

# Prototyp einer natürlichsprachlichen Schnittstelle für Tabellenkalkulation

Alexander Wachtel, Sebastian Weigelt, Philipp Voigt und Walter F. Tichy

Karlsruher Institut für Technologie, Fakultät für Informatik  
Am Fasanengarten 5, 76131 Karlsruhe  
{alexander.wachtel | sebastian.weigelt | walter.tichy}@kit.edu  
philipp.voigt@student.kit.edu

**Abstract:** Nicht zuletzt zeigen uns Siri und Google Now die zunehmende Wichtigkeit sprachlicher Nutzerschnittstellen. Der vorliegende Artikel beschreibt die Konzeption und Umsetzung einer natürlichsprachlichen Schnittstelle für Tabellenkalkulationen. Im Zuge einer Pilotstudie wurden die zur Realisierung notwendigen Anforderungen erhoben. Basierend darauf wurde der Prototyp JustLingo<sup>1</sup> entwickelt, der textuelle Rechenaufgaben, wie *Please add 4 to A1. Then multiply it by 7.*, mithilfe einer musterbasierten Syntax- und Semantikanalyse verarbeitet. Eine Sprachsteuerung befindet sich bereits in der Entwicklung. JustLingo erkennt Rückbezüge sowie fehlende oder mehrdeutige Informationen, fragt entsprechend nach und wandelt die Eingabe zu 79% erfolgreich in die Excel-Formelsprache um.

## 1 Einführung

Seit dem Beginn der Entwicklung digitaler Rechnersysteme spielen Nutzerschnittstellen eine entscheidende Rolle bei der Akzeptanz von Softwaresystemen. Angefangen bei Kommandozeileingaben über graphische Benutzeroberflächen, sind es insbesondere Systeme mit Spracheingabe, die in Anbetracht der steigenden Mobilitätsanforderung eine immer größere Bedeutung gewinnen [Ort14]. Sprachsysteme, wie Siri oder Google Now, sind bereits fester Bestandteil des alltäglichen Lebens. Wir erwarten daher, dass Software-Entwickler zunehmend natürlichsprachliche Schnittstellen entwickeln müssen. Die vorgestellte Arbeit soll helfen, im Voraus Anforderungen an derartige Systeme zu formulieren und Auswirkungen auf den Entwicklungsprozess abzuleiten. Anhand der prototypischen Entwicklung der Excel-Erweiterung JustLingo wurden Erfahrungen in einem technischen, jedoch nutzernahen Umfeld gesammelt. Der Prototyp ermöglicht es den Nutzern, Tabellenkalkulation natürlichsprachlich durchzuführen. Die folgenden Kapitel beschreiben zunächst eine Pilotstudie, die durchgeführt wurde, um die Anforderungen an das System zu erheben. Anschließend wird der konzeptionelle Ansatz und die bisherige prototypische Umsetzung vorgestellt und evaluiert, bevor verwandte Arbeiten präsentiert werden. Abschließend werden die Erkenntnisse zusammengefasst sowie laufende und zukünftige Arbeiten umrissen.

---

<sup>1</sup><http://ps.ipd.kit.edu/JustLingo.php>

## 2 Pilotstudie

Den Ausgangspunkt für die Entwicklung von JustLingo bildete eine von uns durchgeführte Pilotstudie<sup>2</sup>, die das Benutzerverhalten im Umgang mit einem fiktiven natürlichsprachlichen Assistenten für Excel dokumentiert. 57 Probanden sollten textuelle Lösungen zu 35 Aufgaben in alltäglichem Englisch formulieren<sup>3</sup>. Laut Selbsteinschätzung entsprach der Excel-Kennntnisstand der Teilnehmer zu 16% dem eines Anfängers, zu 72% eines Fortgeschrittenen und zu 18% eines Experten. Die Erfahrung im Umgang mit Excel lag zwischen einem und 25 Jahren, bei einem Durchschnitt von 9,5 Jahren.

