UNIVERSITÄT DUISBURG ESSEN

Offen im Denken

RAT RESULT ASSESSMENT TOOL

SEARCH STUDIES HAW HAMBURG

Bachelor's Thesis

on the Topic of

Measuring the Readability of Web Documents

Presented to the Faculty of Engineering
Duisburg-Essen University

by

Mohamed Elnaggar
Bülowstraße 55
45479, Mülheim an der Ruhr
Matrikelnummer: 3086790

Supervised by:	Prof. Dr. Dirk Lewandowski
Second Examiner:	Prof. Dr. Mohamed Chatti
Study Course:	Software Engineering PO15
Studiensemester:	Summer Semester 2023
Datum:	07.09.2023

Contents

List of Figures

List of Tables

List of abbreviations

APP Application

COVID-19 Coronavirus Disease 2019

DVC Dominant Visual Content

en English

es Spanish

ETA Extracted Text Accuracy

fr French

FRE(s) Flesch Reading Ease Score

FKGL Flesch-Kincaid Grade Level

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

nl Dutch

RAT Result Assessment Tool

OSMAN Open Source Metric for Measuring Arabic Narratives

pl Polish

SMOG Simple Measure of Gobbledygook

URL Uniform Resource Locator

WWW World Wide Web

Abstract

The rapid growth of online information has highlighted the Importance of ensuring that readers can understand online text content. This bachelor thesis describes the theory, design, implementation, and evaluation of a tool aimed at objectively measuring the readability of the web by extracting the main component of web documents and analyzing the text's readability. The tool integrates famous readability score formulas to assess texts readability, improving user experience and information accessibility. The paper begins with a review of the existing literature on readability measurment methods and Theoretical foundations, including the selection process of readability formulas. These findings form the basis for developing the software, which has a modular and extensible architecture to allow for future extensions and the possibility of incorporating the tool into the existing RAT project. The tool's core functionality is centered around two main components: web content extraction and readability analysis. The tool extracts textual content from web pages, effectively bypassing navigation menus, advertisements, and other peripheral elements, while the readability analysis component analyzes the complexity of the text using chosen readability formulas depending on the detected language. To validate the measurment, the tool is tested against a provided dataset. The output is evaluated through imperial assessments. This bachelor's thesis enhances user experience by providing a valuable resource for content creators, web developers, and researchers looking to optimize the accessibility and readability of their online content.
Keywords: **Readability measurement, web documents, linguistic analysis, content extraction, web scraping, user experience, information accessibility.**

Chapter 1

Introduction

Readability is a measure of how easy a piece of text is to read.[11] It is measured by different factors, including the length of sentences, the usage of multi-syllabic words, and the complexity of the grammar used in the passage.[12]

1.1 Problem Statement

Measuring the readability of web documents is essential for enhancing user experience, improving accessibility, search engine optimization, and content improvement. Readable web content can also improve user engagement [13] and help with search engine optimization, as search engines can favor well-structured and readable content [14]. By measuring readability content creators can target the intended audience by adjusting the text to their comprehension level [15]. Prioritizing readability leads to, increased user satisfaction, and a positive overall online experience.

1.2 Research Goals

This paper aims to address the topic of assessing the readability of web documents thoroughly, with the following goals in scope:

1. **Readability Formula Selection**: Identify appropriate formulas to calculate readability scores for English, German, and Arabic.

2. **Development for Main Component Extraction**: A software that can extract the main content (body text) from web documents using various Python libraries shall be developed. The software should be

capable of handling various document structures and formats, ensuring accurate extraction of text content while excluding irrelevant elements such as advertisements and side toolbars.

3. **Readability Analysis and Computation**: Implement a tool to analyze extracted test using selected readability formulas. This tool should accurately compute readability scores.

4. **Multi-Language Support**: Expanding the tool's functionality to automatically detect the input language and support English, German, and Arabic, allowing the addition of more languages in the future.

5. **Prefromance Evaluation with Test Data**: Utilize test data to evaluate the accuracy of main component extraction and the differences in computed readability scores using different formulas. This evaluation shall verify the measured scores and the general performance of the developed tool.

1.3 Thesis Outline

This thesis is divided into the following chapters:

1. **Chapter 1: Introduction**
   This chapter presents a quick introduction to the topic, demonstrating the significance of web readability and its impact on usability, stating the research goals, and summarizing the paper's structure.

2. **Chapter 2: Literature Review**
   This chapter reviews the literature on readability assessment formulae, presents previous studies on web documents readability, and provides an overview of current tools and solutions, while diving into the theory behind the selected formulas.

3. **Chapter 3: Methodology**
   This chapter presents an overview of the tool's design and implementation, mentioning used Python libraries and packages.

4. **Chapter 4: Tool Evaluation**
   In this chapter, the developed tool is evaluated using test data.

5. **Chapter 5: Discussion**
   This chapter discusses the implemented solution's possibilities and limitations.

6. **Chapter 6: Conclusion**
   This final chapter summarises the work's main findings and future work possibilities.

7. **Acknowledgment**
   Acknowledgment that this paper is written without the aid of any third party.

8. **Bibliography**
   The Bibliography section lists all the sources cited throughout the paper.

Chapter 2

Literature Review

2.1 History

The first readability formula was presented by Lively and Pressey (1923) when they wanted to choose books for middle school children and became aware of the challenges posed by complicated technical terms. Their approach was to mathematically relate the complexity and the frequency of each word. Even though their approach lacked the potential for widespread use, it served as the cornerstone for future studies. Since then researchers have come a long way in enhancing readability assessment formulae and making them ready for large-scale use.[12]

Afterward came Rudolph Flesch (1948) who is recognized for developing the readability formula that is arguably the most popular and well-known. Due to the success of his formula, Flesch became a significant influence in the readability field. Even now, Microsoft Office Word uses its formula, referred to as the Flesch Reading Ease Readability Formula. The readability of written materials can also be evaluated using computerized methods by modern grammar and editing software. For instance, Microsoft Office Word not only runs spell and grammar checks but also offers details on the passage's readability level. This evaluation is based on the average number of words and sentences per sentence. According to the Flesch Reading Ease Readability Formula, a rating is given on a scale of 100 points, with higher scores indicating easier comprehension.[12]

After that, the Fog-Index and Dale-Chall (1948) Formula were presented. The Dale-Chall Formula adopts sentence length and the presence of "hard words" into account and was developed as an improvement over the Flesch Reading

Ease Formula. Gunning (1952) developed the Fog-Index, which counts the number of words with more than two syllables and the length of sentences.[11]

In 1976, J. Peter Kincaid recognized the potential in Rudolph Flesch's work, as he wanted to derive a universal formula that could be used by the United States Navy for their learning material.[16]

Flesch-Kincaid Grade introduced a grading system replacing the scoring system of the Flesch Reading Ease formula, which allowed more educators and researchers to understand and apply the formula.[16]

2.2 Readability of Web

As people consume more media online than anything else these days, including written content[17], a lot of research has been done to find out how readable this content is, especially when it comes to sensitive content, such as medical articles.

One attempt was by Donna M. D'Alessandro, MD; Peggy Kingsley, BA as they tried To determine the general readability of pediatric patient education materials designed for adults on the World Wide Web by examining a digital library serving the medical information needs of pediatric healthcare providers, patients, and families. Documents from 100 different authoritative Web sites designed for laypersons were evaluated using a built-in computer software readability formula (Flesch Reading Ease and Flesch-Kincaid reading levels)[18].

The results were quite startling as Eighty-nine documents made up the final sample; they covered a wide range of pediatric topics and The average score for Flesch Reading Ease was 57.0, which means that Pediatric patient education materials on the WWW are not written at a reading level appropriate for the average adult[18].

