FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Wilangyman

Eine Google-Chrome Erweiterung die Wikipedia-Artikel um fremdsprachliche Inhalte ergänzt

BACHELORARBEIT

zur Erlangung des akademischen Grades

Bachelor of Science

im Rahmen des Studiums

Medieninformatik und Visual Computing

eingereicht von

Wolfgang Gundacker
Matrikelnummer 08908802

an der Fakultät für Informatik
der Technischen Universität Wien
Betreuung: Dr. tech. Manuela Waldner, MSc.

Wien, 1. Juli 2019

Wolfgang Gundacker	Manuela Waldner

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Wilangyman

A Wikipedia Cross Language Chrome Extension for Google Chrome

BACHELOR'S THESIS

submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science

in

Media Informatics and Visual Computing

by

Wolfgang Gundacker
Registration Number 08908802

to the Faculty of Informatics

at the TU Wien

Advisor: Dr. tech. Manuela Waldner, MSc.

Vienna, 1st July, 2019

Wolfgang Gundacker	Manuela Waldner

Erklärung zur Verfassung der Arbeit

Wolfgang Gundacker

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 1. Juli 2019

Wolfgang Gundacker

Danksagung

Ein herzliches Vergelt's Gott an meine Betreuerin Manuela Waldner. Sie hat mich bei dieser Arbeit mit Ihrem nahezu unerschöpflichem Wissen und wertvollen Inputs unterstützt. Weiters danke ich all meinen Studienkollegen, die mich trotz des großen Altersunterschieds in ihren Kreis aufgenommen haben. Sie alle haben mein Studium zu einem unvergesslichen Erlebnis gemacht.

Acknowledgements

A heartfelt thank God to my supervisor Manuela Waldner. She has supported me in this work with her almost inexhaustible knowledge and valuable input. Furthermore, I thank all my fellow students, who have taken me into their circle despite the great age difference. They all made my studies an unforgettable experience.

Kurzfassung

Wikipedia-Artikel unterscheiden sich in den unterschiedlichen Sprachversionen oft in Struktur und Inhalt. Manche Informationen sind nicht in allen Sprachen verfügbar. Das hat zur Folge, dass NutzerInnen wichtige Daten aus der Online Enzyklopädie entgehen, wenn sie sich auf eine Sprache beschränken. Ziel von Wilangyman ist es, diese Informationen zusammenzuführen und sie in übersichtlicher Art dem Nutzer oder der Nutzerin zu präsentieren. Die Artikel werden mittels Natural Language Processing (NLP) verglichen und und anhand ihrer Ähnlichkeiten miteinander verknüpft. Korrespondierende Passagen mit zusätzlichem Informationsgehalt werden absatzweise dargestellt. Inhaltliche Redundanzen sollen dabei vermieden werden.

Keywords: Wikipedia, Chrome-Extension, Übersetzung, Mehrsprachig, CLIR

Abstract

Wikipedia articles often differ in structure and content in the different language versions. Some information is not available in all languages. As a result, users are missing important data from the online encyclopedia if they confine themselves to one language. The aim of Wilangyman is to bring this information together and present it in a clear way to the user. The articles are compared using Natural Language Processing (NLP) and linked by their similarities. Corresponding passages with additional information are displayed paragraph by paragraph. Content redundancies should be avoided.

Keywords: Wikipedia, Chrome-Extension, Translation, Multilingual, CLIR

Inhaltsverzeichnis

KAPITEL 1

Einleitung

Auf der Suche nach Informationen wird heutzutage gerne die frei verfügbare Enzyklopädie 'Wikipedia' verwendet. Diese enthielt im Juni 2019 mehr als 50 Millionen Artikel [Raba] über sämtliche verfügbaren Sprachen hinweg – Tendenz stark steigend. Viele behandelte Themen existieren in mehreren Übersetzungen oder haben unterschiedliche Sprachversionen [Rabb]. Es kann vorkommen, dass die gesuchte bzw. relevante Information nicht in dem gerade gelesenen Artikel vorhanden ist. Somit ist man verleitet – sofern vorhanden – eine fremdsprachliche Ausgabe zum selben Thema durchzusehen. Natürlich setzt das voraus, dass man dieser Sprache auch mächtig ist. Alternativ kann man vorhandene Übersetzungsdienste (wie z.B. Google Translate [Gooa] oder DeepL [Dee]) bemühen, um eine verständliche Version zu erhalten. Da es sich hierbei aber um sozusagen verwandte Artikel unter dem selben Titel handelt, ist es sehr wahrscheinlich, dass sich zumindest einzelne Absätze ähneln. Man wird also die selbe Information mehrfach vorfinden.

Wikipedia ist ein gemeinnütziges Projekt zur Erstellung einer multilingualen Enzyklopädie auf Basis des Wiki-Konzepts [Wik19]. Eine Vielzahl von Autoren verfasste und übersetzte seit 2001 Artikel zu allen nur erdenklichen Themen. Die dadurch generierten Beiträge können durch Wissen und Vorlieben des jeweiligen Verfassers stark von einander abweichen, auch wenn der Titel derselbe ist. Außerdem kann auch bei gleicher Basis eine landes- bzw. sprachspezifische Sicht entstehen, die zu anderen Versionen vielleicht sogar widersprüchlich ist [GD17]. Es kann also von Interesse sein, die unterschiedlichen Standpunkte übersichtlich präsentiert zu bekommen.

Wilangyman versucht dem Nutzer das in Wikipedia enthaltene Wissen über die Sprachbarriere hinweg zur Verfügung zu stellen. Jedoch sollen nicht einfach Übersetzungen eingefügt oder Artikelversionen gegenübergestellt werden, sondern es soll auch auf den Inhalt eingegangen werden. Auf Knopfdruck werden die angeforderten Sprachversionen geladen, mit Hilfe der Google Translate API übersetzt und schließlich unter Verwendung von Natural Language Processing verglichen. Zu guter Letzt werden die passenden Absätze eingefügt und je nach berechneter Ähnlichkeit dargestellt. Es wird davon ausgegangen,

dass Absätze mit hoher Ähnlichkeit nicht im eigentlichen Leseinteresse sind. Daher werden Texte ab einem selbst zu definierendem Schwellwert nur auf Verlangen zur Verfügung gestellt. Die versteckten Abschnitte enthalten die deutsche Übersetzung und können auf User-Interaktion dargestellt werden. Für Leser, die sich nicht vollständig auf die maschinelle Übersetzung verlassen können oder wollen, gibt es auch die Option den Absatz in der Originalsprache darzustellen. Ziel ist es also, den Ursprungsartikel um weitere, komplementäre Textteile zu erweitern und gleichzeitig vor unnötigen Informations-Redundanzen zu bewahren.

In dieser Arbeit werden unterschiedliche "Arten von Sprachen" erwähnt (siehe Abb. 1.1):

- Ein Wikipedia-Artikel wurde in einer **Quellsprache** verfasst. Das betrifft sowohl den Ausgangsartikel als auch den Ergänzungsartikel.
- Die **Zielsprache** ist definiert durch die Sprache des Ausgangsartikels, also der Sprache der Wikipedia-Seite, die der Nutzer aufgerufen hat. Diese Sprache wird auch zur Darstellung der eingefügten Teile verwendet.
- Alle Sprachversionen werden in die **Vergleichssprache** übersetzt. Diese Übersetzungen dienen dem inhaltlichen Vergleich. Derzeit wird immer Englisch als Vergleichssprache verwendet, da diese Sprache am besten von Natural Language Processing (NLP) Tools unterstützt wird.

Beispielsweise verwendet ein Nutzer das Tool bei der deutschen Seite `https://de.wikipedia.org/wiki/Hilma_Hooker` und möchte diese um Informationen aus der französischen Version erweitern:

- Die Seite wird auf Deutsch (Zielsprache) geladen.
- Die deutsche Version wird ins Englische (Vergleichssprache) übersetzt.
- Die französische (Quellsprache dieses Artikels) Version wird geladen.
- Die französische Version wird ins Englische (Vergleichssprache) übersetzt.
- Die französische Version wird ins Deutsche (Zielsprache) übersetzt.
- Die beiden, in die Vergleichssprache übersetzten Texte, werden verglichen und die Ähnlichkeiten berechnet.
- Die deutsche Übersetzung der französischen Version wird in den Originalartikel eingefügt.

Das Tool besteht aus einer Erweiterung für den Internetbrowser "Google Chrome", über dessen Benutzeroberfläche weitere Informationen angefordert werden können. So können NutzerInnen ohne die gewohnte Umgebung zu verlassen diesen Dienst nutzen.

Die Anfrage wird vom zweiten Teil dieses Tools, einem Serverdienst abgearbeitet. Dieser fordert Übersetzungen an und berechnet die Ähnlichkeiten. Abschließend werden die Daten an die Chrome-Extension zurückgegeben, die diese dann in den gerade angezeigten Wikipedia-Artikel einfügt. Dabei handelt es sich um temporäre Erweiterungen der Seite, die nur in der Browserinstanz des Nutzers angezeigt werden. Die eigentlichen Wikipedia-Artikel bleiben unberührt.

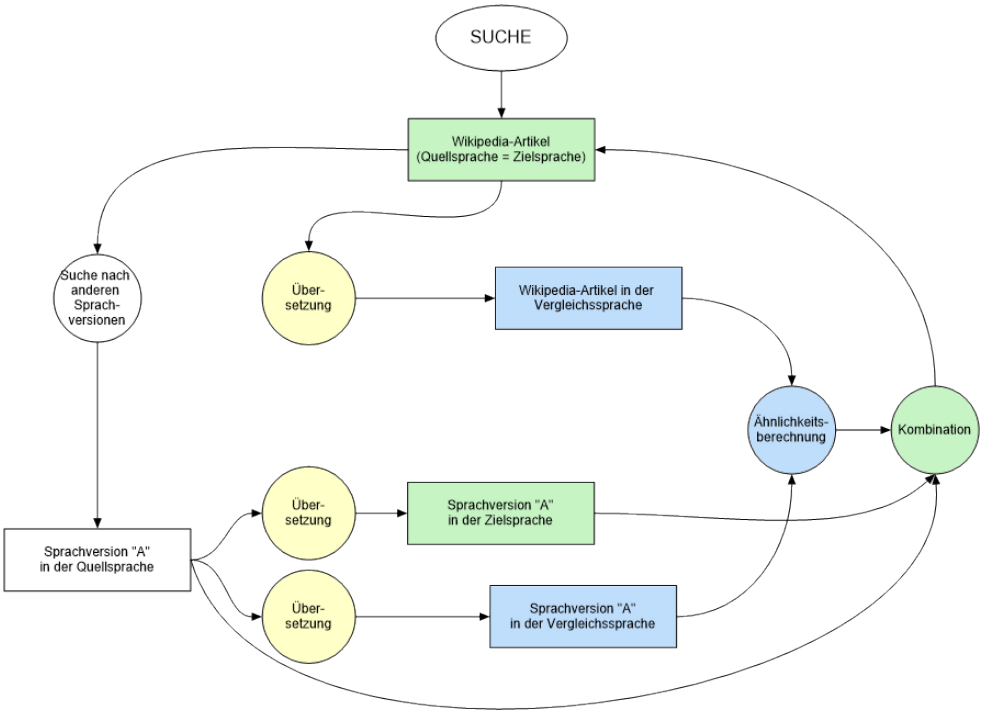

Abbildung 1.1: Übersetzungs-Konzept: blau zeigt Übersetzungen in der Vergleichssprache, grün in der Zielsprache

KAPITEL 2

Ähnliche Arbeiten

Bei der Suche nach diesem Thema werden unter anderem Plugins für Brower mit Interlingualer Unterstützung angeboten. Eines der interessantesten davon ist die Chrome Extension "Google Translate". Diese erlaubt es dem Anwender auf Knopfdruck die gerade betrachtete Webseite unter Nutzung der Google Translate API zu übersetzen. (Die Extension ist unter dem Code `aapbdbdomjkkjkaonfhkkikfgjllcleb` im Chrome Web Store erhältlich.)

Ein Tool mit einer ähnlichen Zielsetzung wie das der vorliegende Arbeit wird unter dem Namen Manypedia angeboten [MS]. Es erlaubt die gleichzeitige Darstellung von zwei Sprachversionen eines Wikipedia-Artikel. Im Unterschied zum Werkzeug, das dieser Arbeit zu Grunde liegt, beschränkt man sich auf die Präsentation und Übersetzung dieser Artikel (siehe Abb. 2.1). Wilangyman integriert die gefundenen Textblöcke anhand der berechneten Ähnlichkeiten in den Ursprungsartikel. Somit muss sich der Nutzer nicht mit der Filterung annähernd gleichen Textblöcke befassen. Manypedia scheint sich auch mit Textähnlichkeiten zu befassen, was aber mit aktuellen Browsern nicht erfolgreich getestet werden konnte.

Wikitranslate [NOH+09] beschreibt ein System, das die Abfrageübersetzung für den Abruf von mehrsprachigen Informationen (CLIR – Cross Language Information Retrieval) nur mit Hilfe von Wikipedia durchführt, um Übersetzungen zu erhalten. Mit CLIR und dessen Werkzeugen, Herausforderungen und Übersetzungsansätzen beschäftigt sich auch eine Arbeit von Sharma und Mittal [SM16]. Diese hebt hervor, wie wichtig die multilinguale Datensammlung sowie die Sprachbarriere übergreifende Suche nach Informationen ist – besonders unter dem Gesichtspunkt der immer weiter fortschreitenden Globalisierung.

