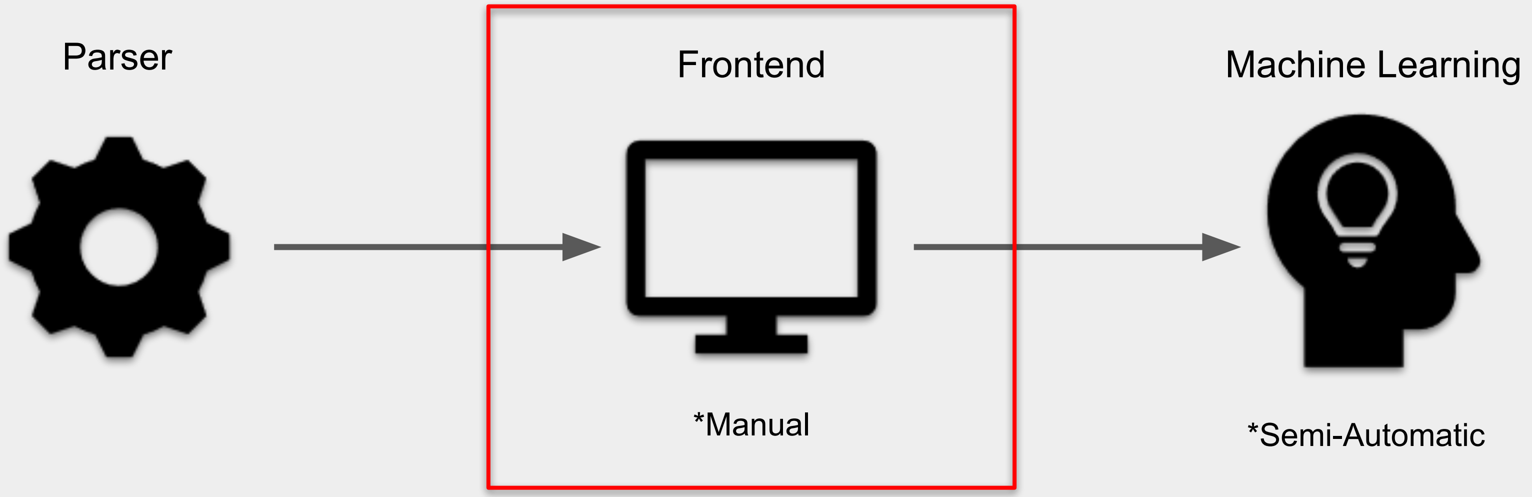
Natural Language Processing and Taxonomy Creation Tool

Brandon Christensen, Kadir Nour, Riley Hunter
The Boeing Company: Don Brancato & Rocky Bhatt

