DATA.STAT.840 Statistical Methods for Text Data Analysis Exercise set 2

Task 2.1

Improved code is on the next page. Improvements has been highlighted. I created a set named crawled_urls_set which contains all the URLs that has already been crawled through. At the end only URLs which are not in this set are added to pagestocrawl, which is the list we are looping in the code. On top of that the maximum number of links per page is limited with MAX_LINKS_PER_PAGE. Number of links to be added at the end of the loop is minimum value between pagetocrawl_urls and MAX_LINKS_PER_PAGE. Then random sample of size num_links_to_add is selected. Out of the random sample URLs which has not been crawled are added to pagestocrawl.

```python
def basicwebcrawler(seedpage_url,maxpages):
    # Store URLs crawled and their text content
    num_pages_crawled=0
    crawled_urls=[]
    crawled_texts=[]

    # urls which has already been visited
    crawled_urls_set = set()

    # Remaining pages to crawl: start from a seed page URL
    pagestocrawl=[seedpage_url]

    MAX_LINKS_PER_PAGE = 10

    # Process remaining pages until a desired number
    # of pages have been found or until there's no pages to crawl
    while num_pages_crawled<maxpages and len(pagestocrawl)>0:
        # Retrieve the topmost remaining page and parse it
        pagetocrawl_url=pagestocrawl[0]
        print('Getting page:')
        print(pagetocrawl_url)
        pagetocrawl_html=requests.get(pagetocrawl_url)
        pagetocrawl_parsed=bs4.BeautifulSoup(pagetocrawl_html.content,'html.parser')

        # Get the text and URLs of the page
        pagetocrawl_text=getpagetext(pagetocrawl_parsed)
        pagetocrawl_urls=getpageurls(pagetocrawl_parsed)

        # Store the URL and content of the processed page
        num_pages_crawled+=1
        crawled_urls.append(pagetocrawl_url)
        crawled_texts.append(pagetocrawl_text)
        crawled_urls_set.add(pagetocrawl_url)

        num_links_to_add = min(len(pagetocrawl_urls), MAX_LINKS_PER_PAGE)
        selected_links = random.sample(pagetocrawl_urls, num_links_to_add)

        # Remove the processed page from remaining pages,
        # but add the new URLs
        pagestocrawl.pop(0)
        for url in selected_links:
            if url not in crawled_urls_set:
                pagestocrawl.append(url)

    return(crawled_urls,crawled_texts)

crawled_urls, crawled_texts = basicwebcrawler('https://www.sis.uta.fi/~tojape/',10)
```

Task 2.2

Exercise code can be found in a file 'gutenberg_crawler.ipynb'. In a first cell I have imports and helper functions. Second cell contains fetching top k books and downloading them in a text format. Third cell contains processing (tokenizing and lemmatizing) the downloaded books.

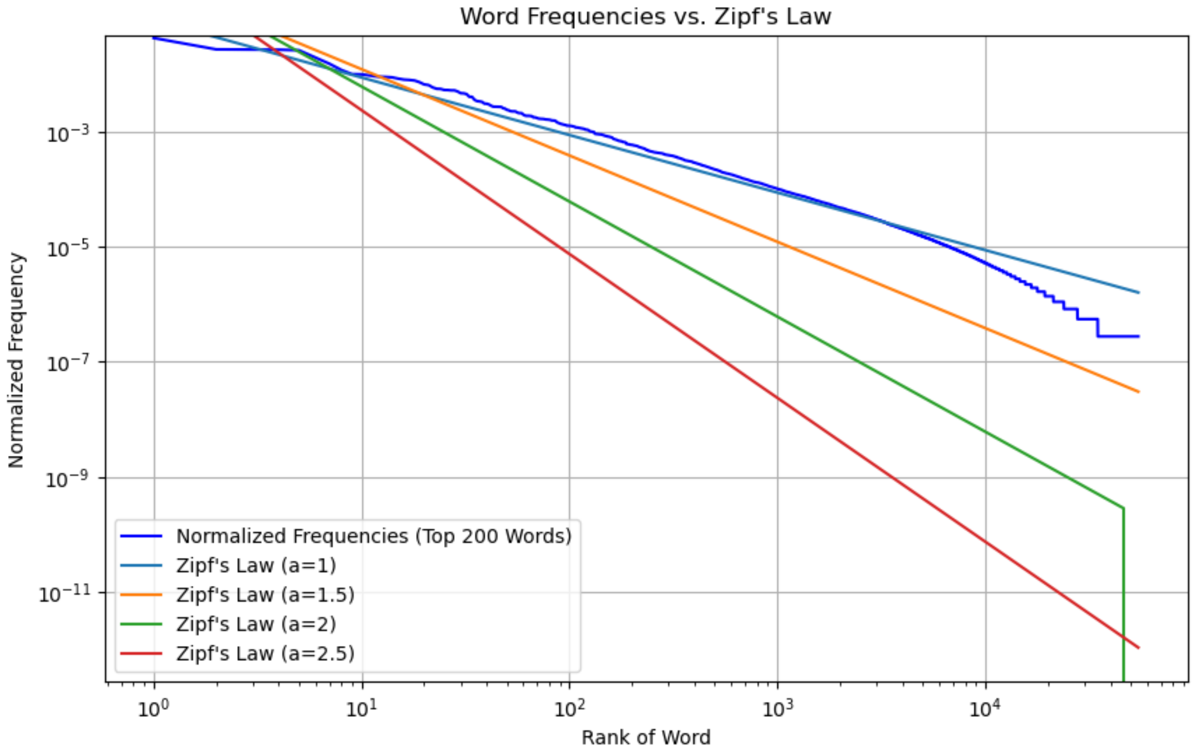
Here's truncated output of cell 3 when k=20:

```
Top 100 words in the downloaded books:
the: 153898
of: 97619
to: 96664
and: 96274
a: 94100
I: 70335
in: 54990
that: 42800
wa: 36177
his: 35919
he: 34156
with: 32679
you: 32406
it: 31904
my: 30270
her: 29121
not: 28712
for: 28057
had: 26241
is: 24019
be: 23400
me: 21275
s: 20157
have: 20037
...
never: 4613
see: 4613
well: 4609
u: 4576
```