Another research was done by Luke S. Bothun, Scott E. Feeder, Gregory A. Poland took a similar approach as they tried to measure the Readability of COVID-19 vaccine information for the general public. The fact sheets and informational materials are appropriate for the general U.S. population.[19]

The interesting part is that they also reached a similar conclusion as The majority of the informational materials related to the COVID-19 vaccine did not meet appropriate readability levels for the American population, according to this study, which examined fact sheets for three of the major U.S. vaccine manufacturers and other informational materials.[19]

Both studies mentioned above reached the same conclusion, which is that the readability of online documents needs to be improved.[18][19]

2.3 Current Solutions

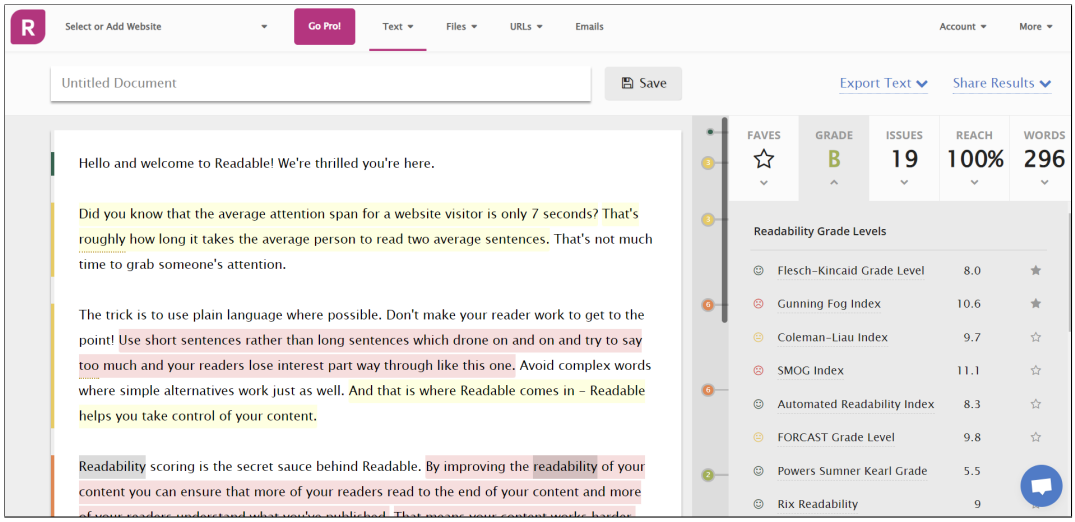


Figure 2.1: Readable.com User Interface

Source: *https://app.readable.com/text/?demo*

- The figure above shows one of the current solutions available to researchers and content creators to measure the readability of any text. The website also provides a solution for extracting text from web pages and measuring the readability of that text, which is similar to the goal of this work, but it is a paid service and not free.[20]
- Multiple free websites such as *readabilityFormulas.com* and textitcharactercalculator.com allow the user to calculate the readability of a piece of text.[5][6]

2.4 Readability Formula Selection

2.4.1 Flesch Reading Ease & Flesch–Kincaid Grade

$$FRE = 206.835 - 1.015\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right) \quad (2.1)$$

(Rudolph Flesch 1948) developed the formula above to measure readability depending on Two factors: the average count of words per sentence and the average count of syllables per word. [16].

The formula yields a score that determines the degree of readability of the text, as shown in Table 2.1.

Score	School Level	Notes
100.0–90.0	5th Grade	Very easy to read. Easily understood by an average 11-year-old student.
90.0–80.0	6th Grade	Easy to read. Conversational English for consumers.
80.0–70.0	7th Grade	Fairly easy to read.
70.0–60.0	8th & 9th Grade	Plain English. Easily understood by 13- to 15-year-old students.
60.0–50.0	10th to 12th Grade	Fairly difficult to read.
50.0–30.0	College	Difficult to read.
30.0–10.0	College Graduate	Very difficult to read. Best understood by university graduates.
10.0–0.0	Professional	Extremely difficult to read. Best understood by university graduates.

Table 2.1: Flesch Reading Ease Readability Score Ranges and School Levels[8].

(Kincaid 1975) worked with Flesh to drive an equation that gives a more understandable result. As shown in Table 2.2 The formula (2.2), the equation yields a number corresponding to one of the grade levels of the U.S. educational system.[16]

$$FKGL = 0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total sentences}}{\text{total words}}\right) - 15.58 \quad (2.2)$$

Index	Reading Level by Grade
20+	Post-graduate plus
17-20	Post-graduate
16	College senior
15, 14, 13	College junior, sophomore, freshman
11-12	High school senior, junior
10	High school sophomore
9	High school freshman
8	8th grade
7	7th grade
6	6th grade

Table 2.2: U.S Grade Level [9]

Both formulas were chosen and used in the tool because they are considered to be one of the most well-known and cited readability formulas, and also since there are a variety of interpretations of the same formulas in other languages with minor changes.[12][21]

For example Toni Amstad (1978) successfully transferred the Flesch Reading Ease formula to the German language by recalculating the word factor, as seen in formula 2.3, because German words are often longer than English words, while sentences are about the same length.[21]

$$FRE(DE) = 180\left(\frac{\text{total words}}{\text{total sentences}}\right) - 84.6\left(\frac{\text{total syllables}}{\text{total words}}\right) \tag{2.3}$$

2.4.2 The Gunning's Fog Index

$$FOG = 0.4\left[\left(\frac{\text{words}}{\text{sentences}}\right) + 100\left(\frac{\text{complex words}}{\text{words}}\right)\right] \tag{2.4}$$

Gunning's Fog index, calculated using the formula (2.4) above and also known as the FOG index, takes a different approach from Flesch's reading ease by measuring average sentence length and introducing a new term: complex words. A word is classified as complex if it has three or more syllables.[22]

The main reason behind including Gunning in the software is the different approach to calculating readability, which facilitates the use of the formula for writing scientific papers, such as abstracts, that need to be clear and concise. [23]

2.4.3 SMOG Index

$$\text{SMOG} = 1.0430\sqrt{\text{number of polysyllables} \times \frac{30}{\text{number of sentences}}} + 3.1291 \tag{2.5}$$

Named by one case study assessing online Parkinson's disease information as "the golden standard, " it is necessary to include this formula due to its wide application and usage, especially in the medical field. [24]

SMOG measures the number of words with more than one syllable and their relative proportion to the number of sentences in a text. The result is a score based on the American educational system, which can be understood using Table 2.2.

2.4.4 Coleman-Liau Index

$$CLI = 0.0588 * L - 0.296 * S - 15.8 \tag{2.6}$$

What makes the formula 2.6 unique is that it takes only two quantities, L, and S in consideration. The first is the average number of letters per hundred words, and the second is the average number of sentences per 100 words.[25]

Like other formulas, its output approximates the U.S. grade level (table 2.2).

2.4.5 Wiener Sachtextformel

Richard Bamberger and Erich Vanecek developed the Wiener Sachtextformel to evaluate non-fiction texts. The calculation is based on the proportion of words with three or more syllables (MS), the average sentence length (SL), the proportion of words with more than six letters (IW), and the proportion of monosyllabic words (ES). There are four different Wiener formulas (2.7 : 2.10) with different weighting of the different criteria.[26]

$$\text{WSTF}_1 = 0,1935 \cdot \text{MS} + 0,1672 \cdot \text{SL} + 0,1297 \cdot \text{IW} - 0,0327 \cdot \text{ES} - 0,875 \tag{2.7}$$

$$\text{WSTF}_2 = 0,2007 \cdot \text{MS} + 0,1682 \cdot \text{SL} + 0,1373 \cdot \text{IW} - 2,779 \tag{2.8}$$

$$\text{WSTF}_3 = 0,2963 \cdot \text{MS} + 0,1905 \cdot \text{SL} - 1,1144 \tag{2.9}$$

$$\text{WSTF}_4 = 0,2744 \cdot \text{MS} + 0,2656 \cdot \text{SL} - 1,693 \tag{2.10}$$

The formulas are oriented toward German grade levels. The results of the index have a scale that starts at school level four and ends at fifteen, whereby from level twelve onwards one should rather speak of difficulty levels than of school levels. Thus, a value of four indicates a very easy text, while fifteen indicates a very difficult text.[26]

2.4.6 OSMAN Index

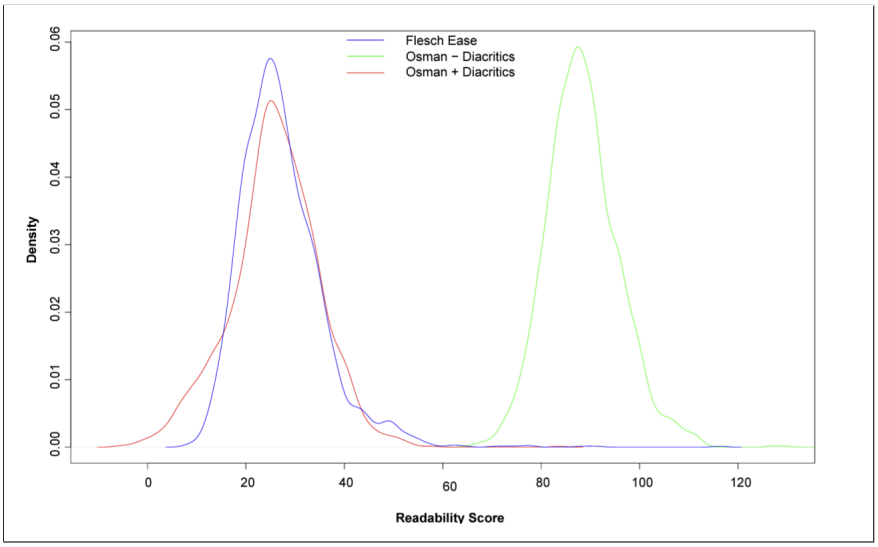


Figure 2.2: Density Distribution Comparison between Flesh Ease and Osman.[1]