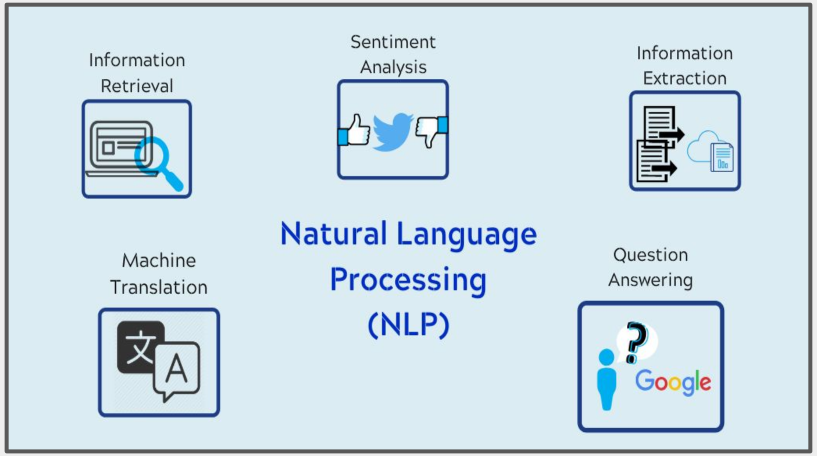
Scope

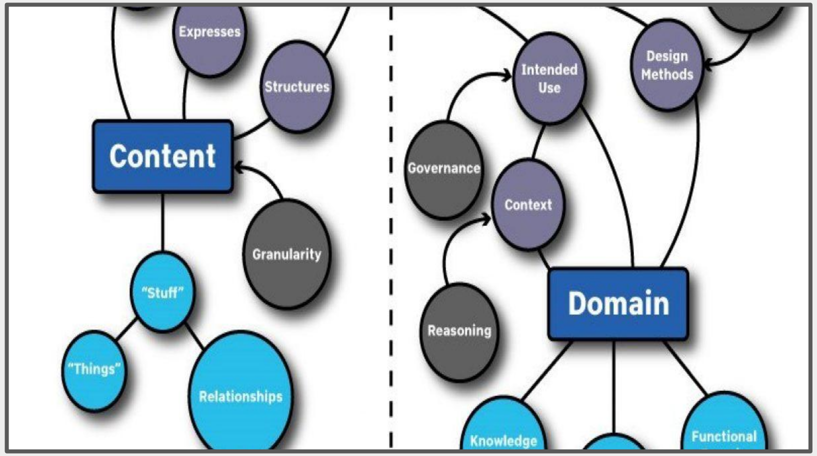


Motivation

Goal:

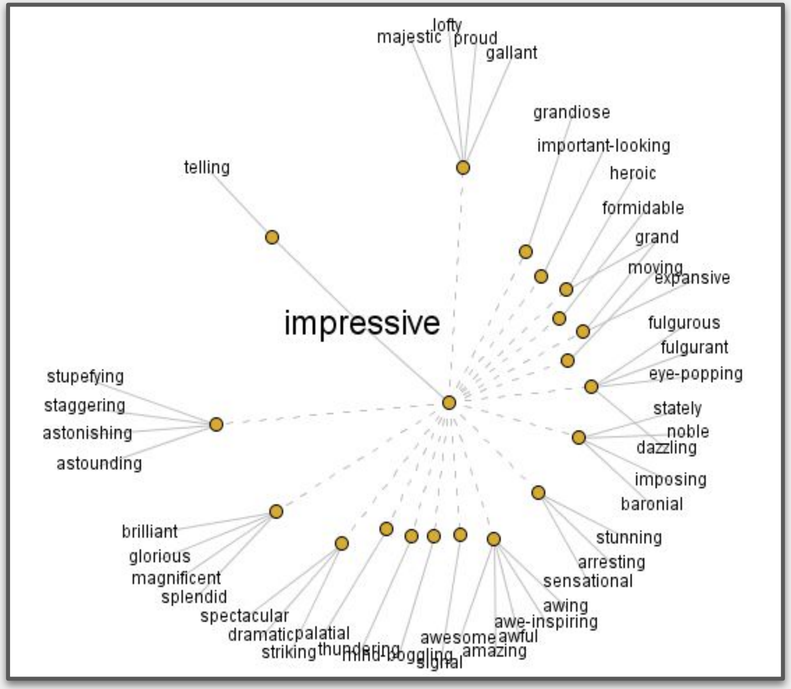
Create a state-of-the-art automatic taxonomy service that can parse and generate taxonomies for any constrained vertical or corpus based on semantic rule based system.

