Chloe Tu
ITAI 2373 - Natural Language Processing
Professor Anna Devarakonda
November 10, 2025

L10 Manual Natural Language Processing (NLP) Pipeline from Corpus to Vectorization

The objective of this assignment is to manually implement a typical Natural Language Processing (NLP) pipeline. This process demonstrates how I take raw, unstructured text (my corpus) and systematically clean, process, and transform it. The final goal is to turn the text into a structured, numerical format that a computer can understand and analyze. This document outlines each step I have taken, from my corpus selection to the final vectorization. I have included my reflections on the importance of each stage.

1. Selection of Corpus

For this assignment, I selected the short Aesop's fable, **"The Tortoise and the Hare."**

I chose this text for a few specific reasons. First, it is very short, which makes the manual steps of this assignment manageable. Second, the language is simple and direct, which makes it a good example for text cleaning and tokenization. The story is well-known, and it's clear narrative helps in understanding the context of the words.

Corpus Details	
Corpus Source	Aesop's Fables
Title	"The Tortoise and the Hare"
Content Type	Short story / Fable
Reason for Choice	Manageable length and simple language are ideal for a manual project.
Application Context	This demonstrates a foundational pipeline on a classic narrative text.

Corpus: Original Text ("The Tortoise and the Hare")

- "A Hare was once boasting of his speed before the other animals. 'I have never yet been beaten,' he said, 'when I put forth my full speed. I challenge anyone here to race with me.'
- The Tortoise said quietly, 'I accept your challenge.'
- 'That is a good joke,' said the Hare; 'I could dance round you all the way.'
- 'Keep boasting till you've beaten,' answered the Tortoise. 'Shall we race?'
- So, a course was fixed, and a start was made. The Hare darted almost out of sight at once, but soon stopped and, to show his contempt for the Tortoise, lay down to have a nap. The Tortoise plodded on and plodded on, and when the Hare awoke from his nap, he saw the Tortoise just finishing the course, and soon after crossed the finish line."

2. Text Cleaning

The first step in the pipeline was to clean the raw text. This is done to prepare the text for tokenization by removing noise. First, I converted the entire text to a lowercase. This is a very important step. It ensures that a word like "Hare" at the beginning of a sentence is treated the same as "hare" in the middle of a sentence. Next, I removed all punctuation, numbers, and special characters. This included all periods, commas, quotation marks, and apostrophes. For this manual project, I treated the contraction "you've" as "you've" after removing the apostrophe.

Objective: To standardize the text and remove noise by converting it to lowercase and removing all punctuation and special characters.

This is the first processing step. I converted the entire text to lowercase to ensure that words like "Hare" and "hare" are treated as the same word. Then, I removed all punctuation like periods, commas, and apostrophes.

Text after Lowercasing:

"a hare was once boasting of his speed before the other animals. 'i have never yet been beaten,' he said, 'when I put forth my full speed. i challenge anyone here to race with me.' The tortoise said quietly, 'i accept your challenge.' 'That is a good joke,' said the hare; 'i could dance round you all the way.' 'Keep boasting till you've beaten,' answered the tortoise. 'Shall we race?' so a course was fixed, and a start was made. The hare darted almost out of sight at once, but soon stopped and, to show his contempt for the tortoise, lay down to have a nap. The tortoise plodded on and plodded on, and when the hare awoke from his nap, he saw the tortoise just finishing the course, and soon after crossed the finish line."

Final Cleaned Corpus (Lowercase and No Punctuation):

"a hare was once boasting of his speed before the other animals i have never yet been beaten said he when i put forth my full speed i challenge anyone here to race with me the tortoise said quietly i accept your challenge that is a good joke said the hare i could dance round you all the way keep your boasting till you've beaten answered the tortoise shall we race so a course was fixed and a start was made the hare darted almost out of sight at once but soon stopped and to show his contempt for the tortoise lay down to have a nap the tortoise plodded on and plodded on and when the hare awoke from his nap he saw the tortoise just finishing the course and soon after crossed the finish line"

3. **Tokenization**

After cleaning the text, I performed a tokenization. This is the process of breaking the long string of text into a list of individual words, or tokens. I did this by splitting the text at every space.

Tokens (Full List):

['a', 'hare', 'was', 'once', 'boasting', 'of', 'his', 'speed', 'before', 'the', 'other', 'animals', 'i', 'have', 'never', 'yet', 'been', 'beaten', 'said', 'he', 'when', 'i', 'put', 'forth', 'my', 'full', 'speed', 'i', 'challenge', 'anyone', 'here', 'to', 'race', 'with', 'me', 'the', 'tortoise', 'said', 'quietly', 'i', 'accept', 'your', 'challenge', 'that', 'is', 'a', 'good', 'joke', 'said', 'the', 'hare', 'i', 'could', 'dance', 'round', 'you', 'all', 'the', 'way', 'keep', 'your', 'boasting', 'till', 'youve', 'beaten', 'answered', 'the', 'tortoise', 'shall', 'we', 'race', 'so', 'a', 'course', 'was', 'fixed', 'and', 'a', 'start', 'was', 'made', 'the', 'hare', 'darted', 'almost', 'out', 'of', 'sight', 'at', 'once', 'but', 'soon', 'stopped', 'and', 'to', 'show', 'his', 'contempt', 'for', 'the', 'tortoise', 'lay', 'down', 'to', 'have', 'a', 'nap', 'the', 'tortoise', 'plodded', 'on', 'and', 'plodded', 'on', 'and', 'when', 'the', 'hare', 'awoke', 'from', 'his', 'nap', 'he', 'saw', 'the', 'tortoise', 'just', 'finishing', 'the', 'course', 'and', 'soon', 'after', 'crossed', 'the', 'finish', 'line']