Source: *http://www.lancaster.ac.uk/staff/elhaj/docs/elhajlrec2016Arabic.pdf*

OSMAN Index is a novel open-source measure of Arabic text readability devel-

oped by Mahmoud El-Hajand and Paul Rayson (2016). It allows researchers to calculate the readability of Arabic texts. [1]

$$\text{OSMAN INDEX } = 200.791 - 1.015 \times \left(\frac{A}{B}\right) - 24.181 \times \left(\frac{C}{A} + \frac{D}{A} + \frac{G}{A} + \frac{H}{A}\right) \text{ [1]} \tag{2.11}$$

Where 'A' is the total number of words, 'B' is the total number of sentences, 'C' is the number of hard words (words with more than 5 letters - long words), 'D' is the average number of syllables in a word, 'E' is the total number of characters without digits, 'G' is the number of words with more than four syllables, 'H' is the number of 'Faseeh' words, Faseeh are words that contain ("ظ", "وء", "ء", "ئ",) or end with ("ون","وا").[1]

As shown in figure 2.2 Osman was tuned to have a similar score range as Flesch Reading Ease mentioned in Table 2.1.

Chapter 3

Methodology

This chapter presents an overview of the tool's design and implementation, mentioning used Python libraries and packages.

3.1 Software Architecture

3.1.1 Context Diagram

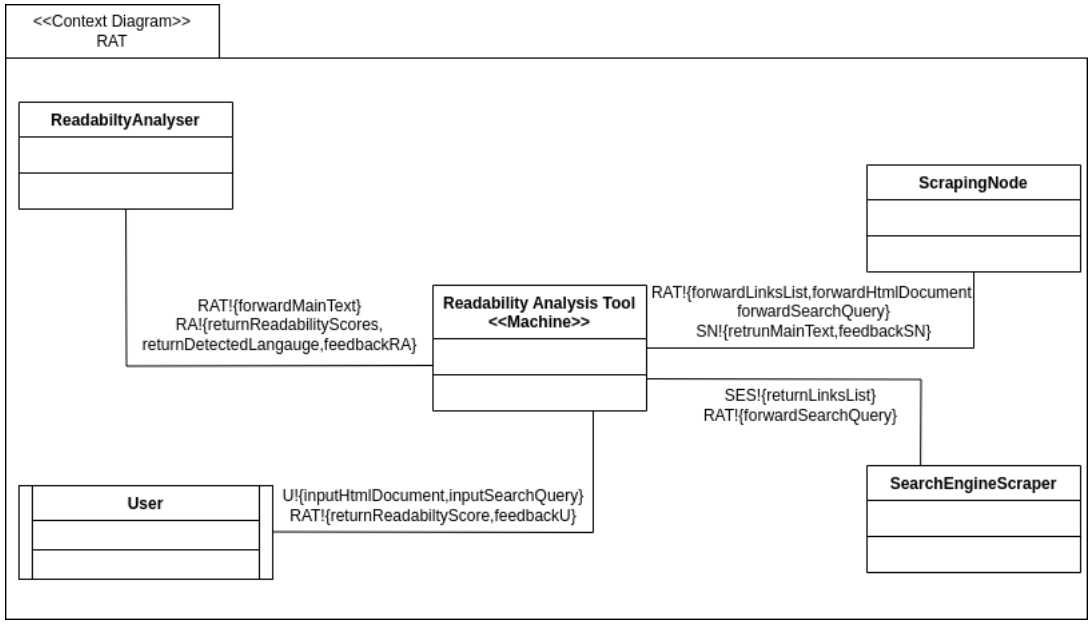


Figure 3.1: Readability Analysis Tool Context Diagram

Figure 3.1 represents the initial design of the software.

- A simple user interface is presented, where users can enter either a quwhiteery or a full URL.
- Depending on user input, the information is either passed directly to the Scraping Node or, in case of a web search, the query is first forwarded to the Search Engin Scraper, which returns a list of URLs that is forwarded to the Scraping Node.
- The scraping node extracts the main content from the web documents(s) and provides clean text passed to the readability analyzer.
- The readability analyzer identifies the extracted text language and returns the calculated readability scores and feedback to the user.

The following sections detail the implementation and tools used for each component shown in Figure 3.1.

3.2 Scraping Node

3.2.1 Trafilatura

Figure 3.2: Trafilatura: A Web Scraping Python Library and Command-Line Tool for Text Discovery and Extraction [2]

Source: *https://github.com/adbar/trafilatura*

Available as both a Python package and command line tool. Trafilatura offers various components for crawling, extraction, and text discovery. Its primary uses include web crawling, downloads, scraping, and extraction of main text, metadata, and comments. Trafilatura is convenient and modular, as it doesn't require a database and can convert the output to several popular formats.

Table 3.1: Web Scraping Library Performance Evaluation. [10]

Source: *https://github.com/scrapinghub/article-extraction-benchmark*

Library Name	Version	F1	Precision	Recall	Accuracy
trafilatura	0.5.1	0.945 ± 0.009	0.925 ± 0.011	0.966 ± 0.009	0.221 ± 0.031
go_readability	bdc-8717	0.943 ± 0.007	0.912 ± 0.009	0.975 ± 0.007	0.210 ± 0.030
readability_js	Feb-2021	0.887 ± 0.012	0.853 ± 0.013	0.924 ± 0.012	0.149 ± 0.026
go_domdistiller	1c90a88	0.927 ± 0.007	0.901 ± 0.010	0.956 ± 0.010	0.066 ± 0.018
news_please	1.5.17	0.911 ± 0.014	0.917 ± 0.013	0.906 ± 0.018	0.249 ± 0.032
goose3	3.1.8	0.887 ± 0.016	0.930 ± 0.015	0.847 ± 0.021	0.227 ± 0.032
inscriptis	1.1.2	0.679 ± 0.015	0.517 ± 0.017	0.993 ± 0.001	0.000 ± 0.000
html2text	2020.1.16	0.662 ± 0.015	0.499 ± 0.017	0.983 ± 0.002	0.000 ± 0.000
justext	2.2.0	0.802 ± 0.018	0.858 ± 0.017	0.754 ± 0.028	0.088 ± 0.021
beautifulsoup	4.9.3	0.665 ± 0.015	0.499 ± 0.017	0.994 ± 0.001	0.000 ± 0.000

Named the Most efficient open-source library According to an evaluation made by K. Lopuhin and nine other contributors, Trafilatura scored the best overall score in terms of precision, recall, and accuracy in comparison to nine other packages.[10]

3.2.2 Simplified Implementation

```
from trafilatura import extract, fetch_url

class ScrapingNode:
    def fetchTextFromURL(self,x):
        self.document = fetch_url(x)   #get HTML-Documents
        print("html fetch success")
        self.text = extract(self.document)  #extract main content
        print("text fetch success")
        return self.text
```

The code snippet above illustrates a simplified implementation of the Scraping node, using only two functions from the library. fetch() accepts a URL as a parameter and returns an HTML document. and then the document is passed to extract(), which returns clean text.

3.3 Search Engine Scraper

Scraping data from search engines can be challenging as different websites and search engines have different structures.[27] Therefore an algorithm is tailored to scrape the first ten results URLs from Google.de. however, the algorithm can be easily modified to work on other search engines

In the following sub-sections, the tools used and the implementation of the algorithm are discussed in detail.

3.3.1 Requests-HTML

Figure 3.3: Requests-HTML: HTML Parsing for Humans (writing Python 3)![3]

Source: *https://requests-html.kennethreitz.org/*

Requests-HTML is a Python library for HTML parsing,[3]. By definition parsing HTML code, means extracting useful information and metadata from a specific HTML code.[28]

For the purpose of extracting the URLs of the first ten search results from google.de requests is used as follows:

```
def get_source(self, url):
    try:
        s = HTMLSession()
        response = s.get(url)
        return response

    except requests.exceptions.RequestException as e:
        print(e)
```

The code snippet above illustrates the usage of requests to retrieve the HTML

code of a URL, which is passed as a parameter to function get() after a new HTML session is created.

