Richtlinie zum „Software-Sprint“

Voice-QL

Schlussbericht

Zuwendungsempfänger:

Frank Börncke

Das diesem Bericht zugrundeliegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS22S30 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.

Kurze Darstellung der Aufgabenstellung und Motivation

Was war Deine Motivation? Welches Problem wolltest Du mit Deinem Projekt lösen? Wie war die geplante Vorgehensweise zur Problemlösung (auch Angabe der wichtigsten Meilensteine)?

Obwohl Tabellen grundsätzlich als triviales Konzept erscheinen, erweist sich der Umgang mit Ihnen bei zunehmender Größe als immer schwieriger. Das liegt vor allem daran, dass es zwar einfach ist, große Datenmengen in Tabellen zu speichern. Aber umgekehrt ist es schwierig, Informationen aus Tabellen zu extrahieren.

Wo dieser Umstand nicht berücksichtigt wird und immer mehr Daten in Tabellen exportiert werden, da besteht die Gefahr, dass wir de facto Datenfriedhöfe schaffen.

Es ist daher wichtig, alternative Zugangsmöglichkeiten zu erforschen, da viele Menschen weder SQL noch Excel beherrschen oder aufgrund von Seh- oder motorischen Einschränkungen nicht in der Lage sind, auf umfangreiche Tabellen-Informationen zuzugreifen.

Letztlich sind alle nicht trivialen Tabellen schon alleine durch ihre Größe implizit barrierebehaftet und damit geht dieses Thema auch alle Menschen an. Erschwerend kommt hinzu, dass Tabellen ein generisches Format sind und keine einheitlichen Standards für das Ablegen von Informationen existieren.

Natürliche Sprache als Interface kann ein Weg sein, der Nutzern einen neuen und einfacheren Zugang eröffnet. Dass dieser Ansatz grundsätzlich funktioniert, wurde mit Voice-QL demonstriert.

Beitrag des Projektes zu den Zielen der Förderinitative „Software-Sprint“

Wer ist die Zielgruppe für Deine Lösung? Wie profitiert sie von den Ergebnissen Deines Projekts? Welche Bezüge gibt es zu den Themenfeldern und Zielen des Software Sprints?

Voice-QL spricht direkt und indirekt drei Zielgruppen an:

(1) Entwickler von Voice-Apps, die dank Voice-QL auch mit begrenztem fachlichen Vorwissen Prototypen für nutzerfreundliche und barrierearme Voice-Apps schnell und einfach erstellen können

(2) Anbieter von Informationen im Tabellenformat, die durch den Einsatz von Voice-QL einen zusätzlichen Zugangspunkt zu ihren Daten schaffen und damit ihre Reichweite erhöhen können

(3) Menschen mit visuellen oder motorischen Einschränkungen, die durch Voice-Apps auf der Basis von Voice-QL eine neue Möglichkeiten des Zugangs zu Informationen erhalten, die ihnen sonst möglicherweise verwehrt geblieben wären

Ausführliche Darstellung der Ergebnisse

Welche konkreten Ergebnisse hast Du erzielt? Konnten alle Meilensteine erreicht werden? Welche zusätzlichen Erkenntnisse hast Du aus der Projektarbeit gewonnen, auch im Hinblick auf die Begleitung durch die Open Knowledge Foundation?

Das Projekt "Voice-QL" generiert passend zu einer gegebenen Datenbank automatisch angepasste Voice-App-Prototypen, die auf verschiedenen Plattformen ausgeführt werden können.

Die generierten Anwendungen bieten den Benutzern eine natürlichsprachliche Schnittstelle für den Zugriff auf Tabellen und Datenbanken. Um dieses Ziel zu erreichen, müssen drei Herausforderungen gemeistert werden: Zunächst muss das System (1) die Intention des Benutzers erkennen, um (2) die passenden Informationen aus der Datenbank zu extrahieren um dann schließlich (3) die Ergebnisse in einer adäquaten natürlichen Sprache zu präsentieren.

Um die Intention des Benutzers zu verstehen, reicht es nicht aus, zu ermitteln, was der Benutzer tun möchte - wie beispielsweise die Berechnung einer Summe. Es ist zudem erforderlich, die Spalte oder den Bereich zu identifizieren, auf den sich die Berechnung beziehen soll („*Spalte mit dem Namen Einwohnerzahl*“). Des Weiteren müssen eventuell gesetzte Filter berücksichtigt werden, die bestimmte Zeilen ausblenden („*Zeige nur Zeilen mit dem Wert Jahr gleich 2023*“).

All dies ist schwerlich mit einem einzigen Sprachbefehl zu leisten. Deshalb haben wir ein **Context**-Konzept entwickelt, dass es ermöglicht, die Benutzerentscheidungen fortlaufend in einer **Session** zu speichern. Dies geschieht iterativ im **Dialog**. Das System merkt sich so die Entscheidungen des Benutzers und reagiert bei unvollständigen Angaben **fehlertolerant**: statt einer Verweigerung der Bearbeitung fragt das System nach und erkundigt sich gezielt nach fehlenden Informationen, bis die ursprüngliche Anfrage bearbeitet werden kann.