Suzuki et al. haben sich mit dem Herausfiltern von komplementären Informationen aus mehrsprachigen Wikipedia-Artikeln in guter Qualität beschäftigt, [SFKN12]. Die Bewertung der Güte steht dabei im Vordergrund. Zur Bewertung wird unter anderem die Informations-Lebensdauer herangezogen. Diese wird in eine Autorenbewertung eingerech-

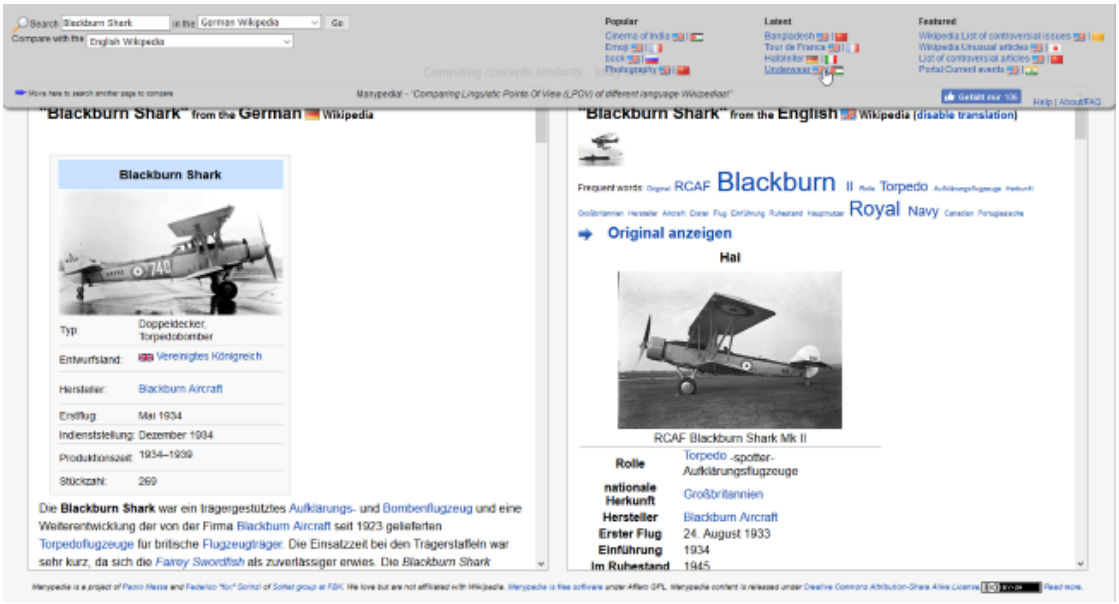

Abbildung 2.1: Screenshot von manypedia.com - Gegenüberstellung zweier Wikipedia Sprachversionen

net. Man geht davon aus, dass Autoren, die langlebige Texte schreiben, höher zu bewerten sind als andere. Da es nicht klar ist, woher diese Daten bezogen werden, bewertet diese Arbeit die Relevanz der Information ausschließlich basierend auf Text-Ähnlichkeiten.

Eine Methode, das sich ebenfalls mit der Gegenüberstellung von Wikipedia-Artikelpaaren beschäftigt, ist Multiwiki [GD17]. Das Ziel der Autoren von der Uni Hannover ist, Passagen gleichen Inhalts zu markieren und die Gemeinsamkeit graphisch aufzuzeigen. Es wird eine Demo angeboten bei der man aus neun vorbereitenen Artikeln auswählen kann [GD] (siehe Abb. 2.2). Von den versprochenen vier Sprachpaaren fanden sich zum Zeitpunkt des Aufrufes allerdings nur die bereits erwähnten neun Vergleichsartikel. Multiwiki vergleicht aber auch die in den Artikel-Paaren vorhandenen Bilder und Links. Es wird gegenübergestellt, welche Autoren aus welchen Ländern wie oft die Ressourcen bearbeitet haben und es wird nachverfolgt, wie ähnlich sich die Artikel im Laufe ihrer Entwicklung waren. Es geht hier also mehr um Analyse des vorhandenen Datenmaterials als um die Präsentation von interessanten Ergänzungen.

Ein ähnliches Problem stellt der Vergleich verschiedener Revisionen eines Dokuments oder eines Artikles dar. Will man einen Verlauf in der Entwicklung eines Dokumentes visuell aufbereiten, bieten sich Animationen an [CDBF10]. Hier werden animierte Darstellung von Textänderungen behandelt. Als Anwendungsbereich wird unter anderem die Wikipedia-Artikel-Historie genannt. Durch die animierte Darstellung werden die Änderungen zwischen Versionen dem Nutzer leicht erfassbar zur Verfügung gestellt.

Aber auch bei Dokumenten gibt es eine Entwicklungsgeschichte. Für Google-Docs bietet

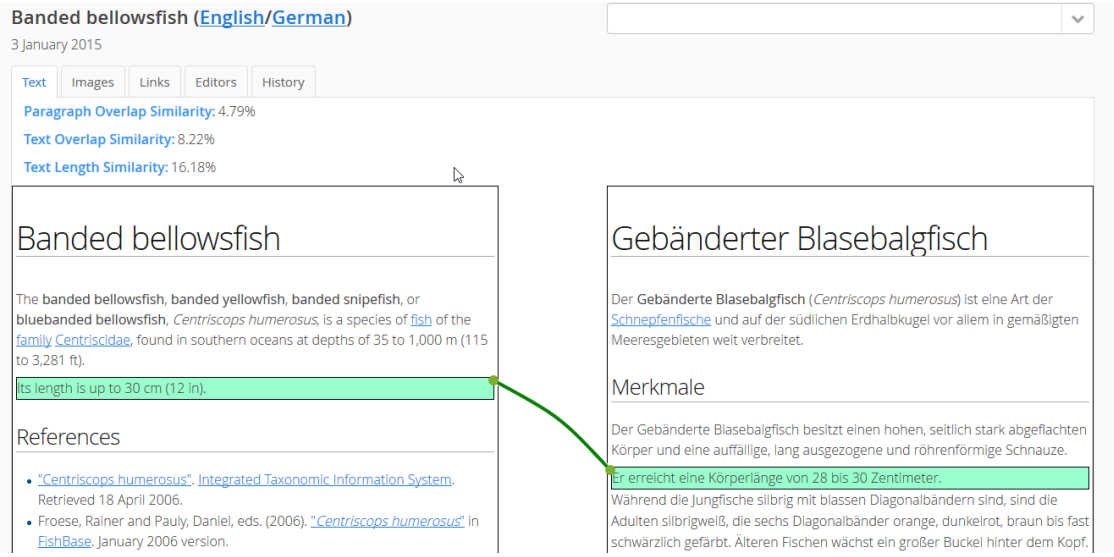

Abbildung 2.2: Screenshot von der Multiwiki Demoseite - Hervorhebung gleicher Information.

sich ein Tool namens DocuViz [WOZ+15] an. Es visualiert die Änderungsgeschichte dieser Online-Textverarbeitung.
Bei den zwei letzteren zeigt sich der große Unterschied zwischen der Differenz von linear entstehenden Dokumenten und parallel bearbeiteten Sprachversionen. Das Hervorheben von Unterschieden kann nicht in der selben Art funktionieren. Die einzelne Sprachversion entwickelt sich zwar linear weiter, aber das selbe gilt auch für jede andere Version. Dadurch werden die Unterschiede im Laufe der Zeit wahrscheinlich immer größer werden. Will man dennoch Vergleiche anstellen, muss man versuchen, den Inhalt automatisch zu interpretieren.

KAPITEL 3

Konzept

Wilangyman soll es erleichtern, Informationen aus Wikipedia über Sprachbarrieren hinweg darzustellen. Dazu werden fremdsprachliche Artikel aus der Enzyklopädie nach zusätzlichen Inhalten durchsucht. Diese werden an der Stelle in den Ausgangsartikel eingefügt, wo sie am besten dazu passen. Um diese Stellen zu finden werden die im Artikel enthaltenen Überschriften beider Sprachversionen miteinander verglichen. Der Absatztext wird bei der am besten passenden Überschrift des Originalartikels eingefügt. Um mögliche Redundanzen zu vermeiden werden diese Texte auf Ähnlichkeiten überprüft und nur unter einem definierten Schwellwert dargestellt. Da Wikipedia Artikel typischerweise eine Einleitung ohne dedizierte Überschrift haben, wird diese direkt zugeordnet.

Da die Installation zusätzlicher Software vermieden werden soll, bietet sich die Plugin-Technik an, um Browser um die gewünschte Funktionalität zu erweitern. Diese Browser-Extension ist nur bei Wikipedia-Webpages aktiv. Dadurch wird verhindert, dass der Aufruf zu Fehlern bei anderen Seiten führt. Bei Wikipedia-Seiten wird der Titel und die Sprache dieser Seite (Ausgangssprache, Zielsprache) zur Verarbeitung weitergegeben. Nun wird via Wikipedia-API abgefragt welche anderen Sprachversionen verfügbar sind. Diese werden - entsprechend der Erweiterungs-Konfiguration abgerufen - und, ebenso wie die Ursprungsversion in Absätze unterteilt. Hierbei wird der Text zwischen zwei Hauptüberschriften (H2-Tags, siehe Abb. 3.1) als Absatz verstanden. In der vorliegenden Version werden dabei die HTML-Tags *P, H3, H4, UL, OL, LI* erkannt. Alle übrigen werden ignoriert. Die Unterscheidung ist nötig, damit die Tags nicht in die Übersetzung einfließen.

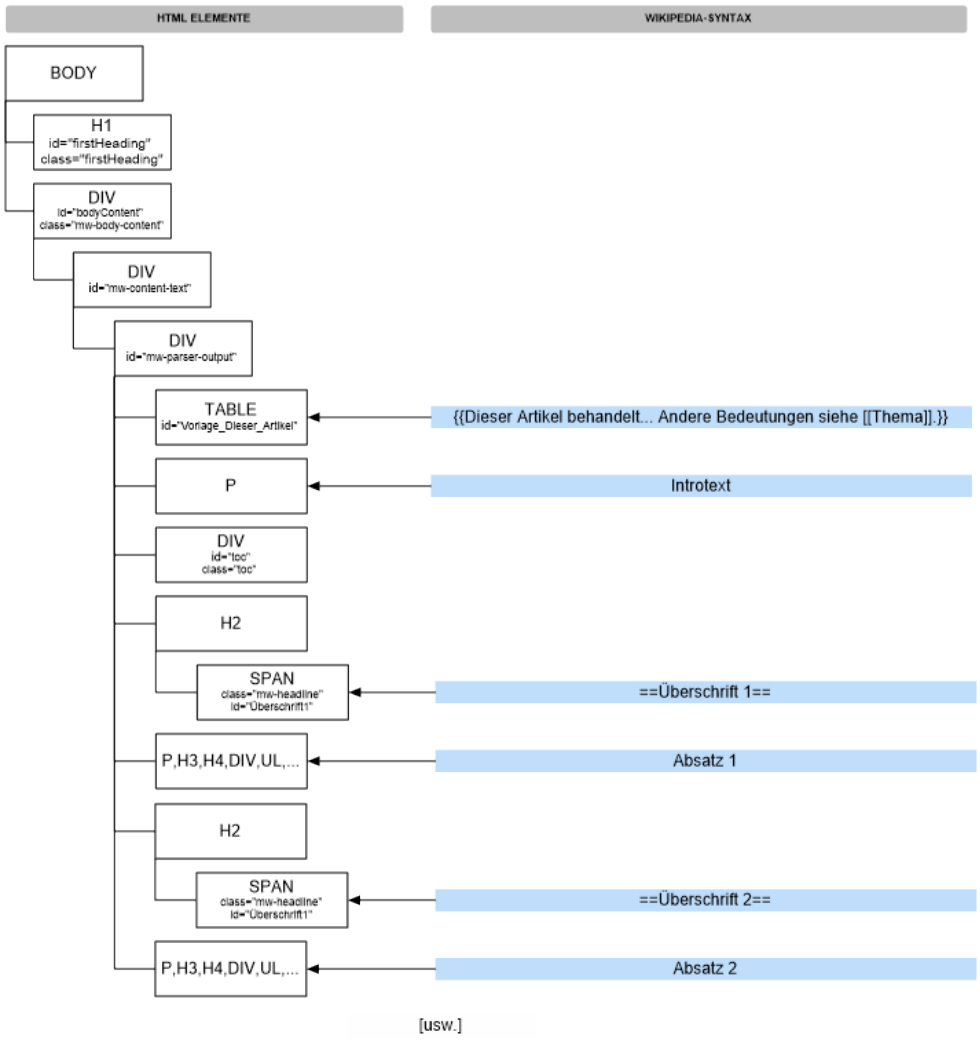

Abbildung 3.1: Aufbau Wikipedia-Artikel.

Die Überschriften und Textblöcke werden mit Hilfe von Maschinen-Übersetzung sowohl in die Zielsprache als auch in der Vergleichssprache übersetzt. (siehe auch Abb. 1.1) Als Vergleichssprache wird Englisch bevorzugt, da die verwendeten Natural Language Processing-Tools diese vorrangig nutzen. Natürlich könnte bei entsprechender Unterstützung auch jede andere Sprache dazu verwendet werden. Die Absätze werden anschließend in der Vergleichssprache miteinander verglichen und deren Ähnlichkeit bewertet. Es kann also auch notwendig sein Artikel zweimal zu übersetzen, was die folgenden Szenarien verdeutlichen sollen.

- Kommt der Ausgangsartikel von der deutschen Wikipedia (DE), der fremdsprachliche aus der englischen (EN), sind folgende Übersetzungen nötig:

1. DE in EN, da die englische Übersetzung des deutschen Artikels für den Vergleich benötigt wird.
2. EN in DE für die Augmentierung.

- Kommt der Ausgangsartikel aus der deutschen (DE) Wikipedia, der fremdsprachliche aus der französischen (FR), sind folgende Übersetzungen nötig:

  1. DE in EN, da die englische Übersetzung des deutschen Artikels für den Vergleich benötigt wird.
  2. FR in EN, da die englische Übersetzung des deutschen Artikels für den Vergleich benötigt wird.
  3. FR in DE für die Augmentierung.