3.3.2 Urllib

Figure 3.4: urllib — URL handling modules.[4]

Source: *https://github.com/urllib3/urllib3*

Urllib package is a URL handling module for Python. It is used to fetch URLs using a variety of different protocols and it is split to serval modules, such as:

- urllib.request for opening and reading.
- urllib.parse for parsing URLs
- urllib.error for the exceptions raised
- urllib.robotparser for parsing robot.txt files

The goal is to pass the user's query to "google.de/search, therefore urlib.parse is chosen and used as follows:

```
def search_google(self, query):
    query = urllib.parse.quote_plus(query)
    response = self.get_source("https://www.google.de/search?q=" + query)
    return response
```

3.3.3 Implementation

In the last subsection, we reviewed the usage of requsts and urllib to forward the user query and get the HTML code of the search result page. What is now needed is to filter the code and extract the URLs from it.

```
links = list(response.html.absolute_links)
```

By calling the function above we extract all URLs in the response, however the initial tests showed that the list had a lot of Google domains and other web cash URLs.

Therefore the results are filtered after the inspection of Google's search results page source code.

```
googleDomains = (
            'https://www.google.',
            'https://google.',
            'https://webcache.googleusercontent.',
            'http://webcache.googleusercontent.',
            'https://policies.google.',
            'https://support.google.',
            'https://maps.google.',
        )

        for url in urls[:]:
            if url.startswith(googleDomains):
                links.remove(url)
        return urls
```

3.3.4 Other Search Engines

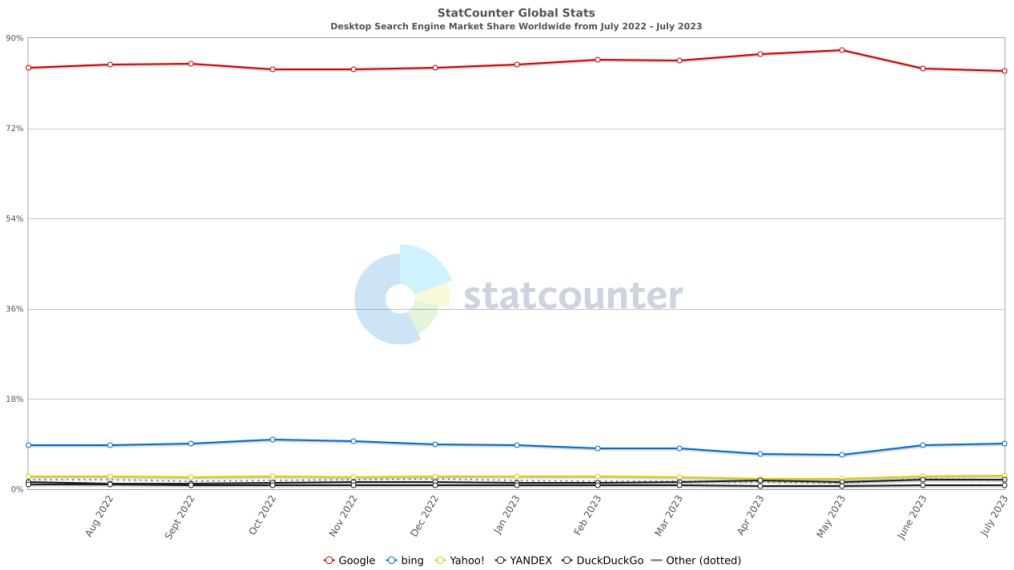


Figure 3.5: Search Engine Usage Globally

Source: *https://gs.statcounter.com/search-engine-market-share*

As shown in Figure 3.5, Google has the highest search engine market share globally[29], so the algorithm is initially tailored to work with Google; however, with exactly two changes, the same algorithm can work with any other search engine.

1. Firstly the URL parsed by Urllib in subsection 3.3.2 needs to be modified depending on the chosen search engine.

2. Similar to "googleDomains" in subsection 3.3.3 a list of internal domains needs to be established to filter the extracted URLs.

3.4 Readability Analyser

3.4.1 py-readability-metrics

The initial suggestion for the implementation of the readability analyzer was to use py-readability-metrics library, as it is very simple to use and it offers the most famous readability formulas.

```
from readability import Readability

r = Readability(text)

r.flesch_kincaid()
r.flesch()
r.gunning_fog()
r.coleman_liau()
r.dale_chall()
r.ari()
r.linsear_write()
r.smog()
r.spache()
```

However, the library had some limitations such as:

1. Supports just English, which is not optimal for the set goals.

2. Does not support text analysis functions, e.g. counting syllables, sentences, and letters, i.e. if another formula is to be used, which the library does not provide, a separate implementation is required.

3.4.2 textstat

Textstat is a ready out-of-the-box library that calculates statistics about text, including readability, grade level, and average reading time. Contrary to other options, it offers a set of functions to count syllables, sentences, and word length.[30]

Function	en	de	es	fr	it	nl	pl	ru
Flesch Reading Ease	✓	✓	✓	✓	✓	✓		✓
Gunning Fog	✓						✓	✓

Table 3.2: Availability of FLesh Reading Ease and Gunning Fog in Textstat.

Source: *https://github.com/textstat/textstat*

Beside offering FLesh Reading Ease and Gunning Fog in eight different languages, the library also offers the following language-specific readabilty formula.

```
>>> import textstat, test_data

>>> textstat.smog_index(test_data) #en
>>> textstat.coleman_liau_index(test_data) #en
>>> textstat.automated_readability_index(test_data) #en
>>> textstat.dale_chall_readability_score(test_data) #en
>>> textstat.difficult_words(test_data) #en
>>> textstat.linsear_write_formula(test_data) #en
>>> textstat.gunning_fog(test_data) #en
>>> textstat.text_standard(test_data) #en
>>> textstat.fernandez_huerta(test_data) #es
>>> textstat.szigriszt_pazos(test_data)  #es
>>> textstat.gutierrez_polini(test_data)  #es
>>> textstat.crawford(test_data)  #es
>>> textstat.gulpease_index(test_data)  #it
>>> textstat.osman(test_data)  #ar
>>> textstat.wiener_sachtextformel(text, variant) #de
```

Additionally, the library offers the following functions:

```
>>> textstat.reading_time(text, ms_per_char=14.69) #Returns the reading
    time of the given text.

>>> textstat.syllable_count(text) #Returns the number of syllables
    present in the given text.

>>> textstat.sentence_count(text) #Returns the number of sentences
    present in the given text.

>>> textstat.char_count(text, ignore_spaces=True) #Returns the number of
    characters present in the given text.

>>> textstat.letter_count(text, ignore_spaces=True) #Returns the number
    of characters present in the given text without punctuation.

>>>textstat.polysyllabcount(text) #Returns the number of words with a
    syllable count greater than or equal to 3.

>>>textstat.monosyllabcount(text) #Returns the number of words with a
    syllable count equal to one.
```

3.4.3 langdetect

langdetect is a language recognition library for Python based on Google's speech recognition module written in Java. The library provides about ninety-nine percent accuracy in recognizing fifty-five languages,[31] which are listed in Table 3.3 below:

Table 3.3: Language Information

Source: *https://github.com/Mimino666/langdetect*

Code	Language	Precision (%)	Items	Details
af	Afrikaans	99.50%	199	en=1, af=199
ar	Arabic	100.00%	200	ar=200
bg	Bulgarian	100.00%	200	bg=200
bn	Bengali	100.00%	200	bn=200
CS	Czech	100.00%	200	cs=200
da	Danish	89.50%	179	da=179, no=14, en=7
de	German	100.00%	200	de=200
el	Greek	100.00%	200	el=200
en	English	100.00%	200	en=200
es	Spanish	100.00%	200	es=200
fa	Persian	100.00%	200	fa=200
fi	Finnish	100.00%	200	fi=200
fr	French	100.00%	200	fr=200
gu	Gujarati	100.00%	200	gu=200
he	Hebrew	100.00%	200	he=200
hi	Hindi	100.00%	200	hi=200
hr	Croatian	100.00%	200	hr=200
hu	Hungarian	100.00%	200	hu=200
id	Indonesian	100.00%	200	id=200
it	Italian	100.00%	200	it=200
ja	Japanese	100.00%	200	ja=200
kn	Kannada	100.00%	200	kn=200
ko	Korean	100.00%	200	ko=200
mk	Macedonian	100.00%	200	mk=200
ml	Malayalam	100.00%	200	ml=200
mr	Marathi	100.00%	200	mr=200
ne	Nepali	100.00%	200	ne=200
nl	Dutch	100.00%	200	nl=200
no	Norwegian	99.50%	199	da=1, no=199
pa	Punjabi	100.00%	200	pa=200
pl	Polish	100.00%	200	pl=200
pt	Portuguese	100.00%	200	pt=200
ro	Romanian	100.00%	200	ro=200
ru	Russian	100.00%	200	ru=200
sk	Slovak	100.00%	200	sk=200
so	Somali	100.00%	200	so=200
sq	Albanian	100.00%	200	sq=200
SV	Swedish	100.00%	200	sv=200
SW	Swahili	100.00%	200	sw=200
ta	Tamil	100.00%	200	ta=200
te	Telugu	100.00%	200	te=200
th	Thai	100.00%	200	th=200
tl	Tagalog	100.00%	200	tl=200
tr	Turkish	100.00%	200	tr=200
uk	Ukrainian	100.00%	200	uk=200
ur	Urdu	100.00%	200	ur=200
vi	Vietnamese	100.00%	200	vi=200
zh-cn	Simplified Chinese	100.00%	200	zh-cn=200
zh-tw	Traditional Chinese	100.00%	200	zh-tw=200
Total		99.77%	9800	

Langdectec is quite simple to use, and a language detection module in the APP can be built using the following code-snipped:

```
>>> from langdetect import detect
>>> detect("War doesn't show who's right, just who's left.")
'en'
```

Chapter 4

Testing & Evaluation

In this chapter, the performance of the implemented tool is tested using a data set of the top fifty domains in Germany[32]. the chapter is split into thre sections, The first section is concerned with testing the content extraction part, while the second focuses on the accuracy of the readability scores measured for the collected text. afterwards, an overall evaluation of the tool's performance is given.