Sobald alle benötigten Informationen vorhanden sind, sollte das SQL-Statement für die Datenbankabfrage problemlos dynamisch aus allen verfügbaren Informationen zusammengestellt werden können.

In der Praxis sind jedoch vielfältige Herausforderungen zu bewältigen:

- Konkurrierende Schreibweisen wie „zwölf“, „Zwoelf“ oder „12“
- Verschiedene Spaltennamen klingen gleich: „Ja“ vs. „Jahr“
- Kryptische Spaltennamen wie „X19Q3“
- Datenqualität: „ grün “ statt „grün“

Abhängig von der Qualität und Beschaffenheit der (Meta)-Daten innerhalb der Datenbank müssen in solchen Fällen möglicherweise Anpassungen an der Struktur vorgenommen oder eine Bereinigung/Vorverarbeitung der Daten durchgeführt werden.

Um Ergebnisse von Abfragen angemessen präsentieren zu können, haben wir uns für ein Cursorkonzept entschieden: Mit diesem Konzept können Nutzer mit Hilfe von Sprachbefehlen durch die Ergebnismenge navigieren und sich gezielt einzelne Zeilen oder Informationen vorlesen lassen.

Da Ergebnismengen oft sehr umfangreich sind, ist diese Möglichkeit der zeilenweisen Navigation und Abfrage von Inhalten wesentlich praktikabler als das vollständige Vorlesen aller Ergebnisse/Zeilen.

Schließlich kann das Vorlesen von Werten mit Markup oder leeren Inhalten wie NULL oder einem Leerstring problematisch sein und im Einzelfall eine Sonderbehandlung erforderlich machen.

Fügen wir all diese Schritte zusammen, so ergibt sich das folgende Gesamtbild:

Voice-QL ist ein Generator, der in der Lage ist, aus einer SQLite-Datenbank Metadaten und Inhalte zu extrahieren, um daraus ein trainierbares Sprachmodell und ausführbaren Code zu generieren.

Die resultierenden Code-Artefakte werden verwendet, um das Open-Source Framework Jovo (https://jovo.tech) aufzurufen, welches es uns ermöglicht, Voice-App-Prototypen für verschiedene Plattformen zu generieren. Diese Prototypen können dann aus der Toolchain entnommen und zu eigenständigen Anwendungen ausimplementiert werden.

Über den Projektantrag hinaus geht die Implementierung dahingehend, dass alle entwickelten Tools und Konfigurationen für einen mehrsprachigen Einsatz vorbereitet und nicht auf den Einsatz in deutscher Sprache beschränkt sind.

Zielgruppe, Nutzen und mögliche Weiterentwicklungen

Welcher Nutzen ergibt sich für die Zielgruppe aus den Ergebnissen Deines Projekts?

Entwickler von Sprachanwendungen werden in die Lage versetzt, auch mit vergleichsweise geringen Vorkenntnissen allein durch die Bereitstellung einer Datenbank im SQLite-Format testbare Sprachanwendungs-Prototypen für unterschiedliche Plattformen zu generieren. Das geht sehr viel schneller, als dies manuell möglich wäre.

Diese Prototypen können dann iterativ verfeinert und zu fertigen Voice-Apps weiterentwickelt werden, die auf verschiedenen Plattformen nutzbar sind. Dies könnte zum Beispiel ein Chatbot im Browser sein, eine Alexa Application oder eine Google-Assistant-Anwendung.

Anbieter von Informationen können die Reichweite für Ihr Angebot erweitern, indem sie ihren Nutzern zusätzlich eine Voice-App zum Datenzugriff anbieten. Dazu müssen die Anbieter nur eingeschränkt Know-How im Bereich Voice aufbauen, weil Voice-QL viele Schritte automatisiert und konfigurierbar macht.

Schließlich können mit Voice-QL auch Nutzergruppen erschlossen werden, die mit anderen Zugangswegen möglicherweise überfordert sind. Unabhängig davon profitieren Anwender grundsätzlich von einem reichhaltigeren Angebot an Tools.

Welche weiter-gehenden Effekte ergeben sich aus der Open-Source-Stellung der Ergebnisse?

Die Erstellung und Wartung von umfangreichen Sprachmodellen stellt eine zeitaufwändige und arbeitsintensive Aufgabe dar. Um die Komplexität der im Rahmen von Voice-QL entwickelten Sprachmodelle in den Griff zu kriegen, haben wir im Rahmen des Projekts neuartige Tools konzipiert und implementiert. Diese wurden als eigenständige Plugins für das Open Source Framework Jovo veröffentlicht und sind komplett unabhängig von Voice-QL für andere Anwendungsfälle nutzbar. Durch die Entwicklung dieser Werkzeuge konnten wir nicht nur die Sprachmodellierung effektiver gestalten, sondern auch die Entwicklungszeit verkürzen und die Qualität unserer Arbeit verbessern.

