1. Exercise

TECHNISCHE UNIVERSITÄT DARMSTADT

The first exercise

Knowlege Engineering
Fachbereich Informatik
Technische Universität Darmstadt

Exercise Presentation:
Frank Englert
Jens Haase

2. Exercise

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Language Detection via character distribution
   - How it works
   - Results of the language detection challenge
2. Web crawler
   - Introduction
   - New URLs found
   - URLs per Page Statistics
   - Frequency of links
   - Further results
   - Experiences

Task 1 - Language detection

Language Detection via letter distribution

TECHNISCHE UNIVERSITÄT DARMSTADT

- The Firefox Plugin uses two detection modes
  - Via letter frequency analysis
  - Via syllable frequency analysis
- The language detection algorithm is the same for both cases
- Advantages of using two detection modes:
  - Double check the language detection results
  - Collect information which mode works better
- The Source of the frequency tables is http://bit.ly/jZHf0H

Task 1 - Language detection

Algorithm details

TECHNISCHE UNIVERSITÄT DARMSTADT

- **The algorithms works with the following steps**
- A chunk is either a letter or a syllable
- dict contains the most important chunks of a language sorted by rank
  1. Take the text an split it to chunks(letters or syllables)
  2. Remove all chunks which are not in the dict
  3. Count the chunks and sort them by the count value. The result of this step is further called rankedChunks
  4. The weighted difference between the dictionary and the rankedChunks is
     - $\sum_{i=0}^{len(dic)} \frac{|i - rankedChunks.indexOf(dict[i])|}{log_2(i+2)}$
     - If dict and rankedChunks are equals the weighted difference is 0
- repeat the steps 1-4 for all available languages. Take the language with the lowest rank.