Sobald die Artikel in der Vergleichssprache vorliegen, werden die Texte verglichen. Dieser Vergleich wird sowohl auf Überschriften angewandt als auch auf den Absatz selbst. Dabei zielt ersteres darauf ab, ein direktes Matching zu finden - also zB. "Publications" und "Veröffentlichungen". Die Zuordnung ergibt sich aus den Überschriften, die die jeweils höchste Ähnlichkeit aufweisen. Die Absatzähnlichkeit wiederum dient dazu, zusätzliche Informationen zu finden. Die Auswertung wird der Browser-Extension übergeben, die anhand der berechneten Ähnlichkeit die übersetzten Texte in den Ursprungsartikel einfügt. Dabei wird beim Ähnlichkeitswert ein vorgegebener Schwellwert berücksichtigt. Dieser dient dazu, dass nur zusätzliche und keine redundante Information angezeigt wird. Die Absatzähnlichkeiten werden nur für Absätze mit der besten Überschriftenzuordnung berechnet. Sollte keine Zuordnung gefunden werden, wird der Absatz am Ende unter "Nicht zugeordnet" eingefügt. Der Einleitungsabsatz des Artikels verfügt über keine Überschrift, dieser Text wird direkt zugeordnet. Eine Zuordnung zum am besten passenden Textblock – also mittels Absatzähnlichkeit – ist derzeit nicht angedacht.

3.1 Natural Language Processing (NLP)

Um einen Text aus einer natürlichen Sprache für einen Computer verarbeitbar zu machen, wird der Eingabetext in eine Datenstruktur umgewandelt. Sozusagen wird eine einfachere Darstellung extrahiert, die auch syntaktische Informationen enthält. So geben Part-of-Speech-Markierungen Auskunft über die Wortart und es wird versucht, Eigennamen zu erkennen. Aber auch semantische Informationen wie der Sinn eines Wortes (Apple als Obst oder als Institution) sind wesentlich. Mehrere Frameworks können dazu genutzt werden. Für diese Arbeit kamen NLTK (Natural Language Toolkit) für Python, das sich besonders für die Lehre im Bereich Computational Lingustics empfielt, als auch SpaCy, das einen eher praxisorientierten Ansatz verfolgt, in Frage.

Der Text durchläuft ein Abfolge von NLP-Operationen ("NLP-Pipeline"), wodurch nach und nach die wichtigen Elemente aus dem Text extrahiert werden [CEE+10, 264ff]. Unter anderem können folgende Verarbeitungsschritte und Methoden dazu genutzt werden:

- Tokenisierung
  Das Aufteilen eines Textes in Tokens, d.h. in die sinnvollen Teile einer Eingabe. Dabei ergeben sich schon beim Erkennen der Satzgrenzen Fragen, wie z.B. bedeutet jeder Punkt ein Satzende (“10. Oktober”, “10 a.m.”,“bzw.”, “Mag.”, “Web 2.0”)? Dabei sind auch sprachspezifische Feinheiten zu berücksichtigen z.B. hat 9,876.543,00 in deutscher Schreibweise dieselbe Bedeutung wie 9 876 543,00 im Französischen, wo man große Zahlen mit Leerzeichen trennt. Existierende Implementierungen von Tokenizer verwenden sprachspezifische Regeln um die Punktuation richtig zu erkennen.

- Stopwords entfernen
  Stopwords sind Wörter, die häufig auftreten und meist keine Relevanz für die Erfassung haben. Übliche Stopwords sind bestimmte und unbestimmte Artikel, Konjunktionen und Präpositionen.

- Wortarterkennung “PoS” - Part-of-Speech-Tagging
  Dabei werden den gefundenen Tokens grammatikalische Informationen - eine Klassifikation anhand der Wortart - beigefügt. Eine Liste der von SpaCy verwendeten PoS-Tags findet man auf der SpaCy-Homepage [Expa]. Von diesen werden zur Zeit “SPACE” und “PUNCT” aus dem Text entfernt.

- Eigennamenerkennung “Named Entity Recognition”
  Dabei kann man auf gewisse Schwierigkeiten stoßen. Im Gegensatz zu “IBM” oder “Charles Darwin” ist es bei “Apple” nicht ganz so eindeutig. Ist nun die Frucht gemeint oder doch der Computerriese? Eigennamen sind wichtige Anhaltspunkte zum Vergleichen von Texten. Wenn die selben Personen, Institutionen oder Gegenstände in zwei Texten genannt werden, lässt das auf eine inhaltliche Nähe schließen.

- Lemmatisierung
  Wörter werden auf ihre Grundform, also Verben zur Infinitiv, Substantive zu Nominativ Singular zurückgeführt. Dieser Bearbeitungsschritt findet bei Wilangyman derzeit nur für Überschriften Anwendung, da gerade bei besonders kurzen Texten sonst Übereinstimmungen schwerer zu finden wären.

Der Vergleich von Überschriften verfolgt andere Ziele als der Vergleich von Fließtext.

3.1.1 Vergleichsmethoden

Es gibt unterschiedliche Methoden die Ähnlichkeit von Texten zu berechnen. Meist werden erst die Wörter erfasst und gezählt. Zuvor werden Stopwords entfernt. Diese “Bag-of-Words”-Methode ist weit verbreitet und resultiert in einem n-dimensionalen Feature-Vektor. Diese Vektoren können dann mit unterschiedlichen Abstandsmaßen verglichen werden [MCS+06]. Die am häufigsten verwendeten Distanzmaße sind:

- Jaccard-Index
  Hier handelt es sich um eine Kennzahl für die Ähnlichkeit von Mengen. Er berechnet sich durch Teilung der Schnittmenge durch die Vereinigungsmenge.

  $$J(A, B) = \frac{A \cap B}{A \cup B}$$

  Demnach liegen die errechneten Werte zwischen 0 (keine gemeinsamen Elemente) und 1 (nur gemeinsame Elemente). Es wird die ungewichtete Variante eingesetzt. Daher wird die Häufigkeit der Tokens außer Acht gelassen. Beispiel: “Ehrungen” wird zu “honors” übersetzt. A enthält nun den Token “honor”. Im englischen Artikel wird ein Absatz mit der Überschrift “awards and honors” gefunden. Nach Entfernen der Stop Words und Lemmatisierung enthält B die Tokens “award” und “honor”.

  $$J(A, B) = \frac{1}{2} = 0.5$$

- Kosinus-Ähnlichkeit
  Der Winkel zwischen zwei Vektoren wird bestimmt wobei ein Wertebereich von -1 (entgegengesetzt) und +1 (gleichgerichtet) abgedeckt wird. 0 (orthogonal) repräsentiert Unabhängigkeit.

  $$\cos \vartheta = \frac{A \cdot B}{|A|\,|B|}$$

  Da hier A und B Häufigkeitsvektoren darstellen, deren Werte nie negativ sein können, liegt auch die Kosinus-Ähnlichkeit stets zwischen 0 und 1.

A stellt hier die Tokenmenge des Ursprungsartikels dar, B die des Fremdsprachlichen Artikels. Die beiden Tokenlisten liegen in der Vergleichssprache vor.

3.1.2 Überschriften

Mit dem Vergleich von Überschriften unterschiedlicher Sprachversionen soll ein möglichst gutes Matching erreicht werden. Optimaler Weise sollen - im einfachsten Fall - der Absatz “Biographie” einer Seite aus der deutschen Wikipedia mit dem Absatz “Biography” im englischen Pendant auf Überschriftenebene übereinstimmen. Die meisten Überschriften bestehen aus Schlagwörtern, die dem Leser eine Ahnung des nachfolgenden Textes geben. Die Einbeziehung von Named Entities sowie die Entfernung von Stopwords (“Die Geschichte” vs “Geschichte”) sind von Interesse. Zur Anwendung kommt der ungewichtete Jaccard-Index.

Der Überschriftenvergleich liefert die Zuordnung zu den entsprechenden Stellen im Ausgangsartikel. Sollte keine Übereinstimmung gefunden werden, wird der Absatz am Ende unter “Nicht zugeordnet” eingefügt. Es wird also jedem Absatz aus dem Ergänzungsartikel genau eine Überschrift im Originalartikel (ggf. “Nicht zugeordnet”) zugewiesen. Jeder Abschnitt im Originalartikel kann $0..n$ Absätze aus dem Ergänzungsartikel aufnehmen (n steht für die Anzahl der Absätze im Ergänzungsartikel).

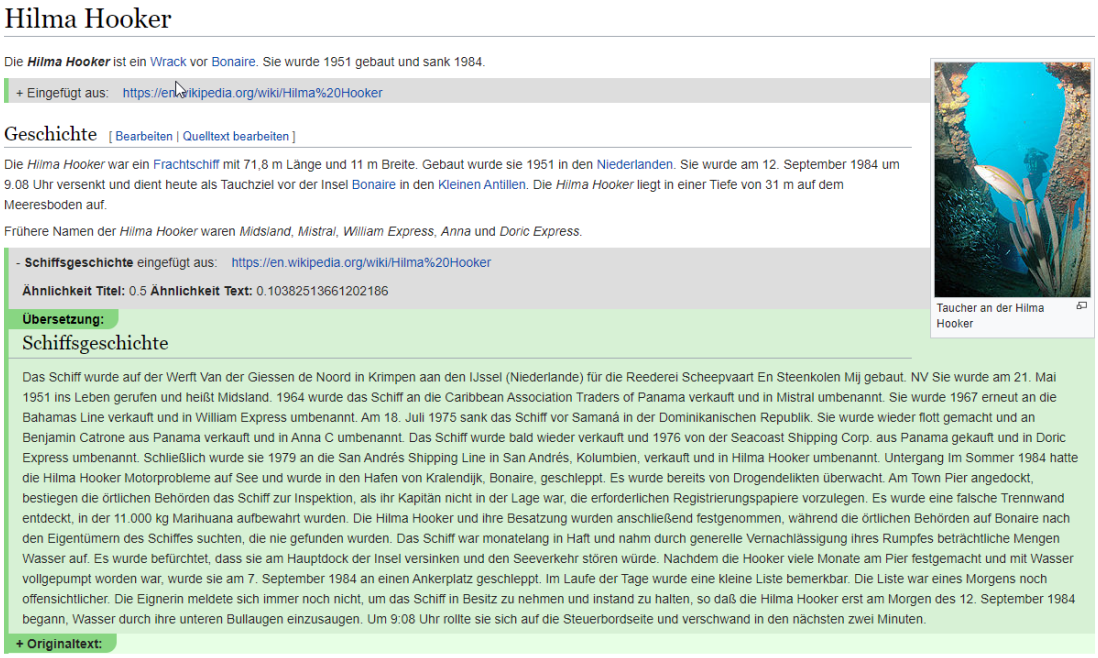

Abbildung 3.2: Screenshot Wilangyman-Ausgabe.

3.1.3 Fließtext

Ist eine gut passende Überschrift für einen Textblock gefunden, wird die Ähnlichkeit des Fließtextes berechnet. Sinnvolle Ergänzungen - also Texte mit geringer Ähnlichkeit - sollen angezeigt werden. Texte mit hoher Ähnlichkeit bieten vermutlich wenig neue Information und werden daher ausgeblendet. Dafür wird die similarity-Methode von SpaCy verwendet. Diese nutzt die Kosinus-Ähnlichkeit – je Sprachmodell – mit Wort-Tensoren oder Vektoren (ab Modell “en_core_web_md” [Expb]).

3.2 Präsentation

Die übersetzten Texte werden direkt in die HTML-Seite des Wikipedia-Artikels eingefügt. Ob ein Text dargestellt werden soll wird mittels Schwellwert entschieden. Ist Ähnlichkeit des Fließtextes unter diesem – mit 0.5 voreingestellten – Schwellwert, wird dieser dargestellt. Somit wird sicher gestellt, dass nur Texte, die zusätzliche Informationen enthalten auch sofort sichtbar dargestellt werden. Für BenutzerInnen, die der Maschinenübersetzung nicht ganz vertrauen bzw. wenn es zu Unschärfen in der Übersetzung kommen sollte, kann der Originaltext sichtbar geschaltet werden. Beide unsichtbaren Texte (übersetzter Text und Originaltext) werden nur auf Nutzeranforderung angezeigt (siehe Abb. 3.2).

KAPITEL 4

Implementierung

Wilangyman unterteilt sich in einen Clientteil, der aus einer Extension für den Web Browser Chrome (Version 79.0.3945.130, 64-Bit) besteht, sowie einen Serverprozess. Der Client übermittelt dem Server den Namen des Wikipedia-Artikels sowie dessen Sprache und fordert damit die Ergänzungen an. Am Server wird mittels Übersetzungstools und Natural Language Processing eine entsprechende Antwort generiert, die an den Client zurückgesandt und von diesem in den Originalartikel integriert wird. Für den Serverdienst bietet sich Python, dank der verfügbaren Libraries für Natural Language Processing (NLP), als Sprache an. Python ist aber nicht im Brower verfügbar. Ein Programm, geschrieben in dieser Programmiersprache, kann daher für diese Anwendung nur als Serverdienst ausgelegt werden. Außerdem wird durch die Auslagerung der wesentlichen Logikteile in einen Serverprozess eine spätere Vermarktung vorbereitet.

4.1 Chrome Extension

Wir brauchen die Möglichkeit, die aktuelle Webseite des Benutzers auszulesen und zu verändern. Dies lässt sich mit einem Webbrowser-Plugin realisieren. Die Wahl des Browsers fiel auf Google Chrome, da dieser der am weitesten verbreitet ist [Ten].

