Zipf's and Heap's law.

Zipf's law.

Zipf's law is a law about the frequency distribution of words in a language (or in a collection that is large enough so that it is representative of the language). To illustrate Zipf's law let us suppose we have a collection and let there be $V$ unique words in the collection (the vocabulary).

For each word in the collection we need to compute the $\text{freq}(\text{word}) =$ how many times word occurs in the collection. Then we rank the words in descending by their frequency (most frequent word has rank 1, next frequent word has rank 2, ...)

The slides provide an example, which we reproduce here:

<table>
<thead>
<tr>
<th>Word</th>
<th>Freq</th>
<th>r</th>
<th>Pr(%)</th>
<th>$r^*Pr$</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>2,420,778</td>
<td>1</td>
<td>6.488</td>
<td>0.0649</td>
</tr>
<tr>
<td>of</td>
<td>1,045,733</td>
<td>2</td>
<td>2.903</td>
<td>0.0561</td>
</tr>
<tr>
<td>to</td>
<td>968,892</td>
<td>3</td>
<td>2.597</td>
<td>0.0779</td>
</tr>
<tr>
<td>a</td>
<td>892,429</td>
<td>4</td>
<td>2.392</td>
<td>0.0957</td>
</tr>
<tr>
<td>and</td>
<td>865,644</td>
<td>5</td>
<td>2.32</td>
<td>0.116</td>
</tr>
<tr>
<td>in</td>
<td>847,825</td>
<td>6</td>
<td>2.272</td>
<td>0.1363</td>
</tr>
<tr>
<td>said</td>
<td>504,593</td>
<td>7</td>
<td>1.352</td>
<td>0.0947</td>
</tr>
<tr>
<td>for</td>
<td>363,865</td>
<td>8</td>
<td>0.975</td>
<td>0.078</td>
</tr>
<tr>
<td>that</td>
<td>347,072</td>
<td>9</td>
<td>0.93</td>
<td>0.0837</td>
</tr>
<tr>
<td>was</td>
<td>293,027</td>
<td>10</td>
<td>0.785</td>
<td>0.0785</td>
</tr>
<tr>
<td>on</td>
<td>291,947</td>
<td>11</td>
<td>0.783</td>
<td>0.0861</td>
</tr>
<tr>
<td>he</td>
<td>250,919</td>
<td>12</td>
<td>0.673</td>
<td>0.0807</td>
</tr>
<tr>
<td>is</td>
<td>245,843</td>
<td>13</td>
<td>0.659</td>
<td>0.0857</td>
</tr>
<tr>
<td>with</td>
<td>223,846</td>
<td>14</td>
<td>0.6</td>
<td>0.084</td>
</tr>
<tr>
<td>at</td>
<td>210,064</td>
<td>15</td>
<td>0.563</td>
<td>0.0845</td>
</tr>
<tr>
<td>by</td>
<td>209,586</td>
<td>16</td>
<td>0.562</td>
<td>0.0899</td>
</tr>
<tr>
<td>it</td>
<td>195,621</td>
<td>17</td>
<td>0.524</td>
<td>0.0891</td>
</tr>
<tr>
<td>from</td>
<td>189,451</td>
<td>18</td>
<td>0.508</td>
<td>0.0914</td>
</tr>
<tr>
<td>as</td>
<td>181,714</td>
<td>19</td>
<td>0.487</td>
<td>0.0925</td>
</tr>
<tr>
<td>be</td>
<td>157,300</td>
<td>20</td>
<td>0.422</td>
<td>0.0843</td>
</tr>
<tr>
<td>were</td>
<td>153,913</td>
<td>21</td>
<td>0.413</td>
<td>0.0866</td>
</tr>
<tr>
<td>an</td>
<td>152,576</td>
<td>22</td>
<td>0.409</td>
<td>0.09</td>
</tr>
<tr>
<td>have</td>
<td>149,749</td>
<td>23</td>
<td>0.401</td>
<td>0.0923</td>
</tr>
<tr>
<td>his</td>
<td>142,285</td>
<td>24</td>
<td>0.381</td>
<td>0.0915</td>
</tr>
<tr>
<td>but</td>
<td>140,880</td>
<td>25</td>
<td>0.378</td>
<td>0.0944</td>
</tr>
</tbody>
</table>