Task 1 - Language detection

Letter frequency revisited

TECHNISCHE UNIVERSITÄT DARMSTADT

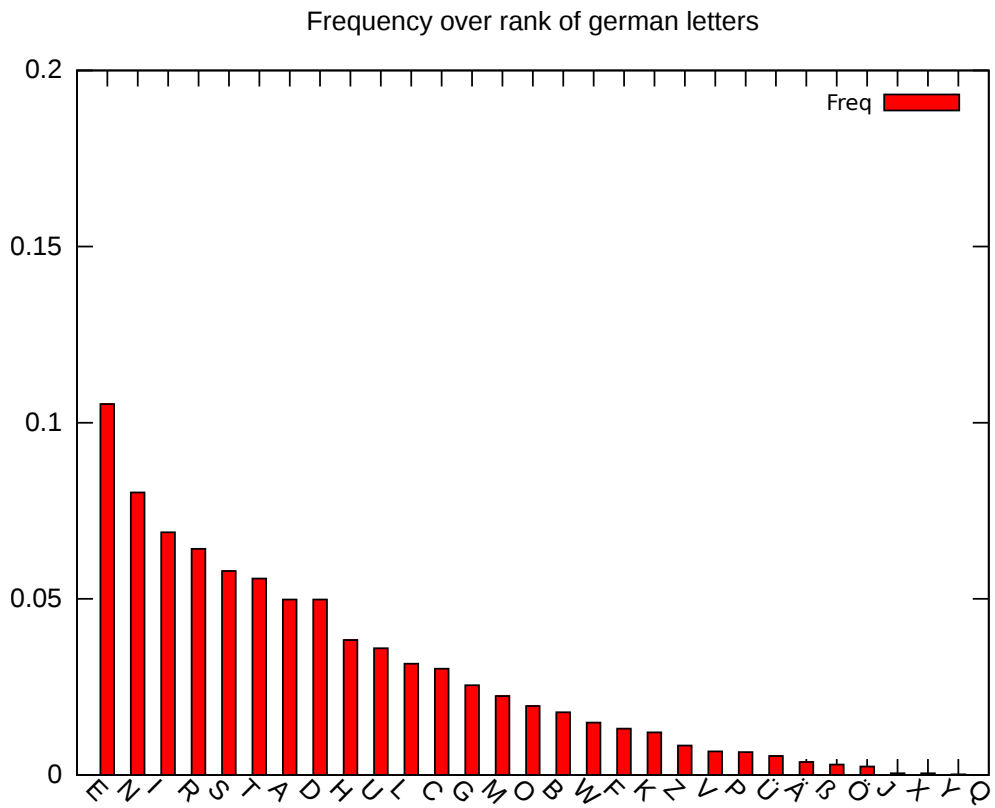


Abbildung: The frequency of german letters used for the Firefox plugin

Task 1 - Language detection

Syllable frequency revisited

TECHNISCHE UNIVERSITÄT DARMSTADT

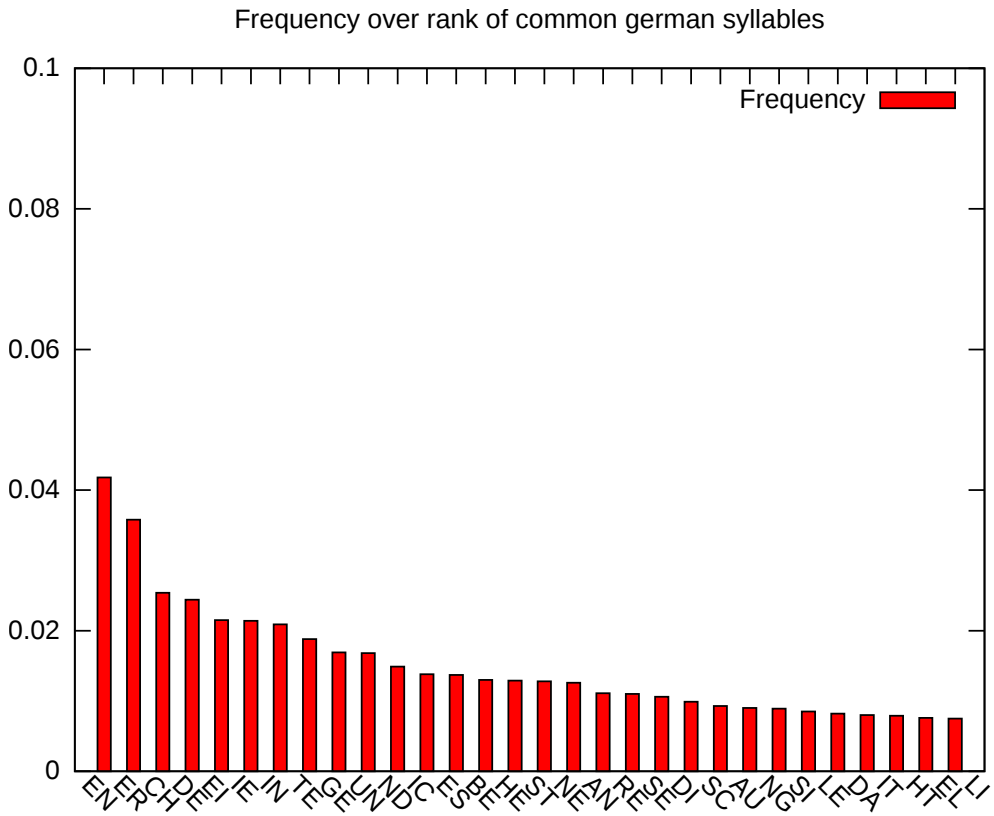


Abbildung: The frequency of common german syllables used for the Firefox plugin

Task 1 - Language detection

Results of the language challenge

TECHNISCHE UNIVERSITÄT DARMSTADT

Rank	letter lang	syllable lang
1	englisch	-
2	englisch	-
3	deutsch	-
4	französisch	-
5	deutsch	-
6	deutsch	deutsch
7	französisch	französisch
8	französisch	französisch
9	englisch	englisch
10	deutsch	deutsch

Tabelle: Detection results of the firefox plugin

Task 1 - Language detection

Further improvement

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Easy:**
  - Add more languages
- **A lot of work:**
  - The Plugin checks already p, div and span tags. It would be better to check the text content of all tags.
  - Try to estimate the best detection result if the syllable and the letter mode returns different results
- **Most Interesting:**
  - Improve the weighting algorithm to reduce the amount of needed text
  - Implement a learning mode to train new languages