In der Erweiterung wird eine Befehlsfläche definiert, die den Abruf der zusätzlichen Informationen startet. Dabei werden die nötigen Daten - Artikelname, geforderte Sprachen u.s.w. - an den Serverprozess übertragen. Dessen Antwort enthält die übersetzten und originalen Absätze der fremdsprachigen Wikipediaseite sowie die Ähnlichkeitswerte, an Hand derer die Texte in den Originalartikel eingefügt werden (siehe Abb 4.1). Zur Zeit ist die Sprachauswahl im Sourcecode hinterlegt. Als Ausgangsartikelsprache wird Deutsch erwartet, zur Informationserweiterung muss ein englischsprachiger Artikel vorliegen.

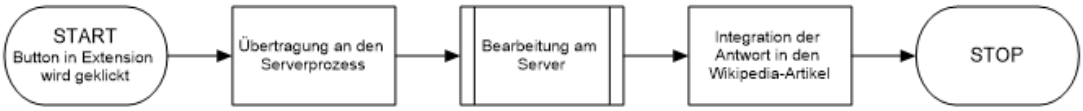

Abbildung 4.1: Ablaufdiagramm Chrome Extension

Dazu können Komponenten, wie Backgroundskripts, Contentscripts, Konfigurationsseiten und andere erstellen. Die verwendenten Technologien dahinter sind HTML, CSS und Javascript. Den Beginn macht in jedem Fall die Datei “manifest.json” (Siehe Listing 4.1)

```
{
    "manifest_version": 2,
    "name": "Wilangiman",
    "description": "Füge passende Absätze aus anderen Wikipedia-
    Sprachversionen hinzu",
    "author": "Wolfgang Gundacker",
    "version": "0.1",
    "permissions": ["declarativeContent", "activeTab", "*://127.0.0.1/*"],
    "background": {
      "scripts": ["lib/jquery-3.4.1.js", "background.js"]},
  "browser_action": {
    "default_title": "Wilangyman"},
  "content_scripts": [{
     "matches": ["https://*.wikipedia.org/wiki/*"],
     "all_frames": true,
     "css": ["content.css"],
     "js": ["content.js"]}]
}
```

Listing 4.1: manifest.json

Im Manifest werden alle Komponenten und Informationen des Plugins erfasst. Man findet die verwendeten Background-Skripts, die Komponenten des Content-Skripts (Javascript-Datei und CSS-Datei). Natürlich sind auch Autor und Version vermerkt. Da Webseiten nicht HTTP-Requests an andere als den gerade besuchten Server absetzen dürfen, Wilangyman aber mit einem Serverdienst auf einem anderen Server kommuniziert, muss diese Erlaubnis gesetzt werden. Das Attribut “permissions” erlaubt dies für einen lokal betriebenen Serverdienst. Sollte Wilangyman auf einem, im Web erreichbaren Server gehostet werden, muss dieser Eintrag angepasst werden.

Das Background-Skript ist für das Management von Browserevents zuständig. Es dient der Kommunikation mit dem Browser aber nicht der Seite. In diesem Skript kann wird unter anderem der Zugriff auf Custom-Buttons genommen werden und es kann auch ausgefiltert werden, von welchen URLs man die Extension verwenden darf.

```
1 var host = "http://127.0.0.1";
2 var port = 5000;
3 var basepath = "https://de.wikipedia.org/wiki/";
4
```

```
5  chrome.runtime.onInstalled.addListener(function() {
6      chrome.declarativeContent.onPageChanged.removeRules(undefined, function()
       {
7          chrome.declarativeContent.onPageChanged.addRules([ {
8              conditions: [
9                  new chrome.declarativeContent.PageStateMatcher({
10                     pageUrl: { urlContains: basepath },
11                 })],
12             actions: [ new chrome.declarativeContent.ShowPageAction() ]
13         }]);
14     });
15 });
16
17 // Called when the user clicks on the browser action.
18 chrome.browserAction.onClicked.addListener(function(tab) {
19     if ( tab.url.includes(basepath) ) {
20     $(document).ready( function() {
21         $.ajax({
22         url: host+":"+port+"/wikiurl/"+url+"?uselang=en&;srclang=de"
23       }).then(function(data) {
24         chrome.tabs.sendMessage(tab.id,data);
25       })
26     })
27     }
28 });
```

Listing 4.2: background.js

Die Conditions ab Zeile 9 (Listing 4.2) beschränken den Zugriff auf Seiten des basepaths, auf die deuschen Wikipedia ein. Die Abfrage-URL wird anschließend aufgebaut und mit den Argumenten “uselang” und “srclang” ausgestattet. Diese sind in der vorliegenden Version fix und können nur über den Sourcecode geändert werden. Nach Einlangen der Antwort, einem vom Serverteil generierten JSON-Objekt, wird diese an den nächsten Baustein weitergereicht. Denn um Zugriff auf das HTML-Document-Objekt zu erlangen, muss man sich eines Content-Skripts bedienen. Dieses markiert als ersten Schritt alle in der geladen Seite vorhanden H2-Tags mit neuen -Elementen. Diese werden dann mit den eben erhaltenen Zusatzinformationen - entsprechend der Titel-Ähnlichkeit befüllt. Die Absatzähnlichkeit bestimmt ob die Passage sofort angezeigt wird oder vorerst eingeklappt bleibt. Dazu wird ein, zur Zeit hardcoded gesetzter Schwellwert verwendet. Ist die Textähnlichkeit über dem Schwellwert, wird eingeklappt.

4.2 Serverdienst

Der Serverprozess erwartet die Übermittlung von dem Titel der Wikipediaseite sowie die Sprachcodes der anzufordernden Sprachversionen. Wikipedia wird daraufhin nach dem Vorhandensein dieser Versionen überprüft und die fremdsprachigen Artikel werden geladen. Diese werden absatzweise einer Übersetzungs-Engine zugeführt und einer Abfolge von NLP-Berechnungen unterzogen. Am Ende wird die Antwort im JSON-Format an den Client zurückgegeben (siehe Abb. 4.2).

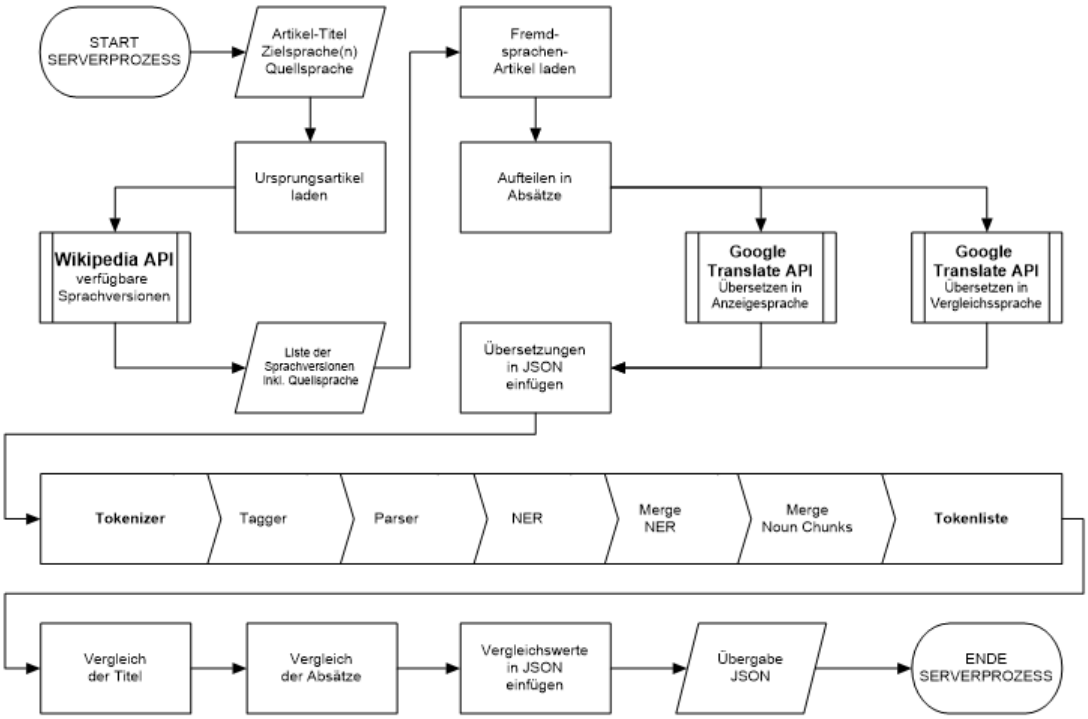

Abbildung 4.2: Ablaufdiagramm Serverprozess inkl. NLP-Pipeline.

Für die Webanfragen wird auf ein bestehendes Framework zurückgegriffen. “Flask” erlaubt die einfache Verarbeitung von REST-Anfragen (REpresentational State Transfer) [Ron10] (Siehe Listing 4.3) .

```
1 from flask import Flask, request
2 app = Flask(__name__)
3
4 @app.route('/wikiurl/<url>')
5 def wiki(url):
6     arg = request.args.get('uselang')
7     uselang = '["en","de"]' if arg is None else arg.split(",")
```

Listing 4.3: Flask Server Code

Durch die Verwendung des Decorators `@app.route('/wikiurl/<url>')` sowie einer passenden Methode (wiki) kann beispielsweise auf den Aufruf `127.0.0.1:5000/wikiurl/Herbert_Benson?uselang=en` reagiert werden. Die übermittelten Argumente (im Beispiel ’uselang’) können mit `request.args.get` abgefragt werden.

4.3 Wikipedia API

Die Programmierschnittstelle von Wikipedia erlaubt eine große Zahl von Abfragen. Die veröffentlichte Gesamtmenge ist auf der MediaWiki-Seite [Wikb] zu finden. In Wilangyman gelangt die Abfrage nach vorhandenen Sprachversionen des vorliegende Artikels zum Einsatz.

Auf der Suche nach dem, im Kapitel 5.1 (Seite 28ff) genutzten Artikels zu Herbert Benson ergäbe sich daher folgende Abfrage:
`https://de.wikipedia.org/w/api.php?action=query&prop=langlinks&titles=Herbert_Benson&format=json&lllimit=100`

Auch hier erhält man - wie in der Abfrage verlangt - die Antwort im JSON-Format:

```
{
    "batchcomplete":"",
    "query": {
        "normalized":[{
            "from":"Herbert_Benson",
            "to":"Herbert Benson"}],
        "pages":{
            "9877977":{
                "pageid":9877977,
                "ns":0,
                "title":"Herbert Benson",
                "langlinks":[
                    {"lang":"ar","*":"\u0647\u0631\u0628\u0631\u062a \u0628\
    u0646\u0633\u0648\u0646"},
                    {"lang":"en","*":"Herbert Benson"},
                    {"lang":"fr","*":"Herbert Benson"},
                    {"lang":"ru","*":"\u0411\u0435\u043d\u0441\u043e\u043d, \
    u0413\u0435\u0440\u0431\u0435\u0440\u0442"}
                ]
      }
        }
    }
}
```

Listing 4.4: Wikipedia Sprachversionen Antwort

Unter “langlinks” in Listing 4.4 finden sich die vorhandenen Sprachen. Im konkreten Fall existieren also Seiten in Arabisch (ar), Englisch (en), Französisch (fr) und Russisch (ru). Der Seitentitel ist im Falle von Englisch und Französisch mit der deutschen Variante ident, auf Arabisch und in Russisch werden kodierte Unicode-Zeichen angegeben. Die Ausgangssprache (hier Deutsch) ist in der Antwort ausgenommen.

Die Beschränkung der zurückgegebenen Links hat den Default-Wert 10. Will man mehr, kann man diesen Wert mit dem Parameter “lllimit” anpassen. Da nicht garantiert ist, dass sich die gesuchte Sprache (hier Englisch) innerhalb der ersten 10 Ergebnisse befindet, wird ein Limit von 100 verwendet.

Für die Benchmarks in Kapitel 5 wird eine Abfrage zur Anzeige der Wortanzahl eines Artikels verwendet. Mit der URL `https://de.wikipedia.org/w/api.php?action=query&list=search&srsearch=Herbert_Benson&srprop=wordcount&format=json` erhält man als Antwort folgende Datenstruktur:

```
{
    "batchcomplete": "",
    "query": {
        "searchinfo": {
            "totalhits": 379
        },
        "search": [
        {
            "ns": 0,
            "title": "Herbert Benson",
            "pageid": 9877977,
            "wordcount": 1726
        },
            ...
        ]
    }
}
```

Listing 4.5: Wortanzahl

Die Anzahl der Treffer mit 379 enthält auch inhaltlich passende Seitentitel wie “Benson-Meditation” oder sogar “Arterielle Hypertonie” (aus Listing 4.5 entfernt, da nicht relevant). Der gewünschte Wert findet sich unter "wordcount im konkreten Fall also 1726.

Diese Ergebnisse dieser Abfrage werden für das Kapitel 5 verwendet, sind für die eigentliche Funktionalität aber nicht erforderlich.

4.4 Google Translate API

Im Internet finden sich bereits einige Werkzeuge, die eine automatische Übersetzung von natürlicher Sprache erlauben. Nicht alle bieten eine API. Sofern eine Programmierschnittstelle angeboten wird, ist diese meist kostenpflichtig. So auch die, für dieses Projekt gewählte – Google Translate API [Goob]. Die angebotenen Werkzeuge nutzen unterschiedliche Techniken für die Übersetzung. Google beispielsweise verwendetet bis Ende 2016 noch eine statistische Methode zur Übersetzung. Seither setzt man bei dem Internetgiganten auf neuronale maschinelle Übersetzung [Tur16]. Das Thema der maschinellen Übersetzungsmethoden sowie deren Vor- und Nachteile sind ein umfassendes Thema und würden den Umfang dieser Arbeit bei weitem sprengen. Exemplarisch sei hier ein Artikel erwähnt der sich mit dem Vergleich statistische Machine Translation und dem neuralen Ansatz auseinandersetzt [BBCF16].