Top 50 words from 84,678 Associated Press 1989 articles
Let \( r \) be the rank of word, \( \text{Prob}(r) \) be the probability of a word at rank \( r \). We do not care about the names of the words, we care only about their ranks and frequencies. By definition \( \text{Prob}(r) = \frac{\text{freq}(r)}{N} \) where \\
 \( \text{freq}(r) = \text{the number of times the word at rank } r \text{ appears in the collection} \) and \\
 \( N = \text{total number of words in the collection} \) (not number of unique words).

Then Zipf's law states that

\[
r \times \text{Prob}(r) = A,
\]

where \( A \) is a constant which should empirically be determined from the data. In most cases \( A = 0.1 \). Zipf's law is not an exact law, but a statistical law and therefore does not hold exactly but only on average (for most words).

Taking into account that \( \text{Prob}(r) = \frac{\text{freq}(r)}{N} \) we can rewrite Zipf's law as

\[
r \times \text{freq}(r) = A \times N
\]

To establish that Zipf's law holds we need to compute \( \text{freq}(r) \), which involves computing the frequency of each word and then ranking the words. Then we need to compute \( r \times \text{freq}(r) \) and see if \( r \times \text{freq}(r) \) is approximately a constant. This does not mean that for all words \( r \times \text{freq}(r) \) has to be exactly the same, but it has to be close to the same number for most words. The simplest way to show that Zipf's law holds is to plot the data. Remember that looking at most frequent and least frequent words only is misleading. For those types of words Zipf's law has the highest errors.

Instead of plotting \( r \) vs. \( \text{freq}(r) \), it is better to plot \( \log(r) \) on the x-axis and \( \log(\text{freq}(r)) \) on the y axis. If Zipf's law holds we should see a line with slope \(-1\) (this means if \( A \) is the point where the line crosses the x-axis and \( B \) is the point where the line crosses the y-axis and \( O \) is the origin of the coordinate system then \( OA = OB \)).

Another, equivalent way is to plot \( \log(r) \) on the x-axis and \( \log(\text{Prob}(r)) \) on the y axis.
Notice the slope of the line is -1.

Zipf's law make most errors for highest frequency and lowest frequency words. Need to plot all the data to see if Zipf's law applies.

Zipf curve for the unigrams extracted from a 250,000 word tokens corpus

Source: Extension of Zipf's law to words and Phrases by Ha, Garcia, Smith
We can use Zipf's law to calculate the number of words that appear \( n \) times in the collection.

Let \( \text{MaxRank}(n) \) = among all words that appear \( n \) times let \( \text{MaxRank}(n) \) be the maximum of the ranks of those words. For example, if \( n = 90 \), the words that appear \( n = 90 \) times are “and”, “in”, “said” with ranks 5, 6, 7. Then \( \text{MaxRank}(90) = \text{max}(5, 6, 7) = 7 \)

Another example: \( \text{MaxRank}(79) = \text{max}(18, 19, 20) = 20 \)

Notice that the number of words that appear \( n \) times is
\[
\text{NumberWordsOccur}(n) = \text{MaxRank}(n) - \text{MaxRank}(n + 1).
\]

For example, the number of words that appear 79 times is \( \text{MaxRank}(79) - \text{MaxRank}(80) = \text{max}(18, 19, 20) - \text{max}(17) = 20 - 17 = 3 \)
We can look at the picture on the right side to see that exactly 3 words appear 79 times.

We know from Zipf's law that for the frequency and the rank are related.