Task 2 - Crawler

Introduction

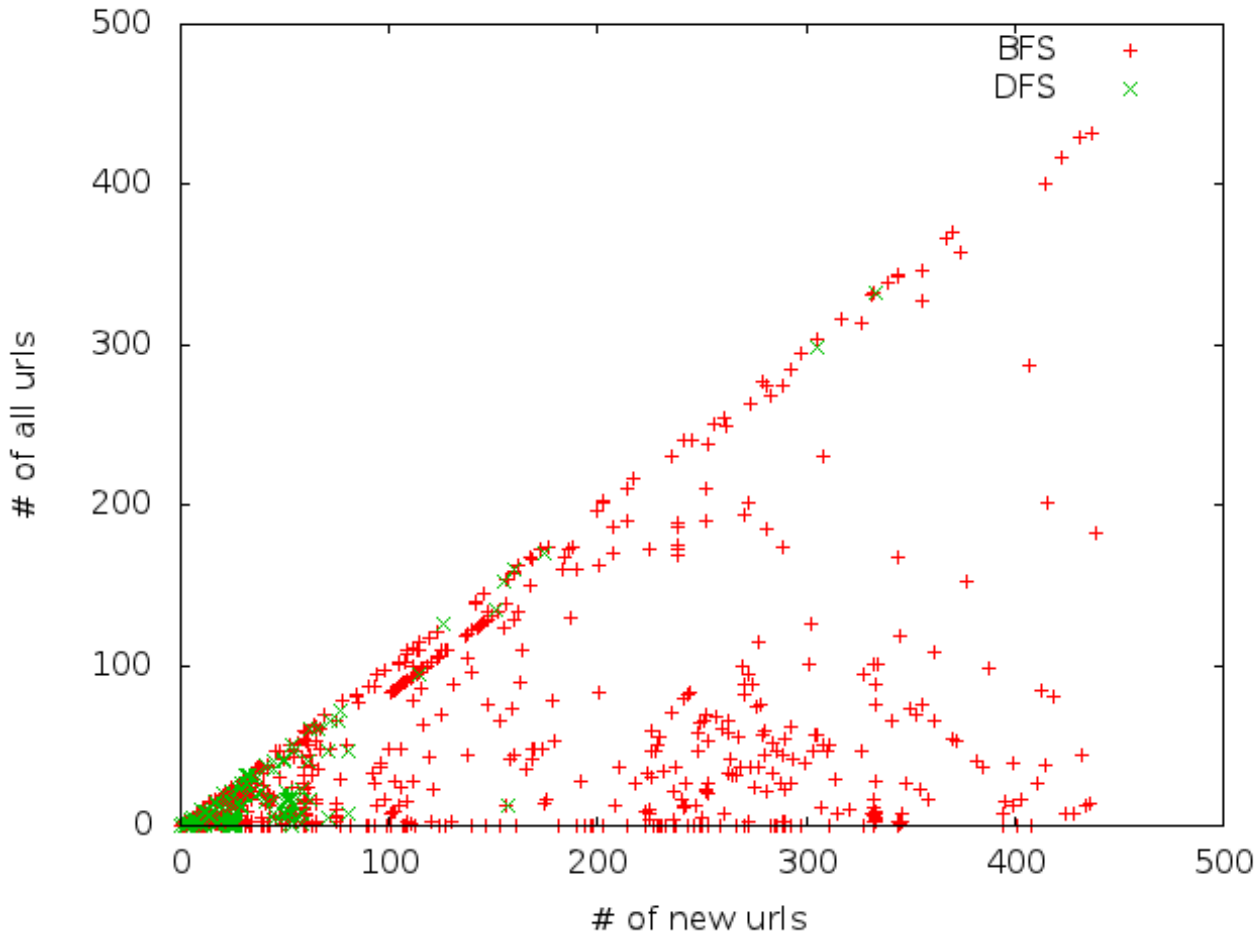
TECHNISCHE UNIVERSITÄT DARMSTADT

- **Implemented in Scala**
- **Currently, runs in a single thread.**
  - Therefore we need not to worry about to many access on the same host
  - But it can be easily moved to multithread with the Akka middleware
- **Started crawling at** `http://news.google.de`
- **Indexed 1000 pages with BFS queue and DFS queue**

