015.264`. – DOI 10.1109/ICSE.2015.264. – ISBN 9781479919345