RECHERCHEBERICHT

APRT-17

Im Rahmen dieses Projektes soll eine Software erstellt werden, die zu einem Themengebiet (bspw. Städte, Künstler, Pflanzenarten) Informationen aus dem Internet bezieht und daraus Präsentationen generiert. So könnte bereits aus dem Wikipedia Artikel zu Leipzig eine Menge an Informationen abgeleitet werden. Diese strukturierten Informationen, wie die historische Entwicklung (incl. Bildern), berühmte Persönlichkeiten und Zahlen / Fakten zur Stadt, sollen anschließend in einem für das Projekt sinnvollen Datenform in Präsentationen überführt.

Teilnehmer: Lucas Lange, Andrea Niebuhr, Johannes Bräuer, Kilian Schwabe, Hunter Foo und Francesco Mandry

Inhaltsverzeichnis

Begriffe (Glossar)

Python[1]
Eine universelle, üblicherweise interpretierte höhere Programmiersprache.
Jython[2]
eine reine Java[30]-Implementierung der Programmiersprache Python[1] und ermöglicht somit die Ausführung von Python[1]-Programmen auf jeder Java[30]-Plattform[6].
GUI[3]
eine grafische Benutzeroberfläche.
Webanwendung[4]
Die Datenverarbeitung und -auswertung findet […] hauptsächlich auf einem entfernten Webserver statt.
Desktopanwendung[5]
[…] [werden] lokal auf dem Rechner des Benutzers installiert und dort ausgeführt.
Plattform[6]
bezeichnet in der Informatik eine einheitliche Grundlage, auf der […] [Programme] ausgeführt. und entwickelt werden können.
PowerPoint[7]
Präsentationssoftware[8] von Microsoft.
Präsentationssoftware[8]
Computerprogramm, das die Erarbeitung und Präsentation eines Vortrages oder Referates unterstützt.
API[9]
(engl. „application programming interface“) - ein Programmteil, mit dessen Hilfe sich andere Programme an das Softwaresystem anbinden können.
HTML[10]
(engl. Hypertext Markup Language) - textbasierte Auszeichnungssprache zur Strukturierung digitaler Inhalte wie Texte, Hyperlinks[29], Bilder und anderen.
Office Open XML[11]
von Microsoft entwickelte Dateiformate zur Speicherung von Bürodokumenten auf XML[15]-Basis.
Library[12]
(auch (Programm-)bibliothek) - Sammlung von Unterprogrammen/-Routinen; nicht eigenständig lauffähig.
parsen/Parser[13]
Programm zur Zerlegung und Umwandlung einer Eingabe in ein bestimmtes Format.
String[14]
eine Folge von Zeichen aus einem definierten Zeichensatz; in Java[30] ein Datentyp.
XML[15]
(Extensible Markup Language) eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien.
automatic summarization[16]
automatisierte Zusammenfassung von Texten.
extraction[17]
wichtige Thesen aus einem Text werden extrahiert, ohne dabei die Sätze oder Wörter zu verändern.

abstraction[18]
Umschreibung von wichtigen Kernpunkten („paraphrasing“).
NLP[19]
natural language processing, automatisierte Textverarbeitung und/oder –generierung.
JSON[20]
JavaScript Object Notation, kompaktes Datenformat in einer einfach lesbaren Textform zum Zweck des Datenaustauschs zwischen Anwendungen.
URL[21]
Uniform Resource Locator, lokalisiert eine Ressource, z.B.: eine Webseite.
String Operatoren[22]
Funktionen zur Manipulation von Zeichenketten.
Framework[23]
ist ein Programmiergerüst, das in der Softwaretechnik […] verwendet wird.
Folie[24]
eine einzelne Seite einer am PowerPoint[7]-Präsentation(Präsentations-) Thema: eine Kategorie, zu der das zu erstellende Programm Präsentationen anbietet, z.B. „Städte“.
(Präsentations-) Instanz[25]
Subjekt einer spezifischen Präsentation, z.B. „Leipzig“.
Datenbank[26]
eine strukturierte Sammlung von Information; Daten sind systematisch katalogisiert und können gezielt abgerufen werden.
SPARQL[27]
eine graphenbasierte Abfragesprache für RDF[28].
RDF[28]
(Resource Description Framework) bezeichnet eine technische Herangehensweise im Internet zur Formulierung logischer Aussagen über beliebige Dinge (Ressourcen).
Hyperlink[29]
ist ein Querverweis […], der funktional einen Sprung zu einem anderen elektronischen Dokument oder an eine andere Stelle innerhalb eines Dokuments ermöglicht.
Java[30]
eine objektorientierte Programmiersprache […].
Apache Jena[31]
Java[30] Bibliothek zur Arbeit mit unter Anderem RDF[28] und SPARQL[27].