Task 2 - Crawler

New URLs found

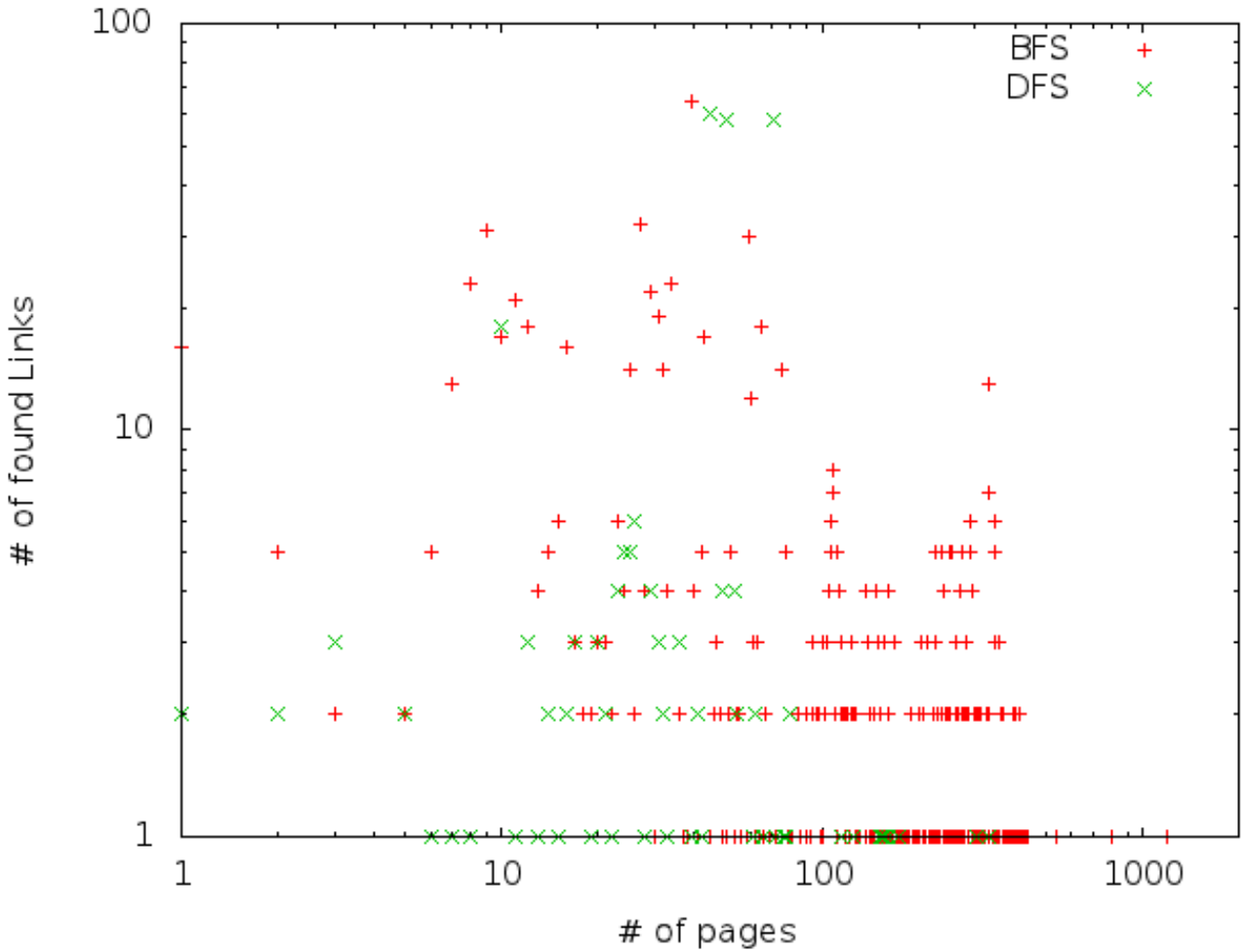
TECHNISCHE UNIVERSITÄT DARMSTADT



Task 2 - Crawler

URLs per Page Statistics

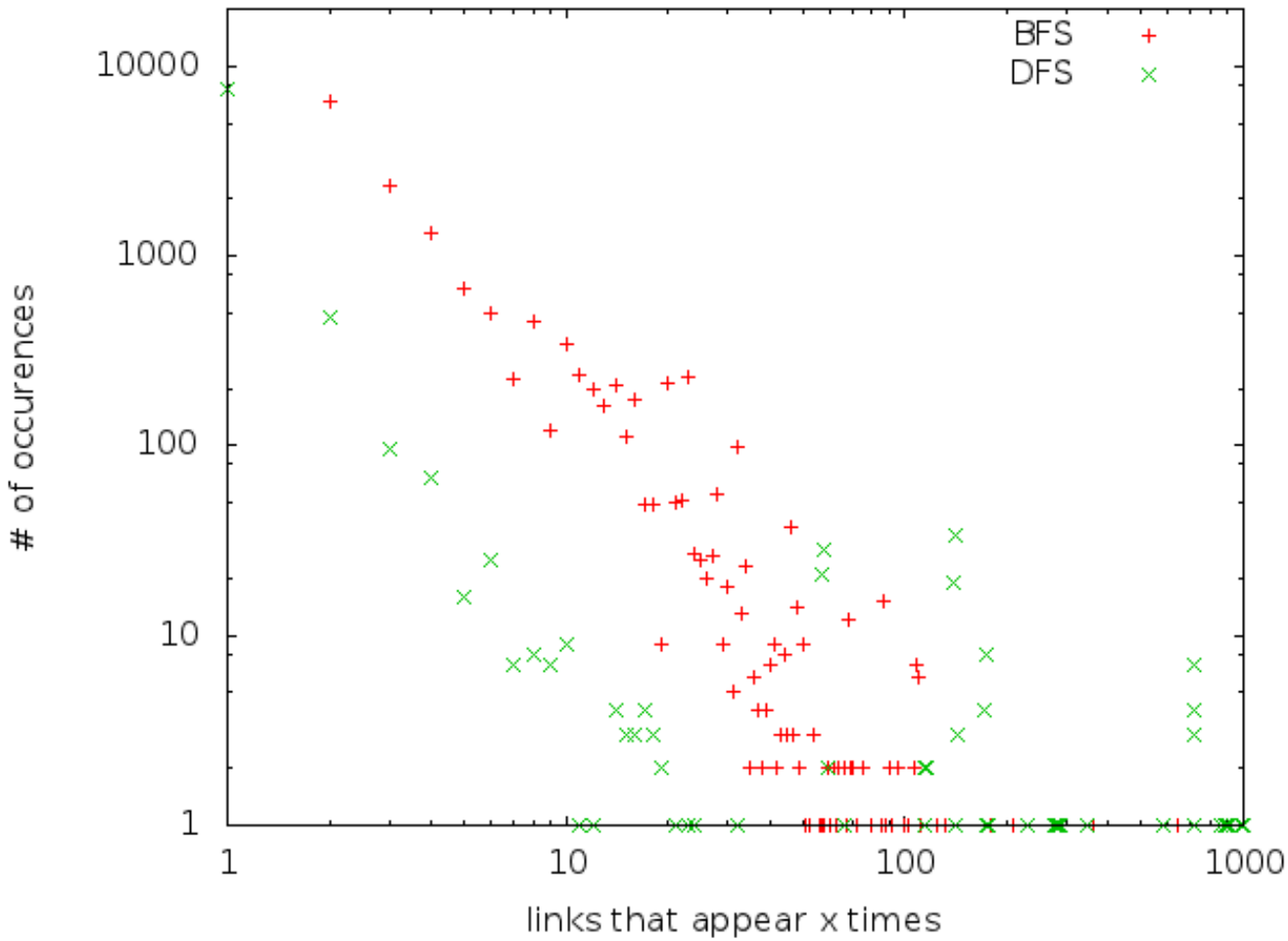
TECHNISCHE UNIVERSITÄT DARMSTADT



Task 2 - Crawler

Frequency of links

TECHNISCHE UNIVERSITÄT DARMSTADT



Task 2 - Crawler

Further results

TECHNISCHE UNIVERSITÄT DARMSTADT

BFS	DFS
136	20

Tabelle: Different Hosts found

Language	DFS	BFS
german	55	*935*
english	*943*	43
french	0	1
portuguese	0	1
<unknown>	2	20
$\sum$	1000	1000

Tabelle: Found languages in all pages

Task 2 - Crawler

Experiences

TECHNISCHE UNIVERSITÄT DARMSTADT

- **With Javas `URL` class a URL can easily brought to the same form**
- **But it has problems with `javascript:` and other *"protocols"***
- **Solution: simple wrap with a try catch block**
- **For crawling exception handling is a must! Else the crawler will stop in near time**