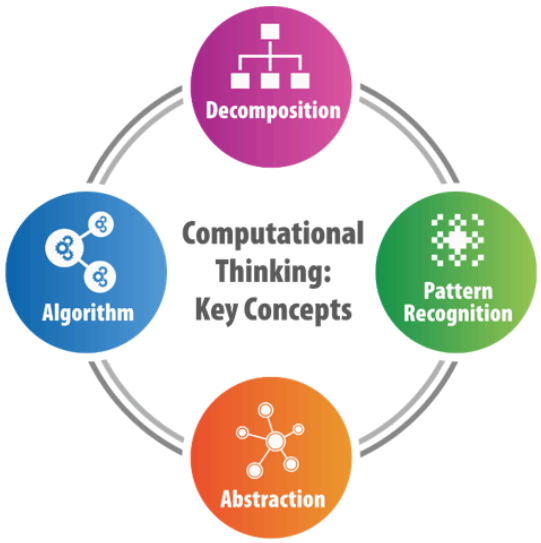
COMPUTATIONAL THINKING Part 1
UNIVERSITY OF PENNSYLVANIA | COURSERA

By Rayane Rocha



Project outline

In this project, you will use computational thinking to develop and then implement an algorithm to solve the problem of **counting the number of occurrences of a word and its synonyms in a corpus of text documents.** This project consists of four parts; you will complete one part at the end of each course module:

1. **Apply the four pillars of CT and describe the results of each**
   Completed · Part 1 - Pillars of CT
2. Express the algorithm used in the solution using a flowchart
3. Express the algorithm using a structured notation known as pseudocode
4. Implement the solution in Python

1. Four pillars of computational thinking

In this part of the project, you will apply the four pillars of CT to this problem by answering the following questions:

1. Using decomposition, what are the primary sub-problems that need to be solved in solving the overall problem?
2. Using pattern recognition, what patterns do you see in the solution, i.e., what processes need to be repeated?
3. Using data abstraction and representation, how would you represent the thesaurus, the corpus, and each of the documents in the corpus?
4. Using the results of the first three pillars, what is the algorithm that you would use to solve this problem? Describe it in as much detail as possible.

2. Additional case – Problem

Describe a problem that you may face — either in your career or in everyday life — that involves determining the number of occurrences of a word and its synonyms in a corpus of documents. The problem you face may be much bigger than that and require that calculation as only a small part of the solution, but should involve looking through some collection of text and looking for certain words.

---

1. Decomposition

1.1. Organization

Breaking down the entire body of each text into units, words or phrases.

1.2. Counting

Count the occurrences of each word in the preprocessed text. Repeat this process on each text within each document of the corpus.

1.3. Finding target words

Identify a given target word's synonyms and equal occurrences.

1.4. Matching words

Integrate the counts of the target word and its synonyms to obtain a consolidated frequency.

1.5. Result visualization

Generate a visual output to illustrate the word and synonym occurrences.

2. Pattern recognition

2.1. Repetition

The solution to this problem will be inherently iterative, since pattern recognition will identify and assemble recurring processes across the entire dataset (in this case, the texts). The overarching goal is to streamline and automate the analysis by recognizing patterns in the data.

In the solution, the process of synonym identification using a lexical database needs to be repeated constantly. The pattern involves querying the database to find equal or synonymous words for a target word. This process is repeated for multiple target words across various documents in the corpus. The identified equal or similar words are then expanded, counted, combined, analyzed, and represented in the output.

2.2. Precision of word frequency

Wait a minute... But what if two words are equal but have different meanings in the text? How can we handle them while counting?

Homographs, words with the same spelling but different meanings can introduce ambiguity that can impact the precision of word frequency analysis. What will happen if the program run through this passage:

> “Arriving in the village, the anthropologist heads to the chief's tent, a bit concerned about whether he would be welcomed. An archer stands sternly by the entrance, holding his arrow and **bow** firmly, and narrowing his eyes as Dr. Bienvenu goes through with an uneasy smile. The darkness inside the tent, in contrast to the bright day outside, obscures Dr. Bienvenu's eyes, who doesn't notice the chief's greeting **bow**.”

In the given passage:

> “An archer stands sternly by the entrance, holding his arrow and **bow** firmly…”
>
> “…doesn't notice the chief's greeting **bow**.”

The program needs to recognize that the two instances of "bow" have different meanings and should be counted separately.

So, when counting occurrences, it's essential to consider not only the spelling of the word but also the context in which it is used to ensure a more precise representation of word frequencies.

In conclusion, the success of the word occurrence counting process hinges on the program's ability to recognize patterns, adapt to contextual intricacies, and implement iterative strategies that account for the inherent complexities of language.

3. Data representation and abstraction

3.1. Thesaurus

- Abstraction: Focus on the essential information about word synonyms. Disregard unnecessary details and capture the relationships between words and their synonyms.
- Data Representation: Represent the thesaurus as a data structure that associates each word with its synonyms. This could be implemented using a dictionary or a similar key-value data structure, where each word is a key, and the corresponding value is a list of synonyms