Von diesen neuen Tools profitieren auch andere Entwickler von Sprachanwendungen.

Gibt es Ideen für die Weiterentwicklung Deiner Lösung und Pläne zu deren Umsetzung?

Nach der erfolgreichen Implementierung der Basisfunktionen ergaben sich in der zweiten Phase des Projekts zahlreiche Möglichkeiten für zusätzliche Funktionen und Erweiterungen. Zur Erreichung der Projektziele im Zeitrahmen war es notwendig, eine Priorisierung vorzunehmen, bei der sowohl die Machbarkeit als auch die verfügbare Restzeit im Projekt berücksichtigt wurden.

Dies führte dazu, dass einige Funktionen aus der Wunschliste (die nicht im Projektantrag enthalten waren), für zukünftige Ausbaustufen von Voice-QL zurückgestellt wurden. Im Kontext künftiger Entwicklungen für Voice-QL stehen die folgenden Funktionalitäten an vorderster Stelle der Anforderungsliste: Integration von JOIN-Operationen, GROUP BY-Klauseln, Nested Queries, Subselects sowie Range Queries.

Hat die Arbeit in dem Projekt Dich in Deiner persönlichen, fachlichen Weiterentwicklung unterstützt?

Ja, definitiv.

Kurze Darstellung der Arbeiten, die zu keiner Lösung geführt haben

Gab es Arbeiten bzw. Lösungsansätze, die nicht weiter verfolgt wurden? Was waren die Hinter-gründe, und wie bist Du alternativ vorgegangen?

Grundsätzlich verlief das Projekt planmäßig. Während der Evaluierung der Umsetzungsmöglichkeiten zur Anbindung von verschiedenen Datenschnittstellen wurde deutlich, dass die Verwendung des Standardformats SQLite besser den angestrebten Zielen entspricht als ein universeller SQL-Adapter.

Durch die Nutzung von freien Konvertern für andere Tabellenformate kann auf die Entwicklung von maßgeschneiderten Schnittstellenadaptern verzichtet werden, wodurch wertvolle Zeit für die zentralen Aufgaben des Vorhabens eingespart wurde.

Kurze Angabe von Präsentationsmöglichkeiten für mögliche Nutzer

Wo können sich Interessenten detailliert über Deine Projektergebnisse informieren (z.B. Webseite, GitHub, Veröffentlichungen)?

Hauptprojektseite: **Voice-QL**
https://github.com/fboerncke/voice-ql

Nebenprojekt: **Web Client** with SQL protocol display - **Chat Widget**
https://github.com/fboerncke/voice-ql-vue-chat-client

Nebenprojekt: **Talking Browser Demo** - Use browser to talk with your data
https://github.com/fboerncke/voice-ql-vue-talking-client

Plugin für das Framework Jovo: **Model Linter**
https://github.com/fboerncke/jovo-v4-community-hook-model-linter

Plugin für das Framework Jovo: **Magic Model**
https://github.com/fboerncke/jovo-v4-community-hook-magic-model

Plugin für das Framework Jovo: **Magic Prototyper**
https://github.com/fboerncke/jovo-v4-community-hook-magic-prototyper

Kurze Erläuterung zur Einhaltung der Arbeits- und Kostenplanung

Gab es im Projektverlauf Ereignisse, die eine Anpassung der Planung erforderlich machten – z.B. Mehr- oder Minderaufwand bei der Bearbeitung von Teilaufgaben?

Im Projektantrags wurde eine Überlegung hinsichtlich der Entwicklung eigener Tabellenadapter für CSV- und Excel-Dateien angestellt. Im Verlauf des Projekts wurde jedoch entschieden, das Standardformat SQLite zu verwenden, welches den Vorteil bietet, dass freie Konverter für CSV-, Excel-Dateien und andere Tabellenformate verfügbar sind. Aufgrund dieser Entscheidung entfiel die Notwendigkeit, eigene Tabellenadapter zu entwickeln.

Die dadurch eingesparte Zeit konnte in die Implementierung weiterer Features investiert werden.

Kurze Darstellung von etwaigen Ergebnissen bei anderen Stellen

Gab es Entwicklungen anderer Personen oder Institutionen, die Einfluss auf Deine Arbeiten und die Zielsetzung hatten? Wenn ja, worin bestand dieser und wie bist Du damit umgegangen?

Eine der ursprünglich avisierten Zielplattformen für die Sprachschnittstellen war der Google Assistant. Erst nach der Einreichung des Projektantrags wurde bekannt, dass Google am 13. Juni 2023 die Unterstützung dieser Plattform für Drittanbieter Voice-Apps einstellen wird.

Vor diesem Hintergrund haben wir uns dafür entschieden, unsere Ressourcen primär auf andere Zielplattformen zu fokussieren und das Testing auf der Google Assistant-Plattform in begrenztem Umfang abzuschließen.