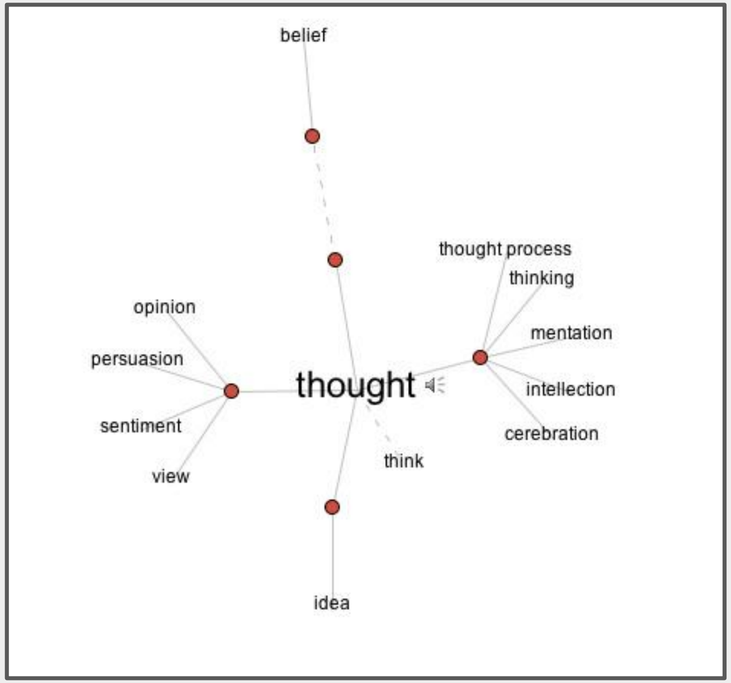




Motivation

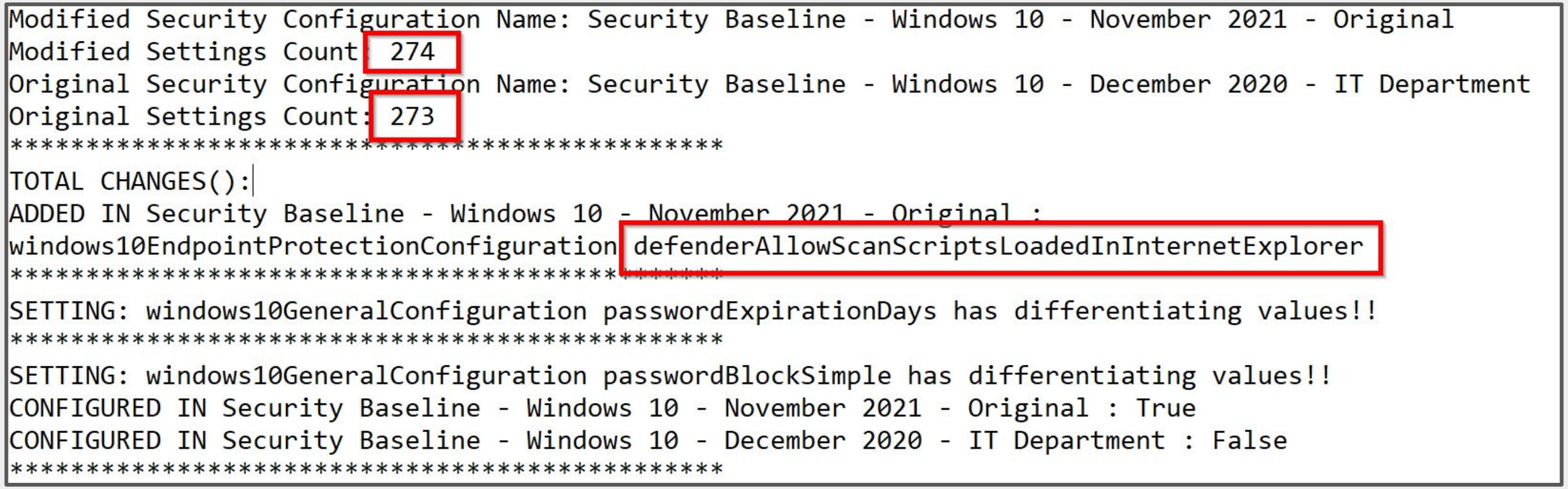
Generalized Text





Motivation

Baseline Security

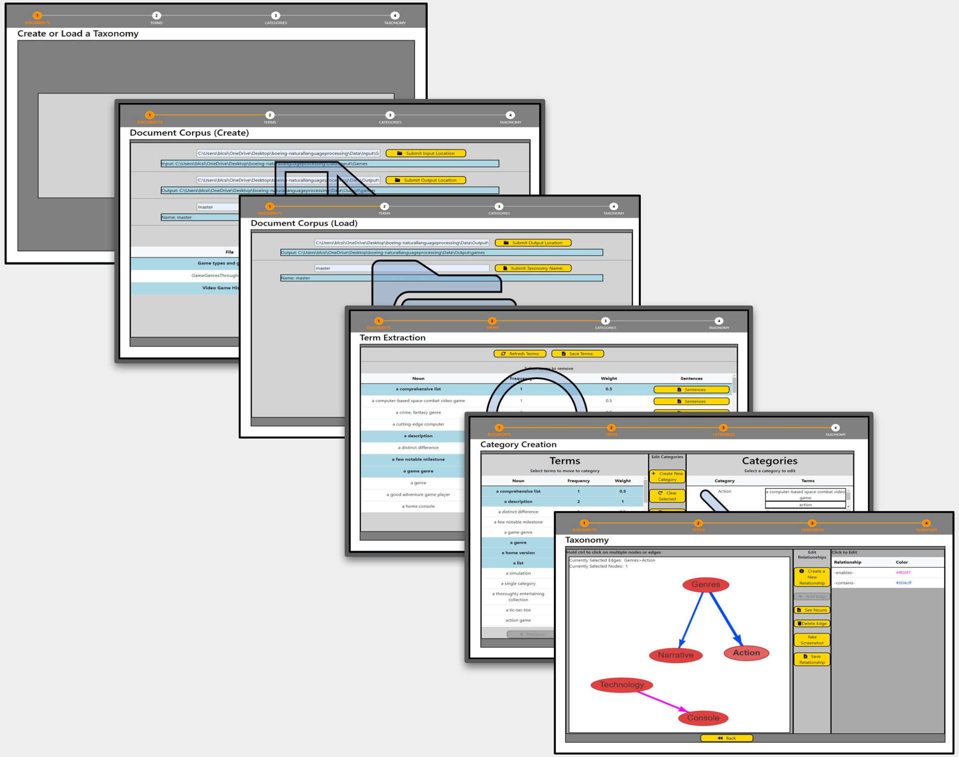


```
Modified Security Configuration Name: Security Baseline - Windows 10 - November 2021 - Original
Modified Settings Count: 274
Original Security Configuration Name: Security Baseline - Windows 10 - December 2020 - IT Department
Original Settings Count: 273
***********************************************
TOTAL CHANGES():
ADDED IN Security Baseline - Windows 10 - November 2021 - Original :
windows10EndpointProtectionConfiguration defenderAllowScanScriptsLoadedInInternetExplorer
***********************************************
SETTING: windows10GeneralConfiguration passwordExpirationDays has differentiating values!!
***********************************************
SETTING: windows10GeneralConfiguration passwordBlockSimple has differentiating values!!
CONFIGURED IN Security Baseline - Windows 10 - November 2021 - Original : True
CONFIGURED IN Security Baseline - Windows 10 - December 2020 - IT Department : False
***********************************************
```