Die Einschätzung wird durch die angebotenen Lösungen der Probanden untermauert: Anfänger formulierten häufig lediglich die Aufgaben in eine Frage um (*I need to have a cell that averages the students grades, how would I do that?*). Keine ihrer angebotenen Lösungsbeschreibungen reichte aus, um das Problem direkt zu lösen. Fortgeschrittene Benutzer stellten dagegen mehr Information für das System zur Verfügung, indem sie Teilschritte zur Bearbeitung der Aufgabe angaben. Meist fehlten jedoch Informationen, wie die Angabe der Operanden, um das Problem vollständig zu lösen (*I need to calculate the average of a row of values into a cell.*). Nur in 20% der Fälle waren diese ausreichend. Beschreibungen von Probanden, die sich selber als Experten einstufen, waren hingegen meist hinreichend genau, um das Problem zu lösen (*Could you insert the average value of Daily grades 1, Daily grades 2 and Final exam into Average?*). Nur 13% ihrer Beschreibungen eigneten sich nicht zur vollständigen Lösung des Problems.

Diese Ergebnisse zeigen die Notwendigkeit eines interaktiven Dialogsystems, welches insbesondere unerfahrene Nutzer sukzessive bei der Lösungsfindung unterstützt. Weitere Anforderungen an JustLingo, die mithilfe der Pilotstudie formuliert werden konnten, sind: (a) Auflösung der Rückbezüge auf bereits getätigte Nutzereingaben, wie *Please add 4 to A1. Then multiply it by 7.* (hierbei stellt das *it* einen Rückbezug auf das Ergebnis der ersten Eingabe dar), sowie (b) Auflösung der Mehrdeutigkeiten in Benutzereingaben. Zum Beispiel kann die natürlichsprachliche Eingabe *Divide A1 by 3 multiplied by A3* entweder als  $(A1/(3*A3))$  oder  $((A1/3)*A3)$  interpretiert werden.

## 3 Prototypenentwicklung

Der Prototyp JustLingo erlaubt Nutzern die Verwendung von Tabellenkalkulationen in alltäglicher englischer Sprache. Momentan erfolgen Eingaben textuell. Die Architektur beinhaltet unter anderem eine interaktive Dialogkomponente und eine Syntax- und Semantikanalyse. Durch die Auflösung von Rückbezügen ist es möglich komplexe Ausdrücke Stück für Stück aufzubauen. JustLingo stellt Rückfragen bei fehlenden Informationen oder mehrdeutigen Eingaben. Der so aufgebaute Kontext unterstützt insbesondere unerfahrene Nutzer bei der Verwendung des Systems. Abbildung 1 zeigt den modularen Aufbau des Gesamtsystems, derzeit bestehend aus der textuellen Benutzerschnittstelle, sowie einer

---

<sup>2</sup><http://mphsoftware.de/justlingo/study>

<sup>3</sup>Die Aufgaben stammen aus dem Buch *Microsoft Excel 2013, Step by Step* von Curtis D. Frye.

Sprachanalyse- und einer Dialogmanagement-Einheit. Im Folgenden wird das Systemverhalten exemplarisch anhand der Eingabe „Please add 4 to A1. Then multiply it by 7.“ erläutert, die über die Benutzeroberfläche eingegeben wurde.

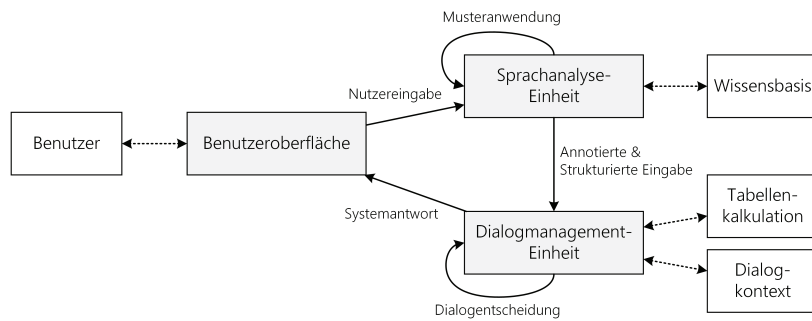


Abbildung 1: Überblick des Systems

Die Domäne der Tabellenkalkulation ist stark eingeschränkt. Daher konnte ein endlicher Wortschatz aufgebaut werden, der fast ausschließlich domänenspezifisches Vokabular enthält. Hierdurch ist es möglich, Schlüsselwörtern eine eindeutige Semantik zuzuordnen (z.B. bedeutet  $add_{Operator}$ , dass das System der Eingabe *add* die Bedeutung eines *Operators* zuweist) und Synonyme unmittelbar durch einen einheitlichen Vertreter zu ersetzen. Darüber hinaus kommen reguläre Ausdrücke zum Einsatz, um Zahlen und Positionsangaben wie Zellen, Zeilen und Spalten erkennen zu können (für Zelle *A1* wird zum Beispiel der reguläre Ausdruck  $(A - Z)(A - Z|\epsilon)(A - Z|\epsilon)(0 - 9)^+$  verwendet). Für das Gesamtbeispiel ergibt sich nach diesem Schritt folgende semantische Annotation:

$$Please_{Bitte} \textit{add}_{Operator} \ 4_{Zahl} \ \textit{to}_{Spez.} \ A1_{Zelle}.$$

$$Then_{Trenner} \ \textit{multiply}_{Operator} \ \textit{it}_{Referenz} \ \textit{by}_{Spez.} \ 7_{Zahl}.$$

Im Anschluss wird innerhalb der Sprachanalyse-Einheit eine kombinierte Syntax- und Semantikanalyse durchgeführt. Diese basiert auf dem Abgleich der Eingabe mit einer wiederum domänenspezifischen Sammlung von Mustern. Ein solches Muster beschreibt ein syntaktisches Schema durch eine Liste von Schlüsselwörtern und Platzhaltern und ordnet diesem eine eindeutige mathematische Operation zu. Die Platzhalter repräsentieren dabei die Operanden der Operation, mit deren Inhalt weitere Muster abgeglichen werden, wodurch eine Baumstruktur aus Operatoren und Operanden entsteht. Im Beispiel aus Abb. 2a kann das Muster *sum* mit der Eingabe abgeglichen werden, da die Schlüsselwörter *add* und *to* in der Eingabe vorkommen. Die Platzhalter erlauben die Operanden  $4_{Zahl}$  und  $A1_{Zelle}$ .

Die baumförmige abstrakte Repräsentation stellt an diesem Punkt bereits den gesamten mathematischen Ausdruck dar. Daher kann durch rekursive Operationen zum einen die mathematische Korrektheit bzgl. Anzahl und Typ der Operanden geprüft werden, zum anderen ist die Erzeugung eines Ausdrucks in Formelsprache unmittelbar durch Tiefensuche realisierbar. Die Baumstruktur wird an die Dialogmanagement-Einheit übergeben, die anhand der Gestalt die folgenden drei Fälle unterscheidet.

**Gültige Formel:** Wurde eine eindeutige Formel gefunden, erzeugt das System eine Ausgabe, in der durch Tiefensuche bestimmt, eine natürlichsprachliche Beschreibung der Berechnung und die zugehörige Formel enthalten ist. Zusätzlich wird die Frage nach einem Speicherziel ausgegeben. Im Beispiel entsteht somit die Systemantwort:

*To calculate the sum of A1 and 4 multiplied by 7 I assume ((A1+4)\*7) is correct.  
Should I write that in your worksheet?*

**Unvollständige Formel:** In diesem Fall sucht JustLingo nach den Informationslücken. Zuerst wird versucht, fehlende Informationen mithilfe eines Rückbezugs zu vervollständigen. Bei expliziten Formulierungen wie *it* (siehe Abb. 2b) wird das Pronomen durch den Baum des passenden Bezugsziels ersetzt. Implizite Rückbezüge (z.B. *Now add 5.*) beziehen sich auf das Ergebnis der vorherigen Berechnung und werden in gleicher Weise aufgelöst. Führt dieser Weg nicht zu einer gültigen Formel, teilt JustLingo dem Nutzer die zu schließende Informationslücke mit. Folgende Rückfragen sind im aktuellen System möglich: Fehlerhafter Wert, inkompatibler Operand, nicht auflösbarer Rückbezug, zu viele bzw. zu wenige Operanden.

**Mehrdeutige Interpretation einer Eingabe:** Das System wendet auf eine Nutzereingabe Muster in unterschiedlicher Reihenfolge an. Sollten hierbei mehrere gültige Bäume entstehen, werden dem Nutzer die zugehörigen Formeln für eine Entscheidung ausgegeben.