Um die Google Translate Dienste nutzen zu können bieten sich zwei Wege an. Der eine ist der über einen HTTP-Aufruf. Der andere ist eine Library zu nutzen. Die Abrechnung basiert auf der Nutzungsmenge, wobei bis zu 500.000 Zeichen kostenlos angeboten werden.

```python
import json, requests

# Routine zum Übersetzen von Text mit Hilfe von Google-Translate API V2
# param text String der zu übersetzende Text
# param source String Sprachcode von text (de, en, ...)
# param target String Sprachcode der Zielsprache
# param key String Google-API Lizenz-Key
# return String translated Text
def translate(text, source, target, key):
    tp = "https://translation.googleapis.com/language/translate/v2?key="+key
    trans = requests.post(tp + "&q=" + text + "&source=" + source + "&target=
    " + target)
    jtrans = json.loads(trans.text)
    trans = jtrans["data"]["translations"][0]["translatedText"]
    return trans
```

Listing 4.6: Übersetzungsroutine nutzt Google Translate

Ein Beispielaufruf:
`https://translation.googleapis.com/language/translate/v2?key=[key]&q=The%20European%20Union%20on%20Wednesday%20fined%20Microsoft%20EUR%20497.2%20million%20($611%20million)%20for%20abusing%20its%20%E2%80%99near%20monopoly%E2%80%99%20with%20its%20Windows%20operatingsystem%20to%20crush%20competitors%20and%20gain%20the%20upper%20hand%20in%20markets%20for%20digital%20media%20players%20and%20low-end%20servers.&source=en&target=de`

(The European Union on Wednesday fined Microsoft EUR 497.2 million ($611 million) for abusing its 'near monopoly' with its Windows operatingsystem to crush competitors and gain the upper hand in markets for digital media players and low-end servers)

Als Antwort erhält man ein JSON-Objekt mit dem übersetzten Text:

```
{
  "data": {
    "translations": [
      {
        "translatedText": "Die Europäische Union hat gegen Microsoft am
    Mittwoch eine Geldbuße in Höhe von 497,2 Mio. EUR (611 Mio. USD) verhängt
    , weil sie ihr &quot;Beinahe-Monopol&quot; mit ihrem Windows-
    Betriebssystem missbraucht hat, um Wettbewerber zu vernichten und die
    Oberhand auf den Märkten für digitale Mediaplayer und Low-End-Server zu
    gewinnen.",
      }
    ]
  }
}
```

Listing 4.7: Google Translate Antwort

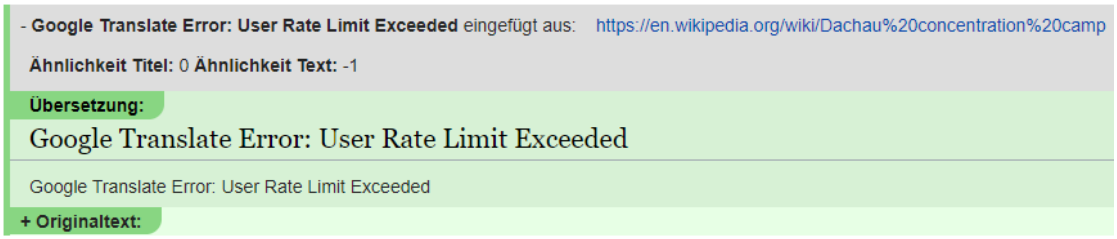

Abbildung 4.3: Google Translate: Kontingent überschritten.

Die Aufrufe erfolgen an den mit “Google Translate API"bezeichneten Stellen in Abb. 4.2. Um die Durchlaufdauer bei wiederholtem Aufruf einer Seite zu reduzieren wurde ein Übersetzungscache implementiert. Bereits übersetzte Stellen können auf diese Art lokal geladen werden und müssen nicht an Google Translate geschickt werden. Dieser Cache kann mit der globalen bool’schen Variable `usecache` gesteuert werden.

Bei vielen aufeinander folgenden Übersetzungsanfragen wird seitens Google ein Fehler (“User Rate Limit Exceeded”) generiert. Wenn dieser Fehler auftritt wird die Fehlermeldung anstelle des Textes in die Antwort integriert. Dies begründet sich damit, dass für die Übersetzungen Inhaltskontigente gelten. So gilt für einen Standard-Nutzer ein Limit von 1,000.000 Zeichen pro 100 Sekunden [Gooc]. Dieser Wert wird bei längeren Artikeln überschritten und kann nur mit einem Upgrade erhöht werden (siehe Tabelle 5.2. Abgesehen von Kontingenten darf eine Anfrage 2000 Zeichen (Codepunkte) nicht überschreiten. Wird dieser Wert überschritten, wird die Anfrage mit “INVALID_ARGUMENT” beantwortet. Daher müssen die einzelnen Absätze weiter unterteilt werden.

4.5 Natural Language Processing

Für die Programmiersprache “Python” findet man mehrere interessante NLP-Libraries. Dazu zählen NLTK (“Natural Language Toolkit”) [NLT], SpaCy [Expc], Scikit-learn [sci], TextBlob [Ste], CoreNLP [MSB+14], Gensim [ŘS10], Polyglot [AR] und vermutlich noch andere. Für die Verwendung in Wilangyman wurden die populärsten Libraries (NLTK und Spacy) in Betracht gezogen. Die deutlich schnellere Bearbeitung sowie der einfachere Einsatz führten schließlich zum Einsatz von SpaCy. Den Unterschied in der Handhabung sowie der Verarbeitungsgeschwindigkeit sollen Listing 4.8 und 4.9 verdeutlichen. SpaCy bietet im Gegensatz zu NLTK bereits vortrainierte Sprachmodelle an, die Taggen, Parsen und der Named Entity Recognition (NER) bedeutend einfacher machen. Man findet derzeit Sprachmodelle für Englisch, Deutsch, Französisch, Spanisch, Portugiesisch, Italienisch, Niederländisch, Norwegisch und Litauisch. Diese Modellpakete unterscheiden sich in Größe und Umfang. Wilangyman verwendet das kleinste englische Modell - en_core_web_sm. Exemplarisch soll an dieser Stelle das umfangreichste Modell - Englisch - die Möglichkeiten aufzeigen:

en_core...	Größe	Vektoren		SA [1]			NERA [5]		
		keys	uv	LAS [2]	UAS [3]	POS [4]	F [6]	P [7]	R [8]
_web_sm	11 MB	n/a		89,71	97,61	97,03	85,08	85,25	84,89
_web_md	91 MB	685k	20k	89,77	91,65	97,14	86,10	86,15	86,05
_web_lg	789 MB	685k	685k	90,16	91,98	97,21	86,30	86,22	86,39

Tabelle 4.1: Spacy CORE Sprachmodelle [Expb]; die uv (Unique Vectors) sind generell 300-dimensional

[1] Syntax Accuracy
[2] Labeled Dependencies
[3] unlabeled Dependencies
[4] Part Of Speech Tags
[5] Named Entities Regocnition Accuracy
[6] Entity F-Score
[7] Entity Precision
[8] Entity Recall

Neben den drei CORE-Modellen existieren noch weitere fünf, also in Summe acht Sprachmodelle für Englisch, davon eines vom Typ “Vectors” und drei vom Typ “PyTorch Transformers”. Die CORE-Modelle arbeiten mit Vokabeln, Syntax, Entities und Vektoren.

Die kleinen Pakete, also die, die mit “sm” enden enthalten keine Wortvektoren sondern nur kontextsensitive Tensoren. Das Ergebnis der similarity-Methode ist damit deutlich schneller, aber eben auch ungenauer.

NLTK bietet auch mittels CoNLL 2002 und CoNLL 2003 [SM03] Sprachmodelle an. Die Anwendung gestaltet sich aber bei weitem nicht so einfach wie bei SpaCy (Vergleiche Listing 4.8 und 4.9).

```
1  import spacy
2  import en_core_web_sm
3  from spacy.pipeline import merge_entities, merge_noun_chunks
4
5  # englisches Sprachmodell in sm(all) laden
6  nlp = en_core_web_sm.load()
7  # Pipeline um NC und NER erweitern
8  nlp.add_pipe(merge_noun_chunks)
9  nlp.add_pipe(merge_entities)
10
11 text = "The European Union on Wednesday fined Microsoft EUR 497.2 million (
       $611 million) for abusing its 'near monopoly' with its Windows operating
       system to crush competitors and gain the upper hand in markets for
       digital media players and low-end servers."
12
13 # wendet das zuvor geladene Sprachmodell mit der definierte NLP-Pipeline an
14 # um eine Tokenliste zu bekommen incl. differenzierter Lemmatisierung.
15 # param text String Ein zu tokenisierender Text
```

```
16 # return Liste der Tokens bereinigt von Stopwords und den Tokens, die die
       Part-Of-Speech (PoS) Markierung SSPACEünd "PUNCTënthalten. siehe
       https://spacy.io/api/annotation
17 def tokenizeWithSpaCy(text):
18     doc = nlp(text)
19     tokenlist = []
20     for token in doc:
21         newtoken = ""
22         if token.is_stop or token.pos_ in ("PUNCT","SPACE"):
23             newtoken = ""
24         elif token.pos_ in ("NOUN", "VERB"):
25             newtoken = token.lemma_ if " " not in token.text else token.text
26         else:
27             newtoken = token.text
28         if newtoken != "":
29             tokenlist.append(newtoken)
30     return tokenlist
31
32 # Erstmalige Durchlaufdauer:  1,9 s, jeder weitere Durchlauf:  32 ms
```

Listing 4.8: SpaCy Tokenize mit NER

Die Spacy-NLP-Pipeline kann erweitert werden. Solche Erweiterungen werden in Zeile acht und neun von Listing 4.8 definiert. Erst wird mit “merge_noun_chunks” eine Funktionalität hinzugefügt, die zusammengehörende Stücke (engl. chunks) zusammenfügt. Damit wird beispielsweise “digital media players” als ein Token erkannt. Nach dem Durchlauf werden ab Zeile 19 unerwünschte Tokens (Stopwords, PoS-Tags PUNCT und SPACE) aussortiert. Außerdem wird statt `token.text` die Grundform des Tokens mit `token.lemma_` genommen - aber nur dann wenn der Tokentext nur ein Wort enthält (kein Leerzeichen) und der Part Of Speech-Tag ein Nomen oder Verb anzeigt. Das ist nötig, da sonst “its Windows operating system” auf “its” reduziert wird.

Die extrahierte Tokenliste
“The European Union”,
“Wednesday”, “fin”, “Microsoft”, “EUR”, “497.2 million”,
“$ 611 million”, “abuse”, “its near monopoly”,
“its Windows operating system”, “crush”, “competitor”,
“gain”, “the upper hand”, “market”,
“digital media players”, “low-end servers”
zeigt noch überflüssige Worte wie “its”. Auch wird EUR nicht als Währungsbezeichnung erkannt.

```
1 import nltk
2 from nltk.corpus import stopwords
3 stopwords = nltk.corpus.stopwords.words('english')
4
5 def ner_tagging(text):
6     chunked = nltk.ne_chunk(nltk.pos_tag(nltk.word_tokenize(text)))
7     continuous_chunk = []
```

```
8     current_chunk = []
9     for i in chunked:
10        if type(i) == nltk.tree.Tree:
11            current_chunk.append(" ".join([token for token, pos in i.leaves()
      ]))
12        elif current_chunk:
13            named_entity = " ".join(current_chunk)
14            if named_entity not in continuous_chunk:
15                continuous_chunk.append(named_entity)
16                current_chunk = []
17            else:
18                continue
19        else:
20            if i[0] not in stopwords and i[0] not in string.punctuation:
21                continuous_chunk.append(i[0])
22    return continuous_chunk
23 # Erstmalige Durchlaufdauer:  17 s, jeder weitere Durchlauf:  55 ms
```

Listing 4.9: NLTK Tokenize mit NER

Die mit Listing 4.9 (NLTK) extrahierte Tokenliste
“The”, “European Union”, “Wednesday”, “fined”, “Microsoft”, “million”, “611”, “million”,
“abusing”, “ńear”, “monopoly”, “Windows”, “operating”, “system”, “crush”, “competitors”, “gain”, “upper”, “hand”, “markets”, “digital”, “media”, “players”, “low-end”, “servers”
lässt Währungen vermissen ebenso wie die Erkennung von Geldbeträgen. Außerdem wurden nur die “European Union” als Organisation und “Microsoft” als Entities erkannt (Output: [(’European’, ’NNP’), (’Union’, ’NNP’)], [(’Microsoft’, ’NNP’)]).

Das Textprocessing in Wilangyman nutzt daher SpaCy.

KAPITEL 5

Resultate

Die nachfolgenden Wikipedia-Artikel wurden zum Vergleich herangezogen. Sie wurden ausgewählt, weil es sich um drei unterschiedlich gestaltete Themenbereichen handelt. Eine Biographie, ein Schiff sowie ein Artikel geographischen Ursprungs.

1. Herbert Benson (siehe 5.1.1)
   Da es sich bei diesem Artikel um eine reine Übersetzung (wurde von mir im Rahmen der LVA Socially Embedded Computing im Sommersemester 2017 übersetzt) einer Biographie handelt, sollten die Ähnlichkeiten entsprechend groß ausfallen. Natürlich gibt es Unterschiede in der Informationspräsentation von deutsch- und englischsprachigen Wikipedia-Mitgestalter und Administratoren, wodurch keine 100%ige Übereinstimmung erreicht werden kann. Für Vergleichszwecke wurde bei diesem Artikel auch eine der ursprünglichen deutsche Versionen - noch vor der Überarbeitung von Wikipedia-Lektoren - verwendet. Der deutsche Artikel ist in 5 Abschnitte unterteilt und umfasst 1651 Wörter, die englische Version 1570 Wörter in 8 Absätzen.

2. Hilma Hooker (siehe 5.1.2)
   Ein Artikel über ein gesunkenes Schiff. Sowohl der englische als auch der deutsche Artikel haben 3 Absätze. Der englische ist mit 558 Wörtern etwas länger als der deutsche.