Konzepte

Teamaufteilung

Wir teilen das Team in zwei Gruppen, wobei sich die Eine für die Extraktion der nötigen Informationen aus dem Netz und die Andere für die Ausgabe dieser Daten als angemessene Präsentation zuständig sieht, so dass diese Bereiche besser untereinander besprochen werden können. Die Bindeglieder dieser Teilaufgaben sind gute Kommunikationswege und das wöchentliche Treffen, welches Raum für regen Austausch untereinander bietet.

Java[30]

Als Hauptsächliche Programmiersprache für das Projekt wählen wir „Java[30]“ auf Grund der guten Vertrautheit aller Gruppenmitglieder mit deren Syntax, sowie der weitreichenden Möglichkeiten, welche auch auf den vielen verfügbaren Bibliotheken[12] beruht.

Anwendungstyp

Wichtig ist die Frage, ob eine Web- oder Desktopanwendung als Ergebnis des Projekts das Ziel sein sollte. Desktopanwendungen[5] bieten meist eine schnellere Laufzeit, interagieren besser mit der PC-Hardware und eine GUI[3] ist nicht speziell nötig, aber auch leicht zu bauen. Webanwendungen[4] sind hingegen sehr portabel und plattformunabhängig einsetzbar. Des Weiteren fällt eine Wartung dieser meist leichter. In Hinsicht auf unser Projekt sollten wir den Kunden entscheiden lassen, doch gib es keinen Grund hier eine Webanwendung zu entwickeln, da dies für unser Projekt eher von geringem Mehrwert wäre.

Präsentationsformat .pptx

Auf jede Präsentationssoftware[8] kommt mindestens ein eigenes Präsentationsformat, jedoch sollte für unser Projekt nur ein Ausgabeformat eine Rolle spielen. Aufgrund der weiten Verbreitung, besonders auch in Bildungseinrichtungen, stehen hier die PowerPoint[7]-Präsentation und ihre Formate an erster Stelle. Im Besonderen das „.pptx“-Format.
Beim Präsentationsformat .pptx handelt es sich um eines der bekanntesten Formate für PowerPoint[7] Präsentationen, welches im Office Open XML[11]-Format gespeichert wird. Die API[9]„python[1]-pptx“, eine sehr gut dokumentierte Python[1] library[12] mit vielen Features, um Präsentationen zu erstellen und zu bearbeiten. (https://python-pptx.readthedocs.io/en/latest) Da unser Programm jedoch in Java[30] wird, sei „python[1]-pptx“ hier nur wegen der guten Kompatibilität von Java[30] und Python[1] aufgeführt.
Eine ebenfalls ausgezeichnet dokumentierte API[9] für Java[30] ist die library[12] docx4j, mit der man neben .docx auch .pptx, sowie .xlsx-Dateien erstellen und Bearbeiten kann.
(http://www.docx4java.org/trac/docx4j)
Mit dieser library[12] erstellte Präsentationen werden vor dem parsen[13] zu .pptx in einem String[14] erstellt und gespeichert, welcher ein ähnliches Layout wie ein HTML[10]-Dokument aufweist.

DBpedia

Im Projekt DBpedia wird versucht Informationen aus Wikipedia strukturiert zu extrahieren und diese online zugänglich zu machen.
Suchvorgänge auf DBpedia werden mit der Abfragesprache „SPARQL[27]“ durchgeführt. Diese ermöglicht es bestimmte Daten aus der Wissensdatenbank zu extrahieren. So lassen sich Beispielsweise alle Informationen zur Stadt „Leipzig“ mit nur einem Zugriff extrahieren. Problematisch ist nur, dass man bis auf einen allgemein gehaltenen Textabschnitt, nur sehr faktenhaft notierte Informationen finden und extrahieren kann. Folgt man den Hyperlinks[29], welche für viele Fakten existieren, kommt man zwar zu tiefreichenden Informationen, doch wird mit steigender Suchtiefe auch der Zusammenhang zum Eigentlichen Thema in der Regel immer geringer. Dies führt dazu, dass man sich themenabhängig nicht alleinig auf DBpedia als Informationsquelle für eine ansprechende Präsentation stützen kann. Der letztlich erwartete Informationsgehalt und dessen geplante Darstellung spielen hier eine wichtige Rolle. Speziellere Themen könnten ein Problem darstellen, da die Datenbank hier möglicherweise nur wenige interessante Fakten bieten kann.
Der sinnvolle Einsatz von DBpedia hängt also von der effektiven Nutzung der Sprache „SPARQL[27]“, den Präsentationsthemen und der Art der zu gewinnenden Informationen ab. Noch zu erwähnen ist, dass der Zugriff mit „SPARQL[27]“ in Java[30] beispielsweise über die Bibliothek „Apache Jena“ einfach möglich gemacht wird, was die Suche auf DBpedia für unser Projekt zugänglich macht.