If \( r = \text{MaxRank}(n) \), this means that the rank is \( r \) and the frequency is \( n \); So \( r \cdot \text{freq}(r) = A \cdot N \) means
\[
\text{MaxRank}(n) \cdot n = A\cdot N,
\]
which implies
\[
\text{MaxRank}(n) = A \cdot N / n.
\]
Applying this formula twice we obtain
\[
\text{NumberWordsOccur}(n) = \frac{A\cdot N}{n} - \frac{A\cdot N}{(n + 1)} = A \cdot N \left( \frac{1}{n} - \frac{1}{n+1} \right) = A \cdot N \left( \frac{1}{n(n+1)} \right).
\]
So,
\[
\text{NumberWordsOccur}(n) = A \cdot N / [n \cdot (n + 1)]
\]
is the number of words that occur \( n \) times.

We need to connect \( A \cdot N \) to the number of unique words in the collection. This is easy because \( V \), which is the number of unique words is simply the rank of the last word in the ranked list of words. We need to assume (quite reasonably) that the least occurring word occurs
only once.
So, Zipf's law applied to the least frequent word
gives (here: \( r = V \), and freq(r) = 1)
\[ r \times \text{freq}(r) = A \times N, \quad V \times 1 = A \times N, \]
\[ A \times N = V \]
Therefore
\[ \text{NumberWordsOccur}(n) = \frac{V}{n \times (n + 1)} \]
is the number of words that occur \( n \) times.
where \( V \) is the number of unique words in the collection

Application of the formula \( \text{NumberWordsOccur}(n) = \frac{V}{n \times (n + 1)} \).
What fraction of all unique words appear only once?
We need “number of words that occur once” / “number of unique words” =
\[ = \frac{\text{NumberWordsOccur}(1)}{V} = \frac{V}{1 \times (1 + 1)} / V = 1 / (1 \times (1 + 1)) = \frac{1}{2} \]

Heap's law.

Heap's law states that the number of unique words \( V \) in a collection with \( N \) words is approximately \( \sqrt{N} \). The more general form of this law is

\[ V \approx K N^\beta (0 < \beta < 1) \]

Typically
\[- K \approx 10-100 \]
\[- \beta \approx 0.4-0.6 \text{ (approx. square-root of } n) \]

Alpha and beta and usually found by fitting the data.

Power laws
Zipf's and Heap's law belong to a class of laws called power laws.
A power law is one that has the form

\[ y = k x^c \]
k and \( c \) are constants that have to be fit from the data.
If we are to write Zipf's law as power low, we notice that
\[ y = \text{freq}(r), \ x = r, \ k = A \times N, \ c = -1 \]
If we are to write Heap's law as a power law we observe that
\[ y = V, \ x = N, \ k = K \text{ (from Heap's law), } c = \beta \]

Power laws have the useful property that if one takes the log of both sides of one obtains a line. See picture.

\[ \log(y) = \log(kx^c) \]

\[ \log(y) = \log(k) + c\log(x) \]

Let \( y' = \log(y) \)
\[ x' = \log(x) \]

\[ y' = m + c \cdot x', \text{ where } m = \log(k) \]

\[ y' = \log(y) \quad x' = \log(x) \]

\[ y' = \log(y) \quad x' = \log(x) \]

\[ \text{Use that } m = \log(k) \text{ to find that} \]
\[ k = \exp(m) \quad [\text{watch the base of your log here}] \]

Note: when you do the line fitting, do not use \( x \) and \( y \) but use \( x' = \log(x) \) and \( y' = \log(y) \) in the formulas for the least squares line fitting.
Fitting Zipf's law:

Plot of term frequency vs. rank.

Log-log plot of term frequency vs. rank.

Source: Modeling Web Data by French

If you fit data that obeys Zipf's law you should get c close to -1.
Fitting Heap's law:

Plot of unique vocabulary terms vs. total terms.

\( n: \text{number of total words encountered} \)
\( V = V(n): \text{number of unique words encountered, is a function of } n \)

(Source: Modeling Web Data by French)

If you fit data that obeys Heap's you should get \( c = \text{slope of the line close to 0.5} \)