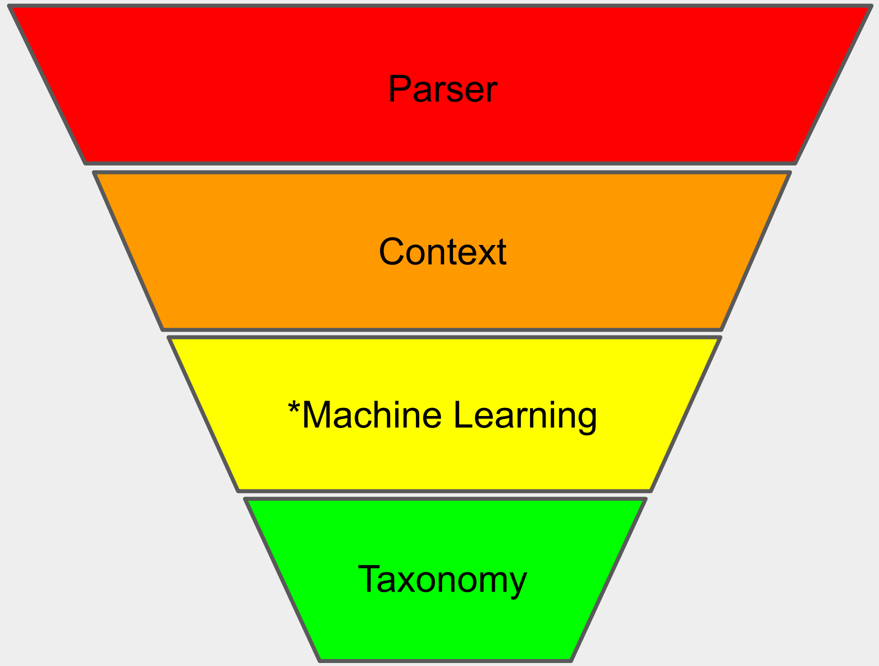
Motivation

Problem:

1. Create a interactive and intuitive frontend.
2. Incorporate context.
3. Expert-driven manipulation of taxonomies.
4. Data saving.



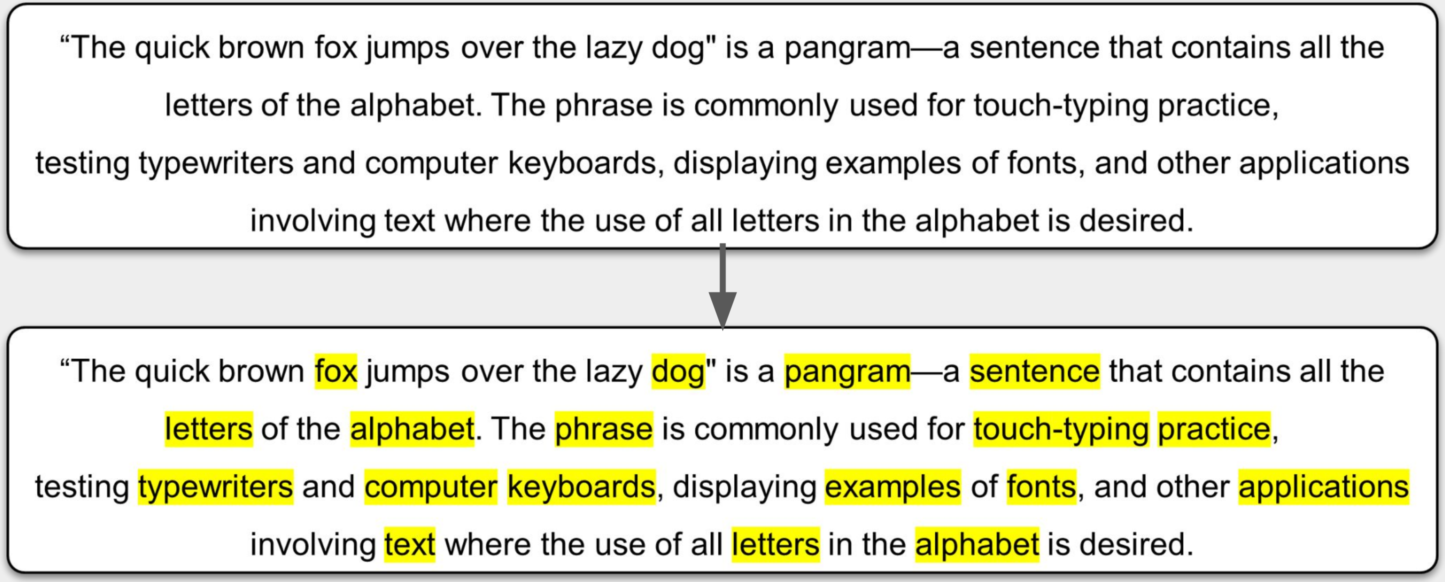
Solution



Frontend

Solution

Why Do We Need Parsers?