Auslesen von Informationen in Wikipedia (1)

Im Web findet sich eine Vielzahl verschiedener Daten die unterschiedlich gut strukturiert sind. Plattformen[6] wie Wikipedia geben verschiedensten Themenbereichen bzw. Artikeln eine Struktur aus Inhaltsverzeichnis, Infobox, Überschriften, Text und Bilder. Mittels zentraler Datenspeicher wie Wikidata und DBPedia, die auf Wikipedia basieren, werden Artikel stichpunktartig zusammengefasst. Auf diese Informationen kann man mittels Datenbanksprachen (SPARQL[27]) oder anderen APIs[9] zugreifen. Für spezifische Texte in Wikipedia reicht es auch Artikel in JSON[20]- oder XML[15]-Formate zu parsen[13].

Recherche zum Aussehen einer guten Präsentation

Wenn man das Internet nach Tipps für gut aussehende PowerPoint[7]-Präsentationen durchsucht, stellt man schnell fest, dass die meisten dieser Hilfen ein spezielles Thema mit einbeziehen. Sie sollen den Leser also unterstützen, seine eigene Präsentation anschaulicher zu gestalten. Nun mögen einige dieser Tipps auch in Bezug auf dieses Projekt durchaus Anwendung finden, andere sind nur schwer umsetzbar.
So liest man immer wieder, wie wichtig große Schrift ist und geringe Textmenge je Folie[24] einzuhalten ist. Dagegen das Design der Folien[24]-Hintergründe oder die Schriftart auf das Thema anzupassen, ist aufgrund der Themenvielfalt, die im Projekt gesetzt werden nicht umsetzbar. Letztendlich hängen Design, so wie einige andere Eckpunkte der Präsentation (z.B. die Einbindung von Bildern) noch von Vorgaben der Betreuer und Umfang des Präsentation-APIs[9] ab.

Präsentationsthemen und Quellen

Für die Präsentationen sind Themen (z.B. „Städte“) zu finden, zu deren einzelnen Instanzen[25] (z.B. „Leipzig“) Präsentationen erstellt werden können. Diese Themen sollen einige Eigenschaften erfüllen. Sie sollen weit gefasst sein, d.h. zu jedem Thema sollen möglich viele Instanzen[25] existieren, um viele Präsentationen daraus erstellen zu können. Die Themen sollen hauptsächlich durch strukturierte Parameter definiert und die einzelnen Instanzen[25] durch ihre verschiedenen Werte abgegrenzt werden, um das automatisierte Einlesen der Information und die sinnvolle Aufarbeitung in der Präsentation zu erleichtern. Information zu den Themen soll aus möglichst wenigen (ideal: einer) vertrauenswürdigen Quellen verfügbar sein, in der Information zu möglichst vielen Instanzen[25] möglichst strukturiert gesammelt ist. Da die Präsentationen Grundstock und Testobjekte für eine wissenschaftlich orientierte Plattform[6] sind, sollen die Themen eine entsprechende wissenschaftliche Relevanz haben. Diese Anforderungen legen nahe, gepflegte Online-Datenbanken[26] als Quellen zu verwenden, wie z.B. DBpedia oder Wikidata. Auch gut strukturierte (Elemente von) Webseiten, wie z.B. Infoboxen auf Wikipedia, sind als Quellen interessant. Für Informationen, die nicht kurz und tabellarisch auffindbar sind, muss auf Natural Language Processing (NLP[19]) zurückgegriffen werden, um Information aus Fließtexten, z.B. Wikipediaartikeln, zu extrahieren. Mögliche Themen, die die genannten Anforderungen erfüllen, sind Städte und Minerale.

Aspekte

GUI[3]

Eine schöne, intuitive GUI[3] sehen wir eher als optionales Ziel des Projekts, da sie für die Arbeit des Programms nicht existenzielle Bedeutung hat. Dabei gehen wir nach dem Verwendungszweck, der von spezieller Natur ist und somit keine massentaugliche Bedienbarkeit vorgibt.