3. Nethermost Pike (siehe 5.1.3)
   Ein Artikel über einen Berg in England, der in der englischen Version 6 Abschnitte mit einer Gesamtlänge von 2064 Wörtern aufweist. Die deutsche Version verfügt nur über 3 Abschnitte mit insgesamt 655 Wörtern.

Dazu folgende Anmerkungen:

- In den Tabellen finden sich horizontal die Überschriften aus der deutschen Seite, darunter die Übersetzung. Vertikal werden die der englischen Seite dargestellt.
- Tokens, die in beiden Sprachversionen vorkommen sind unterstrichen.
- Die Werte geben die berechnete Ähnlichkeit Im Intervall [0,1] an. Die, in den Tabellen verwendetet Farbskala reicht von rot (=0.0, keine Übereinstimmung) bis grün (=1.0, volle Übereinstimmung).
- Die Ähnlichkeiten aller Absätze wurden nur für dieses Dokument berechnet. Wilangyman berechnet dies standardmäßig nur für den Absatz, der die höchste Titelähnlichkeit aufweist.

Bei den in Abschnitt 5.2 erwähnten längeren Artikel (KZ Dachau, Mount Everest) zeigt sich auch der Bedarf an weiterer Unterteilung. Die in diesen Artikeln zahlreich verwendeten `H3`-Überschriften führen zu – im Verhältnis zur Gesamtlänge – geringen Anzahl von Textabschnitten (diese werden mittel `H2`-Tags unterteilt siehe 3). Je mehr Unterüberschriften vorhanden sind, desto länger wird der Abschnittstext und damit verringert sich die Chance auf eine direkte Zuordnung des Absatzes. Für längere Artikel mit stark genutzter Struktur müsste das Konzept erweitert werden.

5.1 Systematische Vergleiche

5.1.1 Herbert Benson

URL der deutschen Version: https://de.wikipedia.org/wiki/Herbert_Benson
URL der englischen Version: https://en.wikipedia.org/wiki/Herbert_Benson
Einleitung in der Ausgangssprache:

> Herbert Benson(* 1935 in Yonkers, New York) ist ein amerikanischer Arzt, Kardiologe und Gründer des Benson-Henry Institute for Mind Body Medicine im Massachusetts General Hospital (MGH) in Boston. Er unterrichtet Mind-Body-Medizin an der Medizinischen Fakultät in Harvard und ist emeritierter Direktor des Benson-Henry Institute (BHI) am MGH. Er ist ein Gründungskurator des American Institute of Stress. Benson prägte den wissenschaftlichen Begriff der Benson-Meditation (engl. relaxation response) 2013 er schrieb auch ein Buch mit demselben Titel 2013 und benutzte diesen Begriff, um die Fähigkeit des Körpers zu beschreiben, Entspannung von Muskeln und Organen zu stimulieren.

Die Google-Übersetzung zu Englisch, also der Vergleichssprache:

> Herbert Benson (born 1935 in Yonkers, New York) is an American physician, cardiologist and founder of the Benson-Henry Institute for Mind Body Medicine at the Massachusetts General Hospital (MGH) in Boston. He teaches

> mind-body medicine at the Harvard School of Medicine and is the emeritus director of the Benson-Henry Institute (BHI) at MGH. He is a founding curator of the American Institute of Stress. Benson coined the scientific notion of Benson Meditation - he also wrote a book of the same title - and used it to describe the body's ability to stimulate relaxation of muscles and organs.

Die extrahierten Tokens sind:

> "Herbert Benson", "bear", "1935", "Yonkers", "New York", "an American physician", "cardiologist", "founder", "the Benson-Henry Institute", "Mind Body Medicine", "the Massachusetts General Hospital MGH", "Boston", "teach", "mind-body medicine", "the Harvard School of Medicine", "the emeritus director", "the Benson-Henry Institute", "BHI", "MGH", "a founding curator", "the American Institute of Stress", "Benson", "coin", "the scientific notion", "Benson Meditation", "write", "a book", "the same title", "describe", "stimulate", "relaxation", "muscle", "organ"

Die englische Version des Artikels hat diesen Einleitungsabschnitt:

> Herbert Benson (born 1935), is an American medical doctor, cardiologist, and founder of the Mind/Body Medical Institute at Massachusetts General Hospital (MGH) in Boston. He is a professor of mind/body medicine at Harvard Medical School and director emeritus of the Benson-Henry Institute (BHI) at MGH. He is a founding trustee of The American Institute of Stress. He has contributed more than 190 scientific publications and 12 books. More than five million copies of his books have been printed in different languages.Started in 1998, Benson became the leader of the so-called 'Great Prayer Experiment,' or technically the 'Study of the Therapeutic Effects of Intercessory Prayer (STEP).' The result published in 2006 concluded that intercessory prayer has no beneficial effect on patients with coronary artery bypass graft surgery. He, however, still believes that prayer has positive health benefits.Benson coined relaxation response (and wrote a book by the same title) as a scientific term for meditation, and he used it to describe the ability of the body to stimulate relaxation of muscle and organs.

Hier die Tokens des englischen Textes:

> "Herbert Benson", "bear", "1935", "an American medical doctor", "cardiologist", "founder", "the Mind/Body Medical Institute at Massachusetts General Hospital", "MGH", "Boston", "a professor", "mind/body medicine", "Harvard Medical School", "director emeritus", "the Benson-Henry Institute", "BHI", "MGH", "a founding trustee", "The American Institute of Stress", "contribute", "more than 190 scientific publications", "12 books", "More than five

million copies", "his books", "print", "different languages", "start", "1998", "Benson", "the leader", "the so-called 'Great Prayer Experiment", "technically the 'Study of the Therapeutic Effects of Intercessory Prayer", "step", "The result", "publish", "2006", "conclude", "intercessory prayer", "no beneficial effect", "patient", "coronary artery bypass graft surgery", "believe", "prayer", "positive health benefits", "Benson", "coin", "relaxation response", "write", "a book", "the same title", "a scientific term", "meditation", "describe", "the ability", "the body", "stimulate", "relaxation", "muscle", "organ"

Die berechnete Ähnlichkeit des Textes ist 0,27. Die Einleitung wird daher ausgeklappt dargestellt.

Damit die Zeilenlänge nicht die Seitenbreite übersteigt wurden die folgenden Überschriften wie folgt in die Tabellen eingetragen:
Ehrungen: Ehrungen und Auszeichnungen, Übersetzung: honors and awards
Veröffentlichungen: Veröff., Übersetzung: publications
Einzelnachweise: Einzelnachw., Übersetzung: references

Der Überschriften-Vergleich ergibt:

	Biografie biography	Ehrungen honors	Veröff. pub	Weblinks web links	Einzelnachw. references
Biography	1,000000	0,000000	0,000000	0,000000	0,000000
Notable projects	0,000000	0,000000	0,000000	0,000000	0,000000
Personal life	0,000000	0,000000	0,000000	0,000000	0,000000
Awards and honours	0,000000	0,333333	0,000000	0,000000	0,000000
Publications	0,000000	0,000000	1,000000	0,000000	0,000000
References	0,000000	0,000000	0,000000	0,000000	1,000000
Additional sources	0,000000	0,000000	0,000000	0,000000	0,000000
External links	0,000000	0,000000	0,000000	0,333333	0,000000

Tabelle 5.1: Vergleich der Überschriften von Herbert Benson mit Jaccard-Index

Betrachtet man nun die beiden Absätze "Ehrungen und Auszeichnungen" (engl. honors and awards) und "Awards and honours" sieht man, dass spaCy bei britischer und amerikanischer Schreibweise nicht den gleichen Inhalt erkennt. Es ergibt sich als gemeinsame Menge nur das Wort "awards", "and" wird als Stopword erkannt und nicht in die Berechnung übernommen. Folglich ergibt sich die Vereinigungsmenge "award honor honour" und die Schnittmenge "award". Daher ist der errechnete Jaccard-Index lediglich 0,33. Trotzdem reicht dieser Wert für eine Zuordnung, da sonst keine Übereinstimmung gefunden wird.

Der Textblock-Vergleich ergibt:

	Biografie biography	Ehrungen honors	Veröff. pub	Weblinks web links	Einzelnachw. references
Biography	0,928206	0,886406	0,782331	0,871810	0,327902
Notable projects	0,916259	0,515598	0,573194	0,717906	0,269821
Personal life	0,590682	0,737001	0,623794	0,657618	0,285110
Awards and honours	0,803105	0,954316	0,819703	0,871799	0,376932
Publications	0,727064	0,859773	0,967212	0,810574	0,415727
References	0,000000	0,000000	0,000000	0,000000	0,000000
Additional sources	0,781338	0,933071	0,854953	0,905350	0,496129
External links	0,895100	0,894552	0,810019	0,955676	0,358773

Tabelle 5.2: Vergleich der Textblöcke von Herbert Benson mit Kosinus-Ähnlichkeit

Die deutsche und die englische Variante des Absatzes “Ehrungen und Auszeichnungen” unterscheiden sich nur geringfügig. Die berechnete Ähnlichkeit (Cosinus Similarity) lag bei 0,954316. Daher wird zwar angezeigt, dass es im englischen Artikel eine korrespondierende Stelle gibt, diese ist aber “eingeklappt” (siehe Abb. 5.1).

Abbildung 5.1: Wikipedia Artikel: eingeklappte Zusatzinformation

Der Nutzer kann jederzeit die Passage aufklappen und sieht dann die, aus dem Englischen übersetzte Version (siehe Abb. 5.2)

Abbildung 5.2: Wikipedia Artikel: ausgeklappte Zusatzinformation

Sollte man Zweifel an der übersetzten Textpassage haben, kann auch der Text in Originalsprache per Klick angezeigt werden.

Für die Absätze “Notable Projects”, “Personal Life” sowie “Additional Ressources” kann keine Zuordnung gefunden werden. Daher werden sie am Ende des Artikels unter “Nicht zugeordnet” eingefügt.

In Summe benötigt der Vorgang bei Installation auf dem Testrechner ca. 10 Sekunden. Die Übersetzungen schlagen dabei mit fast 4 Sek. zu Buche 5.2.

Eine Schwäche des Textvergleiches zeigt sich beim Vergleich der ersten Herbert-Benson-Übersetzung, also vor der Bearbeitung der deutschen Lektoren. Die Titel und Absätze waren reine Übersetzungen. Man könnte also annehmen, dass alle Überschriften des deutschen Artikels ein Pendant im englischen Text haben.

Google Translate übersetzt aber die Überschriften nicht so wie der menschliche Übersetzer. So kommt es, dass der Vergleich nur drei Mal die Übereinstimmung “1” anzeigt. Besonders fatal ist dies beim Titel “Privatleben”, das zu “privacy” übersetzt wurde und dem Vergleich mit dem englischen Originaltitel “Personal life” gar nicht standhält.

	Biographie	Besondere Projekte	Privatleben	Ehrungen und Auszeichnungen	Veröffentlichungen	Weblinks	References
	biography	special projects	privacy	honors and awards	publications	web links	references
Biography	1,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
Notable projects	0,000000	0,333333	0,000000	0,000000	0,000000	0,000000	0,000000
Personal life	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
Awards and honours	0,000000	0,000000	0,000000	0,333333	0,000000	0,000000	0,000000
Publications	0,000000	0,000000	0,000000	0,000000	1,000000	0,000000	0,000000
References	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	1,000000
Additional sources	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000	0,000000
External links	0,000000	0,000000	0,000000	0,000000	0,000000	0,333333	0,000000

Tabelle 5.3: Vergleich der Original Überschriften vom Artikel über Herbert Benson

Die Überschrift “Besondere Projekte”, übersetzt als “special project” passt nur zu 33 Prozent zum englischen Originaltitel “notable projects”. Zu den unterschiedlichen Übersetzungen kommen dann noch Probleme (wie schon zuvor erwähnte) zwischen britischem und amerikanischen Englisch.

Es gibt also sechs richtige Erkennungen, ein Mal wurde keine Übereinstimmung gefunden obwohl eine vorhanden gewesen wäre. Ein weiterer Fall (Additional Sources) brachte - korrekter Weise - keine Übereinstimmung. Die übrigen Absätze (Bemerkenswerte Projekte, Persönliches Leben und zusätzliche Quellen) werden am Ende unter “Nicht zugeordnet” angezeigt.a

5.1.2 Hilma Hooker

URL der deutschen Version: https://de.wikipedia.org/wiki/Hilma_Hooker
URL der englischen Version: https://en.wikipedia.org/wiki/Hilma_Hooker

Einleitung in der Ausgangssprache:

> Die Hilma Hooker ist ein Wrack vor Bonaire. Sie wurde 1951 gebaut und sank 1984.

Die Google-Übersetzung zu Englisch, also der Vergleichssprache:

> The Hilma Hooker is a wreck before Bonaire. It was built in 1951 and sank in 1984.

Die extrahierten Tokens sind:

> “The Hilma Hooker”, “a wreck”, “Bonaire”, “It”, “built”, “1951”, “sink”, “1984”

Die englische Version des Artikels hat diesen Einleitungsabschnitt:

> The Hilma Hooker is a shipwreck in Bonaire in the Caribbean Netherlands. It is a popular wreck diving site.

Hier die Tokens des englischen Textes

> “The Hilma Hooker”, “a shipwreck”, “Bonaire”, “the Caribbean Netherlands”, “It”, “a popular wreck diving site”

Die berechnete Ähnlichkeit des Textes ist 0,78. Da dies über dem Schwellwert liegt, wird der übersetzte englische Introtext eingeklappt angezeigt.