4.1 Text Extraction Testing & Evaluation

For testing content extraction, the data set is compiled, and websites with dominant textual content are included.e.g. Youtube, Instagram, and other domains are not considered due to their dominant visual content.

1. TEST CASE: Wikipidia.org

```
    source: https://en.wikipedia.org/wiki/Eryxias_(dialogue)
        \\\\\\\\\\\\\\\\|EXTRACTED TEXT|\\\\\\\\\\\\\\\\
Eryxias (//; Greek:       )is a Socratic dialogue attributed to
    Plato, but which is considered spurious. It is set in the Stoa
    of Zeus Eleutherios, and features Socrates in conversation with
    Critias, Eryxias, and Erasistratus (nephew of Phaeax).[1]
The dialogue concerns the topic of wealth and virtue. The position
    of Eryxias that it is good to be materially prosperous is
    defeated when Critias argues that having money is not always a
    good thing. Socrates then shows that money has only a
    conventional value.[2] In an argument addressed to Critias,
    Socrates concludes that money can never be considered useful,
    even when it is used to buy something useful.[2] The final
    conclusion of the Eryxias is that the most wealthy are the most
    wretched because they have so many material wants.[3]
References\[edit]
- A. E. Taylor, (2001), Plato: the man and his work, page 548. Dover
- John Madison Cooper, D. S. Hutchinson, (1997), Plato, Complete
```

```
    works, page 1718. Hackett Publishing
- William Keith Chambers Guthrie, (1986), A history of Greek
    philosophy, page 397. Cambridge University Press
```

Eryxias (/ɪˈrɪksiəs/; Greek: Ἐρυξίας) is a Socratic dialogue attributed to Plato, but which is considered spurious. It is set in the Stoa of Zeus Eleutherios, and features Socrates in conversation with Critias, Eryxias, and Erasistratus (nephew of Phaeax).[1]

The dialogue concerns the topic of wealth and virtue. The position of Eryxias that it is good to be materially prosperous is defeated when Critias argues that having money is not always a good thing. Socrates then shows that money has only a conventional value.[2] In an argument addressed to Critias, Socrates concludes that money can never be considered useful, even when it is used to buy something useful.[2] The final conclusion of the *Eryxias* is that the most wealthy are the most wretched because they have so many material wants.[3]

References [edit]

1. ^ A. E. Taylor, (2001), *Plato: the man and his work*, page 548. Dover
2. ^ a b John Madison Cooper, D. S. Hutchinson, (1997), *Plato, Complete works*, page 1718. Hackett Publishing
3. ^ William Keith Chambers Guthrie, (1986), A history of Greek philosophy, page 397. Cambridge University Press

External links [edit]

- Works related to Eryxias at Wikisource
- *Eryxias*, in a collection of Plato's Dialogues at Standard Ebooks
- *Eryxias* public domain audiobook at LibriVox

Part of a series on

Platonism

Plato from Raphael's *The School of Athens* (1509–1511)

Life · Allegory of the cave · Theory of forms · Form of the Good · Theory of soul · Epistemology · Analogy of the sun · Analogy of the divided line · Political philosophy · Philosopher king · Ship of State · Euthyphro dilemma · Ring of Gyges · Myth of Er · Demiurge · Atlantis

The works of Plato

Euthyphro · *Apology* · *Crito* · *Phaedo* · *Cratylus* · *Theaetetus* · *Sophist* · *Statesman* · *Parmenides* · *Philebus* · *Symposium* · *Phaedrus* · *First Alcibiades* · *Second Alcibiades* · *Hipparchus* · *Rival Lovers* · *Theages* · *Charmides* · *Laches* · *Lysis* · *Euthydemus* · *Protagoras* · *Gorgias* · *Meno* · *Hippias Major* · *Hippias Minor* · *Ion* · *Menexenus* · *Clitophon* · *Republic* · *Timaeus* · *Critias* · *Minos* · *Laws* · *Epinomis* · *Epistles* · *Definitions* · *On Justice* · *On Virtue* ·

Figure 4.1: Snapshot of the Original Wikipedia Article.

Source: *https://en.wikipedia.org/wiki/Eryxias(dialogue)*

After testing a set of documents from Wikipedia.org in different languages, the tool showed word-to-word accuracy in extracting the main article content, however "[Edit]", or "[Bearbeiten | Quelltext bearbeiten]" for German articles need to be manually filtered. After applying some filters to retrieved text the the results are accurate.

```
source:https://de.wikipedia.org/wiki/Natrium
    \\\\\\\\\\\\\\\|EXTRACTED TEXT|\\\\\\\\\\\\\\\
Geschichte
Die Herstellung von elementarem Natrium gelang erst im Jahre 1807
    Humphry Davy durch Elektrolyse von geschmolzenem
    Natriumhydroxid ("Atznatron) unter Verwendung von Voltaschen
    S ulen als Stromquelle. Wie er am 19. November 1807 vor der
    Royal Society in London berichtete, gewann er zwei verschiedene
    Metalle: Das in Soda enthaltene Natrium nannte er Sodium, was
    die noch gebr uchliche Bezeichnung des Metalls im
    franz sischen und englischsprachigen Raum ist; das andere
    Metall nannte er Potassium (Kalium). Den Namen Natrium schlug
    1811 Berzelius vor.[15]
```

Geschichte

Humphry Davy stellte elementares Natrium her

Die Herstellung von elementarem Natrium gelang erst im Jahre 1807 Humphry Davy durch Elektrolyse von geschmolzenem Natriumhydroxid (*Ätznatron*) unter Verwendung von Voltaschen Säulen als Stromquelle. Wie er am 19. November 1807 vor der *Royal Society* in London berichtete, gewann er zwei verschiedene Metalle: Das in Soda enthaltene Natrium nannte er *Sodium*, was die noch gebräuchliche Bezeichnung des Metalls im französischen und englischsprachigen Raum ist; das andere Metall nannte er *Potassium* (Kalium). Den Namen *Natrium* schlug 1811 Berzelius vor.[15]

Natriumverbindungen sind im Gegensatz zum elementaren Metall schon sehr lange bekannt. Schon früh wurden die aus Meerwasser, Salzseen oder Erdlagerstätten gewonnenen Produkte teils bis in ferne Regionen gehandelt. Sie enthielten überwiegend das Natriumchlorid (*Kochsalz*) als die wichtigste Natriumverbindung in fester Form. Deren wässrige Lösung wird vornehmlich für Speisesalz bei der Salzgewinnung zunächst im Einengen zunehmend konzentriert (Gradierwerke) und das Lösungsmittel schließlich verdampft (Salinen) – durch Verdunsten von Meerwasser oder durch Eindampfen von Sole salzhaltiger Quellen oder aus Salzbergwerken. Der Handel mit Salz machte manche Städte reich und prägte ihren Namen, so Salzgitter und Salzburg. Bei anderen weist *Hall* im Ortsnamen auf die Salzgewinnung[16] hin (z. B. Hallstatt, Hallein, Niedernhall, Bad Hall, Bad Reichenhall, Schwäbisch Hall, Hall in Tirol. Halle und Schweizerhalle). Daneben wurden auch andere natürlich vorkommende Natriumverbindungen wie Natriumcarbonat (*Soda*) und Natriumhydrogencarbonat (*Natron*) sowie Natriumnitrat (*Natronsalpeter*) schon in der Antike gewonnen und gehandelt. Die Ägypter bezeichneten das aus Natronseen gewonnene Soda bzw. Natron als *netjerj* (*ntr*). Die Griechen entlehnten dieses Wort als νίτρον *nitron*, die Römer als *nitrium*, die Araber als *natrun*.[15]

Figure 4.2: Snapshot of the Original Wikipedia Article

Source: *https://de.wikipedia.org/wiki/Natrium*

2. TEST CASE: Bild.de

```
Source:https://www.bild.de/unterhaltung/leute/leute/mohamed-al-fayed
    -tot-sein-sohn-dodi-al-fayed-verungluecke-mit-
    prinzessin-diana-85265908.bild.html
        \\\\\\\\\\\\\\\|EXTRACTED TEXT|\\\\\\\\\\\\\\\

Der I'm "agyptischen Alexandria (5,2 Mio. Einwohner) geborene
    Milliard'"ar arbeitete erst als Coca-Cola-Strassenh"andler,
    Verkufer von N"ahmaschinen und Lehrer. Reich wurde er, als er
    in die Waffenh"andler-Familie Khashoggi einheiratete und dort
    eine f"uhrende Position in einer saudischen Import-Firma
    erhielt. Aus der Ehe mit Gattin Samira stammte Sohn Dodi
    Al-Fayed.
```

Text extraction works perfectly for the website in question, and no post-filtration is needed.