Token Counts:

- **Total Tokens:** 143
- **Total Unique Tokens:** 78

Reflection on Tokenization:

I found that tokenization is a very important step. It turns the unstructured block of text into a list of items that I can count and work with. Without tokenization, I could not perform any of the later steps like removing stop words or stemming. It is the foundation for turning language into data that a computer can analyze.

Key Benefits of Tokenization:

- Converts unstructured text to structured data.
- Enables counting and analysis of individual words.
- Prepares text for stopping word removal and stemming.

4. **Stop Word Removal**

Next, I removed the stop words. Stop words are very common words that do not add much meaning to the text. Words like "the," "is," "a," and "and" appear often but do not tell me what the story is about.

I created my own stop word list based on the tokens from my corpus:

Stop Words List I Used: This is the list of common stop words I researched and compiled for removal. | | | | | | :--- | :--- | :--- | :--- | | a | and | at | been | | before | but | by | could | | down | for | from | had | | has | have | he | here | | his | i | in | is | | it | its | just | me | | my | never | of | on | | other | out | put | round | | said | shall | so | that | | the | their | then | this | | till | to | up | was | | we | when | with | yet | | you | your | | |

I then went through my token list and removed every word that was on my stop word list.

Tokens after Stop Word Removal:

['hare', 'once', 'boasting', 'speed', 'animals', 'beaten', 'full', 'speed', 'challenge', 'race', 'tortoise', 'quietly', 'accept', 'challenge', 'good', 'joke', 'hare', 'dance', 'keep', 'boasting', 'you've', 'beaten', 'answered', 'tortoise', 'shall', 'race', 'course', 'fixed', 'start', 'made', 'hare', 'darted', 'almost', 'out', 'sight', 'once', 'soon', 'stopped', 'show', 'contempt', 'tortoise', 'nap', 'tortoise', 'plodded', 'plodded', 'hare', 'awoke', 'nap', 'saw', 'tortoise', 'finishing', 'course', 'soon', 'crossed', 'finish', 'line']

New Token Counts:

- **New Total Tokens:** 56
- **New Unique Tokens:** 36

Reflection on Stop Word Removal:

This step really cleaned up my data. It reduced the total number of tokens by a lot, from 143 to 56. The new list of tokens is much more focused on the real content of the story. Words like "hare," "tortoise," "speed," and "race" are now more prominent. This step helps focus on the analysis of the words that carry the most meaning.

5. Stemming/Lemmatization

For this step, I chose to perform **stemming**.

I also researched **lemmatization**. I learned that is a more advanced process. It uses a dictionary to find the root form of a word, which is called the lemma. For example, lemmatization would turn "beaten" into "beat" and "awoke" into "awake." It is more accurate but also more complex.

Stemming is a simpler process. It just chops off the end of words using a set of rules, without knowing what the word means. I chose stemming because it is easier to apply manually. For my process, I applied common rules, like removing "ing," "ed," and "s" from the ends of words.

Why I Chose Stemming Over Lemmatization: I also researched lemmatization. Lemmatization is more advanced and uses a dictionary to find the true root word. I chose stemming because it is a more straightforward, rule-based approach that is easier to implement manually.

Stemming Transformations: Before vs. After Here are examples of changes from the stemming process. | Original Word | Stemmed Word | | :--- | :--- | | boasting | boast | | animals | anim | | beaten | beaten | | answered | answer | | fixed | fix | | darted | dart | | stopped | stop | | plodded | plod | | awoke | awok | | finishing | finish | | crossed | cross |

Final Stemmed Tokens:
['hare', 'onc', 'boast', 'speed', 'anim', 'beaten', 'full', 'speed', 'challeng', 'race', 'tortois', 'quietli', 'accept', 'challeng', 'good', 'joke', 'hare', 'danc', 'keep', 'boast', 'youv', 'beaten', 'answer', 'tortois', 'shall', 'race', 'cours', 'fix', 'start', 'made', 'hare', 'dart', 'almost', 'out', 'sight', 'onc', 'soon', 'stop', 'show', 'contempt', 'tortois', 'nap', 'tortois', 'plod', 'plod', 'hare', 'awok', 'nap', 'saw', 'tortois', 'finish', 'cours', 'soon', 'cross', 'finish', 'line']

Noticeable Changes:

- 'once' became 'onc'
- 'boasting' became 'boast'
- 'animals' became 'anim'
- 'challenge' became 'challeng'
- 'tortoise' became 'tortois'
- 'quietly' became 'quietli'
- 'answered' became 'answer'
- 'course' became 'cours'
- 'fixed' became 'fix'
- 'darted' became 'dart'
- 'stopped' became 'stop'
- 'plodded' became 'plod'
- 'awoke' became 'awok'
- 'finishing' became 'finish'
- 'crossed' became 'cross'

(Note: My simple stemming did not change 'beaten' or 'made', which shows the limits of this manual process.)

