Wordmonkey

Inhalt

[Allgemeines 1](#_Toc461546307)

[Beschreibung 1](#_Toc461546308)

[Umgebung 2](#_Toc461546309)

[Zweck und Funktionen 2](#_Toc461546310)

[WordExtractor 2](#_Toc461546311)

[ListComparator 2](#_Toc461546312)

[MinimalPairs 2](#_Toc461546313)

[Groupier 2](#_Toc461546314)

[Crawler 2](#_Toc461546315)

[Funktionsweise 3](#_Toc461546316)

[Fehlerbehandlung 3](#_Toc461546317)

[Logging 3](#_Toc461546318)

[Struktur 3](#_Toc461546319)

[User Manual 4](#_Toc461546320)

[Basics 4](#_Toc461546321)

[WordExtractor 4](#_Toc461546322)

[ListComparator 4](#_Toc461546323)

[MinimalPairs 5](#_Toc461546324)

[Groupier 5](#_Toc461546325)

[Crawler 5](#_Toc461546326)

[Programmer Manual 5](#_Toc461546327)

[Technik 5](#_Toc461546328)

[Testfälle & Testprotokoll 5](#_Toc461546329)

[Flaschenhälse 5](#_Toc461546330)

[Module 5](#_Toc461546331)

# Allgemeines

## Beschreibung

Wordmonkey ist ein Tool welches den Umgang mit Wortlisten vereinfachen und automatisieren soll. Geschrieben wurde es mit dem Fokus Worte die zu einander eine Verbindung haben in den selben Kontext zu bringen um diese besser verarbeiten zu können. Die Hauptaufgabe liegt in der heraus Filterung von Deutschen Wörtern und deren Gruppierung nach Ähnlichkeit. Das ganze Programm lässt sich in drei Sektoren unterteilen. Im ersten Sektor befinden sich Werkzeuge zum einlesen von Ressourcen aus denen dann die entsprechenden Worte erkannt werden so dass Wordmonkey im weiteren Verlauf mit Ihnen arbeiten kann. Wordmonkey ist Modular aufgebaut und versucht möglichst viele Funktionen wieder zu verwenden. In dem Sinne gibt es nur ein Modul für den Import und ein den Export. Diese werden dann bei allen weiteren Funktionen eingesetzt. Der Zweite Sektor befasst sich mit der Gruppierung dieser Wörter. Dabei hat man zwei Möglichkeiten. Entweder man gruppiert nach Unterschieden. Dies heisst ein Zeichen ist anders oder es hat eins mehr oder weniger. Natürlich gibt es auch die Möglichkeit den Unterschied zu erweitern und somit auf den Unterschied von 2,3 und mehr Zeichen zu Gruppieren. Man muss allerdings auch damit rechnen das bei einem grösseren Unterscheid auch die rechen Dauer entsprechen länger wird. Der Dritte und letzte Sektor verfügt über Tools zum Erstellen von Wortlisten aus Internet oder lokalen Ressourcen. Dabei wurden auch Tools integriert die bestehende Wortlisten erweitern können durch Phonetische Schrift oder Synthesizer aufnahmen.

### Umgebung

Wordmonkey wurde unter Windows 10 geschrieben und getestet. Aus Gründen der Tendenz zu längeren Methoden verursacht durch mehrere Schritte der Verarbeitung wurde die Skriptsprache Python gewählt. Es muss noch hinzugefügt werden das Wordmonkey die Version 3.4 benutzt und das diese nicht mit der Version 2 Kompatibel ist. Es werden jedoch keine 3th Party Packages verwendet. Somit braucht man nur einen laufenden Python >3 Interpreter um das Programm zum Laufen zu bringen. Es wird auch kein Plattform spezifischer Code benutz. So kann man davon ausgehen das Wordmonkey auch unter anderen Windows Versionen oder Linux wie auch Macintosh laufen würde. Falls die eine Executable von Wordmonkey benötigt wird kann man dies mit dem Package cx\_Freeze machen, welches auch mit Python 3.4 kompatibel ist.