Der im ↗ ägyptischen Alexandria (5,2 Mio. Einwohner) geborene Milliardär arbeitete erst als Coca-Cola-Straßenhändler, Verkäufer von Nähmaschinen und Lehrer. Reich wurde er, als er in die Waffenhändler-Familie Khashoggi einheiratete und dort eine führende Position in einer saudischen Import-Firma erhielt. Aus der Ehe mit Gattin Samira stammte Sohn Dodi Al-Fayed.

Figure 4.3: Snapshot of the Original from BILD.de Article.

Source: *https://www.bild.de/unterhaltung/leute/leute/mohamed-al-fayed-tot-sein-sohn-dodi-al-fayed-verunglueckte-mit-prinzessin-diana-85265908.bild.html*

3. TEST CASE: kicker.de

Figure 4.4: Snapshot of the Original kicker.de Article

Source: *https://www.kicker.de/dortmund-die-geister-der-vergangenheit-sind-zurueck-966735/artikel*

The website prevents the tool from extracting any text, which makes it a failed test case.

from the past 3 cases we can deduce 4 expected results:

0. The tool does not work for the website in question: There are many reasons why this can happen, and in the next chapter, we will talk more about it and how to prevent it.

1. low-quality extracted text: The extraction was successful; however, that data contains a lot of nonsense and can lead to faulty readability scores.

2. high-quality extracted text: The extraction was successful, and the extracted text contains minor problems that can be easily filtered.

3. word-to-word accuracy extracted text. The extracted text is an exact match to the main content on the website.

Table 4.1: Top 50 Used Domains in Germany

Index	**Domain**	**ETA**	**DVC**
1	wikipedia.org	2	
2	spiegel.de	0	
3	stern.de	3	
4	gala.de	3	
5	bild.de	3	
6	t-online.de	3	
7	tagesschau.de	3	
8	kicker.de	0	
9	zdf.de	3	✓
10	sueddeutsche.de	3	
11	bunte.de	3	
12	web.de	3	
13	faz.net	3	
14	augsburger-allgemeine.de	3	
15	sportschau.de	3	
16	ndr.de	3	
17	welt.de	3*	
18	zeit.de	3*	
19	rtl.de	3	
20	swp.de	3*	
21	tagesspiegel.de	3	
22	vip.de	3	✓
23	md.de	2	✓
24	prosieben.de	1	✓
25	kino.de	3	
26	fr.de	3	
27	transfermarkt.de	0	✓
28	filmstarts.de	3	

Continued on the next page

Index	Domain	ETA	DVC
29	sportbuzzer.de	3	
30	sport1.de	3	
31	weltfussball.de	3	
32	focus.de	3	
33	rp-online.de	0	
34	sky.de	1	✓
35	dazn.com	1	✓
36	spox.com	3	
37	br.de	3	
38	mdr.de	3	
39	bundesregierung.de	3	
40	ran.de	3	
41	fernsehserien.de	3	
42	facebook.com	3*	✓
43	tz.de	3	
44	amazon.de	0	✓
45	eurosport.de	3	
46	deutschlandfunk.de	3	
47	news.de	3	
48	dw.com	3	
49	dfb.de	3	
50	moviepilot.de	0*	

- 3* requires a paid subscription or login is required, but the free part is successfully extracted.
- 0* Website is down.

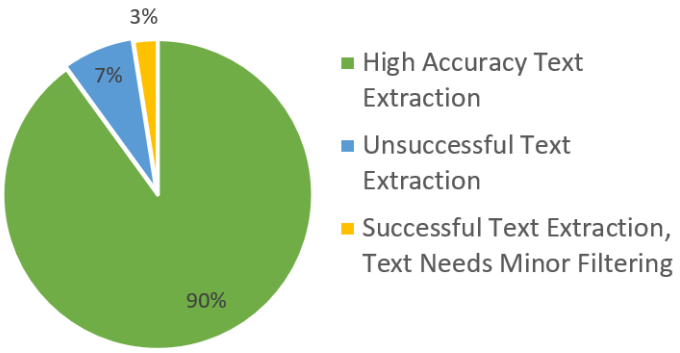


Figure 4.5: Text Extraction Success and Accuracy.

According to the evaluation in Figure 4.5, the content extraction component of the tool had a success rate of 92.5 percent, which is quite promising. The discussion chapter will discuss the reason for this and how we can reach higher success rates.

4.2 Readability Measurement Testing & Evaluation

To test the readability scores produced by the tool, samples of texts with an expected readability range are gathered for each Language measured by the developed software and then measured again by open-source solutions available online. The expected result is not an exact match but a correlation between the measured scores and the expected ranges for each sample.

4.2.1 English

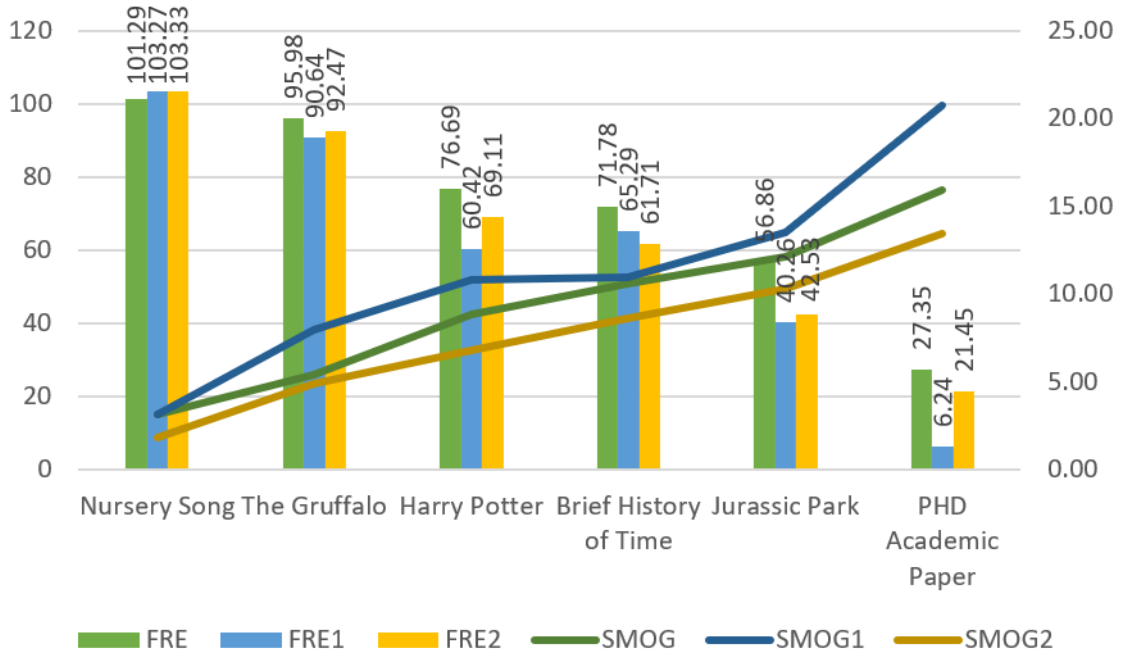


Figure 4.6: Comparison Between Readability Scores (FRE and SMOG) for English Language across the Developed Tool & Two Other Online Solutions. [5] [6]

Although the tool scores for Flesch Readability Ease were a little more optimistic than the other solutions, which can mean that the developed solution needs minor tuning for FRE, a direct correlation between the scores measured by the developed tool, the scores measured by the other solutions and the expected result for the chosen samples can be established, which principally means that the tool works.

On the other hand, for SMOG, we can see the Green line, which presents the measured scores by the developed tool falling exactly in between SMOG1 and SMOG2 measured using other solutions, which, in terms of expected results, is quite satisfying.