6. **Part-of-Speech (POS) Tagging**

In this step, I manually tagged 20 tokens from my stemmed list with their part of speech. This process helps identify the grammatical role of each word (like a noun, verb, or adjective).

Manual POS Tagging (20 Tokens):

1. hare - **Noun (NN)**
2. boast - **Verb (VB)**
3. speed - **Noun (NN)**
4. anim - **Noun (NN)**
5. beaten - **Verb, Past Participle (VBN)**
6. full - **Adjective (JJ)**
7. challeng - **Noun (NN)**
8. race - **Noun (NN)**
9. tortois - **Noun (NN)**
10. quietli - **Adverb (RB)**
11. accept - **Verb (VB)**
12. good - **Adjective (JJ)**
13. joke - **Noun (NN)**
14. danc - **Verb (VB)**
15. keep - **Verb (VB)**
16. answer - **Verb (VB)**
17. cours - **Noun (NN)**
18. fix - **Verb (VB)**
19. start - **Noun (NN)**
20. dart - **Verb (VB)**

Reflection on POS Tagging:

This step was challenging but very useful. It made me think about the job each word does in a sentence. For example, the word "race" could be a noun ("a race") or a verb ("to race"). In my list, it was used as a noun. Knowing the part of speech is critical for understanding the true meaning of a sentence. A computer would need this information to tell the difference between "I challenge you to a race" (noun) and "Shall we race?" (verb).

7. **Vectorization**

Vectorization is the final step, where I turn the processed words into numbers. Machine learning models cannot understand text, but they can understand numbers (vectors).

I chose the Count Vectorization technique. This technique represents text by counting the number of times each word from the vocabulary appears.

I will demonstrate this on a simple subset of my text.

- **Subset:** "hare boast speed"
- **Vocabulary (from this subset):** ['hare', 'boast', 'speed']

With this vocabulary, I can represent each word as a vector. The vector will have a '1' in the position for that word and a '0' in all other positions.

To show how this works, I will use the Count Vectorization method on two small, processed sentences from my text.

Vectorization Process Overview:

- **Text Processing Complete:** All cleaning, tokenization, and stemming steps are finished.
- **Vocabulary Creation:** I build a unique vocabulary from all the processed tokens.
- **Vector Generation:** I convert each sentence or document into a numerical vector based on that vocabulary.
- **Machine Learning Ready:** The text is now suitable for computational analysis.

Processed Sentences:

- **Sentence A (processed):** hare boast full speed
- **Sentence B (processed):** tortois answer quietli

1. Combined Vocabulary: First, I create a single vocabulary list from all the unique words in these two sentences: ['hare', 'boast', 'full', 'speed', 'tortois', 'answer', 'quietli']

Vector Representation:

- 'hare': [1, 0, 0]
- 'boast': [0, 1, 0]
- 'speed': [0, 0, 1]

If my subset was a sentence like "hare boast, hare run", and my vocabulary was ['hare', 'boast', 'run'], the vector for that sentence would be [2, 1, 1] because 'hare' appears twice.

Now, I represent each sentence as a vector. This vector will have a "1" if the word from the vocabulary is present in the sentence and a "0" if it is not.

- **Vector for Sentence A:** [1, 1, 1, 1, 0, 0, 0]
  - (It has 'hare', 'boast', 'full', 'speed', but not the others.)
- **Vector for Sentence B:** [0, 0, 0, 0, 1, 1, 1]
  - (It has 'tortois', 'answer', 'quietli', but not the others.)

Each sentence is now a numerical vector that a machine can process and analyze.

Reflection on Vectorization:

This step is the bridge from language to mathematics. By converting words into vectors, I make it possible for a computer to process the text. A machine learning algorithm could use these vectors to find patterns. For example, it might be learned that the vector for "hare" is often near the vector for "speed," while the vector for "tortois" is near the vector for "plod." This is the most important step for allowing computers to "understand" and learn from the human language.

Conclusion

This assignment demonstrated the complete journey of text from a raw, unstructured story to a clean, structured, and numerical format. Each step in the pipeline was essential.

- **Text Cleaning** and **Tokenization** turned the raw text into manageable units.
- **Stop Word Removal** and **Stemming** refined the data, reduced noise, and focused on the most important words.
- **Vectorization** provided the final bridge from human language to machine-readable numbers.

This manual exercise gave me a strong foundational understanding of why each step is so important. I now see how these core principles are the basis for all modern Natural Language Processing.