Malte Dostal

22. April 2021

The average German word

0. Abstract

Through the use of a simple algorithm written in Java and a vast word list (dictionary), the average word of the German language was calculated to be

Sereeeeeeeeeee

The phrase „average German word“ might be counter intuitive to understand and is defined as follows.

1. Definition of „average word“

The average word, as defined by this work, is a word with $\overline{n}$ number of characters, where

$$\overline{n} = \frac{\sum_{n \in W} N(n)}{|W|}$$

; W is a set containing all the words form the dictionary[1], $N(n)$ is a function that yields the number of characters the given word has

In this average word, each character c_x is the character that most commonly occurs at its index throughout the dictionary. The argument could be made that this is not how an average works, but as characters are not a spectrum, an average of e.g. the ascii values just wouldn't make sense semantically speaking.

2. The algorithm

The average word of the German language was calculated using a simple algorithm that works with a dictionary of over one million German words pulled from a public GitHub Gist[2].

Basically, the algorithm iterates over each word in the dictionary and adds the character data to a map that maps an `Integer` to a map of `Character` to `Integer`. The key of the enclosing map represents the character index, the inner one maps each character to its number of occurrences. Additionally, a `BigInteger` is used to count all characters in order to calcu-

[1] I know that this sounds very mathematically incorrect but 1. I don't care and 2. If you insist, then I have simply defined a „word“ to be a sequence of concatenated „characters“ where a „character“ is the ascii code point of the represented symbol. This way, my math would be correct, checkmate.

[2] source link: https://gist.github.com/MarvinJWendt/2f4f4154b8ae218600eb091a5706b5f4

late the average number of characters per word. After this initial indexing of the dictionary, the algorithm iterates over all entries in the enclosing map and in an inner loop, iterates over all characters and their number of occurrence to figure out the one with the most occurrences for each index, which is then appended to a `StringBuilder`. In the end, said `StringBuilder` is trimmed to the average number of characters and serves as the output of the algorithm.

3. The findings

Running the algorithm with before mentioned dictionary (which takes a few seconds on my late 2017 iMac running an i7 with 4 physical cores), the average number of characters in a German word is found to be

$$\overline{n} = 14$$

According to Duden online, this number is actually correct, as they state[3]

> Nimmt man als Berechnungsgrundlage hingegen nur die gut 18 Millionen Grundformen im Dudenkorpus [sic], kommt man auf ein mehr als doppelt so hohes Ergebnis, nämlich 14,43 Buchstaben.

As mentioned in the abstract, the average word turns out to be „ Sereeeeeeeeeee“, but ignoring the average length, it would be „Sereeeeeeeeeeeeeeeeeeeennnnnnnnnennnneneeen-ten“. Running the same algorithm on an English dictionary with just over 370,000 words, the average word is „sereeeeee“ with length 9 and ignoring that, it would be „sereeeeeeeeeessssssssssicallinene“.

[3] https://www.duden.de/sprachwissen/sprachratgeber/Durchschnittliche-Lange-eines-deutschen-Wortes