Der Überschriften-Vergleich ergibt:

	Geschichte history	Grund der Versenkung reason for sinking	Weblinks web links
ship history	0,500000	0,000000	0,000000
dive site	0,000000	0,000000	0,000000
references	0,000000	0,000000	0,000000

Tabelle 5.4: Vergleich der Überschriften von Hilma Hooker mit Jaccard-Index

Da der Vergleich mit dem Jaccard-Index berechnet wird findet man nur bei dem Überschriftenpaar “ship history” und “Geschichte” (übersetzt “history”) einen Wert größer Null. Das hat zur Folge, dass die anderen beiden Absätze erst am Schluss (unter “Nicht zugeordnet”) eingefügt werden.

Der Textblock-Vergleich ergibt:

	Geschichte history	Grund der Versenkung reason for sinking	Weblinks web links
ship history	0,927597	0,921187	0,378381
dive site	0,854021	0,870360	0,344493
references	0,608645	0,358330	0,424479

Tabelle 5.5: Vergleich der Textblöcke von Hilma Hooker mit Kosinus-Ähnlichkeit

In Summe benötigt der Vorgang bei Installation auf dem Testrechner ca. 5 Sekunden. Die Übersetzungen schlagen dabei mit 1,8 Sek. zu Buche 5.2.

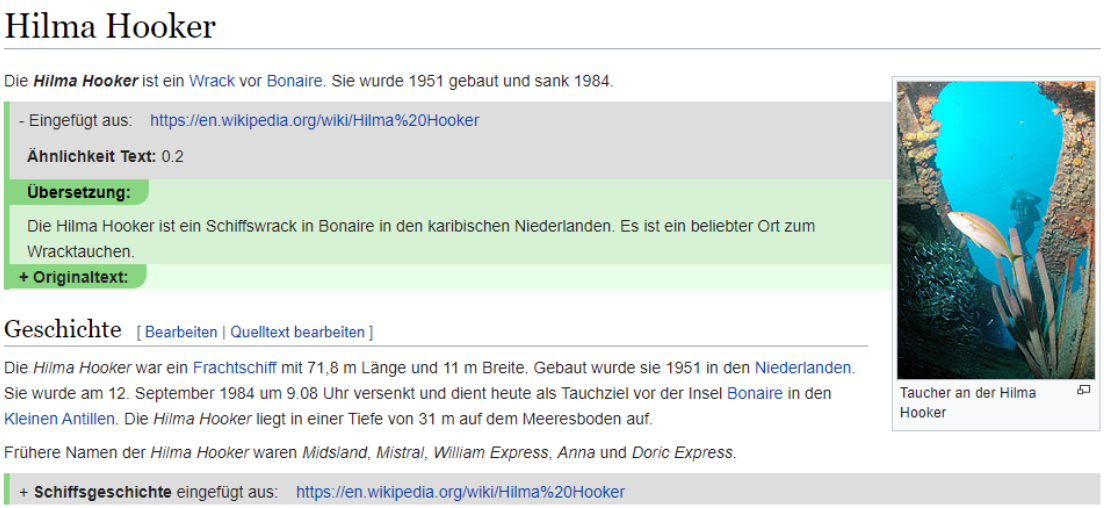

Abbildung 5.3: Wikipedia Artikel: Ergebnis Hilma Hooker

5.1.3 Nethermost Pike

URL der deutschen Version: https://de.wikipedia.org/wiki/Nethermost_Pike
URL der englischen Version: https://en.wikipedia.org/wiki/Nethermost_Pike
Einleitung in der Ausgangssprache:

> Nethermost Pike ist einer der 214 Wainwright genannten Berge (Fell) im nordenglischen Nationalpark Lake District. Er ist der zweithöchste Berg der Helvellyn Range, einer in Nord-Süd-Richtung verlaufenden Bergkette, die zwischen Ullswater im Osten und Thirlmere im Westen liegt, und wird den Eastern Fells zugeordnet. Während die Westseite aus Grashängen besteht, ist die Ostseite steil und felsig. In früheren Jahrhunderten wurde hier Bergbau zur Gewinnung von Blei betrieben, und es sind noch heute zahlreich vorhandene Stollen und Schächte zu finden.

Die Google-Übersetzung zu Englisch, also der Vergleichssprache:

> Nethermost Pike is one of the 214 Wainwright mountains (fur) in the northern English Lake District National Park. It is the second highest mountain in the Helvellyn Range, a north-south running mountain range that lies between Ullswater in the east and Thirlmere in the west, and is assigned to the Eastern Fells. While the west side consists of grassy slopes, the east side is steep and rocky. In earlier centuries, mining was used for the production of lead, and there are still numerous existing tunnels and shafts to find today.

Die extrahierten Tokens sind:

> "Nethermost Pike", "the 214 Wainwright mountains", "fur", "the northern English Lake District National Park", "It", "the second highest mountain", "the Helvellyn Range", "a north-south running mountain range", "lies", "Ullswater", "the east", "Thirlmere", "the west", "assigned", "the Eastern Fells", "While", "the west side", "consists", "grassy slopes", "the east side", "steep", "rocky", "In", "earlier centuries", "mining", "the production", "lead", "numerous existing tunnels", "shafts", "find", "today"

Die englische Version des Artikels hat diesen Einleitungsabschnitt:

> Nethermost Pike is a fell in Cumbria, England, and a part of the Lake District. At 891 metres (2,923̆00a0ft) it is the second highest Wainwright in the Helvellyn range, the highest of which is Helvellyn itself. It is located close to the southern end of the ridge, with Helvellyn to the north, and High Crag and Dollywaggon Pike to the south. Nethermost Pike, along with many of the Eastern Fells, lies between Thirlmere in the west and the Ullswater catchment in the east. The closest villages are Glenridding and Patterdale on the shores of Ullswater, over 8 kilometres (5̆00a0mi) away.Like most fells in the Helvellyn range, Nethermost Pike has grassy western slopes and rocky outcrops on the eastern side. Geologically, Nethermost Pike belongs to the Borrowdale Volcanic Group. Lead was once mined on its eastern slopes, resulting in open workings and underground mines. The eastern slopes are protected as part of a Site of Special Scientific Interest because of the Pike's geological and biological features, which include some of England's best arctic-alpine and tall-herb vegetation.

Hier die Tokens des englischen Textes

> "Nethermost Pike", "a fell", "Cumbria", "England", "a part", "the Lake District", "At", "891 metres", "2,923", "ft", "the second highest Wainwright", "the Helvellyn range", "highest", "Helvellyn", "It", "located", "close", "the southern end", "the ridge", "Helvellyn", "the north", "High Crag", "Dollywaggon Pike", "the south", "Nethermost Pike", "the Eastern Fells", "lies", "Thirlmere", "the west", "the Ullswater catchment", "the east", "The closest villages", "Glenridding and Patterdale", "the shores", "Ullswater", "8 kilometres", "5̆00a0mi", "away", "Like", "most fells", "the Helvellyn range", "Nethermost Pike", "grassy", "western slopes", "rocky outcrops", "the eastern side", "Geologically", "Nethermost Pike", "belongs", "the Borrowdale Volcanic Group", "Lead", "mined", "its eastern slopes", "resulting", "open workings", "underground mines", "The eastern slopes", "protected", "a Site of Special Scientific Interest", "the Pike's geological and biological features", "include", "England's best arctic-alpine and tall-herb vegetation"

Die berechnete Ähnlichkeit des Textes ist 0,95. Der übersetzte englische Introtext wird daher eingeklappt dargestellt.

Der Überschriften-Vergleich ergibt:

	Geologie geology	Topographie topography	Anmerkungen remarks
classification	0,000000	0,000000	0,000000
topography	0,000000	1,000000	0,000000
ascents	0,000000	0,000000	0,000000
geology	1,000000	0,000000	0,000000
biological interest	0,000000	0,000000	0,000000
references	0,000000	0,000000	0,000000

Tabelle 5.6: Vergleich der Überschriften von Nethermost Pike mit Jaccard-Index

Tabelle 5.6 zeigt, dass man bei diesem Artikel-Paar genau zwei Mal, bei Geologie und bei Topographie, eine Übereinstimmung bei den Überschriften findet. Alle anderen lassen sich nicht direkt zuordnen.

Der Textblock-Vergleich ergibt:

	Geologie geology	Topographie topography	Anmerkungen remarks
classification	0,850621	0,908651	0,325176
topography	0,958773	0,982828	0,232577
ascents	0,931861	0,965771	0,245818
geology	0,957780	0,969680	0,212272
biological interest	0,954619	0,950801	0,242220
references	0,000000	0,000000	0,000000

Tabelle 5.7: Vergleich der Textblöcke von Nethermost Pike mit Kosinus-Ähnlichkeit

Tabelle 5.7 zeigt, dass die Absätze zu den beiden gefundenen Überschriftspaaren sehr ähnlich sind. Daher werden sie ausgeblendet dargestellt. Die anderen Absätze werden erwartungsgemäß am Ende unter “Nicht zugeordnet” angezeigt (siehe Abb. 5.4).

In Summe benötigt der Vorgang bei Installation auf dem Testrechner ca. 8 Sekunden. Die Übersetzungen schlagen dabei mit 1 Sek. zu Buche 5.2.

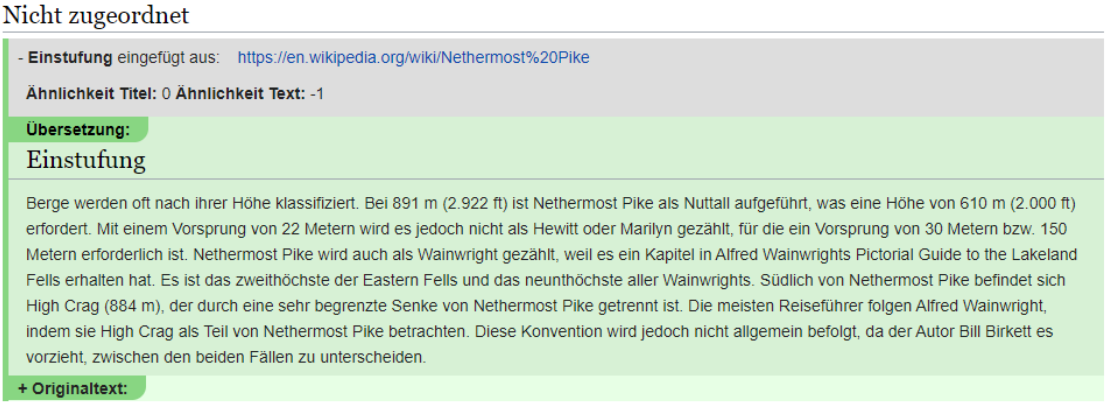

Abbildung 5.4: Wikipedia Artikel: Nicht zugeordnete Absätze Nethermost Pike

5.2 Benchmarks

Natürlich konsumieren die unterschiedlichen Prozesse Rechenleistung und Bandbreite. In Tabelle 5.2 sind die Lade- und Bearbeitungszeiten der vorigen Artikel gegenübergestellt. Klarer Weise wächst die benötigte Zeit mit der Länge der Artikel (siehe Tabelle 5.2). Im Preprocessing sind das Parsing des HTML-Sourcecodes der Artikel, die Übersetzungen in Vergleichs- und Zielsprache sowie die Extraktion der Tokens enthalten. Das entspricht den Berechnungen in der Methode `getParagraphs` des Serverdienstes.

	Herbert Benson	Hilma Hooker	Nethermost Pike
Absätze (de)	5	3	3
Wörter (de)	1726	218	633
Absätze (en)	8	3	6
Wörter (en)	1574	561	1987

Tabelle 5.8: Eckdaten der Artikel, Wortanzahl über Wikipedia API [4.3]

Da für Google Zugriffskontingent gelten (siehe Kapitel 4.4) ist auch die Zeichenanzahl sowie die Menge der nötigen Übersetzungs-Anfragen von Interesse (Tabellen 5.2 und 5.2).

	Herbert Benson	Hilma Hooker	Nethermost Pike
Anfragen	43	21	40
Zeichen	15601	4331	14609
Dauer in Sek.	3,7	1,8	1,0

Tabelle 5.9: Daten zu Google Translate

	Herbert Benson	Hilma Hooker	Nethermost Pike
Laden (de)	0,2	0,2	0,2
Preprocessing (de)	3,7	1,4	2,3
Laden (en)	0,2	0,3	0,2
Preprocessing (en)	3,5	2,2	3,6
Titelvergleich	0,7	0,2	0,3
Textvergleich	0,8	0,2	0,7
Übersetzungen	3,7	1,8	1,0
Gesamt	9,8	5,0	7,9

Tabelle 5.10: Berechnungszeiten in Sekunden.

Die Messungen wurden auf einem PC mit Intel Core m5-6Y54 CPU mit 8GB RAM durchgeführt. Die Internetanbindung erfolgte über Magenta mit max. 25MBit Download. Die Cache-Funktion wurde für die Zeitmessung deaktiviert.

Weitere Tests mit längeren Artikel zeigten, dass GT (Google Translate) nach mehreren Anfragen einen Fehler generiert (siehe 4.4). Dies geschah gelegentlich z.B. bei Bearbeitung von `https://de.wikipedia.org/wiki/KZ_Dachau` und regelmäßig bei `https://de.wikipedia.org/wiki/Mount_Everest`. Diese, beim Artikel über das KZ Dachau reproduzierbare Abweichung (Fehler wird nicht immer generiert) scheint in der Begrenzung von 100 Sekunden und der zeitlichen Überschneidung mit vorhergehenden Abfragen zu liegen.

	KZ Dachau	Mount Everest
Wortanzahl (de/en)	10.175/10.586	9.795/23.864
GT Zeichen	110.591	161.686
GT Anfragen	253	383
Laden (de)	0,2	0,2
Preprocessing (de)	26,7	25,5
Laden (en)	0,2	0,3
Preprocessing (en)	20,0	52,6
Titelvergleich	3,6	8,1
Textvergleich	6,3	13,3
Übersetzungen	7,3	10,7
Gesamt	58,3	102,0

Tabelle 5.11: Berechnungszeiten bei längeren Artikeln(in Sekunden).