## Zweck und Funktionen

### WordExtractor

Der WordExtractor wird mit dem Comandozeilenargument –wx angesteuert und ist in der Lage Wörter aus Files mit verschiedenen Daten heraus zu extrahieren. Wordmonkey geht die Files Zeilenweise durch und schaut ob die entsprechende Line dem Muster eines Wortes entspricht und vor allem auch dem eines deutschen Wortes. Als Argumente muss man das Inputfile oder Inputdirectory angeben und ein Output File (-i,-o). Der Prefile\_analyzer überprüft ob das File nicht korrupt ist, ob es überhaupt existiert und ob die Kodierung stimmt. UTF-8 ist die Standard Kodierung. Es ist auch möglich mehrere Datensätze in diesem Sinn von einer Zeile aus hereinlesen. Ein Beispiel wäre, wenn man dazu noch die phonetische Schrift mitbeachten möchte.

### ListComparator

Der ListComparator wird mit dem Argument –lc angesteuert und vergleicht Wortlisten miteinander oder fusioniert diese zusammen zu einer Liste. Dabei werden beim letzteren wieder die Input und Output Parameter (-i,-o) benötigt.

### MinimalPairs

MinimalPairs wird mit dem Argument –mp angesteuert und gruppiert die Wörter nach deren Unterschied. Diese werden dann auf eine Zeile geschrieben. Für jede Gruppe von minimal Pairs wird in dem Sinn eine Zeile vorhergesehen.

### Groupier

Der Groupier wird mit dem Argument –gp angesteuert und gruppiert die Wörter nach deren Wortstamm. Das heisst: laufen, gelaufen, laufend werden zusammen zu einer Gruppe genommen. Wie bei den Minimalpairs wird auch hier die für jede Gruppe eine Zeile genommen.

### Crawler

Der Crawler wird mit dem Argument –crwl angesteuert und verfügt über folgende Möglichkeiten:

* Wikipedia

Im Wikipedia Modus kann man eine Stammpage angeben von welcher aus sich der Crawler weiter navigiert und Unterwegs von jeder Page Worte sammelt. Falls er dabei in eine Sackgasse kommt aus welcher nichts weiter führt steht ihm ein Sortiment von Begriffen bereit aus denen er sich weiternavigieren kann.

* Book

Im Book Modus kann man ein Buch angeben welches er dann nach Wörtern durchsucht.

* Phonetic

Im Phonetic Modus muss man schon eine bestehende Wortliste angeben die er dann durch die Phonetische Schrift erweitert. Wordmonkey verfügt über keine eigene Funktion für die Generierung phonetischer Schrift. Es wird lediglich ein Bot gestartet welcher die entsprechende Aussprache von einem Service holt.

* Spell

Spell ist der Einzige Modus in dem es nicht um Text geht, sondern um Synthesizer Aussprache. Diese wird wie beim phonetic Modus durch einen Bot von einem Service geholt.

## Funktionsweise

Wordmonkey ist benutzt für die Interaktion mit dem User die Consolen Schnittstelle. Die Argumente werden durch den PrefixManager bearbeitet. Dieser verfügt über die Methoden getDataFromPrefix () und doesPrefixExist (). Durch die Zuweisung dieser Aufgabe einem Modul war eine grössere Flexibilität der möglich und so ist es nicht von belangen an welcher Stelle das Argument steht. Somit gibt es keine vorgeschriebene Reihenfolge.

## Fehlerbehandlung

Da Wordmonkey normaler weise mit grossen Datenmengen zu tun hat, lag unsere Intention vor allem auch an der Richtigkeit der Daten, die man eben nur sehr schwer im Nachhinein alle überprüfen kann. Somit wurden viele Fehlermeldungen vor allem im Importer eingebaut.

## Logging

Der Logger knüpft an das Thema der Fehlerbehandlung an. Geloggt wird jeder Fall der nicht ideal ist. Falls ein Wort zu kurz ist, also in dem Fall einfach 1 char. Oder Zeichen die nicht im Deutschen Verwendet werden beinhaltet wird dies geloggt. Dies befindet sich im File ~log.log

## Struktur

Wordmonkey hat die Typische Struktur eines Python Packages.