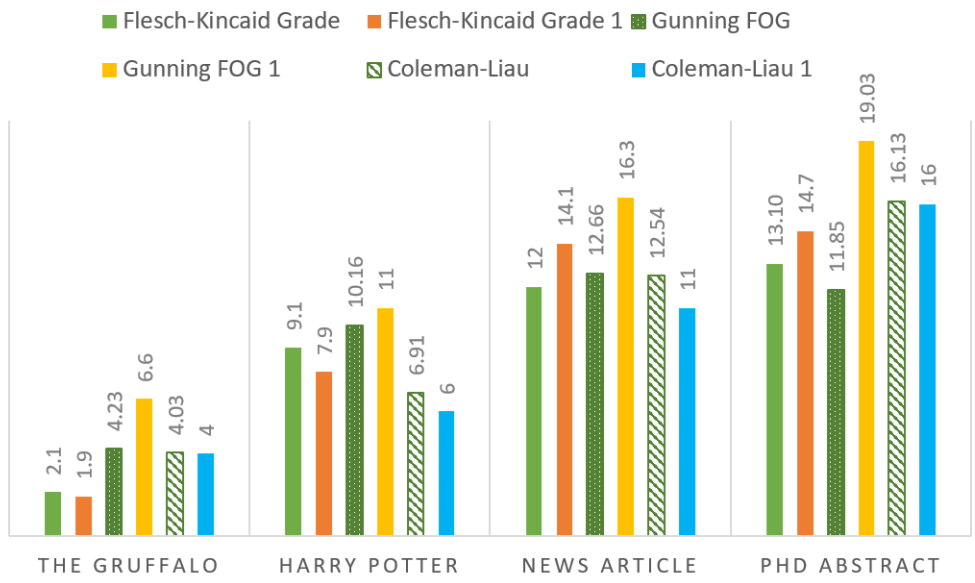
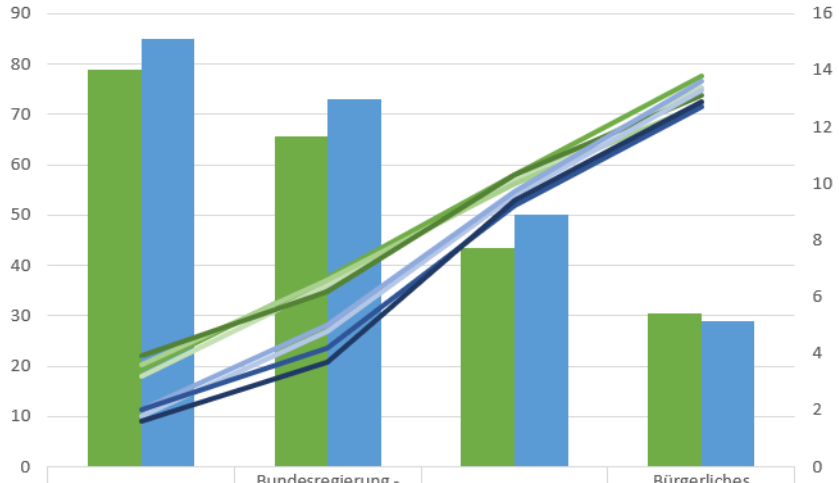


Figure 4.7: Comparison between Readability Scores for Grade-based Formulas in English across the Developed Tool & another Online Solutions.[5]

When testing grade level-based scores, the implemented APP showed more accuracy in terms of expected results than an online tool; for example, a Ph.D. abstract the online tool scored 19.03 on the Gunning FOG index which is not a reasonable result. Another example is Harry Potter. The implemented tool scored 9.1 on FKGL, which falls exactly in the range for the targeted audience for Harry Potter.

4.2.2 German



	Alle Vögel Sind Da!	Bundesregierung - Leichter Sprache	Bild.de	Bürgerliches Gesetzbuch (BGB)
FRE	78.95	65.7	43.4	30.5
FRE1	85	73	50	29
1. Wiener Sachtextformel	3.2	6.4	10.1	13.4
2. Wiener Sachtextformel	3.4	6.7	10.3	13.8
3. Wiener Sachtextformel	3.6	6.6	10	12.8
4. Wiener Sachtextformel	3.9	6.2	10.3	13.1
1. Wiener Sachtextformel **1	1.8	4.8	9.6	13.3
2. Wiener Sachtextformel **1	2	5	9.7	13.6
3. Wiener Sachtextformel **1	2	4.2	9.2	12.7
4. Wiener Sachtextformel **1	1.6	3.7	9.4	12.9

Figure 4.8: Comparison Between Readability Scores (FRE and Wiener Sachtextformel) for the German Language across the Developed Tool & another Online Solution.[7]

For German text, the chosen samples are a children's song, a simplified version of the German government's official website, an online news website (Bild.de), and the German book of the law.

As we can observe in Figure 4.7 there is a direct coloration between the expected results, the measured scores with the developed tool, and the scores measured by the other solutions.

4.2.3 Arabic

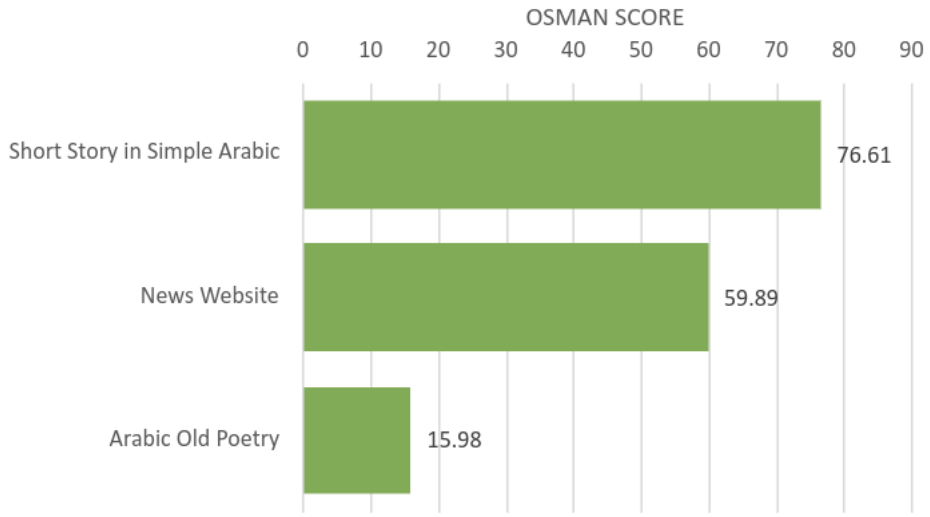


Figure 4.9: Overview of Readability Scores (OSMAN INDEX) for Arabic Language.

No online solution is found to be tested along the implemented software. The software performance is tested directly with three samples with a distinguishable difference between the expected readability scores.

As shown in Figure 4.9 the results match the expected ranges for the samples chosen, with a score of 76.61 for a short story written in simple Language, 59.89 for a news article, and a very complex Arabic poetry that even an educated Arabic speaker would have problems understanding it.

4.3 Overall Performance Evaluation

- The performance of the content extraction component is on point, with a success rate of 93% and 90% word-to-word extraction accuracy. the next chapter will discuss some general content extraction and web scraping limitations.
- The measured readability scores for English, German, and Arabic are accurate and match other open-source solutions online and, most im-

portantly, fall in the expected ranges for each sample. However, some measured scores using Flech Reading Ease for English could be tuned.

- Mentioning the two points above and after testing the tool's functionality in general with a large sum of documents from different resources and languages. The tool shows excellent performance and accuracy in extracting and analyzing text from web documents.

Chapter 5

Discussion

5.1 Limitations

In the last chapter, the software failed to extract the text in the test case: kicker.de

After testing the application with different websites, it was clear that it is possible for some websites to detect if a human or a crawler is sending the request. Some websites even explicitly state that it is not allowed to scrape their content.

Figure 5.1: dict.cc Online German-English Dictionary.