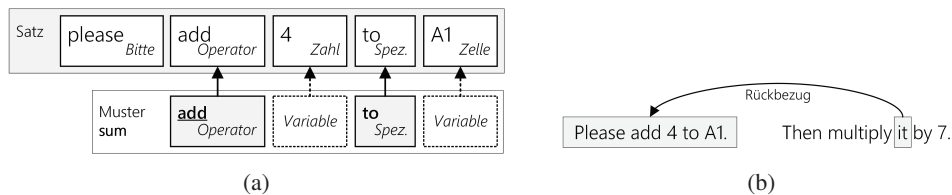


Abbildung 2: Beispiele für eine Musteranwendung (a) und einen Rückbezug (b)

## 4 Evaluation

Inwieweit die erhobenen Anforderungen durch JustLingo erfüllt sind, wurde in einer abschließenden Evaluation ermittelt. Die 17 Teilnehmer waren mit Tabellenkalkulation grundlegend vertraut, hatten jedoch zuvor keine Erfahrung mit JustLingo. Laut Selbsteinschätzung entsprach der Excel-Kenntnisstand der Teilnehmer zu 53% dem eines Anfängers, zu 41% eines Fortgeschrittenen und zu 6% eines Experten. Die Erfahrung im Umgang mit Excel lag zwischen einem und 14 Jahren, bei einem Durchschnitt von 8,4 Jahren. Zur Bearbeitung der Aufgaben wurde ihnen Microsoft Excel mit installierter JustLingo-Erweiterung bereitgestellt. In einer Beispieldatei sollten unter ausschließlicher Verwendung des Dialogsystems zehn mathematische Aufgabenstellungen gelöst werden, die teilweise aufeinander aufbauend formuliert waren, um neben dem korrekten Ergebnis auch die erfolgreiche Auflösung von Mehrdeutigkeiten und Rückbezügen beobachten zu kön-

nen. Tabelle 1 fasst die Ergebnisse des Feldversuchs zusammen. JustLingo weist eine die Erfolgsquote von 79,5% auf. Das System löst die vorhandenen Rückbezüge zu 82% sowie Mehrdeutigkeiten zu 34,8% auf, stellt aktiv kontextbezogene Rückfragen, die in jedem sechsten Fall unmittelbar zur Lösung der Aufgabe führten, und gibt Fehlerbeschreibungen aus. Abschließend wurden die Teilnehmer um eine subjektive Beurteilung des Verhaltens und der Qualität des Systems gebeten. Dabei befanden 82% der Befragten JustLingo als intuitiv einsetzbar und 70% gaben an, einen deutlichen Mehrwert für sich selbst erkannt zu haben. In zwei von drei Fällen wurden die Ausgaben und Rückfragen des System als zielführend und hilfreich wahrgenommen. Hierbei konnte beobachtet werden, wie gerade Teilnehmer mit geringer Erfahrung mit Tabellenkalkulationen von der Dialogführung des Systems profitierten. Bemerkenswert ist zudem die Beobachtung, dass sich viele Nutzer nach nur wenigen Eingaben bereits an das System anpassten, indem sie sich für Formulierungen entschieden, die bereits zuvor erfolgreich verarbeitet wurden.

<b>Gesamt</b>	
Anzahl der Aufgabenstellungen	170
Dialogiterationen (Anzahl Benutzer-System-Interaktionen)	316
Erfolgreiche Lösung	79,5 %
<b>Anforderungen</b>	
Häufigkeit der Systemrückfragen	17,1 %
Erkannte Rückbezüge	82 %
Erkannte Mehrdeutigkeiten	34,8 %

Tabelle 1: Ergebnis des Feldversuchs

## 5 Verwandte Arbeiten

Bereits 1979 stellte Ballard [BB79] eine natürlichsprachliche Schnittstelle für Matrixmanipulation vor. Zwar werden hierbei ebenfalls Rückbezüge aufgelöst, allerdings erfolgt keine Mensch-Computer-Interaktion über ein Dialogsystem. Rückmeldung erhält der Nutzer lediglich über die Ausgabe der Ergebnisse. Ebenso verhält sich NLyze [GM14], eine parallel zu unserer Arbeit entwickelte Erweiterung für Microsoft Excel. Der Ansatz verwendet eine eigene domänenspezifische Sprache zur logischen Interpretation der Nutzereingabe. Die Integration einer Dialogkomponente bietet SmartSynth [LGS13], ein Werkzeug zur Erstellung von Skripten für Mobiltelefone. Seneff [Sen02] stellt ein zielorientiertes Dialogmanagement für das Mercury Flugbuchungssystem vor: Über mehrere Schritte erfragt das System vom Nutzer aktiv alle relevanten Flugdaten und führt im Erfolgsfall eine Buchung durch. Kommerziell erfolgreiche Systeme, wie Apples Siri und Googles Sprachsuche [Bel14] zeichnen sich durch die Abdeckung vieler Domänen aus. Die Auflösung von Rückbezügen lässt die Systeme zudem sehr natürlich wirken. Jedoch findet außer der Ergebnisausgabe sowie der Ausgabe von Fehlermeldungen kein Dialog statt.