\_\_init\_\_.py: wird für den Interpreter gebraucht, damit er weiss das sich dort Python Scripts befinden.

* Crawler

Crawler umfasst die Scripts welche zum Crawlen und erweitern von Wortlisten gedacht sind

* + \_\_init\_\_.py
  + WordSpyder.py
* IO

IO umfasst alle Scripts welche mit Input und Output hantieren wie auch mit dem analysieren von Files oder dem herausfiltern der richtigen Wörter.

* + \_\_init\_\_.py
  + Exporter.py
  + Import.py
  + Prefile\_analyzer.py
  + Word\_catcher.py
* Grouppier

Grouppier umfasst Scripts mit der Stammwort Gruppierung.

* + \_\_init\_\_.py
  + Grouppier.py
* Listcomparator

Listcomparator umfasst die Scripts welche zum Vergleichen und fusionieren mehrerer Wortlisten gebraucht werden.

* + \_\_init\_\_.py
  + Comparator.py
  + Fusion.py
* Mpair

Mpair umfasst Script welche das Gruppieren nach Differenz ermöglichen.

* + \_\_init\_\_.py
  + Mpair\_extractor.py
* Utils

Utils umfasst alle Hilfs Module wie etwa den Logger oder die Progress bar.

* + \_\_init\_\_.py
  + Logger.py
  + prefixManager.py
  + progessbar.py
  + ressource\_manager.py
* /wordmonkey.py

# User Manual

### Basics

Um Wordmonkey ansteuern zu können muss man ihn als Parameter für Python übergeben.

* py wordmonkey.py -[Argument] …

Standard Funktionen:

* Hilfe: wordmonkey.py -h

Wordmonkey verfügt über eine Integrierte Hilfsfunktion. Welche auch im Readme ist.

* Version: wordmonkey.py -v
* Besserer Output --prntOut

### WordExtractor

* py wordmonkey.py –wx –i input.txt –o output.txt

[optional] -start : start index oder wort.

[optional] -end : end index oder wort.

[optional] -ext : Falls ein dir als Input, nur files mit dieser Extension.

Input Möglichkeiten: Files, Directory

### ListComparator

* py wordmonkey.py –lc [-i dir –o out.txt] oder [-i root,-c toCompare, -diff]

[optional] –diff : für differenz, in dem Fall muss auch –c angegeben sein. Sonst wird fusioniert.

### MinimalPairs

* py wordmonkey.py –mp –i input –o output

[optional] –fi : steht für fokus index, falls mehrere Wörter auf einer Zeile stehen zum Beispiel mit Phonetischer schrift noch, wird angegeben welcher index für die bildung der Gruppen berücksichtigt werden muss.

### Groupier

* py wordmonkey.py –gp –i input –o output

### Crawler

* py wordmonkey.py –crwl –[Modus] –[args]

1. –w : webpage crawl braucht –url Argument
2. –b : book crawl ,nimmt keine weiteren Argumente
3. –phon : Erweitert Wörter mit Phonetischer Schrift braucht –i Input und –o Output
4. –spell : braucht –i Input
5. –wkipedia : crawlt wikipedia, optional kann man –start für die rootpage angeben.

# Programmer Manual

### Technik

Mischung aus Prozedural und Objektorientiert. Lässt sich unter dem Begriff «python style» zusammenfassen.

### Testfälle & Testprotokoll

Befinden sich im File Testprotokoll.excl und Testprotokoll.docx

### Flaschenhälse

### Module

Die Module sind mit Docstrings selber schon dokumentiert. Mit der Python built in methode help(Module) kann man dies auch in der Console sichtbar machen.

### Repository

Befindet sich unter : <https://github.com/marcin96/wordmonkey.git>

Dies ist eine Praktikumsarbeit. Aus diesem Grund ist auch ein Lernjournal im Hauptverzeichnis auffindbar mit dem Namen lernjournal.docx