3.2. Corpus

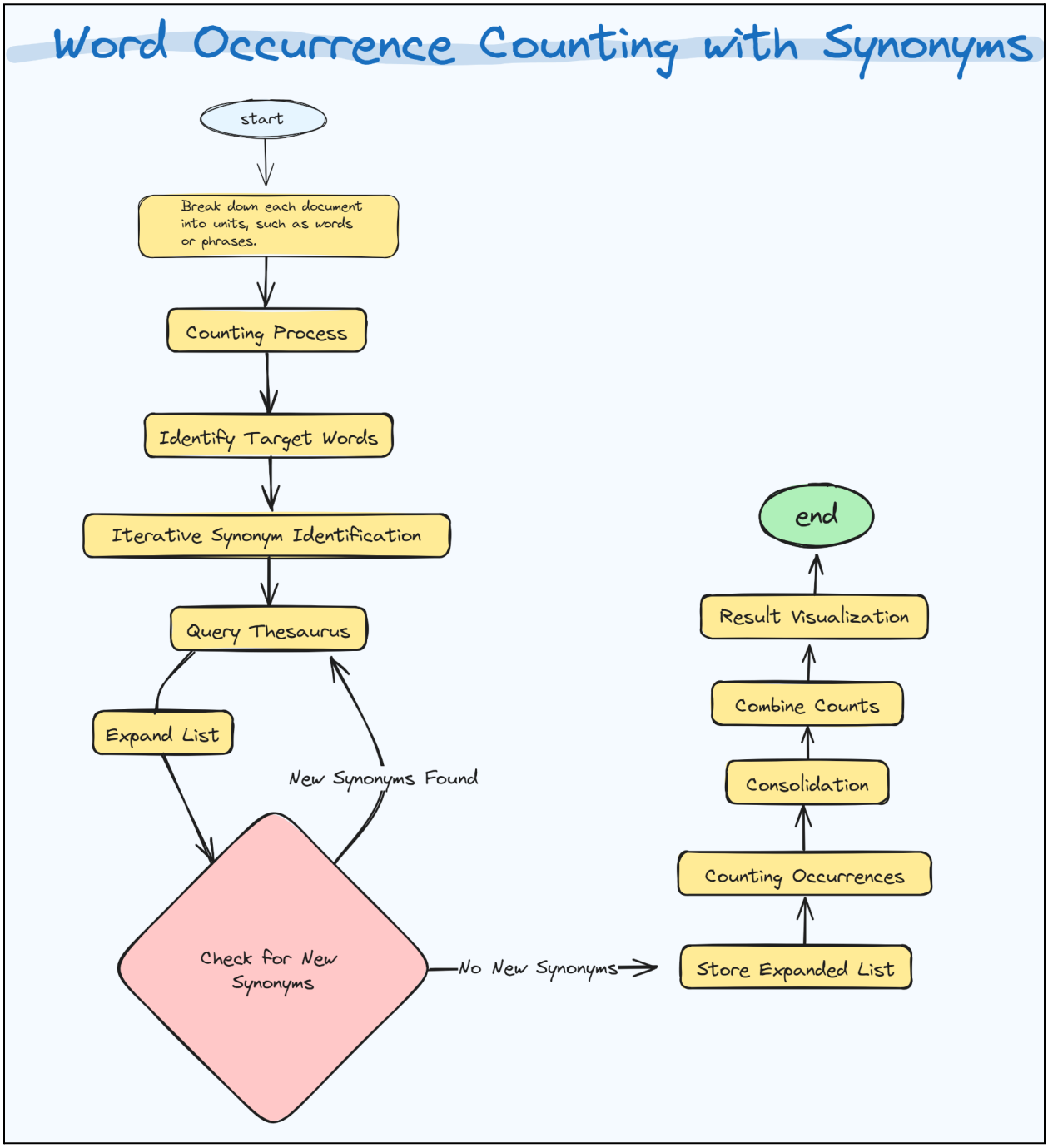
- Abstraction: Ignore document-specific details such as content, writing style, type of document, authors, appearance, overarching themes, abstract’s keywords, etc.
- Data Representation: Represent the corpus as a collection of documents. Each document is represented as a string or a list of words, and the entire corpus can be stored as a list or another appropriate data structure.

3.3. Document

- Abstraction: Disconsider elements unhelpful and unnecessary for counting occurrences of words and their synonyms. Leave out number of words paragraphs, and indentation,
- Data Representation: Each document will be represented as a list of words or tokens (phrases broken down into words).

4. Algorithm

Algorithm is, simply put, a step-by-step procedure or set of rules to solve a problem. Below is a flowchart of how the algorithm for this problem would run.



4.1. Decomposition and Counting

1) Break down each document into units, such as words or phrases.
2) Apply the counting process to each document, counting the occurrences of each word.
3) Identify target words and their synonyms using the thesaurus.

4.2. Iterative Synonym Identification:

1) Start an iterative process for each target word:
    a) Query the thesaurus to find equal or synonymous words for the target word.
    b) Expand the list of identified words by considering the synonyms of the synonyms.
    c) Repeat this process until no new synonyms are found.
    d) Store the expanded list for the target word.

4.3. Counting Occurrences

1) For each document in the corpus:
    a) Iterate through the words in the document.
    b) If a word matches any of the target words or their synonyms, increment the count for that target word.
    c) Handle homographs by considering the context in which the word is used to determine its meaning.

4.4. Consolidation

1) Combine the counts of the target word and its synonyms to obtain a consolidated frequency.
2) Ensure that each occurrence is counted only once, even if a word is a synonym for multiple target words.

4.5. Result Visualization

1) Generate a visual output illustrating the occurrences of target words and their synonyms.

☑ Problem

Let’s go with the French masculine word “bienvenu”, that means “welcome” in English that we used in the previous example.

In a linguistic research project, we could explore the usage and evolution of the word “beinvenu” in a corpus of documents spanning various time periods and contexts. “Beinvenu” is a term with deep linguistic roots, and we can analyze its occurrences and its synonyms across a large collection of texts.

To study this word, we would need to extract instances of "beinvenu" and its synonyms from a corpus, considering variations in spelling, synonyms, and contextual meanings. Then, after a data cleaning and preparation, once the instances are extracted, the frequency of occurrences for "beinvenu" and its synonyms is calculated within the corpus. This analysis helps in understanding the popularity and usage patterns of the term over time.