Python

Beim Blick auf weitere Programmiersprachen lässt sich besonders „Python[1]“ nicht gänzlich ausschließen, da im Hinblick auf dieses Projekt dort ebenfalls viele nützliche Bibliotheken[12] zur Verfügung stehen. Vorteilhaft ist das leichte Zusammenspiel von „Python[1]“-Programmen in „Java[30]“ durch beispielsweise „Jython[2]“.

NLP[19]

„Automatic summarization[16]“, ein Teilgebiet des NLP[19] (natural language processing), dient dazu, Fließtexte möglichst kurz zusammenzufassen, ohne wichtige Schlüsselaussagen des Textes zu vernachlässigen. Es gibt zwei verschiedene Ansätze: die „extraction[17]“ kopiert einzelne Sätze aus einem Fließtext heraus, die „abstraction[18]“ umschreibt Sätze und verknüpft komplexe Sachverhalte logisch miteinander. Im Rahmen dieses Projekts ist letztere zu bevorzugen, da sie möglichst kurze Fakten aus Informationstexten liefert, ohne

dabei den Sinn des Textes zu verändern. Die Web-Applikation „smmry“ (smmry.com) bietet diese Dienste an. Hier lässt der User einen Fließtext einlesen und bestimmt, in wie vielen Sätzen er zusammengefasst werden soll. Die Anzahl der Sätze passend zum Informationsgehalt automatisiert festzulegen ist problematisch, da sich die Menge von wichtigen Fakten nicht anhand von z.B. der Textlänge ermitteln lässt. Es liegt nahe, stattdessen Werte festzulegen, um so eine bestimmte Anzahl von Folien mit einer bestimmten Menge von Fakten pro Folie zu erstellen.
NLP[19] ist ein aktives Forschungsgebiet und die Ergebnisse beider Arten von automatic summarization[16] schwanken in ihrer Qualität. Dennoch ist es eine sinnvolle Methode, um für die Präsentationen relevante Information aus Fließtexten zu extrahieren.

Auslesen von Informationen in Wikipedia (2)

Mittels Query Abfragen (Abfragesprachen) in der URL[21] kann man konkrete Inhalte aus Wikipedia Artikeln filtern. Informationen der Befehle dazu finden sich in der Dokumentation für die MediaWiki API[9]. https://en.wikipedia.org/w/api.php. Das gleiche gibt es auch für die TMDb mit kurzer Dokumentation. https://www.themoviedb.org/documentation/api
Durch Abfrage des geparsten Ergebnisses in JAVA[30], kann der Inhalt in einem String[14] gespeichert werden und mittels String[14]-Operatoren von unnötigen Zeichen und Text befreit werden. Dies kann bei langen Texten umständlich werden, eignet sich jedoch für einzelne Themen eines Artikels, da dieser wenige Text umfassen.
Eine andere Möglichkeit bietet SPARQL[27], das nicht nur DBPedia, sondern auch Wikidata unterstützt. Apache Jena[31] als Framework[23] kann dafür genutzt werden, mit JAVA[30] und SPARQL[27] die Datenbanken[26] auszulesen.

Quellen

https://de.wikipedia.org/wiki/JavaScript_Object_Notation
https://de.wikipedia.org/wiki/Uniform_Resource_Locator
https://de.wikipedia.org/wiki/Programmierschnittstelle
https://de.wikipedia.org/wiki/Hypertext_Markup_Language
https://de.wikipedia.org/wiki/Extensible_Markup_Language
https://de.wikipedia.org/wiki/Office_Open_XML
https://de.wikipedia.org/wiki/Parser
https://de.wikipedia.org/wiki/Zeichenkette
https://de.wikipedia.org/wiki/Programmierschnittstelle
https://de.wikipedia.org/wiki/Webanwendung
https://de.wikipedia.org/wiki/Plattform_(Computer)
https://de.wikipedia.org/wiki/Pr%C3%A4sentationsprogramm
https://de.wikipedia.org/wiki/Extensible_Markup_Language
https://de.wikipedia.org/wiki/Programmbibliothek
https://de.wikipedia.org/wiki/Abfragesprache
https://de.wikipedia.org/wiki/Graph_(Graphentheorie)
https://de.wikipedia.org/wiki/Resource_Description_Framework
https://de.wikipedia.org/wiki/Hyperlink
https://de.wikipedia.org/wiki/Java_(Programmiersprache)
https://jena.apache.org/about_jena/about.html
https://de.wikipedia.org/wiki/SPARQL