Die Artikellänge steht in einem direktem Verhältnis zur Bearbeitungsdauer (siehe Abb. 5.5). Die Wortanzahl summiert sich dabei aus beiden zu vergleichenden Artikeln (Ursprungs- und Ergänzungs-Artikel).

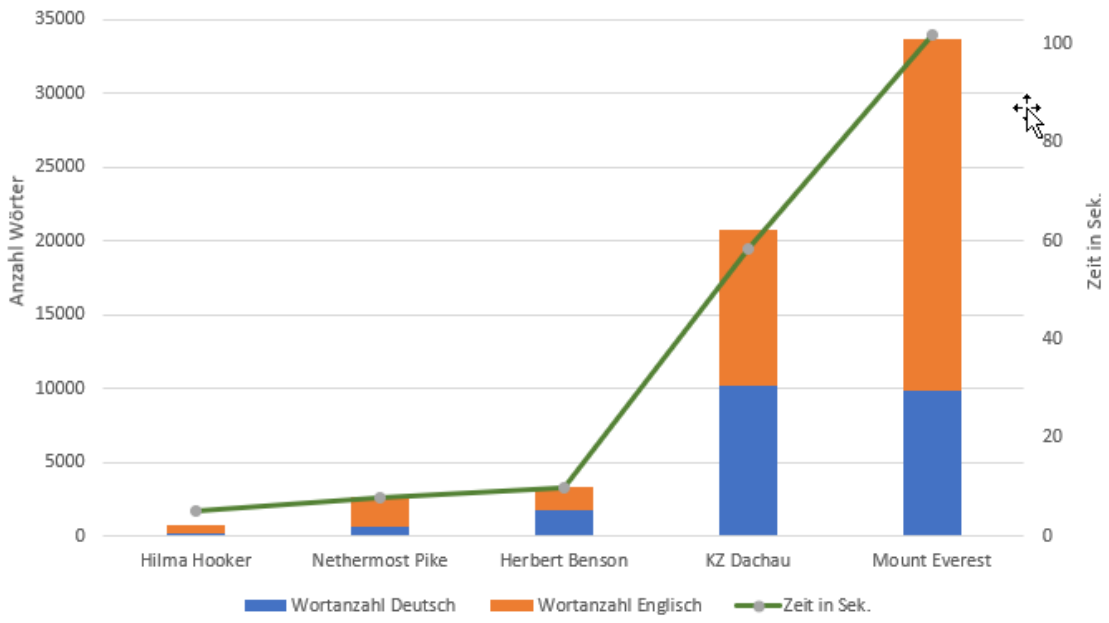

Abbildung 5.5: Zusammenhang Wortanzahl zu Bearbeitungsdauer.

KAPITEL 6

Fazit

Wilangyman soll die Suche nach Informationen in Wikipedia erleichtern. Zusätzliche Information aus anderen Sprachversionen werden dazu in die Ausgangsseite eingebettet (siehe Abb. 3.1). Das funktioniert bei "normal" formatierten Seiten gut, weicht der Verfasser aber vom Standard ab, sinkt die Qualität deutlich. Die Ausführungsdauer steigt linear mit der Länge des Artikels an.

6.1 Bekannte Fehler und Schwächen

Natürlich bleiben bei einem ersten Prototypen immer Wünsche offen. Bei Wilangyman zeigt sich Erweiterungspotential etwa bei Bildern, Referenzen und Links. Generell werden die übersetzten Seiten auf ihren Text reduziert, daher werden Bilder und Tabellen gar nicht dargestellt. Referenzen werden gefiltert, um beim Extrahieren von Tokens keine ungewollten Artefakte zu hinterlassen. So würde aus "... aus vulkanische Quellen gespeist[1]" die Token-Liste "aus, vulkanische, Quellen gespeist[1" wenn man den Verweis unbehandelt ließe.

Wilangyman funktioniert in dieser Version nur ausgehend von der deutschen Wikipedia-Seite (Deutsch ist somit generell die Zielsprache). Konzeptionell sind natürlich alle Zielsprachen möglich. Zur Erweiterung wird werden derzeit nur englische Artikel verwendet. Eine Ausdehnung auf andere Sprachen kann über weitere Parameter im Sourcecode der Chrome-Extension erreicht werden. Durch die zusätzliche Übersetzung ist aber mit einer weiteren Steigerung der Bearbeitungsdauer zu rechnen.

Es zeigt sich, dass das Matching von Überschriften ausbaufähig ist. In nahezu allen Fällen in denen eine direkte Zuordnung von Absätzen gefunden wurde, wurde auch eine hohe inhaltliche Ähnlichkeit errechnet. Dies hat zur Folge, dass es kaum Inhalte gab, die bereits ausgeklappt dargestellt wurden. Es könnte interessant sein, bei fehlender Zuordnung auf

Absatzähnlichkeiten für diesen Zweck heranzuziehen und somit auch die Absätze unter "nicht zugeordnet" zu reduzieren.

6.2 Mangelhafte Ergebnisse

Die Textabschnitte werden anhand ihrer <H2>-Tags erkannt. Daher funktioniert Wilangyman bei anders formatierten Artikeln nur mangelhaft. Dies wurde bei der Seite der Daniel Motor Company [Wika] deutlich. Die deutsche Seite ist korrekt mittels H2-Tag formatiert, die englische jedoch verwendet diese nur drei Mal ("See also", "References" und "Sources"). Der eigentliche Text wird lediglich von H3-Tags gegliedert, was dazu führt, dass Wilangyman den Großteil des Textes als einen Abschnitt erkennt und auch nur einmal - ganz am Anfang - in die deutsche Version einfügt. Ähnlich verhält es sich bei längeren Artikeln (siehe auch Abs. 5).

6.3 Weiterentwicklungspotential

Wie schon aus den vorigen Absätzen ersichtlich, bieten sich einige Bereiche für eine Weiterentwicklung an. Nichtsdestotrotz kam von den bisherigen vier informellen Testern des Tools großteils positives Feedback. Den häufigsten Kritikpunkt stellte die Wartezeit dar. Ein großer Teil der Gesamtzeit wird von Übersetzungen gebraucht (siehe Kapitel 5.2). Optimierungen sind natürlich schon jetzt möglich. Die Google-Translate-Extension lässt auch effizientere Übersetzungsmöglichkeiten vermuten. Als Ansatz könnte man die Übersetzung der gesamten Seite in Einem erledigen und erst dann die Textaufteilung in Abschnitte vornehmen. Das könnte auch dazu führen, dass HTML-Tags für Links, Referenzen ebenso erhalten blieben wie Bilder und Tabellen. Jedoch ist die von Google zugelassen maximale Textmenge zu beachten.

Auf der Seite der Chrome-Extension sind auch einige Entwicklungsmöglichkeiten gegeben. Diese betreffen in erster Linie die Konfigurierbarkeit. Vorstellbar wären Einstelloptionen:

- Flexiblerer Einsatz durch Unabhängigkeit der Ausgangssprache. Derzeit wird nur die deutsche Wikipedia verwendet.
- Flexiblerer Einsatz durch Verwendung älterer Versionen der Artikel.
- Bevorzugte Sprachen: welche Sprachversionen – sofern vorhanden – verwendet werden sollen.
- Unter der Annahme, dass mit der Textmenge auch der Informationsgehalt größer ist, könnten Sprachen anhand der Artikellänge auswählt werden.
- Einstellbarer Schwellwert: ab welchem Ähnlichkeitswert sollen die zusätzlichen Informationen ein- bzw. ausgeblendet werden.

- Welche Vergleichsmethode soll verwendet werden. Reicht der schnell berechnete Jaccard-Index oder soll gar das Large-Sprachmodell inkl. Wordvektoren (Kosinus-Ähnlichkeit) zum Einsatz kommen (siehe Tab. 4.1). Letzteres würde eine Verfeinerung der Ergebnisse zu Lasten der Bearbeitungsdauer bedeuten.
- Beim Vergleich von Bullet-Lists könnten zusätzliche Punkte in der Übersetzung hervorgehoben werden.
- Hervorheben von Entities, die im Ausgangsartikel nicht vorkommen damit der Nutzer Informationen, die tatsächlich neu sind, schneller erfassen kann.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Literaturverzeichnis

[AR] Rami Al-Rfou. https://polyglot.readthedocs.io/. [Online; Stand 29. November 2019].

[BBCF16] Luisa Bentivogli, Arianna Bisazza, Mauro Cettolo, and Marcello Federico. Neural versus phrase-based machine translation quality: a case study. *CoRR*, abs/1608.04631, 2016.

[CDBF10] Fanny Chevalier, Pierre Dragicevic, Anastasia Bezerianos, and Jean-Daniel Fekete. Using text animated transitions to support navigation in document histories. In *International Conference on Human Factors in Computing Systems*, pages 683–692, Atlanta, GA, United States, Apr 2010. ACM, ACM Press.

[CEE+10] Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne J. Jekat, Ralf Klabunde, and Hagen Langer. *Computerlinguistik und Sprachtechnologie*. Spektrum Akademischer Verlag, Heidelberg, 3., überarbeitete und erweiterte auflage edition, 2010.

[Dee] https://www.deepl.com/translator. [Online; Stand 12. November 2019].

[Expa] https://spacy.io/usage/linguistic-features#tokenization. [Online; Stand 21. November 2019].

[Expb] https://spacy.io/models/en#en_core_web_md. [Online; Stand 21. November 2019].

[Expc] https://spacy.io/. [Online; Stand 29. November 2019].

[GD] Simon Gottschalk and Elena Demidova. http://multiwikiwebpage.l3s.uni-hannover.de/. [Online; Stand 12. November 2019].

[GD17] Simon Gottschalk and Elena Demidova. Multiwiki: Interlingual text passage alignment in wikipedia. *ACM Trans. Web*, 11(1):6:1–6:30, April 2017.

[Gooa] https://translate.google.com/. [Online; Stand 12. November 2019].

[Goob] https://cloud.google.com/translate/docs/. [Online; Stand 29. November 2019].

[Gooc] https://cloud.google.com/translate/quotas. [Online; Stand 16. Februar 2020].

[MCS+06] Rada Mihalcea, Courtney Corley, Carlo Strapparava, et al. Corpus-based and knowledge-based measures of text semantic similarity. In *Aaai*, volume 6, pages 775–780, 2006.

[MS] Paolo Massa and Federico Scrinzi. http://www.manypedia.com. [Online; Stand 12. November 2019].

[MSB+14] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit, 2014.

[NLT] https://www.nltk.org. [Online; Stand 29. November 2019].

[NOH+09] Dong Nguyen, Arnold Overwijk, Claudia Hauff, Dolf R. B. Trieschnigg, Djoerd Hiemstra, and Franciska de Jong. Wikitranslate: Query translation for cross-lingual information retrieval using only wikipedia. In Carol Peters, Thomas Deselaers, Nicola Ferro, Julio Gonzalo, Gareth J. F. Jones, Mikko Kurimo, Thomas Mandl, Anselmo Peñas, and Vivien Petras, editors, *Evaluating Systems for Multilingual and Multimodal Information Access*, pages 58–65, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[Raba] L. Rabe. Anzahl der artikel bei wikipedia in den jahren 2002 bis 2019. https://de.statista.com/statistik/daten/studie/195081/umfrage/anzahl-der-artikel-auf-wikipedia-weltweit. [Online; Stand 12. November 2019].

[Rabb] L. Rabe. Top 10 sprachen nach anzahl der artikel auf wikipedia im september 2019. https://de.statista.com/statistik/daten/studie/170265/umfrage/wikipedias-nach-anzahl-der-artikel/. [Online; Stand 12. November 2019].

[Ron10] Armin Ronacher. https://palletsprojects.com/p/flask/, 2010. [Online; Stand 29. November 2019].

[ŘS10] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[sci] https://scikit-learn.org/. [Online; Stand 29. November 2019].

[SFKN12] Yu Suzuki, Yuya Fujiwara, Yukio Konishi, and Akiyo Nadamoto. Good quality complementary information for multilingual wikipedia. In X. Sean Wang, Isabel Cruz, Alex Delis, and Guangyan Huang, editors, *Web Information Systems Engineering - WISE 2012*, pages 185–198, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[SM03] Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *CoRR*, cs.CL/0306050, 2003.

[SM16] Vijay Kumar Sharma and Namita Mittal. Cross lingual information retrieval (clir): Review of tools, challenges and translation approaches. In Suresh Chandra Satapathy, Jyotsna Kumar Mandal, Siba K. Udgata, and Vikrant Bhateja, editors, *Information Systems Design and Intelligent Applications*, pages 699–708, New Delhi, 2016. Springer India.

[Ste] `https://textblob.readthedocs.io/`. [Online; Stand 29. November 2019].

[Ten] F. Tender. `https://de.statista.com/statistik/daten/studie/157944/umfrage/marktanteile-der-browser-bei-der-internetnutzung-weltweit-seit-2009/`. [Online; Stand 22. November 2019].

[Tur16] Barak Turovsky. Found in translation: More accurate, fluent sentences in google translate. `https://www.blog.google/products/translate/found-translation-more-accurate-fluent-sentences-google-translate/`, 2016.

[Wika] `https://en.wikipedia.org/wiki/Daniels_Motor_Company`. [Online; Stand 29. November 2019].

[Wikb] `https://www.mediawiki.org/wiki/API:Query`. [Online; Stand 29. November 2019].

[Wik19] Wikipedia. Wikipedia — wikipedia, die freie enzyklopädie. `https://de.wikipedia.org/w/index.php?title=Wikipedia&oldid=192252757`, 2019. [Online; Stand 24. September 2019].

[WOZ+15] Dakuo Wang, Judith S. Olson, Jingwen Zhang, Trung Nguyen, and Gary M. Olson. Docuviz: Visualizing collaborative writing. In *CHI 2015, Crossings, Seoul, Korea*, Irvine, CA USA, 2015. University of California Irvine.