## 6 Fazit

In dieser Arbeit wurde ein Prototyp eines natürlichsprachlichen Dialogsystems entwickelt und evaluiert. Die Erfahrungen mit dem Prototypen zeigen, dass in einer eingeschränkten Domäne gute Ergebnisse mithilfe eines einfachen, domänenspezifischen Ansatzes erzielbar sind. Es ist zu erwarten, dass ein solcher Ansatz für andere eingeschränkte Anwendungsbereiche (wie Navigationsgerät, Heimautomatisierung, etc.) ebenfalls funktioniert und umfangreiche Sprachanalysewerkzeuge, wie Stanford CoreNLP<sup>4</sup>, nicht notwendigerweise eingesetzt werden müssen. Darüber hinaus wurde deutlich, dass ein solches System stark von einer modularen Architektur profitiert. Dadurch können einzelne Komponenten wie Spracheingabe, Sprachanalyse, Dialogmanagement, Lernsystem und Wissensbasis nach Bedarf ausgetauscht und ergänzt werden. Schließlich stellte sich gerade im Umgang mit unerfahrenen Anwendern heraus, dass eine interaktive Dialogkomponente für die Akzeptanz eines solchen Systems entscheidend ist. Derzeit wird der Dialog für den Nutzer noch natürlicher gestaltet, indem eine Komponente für Spracheingabe entwickelt wird. Die Sprachkomponente profitiert vom robusten musterbasierten Ansatz. Anhand des bekannten Wortschatzes kann der Erkenner zusätzlich mit einer entsprechenden Grammatik versehen werden. So wird die Erkennung auf die Domäne abgestimmt und ist daher genauer. Des Weiteren wird an einem interaktiven Lernsystem für die Erweiterung der Funktionalität gearbeitet. Anhand des Dialogs mit dem Benutzer lernt JustLingo neue Synonyme und Muster. Das System ist zudem in der Lage, interaktiv benutzerdefinierte Formeln zu erlernen und automatisch die entsprechenden Muster zu generieren. Steuerung über Gesten, möglicherweise in Kombination mit Sprache, ist Gegenstand zukünftiger Untersuchungen.

## Literatur

- [BB79] Bruce W. Ballard und Alan W. Biermann. Programming in natural language: NLC as a prototype. In *Proceedings of the 1979 annual conference, ACM '79*, Seiten 228–237, New York, NY, USA, 1979. ACM.
- [Bel14] Jerome R. Bellegarda. Spoken Language Understanding for Natural Interaction: The Siri Experience. In Joseph Mariani, Sophie Rosset, Martine Garnier-Rizet und Laurence Devillers, Hrsg., *Natural Interaction with Robots, Knowbots and Smartphones*, Seiten 3–14. Springer New York, 2014.
- [GM14] Sumit Gulwani und Mark Marron. NLyze: Interactive programming by natural language for spreadsheet data analysis and manipulation. In *SIGMOD Record*, 2014.
- [LGS13] Vu Le, Sumit Gulweani und Zhendong Su. SmartSynth: Synthesizing Smartphone Automation Scripts from Natural Language. *MobiSys'13*, 2013.
- [Ort14] Charles L. Ortiz. The Road to Natural Conversational Speech Interfaces. *Internet Computing, IEEE*, 18(2):74–78, Mar 2014.
- [Sen02] S. Seneff. Response planning and generation in the MERCURY flight reservation system. *MIT Laboratory of Computer Science, Spoken Languages System Group*, 2002.

---

<sup>4</sup><http://nlp.stanford.edu/software/corenlp.shtml>