Source: *https://www.dict.cc/*

For example, when trying to extract text from dict.cc, the following text is extracted instead:

```
Extracted text from dict.cc
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
Please don't run crawlers against dict.cc and don't try to make the
    dictionary available offline.Drop me a short mail - paul@dict.cc - if
    you feel this message to be in error!
```

5.2 Possible Upgrades

- Proxy: A proxy is a server between the user and the target website. The proxy server has its IP address; therefore, when a user requests to access a website via a proxy, the website sends and receives the data to the proxy server IP, which forwards it to the user.[28]

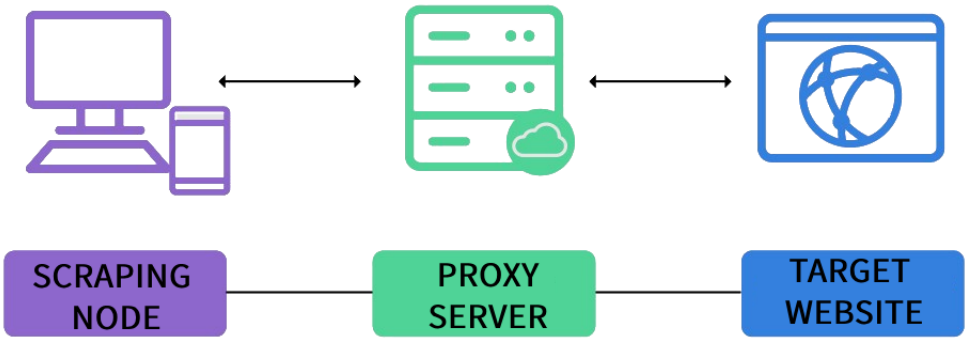


Figure 5.2: Using Proxy instead of Connecting Directly to Target website.

As shown in Figure 5.2 The developed tool can use a proxy server to hide its identity and make its traffic look like regular user traffic.[28]

- GUI: A simple graphical user interface can positively impact the general user experience.
- More Languages: The modular structure of the APP and the libraries used, allow the implementation to be easily extended to support more languages.

Chapter 6

Conclusion

6.1 Summary

The checklist in the table below demonstrates the original list of goals of the paper and what has been achieved:

Table 6.1: Summary of Goals Achieved

Goal	**Achieved**
Readability Formula Selection	✓
Development for Main Component Extraction	✓
Development for Readability Analysis	✓
Multi-Language Support	✓
Performance Evaluation with Test Data	✓

To summarize, research has been done to select a set of suitable readability formulas. A software has been implemented to extract the main text of any web document and analyze its readability using different readability formulas. The implementation has been extended to support Arabic as an example of the tool's capability for multi-language support. Afterwards, the language extraction and readability analysis components were tested and evaluated using test data to ensure the tool is functional and accurate. Later the results of the testing are discussed in the discussion chapter, mentioning the the current limitations and possible solutions and upgrades for the actual implementation.

Figure 6.1: Relevance Assessment Tool Logo.

Source: *https://searchstudies.org/research/rat/*

6.2 Future Work

As mentioned in the discussion chapter, the implemented tool has the potential for expansion by adding support for more languages and adding more readability formulas for different applications.

Due to its modular design and convincing performance, the tool has the potential to be integrated into the existing RAT in the future.

Declaration in lieu of an oath

I hereby solemnly declare that I have independently completed this work or, in the case of group work, the part of the work that I have marked accordingly. I have not made use of the unauthorised assistance of third parties. Furthermore, I have used only the stated sources or aids and I have referenced all statements (particularly quotations) that I have adopted from the sources I have used verbatim or in essence.

I declare that the version of the work I have submitted in digital form is identical to the printed copies submitted.

I am aware that, in the case of an examination offence, the relevant assessment will be marked as 'insufficient' (5.0). In addition, an examination offence may be punishable as an administrative offence (Ordnungswidrigkeit) with a fine of up to €50,000. In cases of multiple or otherwise serious examination offences, I may also be removed from the register of students.

I am aware that the examiner and/or the Examination Board may use relevant software or other electronic aids in order to establish an examination offence has occurred.

I solemnly declare that I have made the previous statements to the best of my knowledge and belief and that these statements are true and I have not concealed anything.

I am aware of the potential punishments for a false declaration in lieu of oath and in particular of the penalties set out in Sections 156 and 161 of the German Criminal Code (Strafgesetzbuch; StGB), which I have been specifically referred to.

Mohamed Elnaggar, Duisburg, den 07.09.2023

Bibliography

[1] M. EL-Haj and P. Rayson, "Osman – a novel arabic readability metric," 05 2016.

[2] A. Barbaresi, "Trafilatura: A Web Scraping Library and Command-Line Tool for Text Discovery and Extraction," in *Proceedings of the Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pp. 122–131, Association for Computational Linguistics, 2021.

[3] K. Reitz, "Requests-html: Html parsing for humans (writing python 3)!¶." `https://github.com/psf/requests-html`, 2019.

[4] "urllib python module." `https://docs.python.org/3/library/urllib.html`. Accessed on: 10.07.2023.

[5] "Free text readability consensus calculator." `https://readabilityformulas.com/free-readability-formula-tests.php/`. Accessed on: 04.08.2023.

[6] "Character calculator." `https://www.charactercalculator.com/`. Accessed on: 04.08.2023.

[7] "Text analyse tool online." `https://gratis-ecke.de/webtools/text-analyse-tool-online.html/`. Accessed on: 07.08.2023.

[8] A. Corazza, V. Maggio, and G. Scanniello, "Coherence of comments and method implementations: a dataset and an empirical investigation," *Software Quality Journal*, vol. 26, pp. 751–777, 2018.

[9] D. Eleyan, A. Othman, and A. Eleyan, "Enhancing software comments readability using flesch reading ease score," *Information*, vol. 11, pp. 1–25, 09 2020.

[10] K. Lopuhin, "Evaluating quality of article body extraction for commercial services and open-source libraries." `https://github.com/scrapinghub/article-extraction-benchmark/releases/download/v1.0.0/paper-v1.0.0.pdf`, 2020.

[11] W. H. DuBay, "The principles of readability.," *Online Submission*, 2004.

[12] M. Zamanian and P. Heydari, "Readability of texts: State of the art.," *Theory & Practice in Language Studies*, vol. 2, no. 1, 2012.

[13] J. T. Richards and V. L. Hanson, "Web accessibility: a broader view," in *Proceedings of the 13th international conference on World Wide Web*, pp. 72–79, 2004.

[14] H. Antunes and C. T. Lopes, "Readability of web content," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–4, 2019.

[15] R. Matsuno, R. Gilbert, K. Matsuo, and Y. Tsutsumi, "Miwit: Integrated esl/efl text analysis and readability tools for content creation in microsoft word," in *EdMedia+ Innovate Learning*, pp. 3093–3102, Association for the Advancement of Computing in Education (AACE), 2011.

[16] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," 1975.

[17] D. F. Roberts and U. G. Foehr, "Trends in media use," *The future of children*, pp. 11–37, 2008.

[18] D. M. D'Alessandro, P. Kingsley, and J. Johnson-West, "The readability of pediatric patient education materials on the world wide web," *Archives of pediatrics & adolescent medicine*, vol. 155, no. 7, pp. 807–812, 2001.

[19] L. S. Bothun, S. E. Feeder, and G. A. Poland, "Readability of covid-19 vaccine information for the general public," *Vaccine*, vol. 40, no. 25, pp. 3466–3469, 2022.

[20] "Readable.com homepage." `https://readable.com/`. Accessed: 20.08.2023.

[21] T. Amstad, *Wie verständlich sind unsere Zeitungen? Universität Zürich.* PhD thesis, Dissertation, 1978.

[22] R. Gunning, "The technique of clear writing," *(No Title)*, 1952.

[23] R. Goldbort, "Readable writing by scientists and researchers," *Journal of Environmental health*, vol. 63, no. 8, pp. 40–40, 2001.

[24] P. R. Fitzsimmons, B. Michael, J. L. Hulley, and G. O. Scott, "A readability assessment of online parkinson's disease information.," *The journal of the Royal College of Physicians of Edinburgh*, vol. 40, no. 4, pp. 292–296, 2010.

[25] M. Coleman and T. L. Liau, "A computer readability formula designed for machine scoring.," *Journal of Applied Psychology*, vol. 60, no. 2, p. 283, 1975.

[26] R. Bamberger and E. Vanecek, *Lesen-Verstehen-Lernen-Schreiben: die Schwierigkeitsstufen von Texten in deutscher Sprache*. Jugend und Volk, 1984.

[27] M. Dogucu and M. Çetinkaya-Rundel, "Web scraping in the statistics and data science curriculum: Challenges and opportunities," *Journal of Statistics and Data Science Education*, vol. 29, no. sup1, pp. S112–S122, 2021.

[28] P. C. Patil, P. M. Chawan, and P. M. Chauhan, "Parsing of html document," *IJARCET In*, 2012.

[29] "Global stats state counter." `https://gs.statcounter.com/search-engine-market-share/desktop/worldwide/`. Accessed: 20.08.2023.

[30] A. Ward, "Textstat: A simple python library to calculate readability, complexity, and grade level of a text." `https://github.com/shivam5992/textstat`, Year of access, 2023. Accessed on: Date, 13.07.2023.

[31] M. Danilák, "langdetect." `https://github.com/Mimino666/langdetect`, 2014.

[32] S. Sünkler, D. Lewandowski, S. Schultheiß, N. Yagci, D. Sygulla, and S. von Mach, "Asist 2022," Apr 2023.