Task 2.3

Code is in Zips_law.ipynb

I think a=1 is close to real frequencies.



Task 2.4

a) I asked: "Why is Zip's law important??", ChatGPT's reply is on the next page
b) manually rewritten (my own points highlighted):

Zip's law is important because:

- Zip's law represents how language is structured, showing that small number of words such as "the", "of", "to" and "and" are used most frequently while most words are used.
- It's useful in NLP, text analysis and language modelling, because Zip's this reduces computational complexity of these models which already require big amounts of computation
- when we know which words are frequent, we can do text compression more efficiently, like in Huffman coding, where assigning shorter codes to frequently used symbols and longer codes to rarer symbol reduces overall size of data while ensuring lossless compression
- in modelling social phenomena, Zip's law applicates to economic distributions, where a small percentage of the population controls large share of wealth, this helps in economic theory
- in urban planning and resource allocation Zip's law can be used to illustrate that a few cities have large populations while the majority are much smaller.
- Principle also applies to Network theory and internet, such as hyperlinks where a few web pages receive a disproportionate number of links, influencing how information is navigated online and user interactions on social media platforms, where small number of users such as celebrities have large following which results to small amount of content having more likes and comments
- it also aids in building models that can predict trends across various datasets, from consumer behaviour to web traffic
- it helps identify unusual patterns or outliers in data, for example corpus may have some typo and by examining which words only occur once we can find these typos.
- Zip's law provides an efficient way to approximate complex distributions, saving both time and computational resources, by focusing on the most frequent elements (words, cities, links) and approximating the rest, systems and models can function more efficiently without sacrificing too much of the accuracy

- it can help us in text compression, oim

  Zipf's Law is important for several reasons across various fields of study. Here are some key points that illustrate its significance:

  1. **Understanding Language and Linguistics**:
     - **Word Frequency Distribution**: Zipf's Law reveals how language is structured, showing that a small number of words are used frequently while most words are rarely employed. This has profound implications for natural language processing (NLP), text analysis, and language modeling.
     - **Language Efficiency**: The law underscores the efficiency of language, where commonly used words tend to be shorter and easier to pronounce, aiding in quicker communication.
  2. **Applications in Information Retrieval**:
     - **Search Engine Optimization**: Knowledge of word frequency can enhance search algorithms, leading to better indexing and improved search results.
     - **Text Compression**: Understanding which words are most frequent allows for more efficient text compression algorithms, optimizing data storage and transmission.
  3. **Modeling Social Phenomena**:
     - **Wealth Distribution**: Zipf's Law is applicable to economic distributions, where a small percentage of the population controls a large share of wealth. This has implications for economic theory and social policy.
     - **City Size Distribution**: The law is observed in urban studies, illustrating that a few cities have large populations while the majority are much smaller, affecting resource allocation and urban planning.
  4. **Network Theory and Internet**:
     - **Web Page Link Structures**: The principle applies to hyperlinks, where a few web pages receive a disproportionate number of links, influencing how information is navigated online.
     - **Social Media Dynamics**: Zipf's Law explains user interactions on social platforms, where a small number of users command a large following, impacting how content spreads and influences behavior.
  5. **Implications for Data Science**:
     - **Predictive Modeling**: It aids in building models that predict trends across various datasets, from consumer behavior to web traffic.
     - **Anomaly Detection**: Zipf's Law helps identify unusual patterns or outliers in data, signaling potential issues or areas for further exploration.
  6. **Cognitive Science**:
     - **Information Processing**: The law provides insights into human cognitive processes, influencing theories about how we process language and information efficiently.

  Conclusion

  Zipf's Law serves as a foundational principle in understanding complex systems across disciplines, offering insights into patterns of usage, distribution, and behavior. Its applications can lead to enhanced algorithms, a deeper understanding of social structures, and advancements in technology related to language and data processing. Understanding Zipf's Law allows researchers and practitioners to make informed decisions based on empirical data, driving innovation and efficiency.