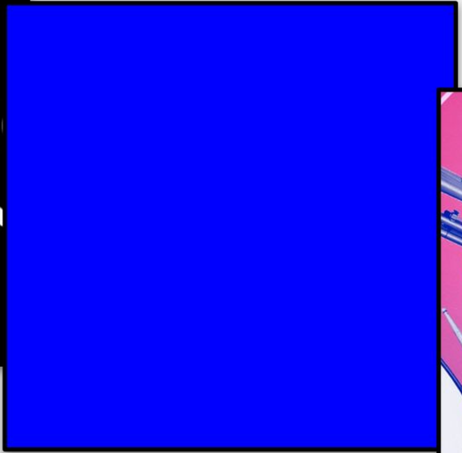


“The quick brown fox jumps over the lazy dog" is a pangram—a sentence that contains all the letters of the alphabet. The phrase is commonly used for touch-typing practice, testing typewriters and computer keyboards, displaying examples of fonts, and other applications involving text where the use of all letters in the alphabet is desired.

“The quick brown fox jumps over the lazy dog" is a pangram—a sentence that contains all the letters of the alphabet. The phrase is commonly used for touch-typing practice, testing typewriters and computer keyboards, displaying examples of fonts, and other applications involving text where the use of all letters in the alphabet is desired.

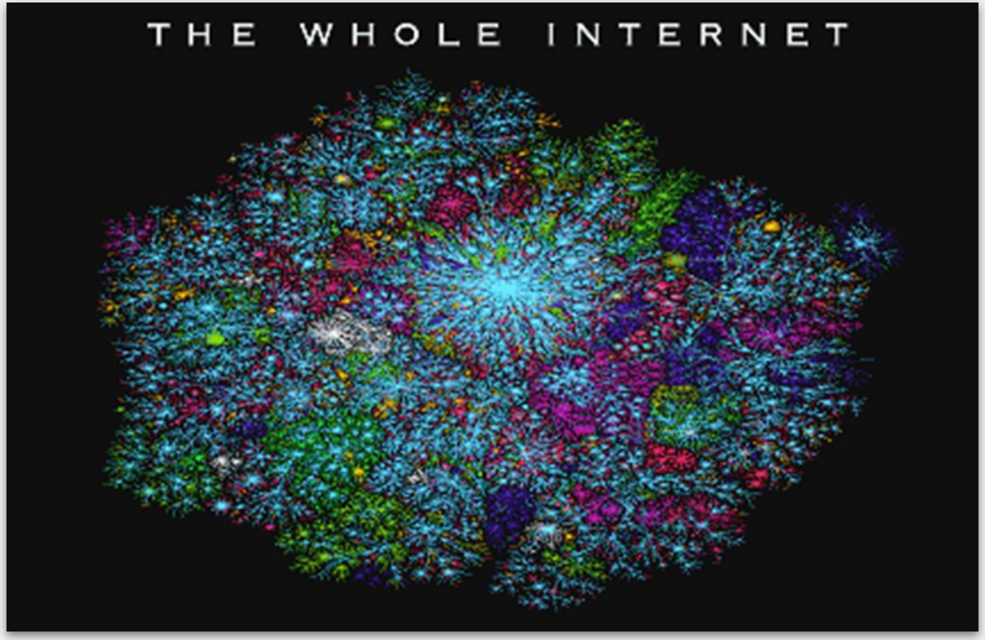
Solution

<u>Why Do We Need Context?</u>



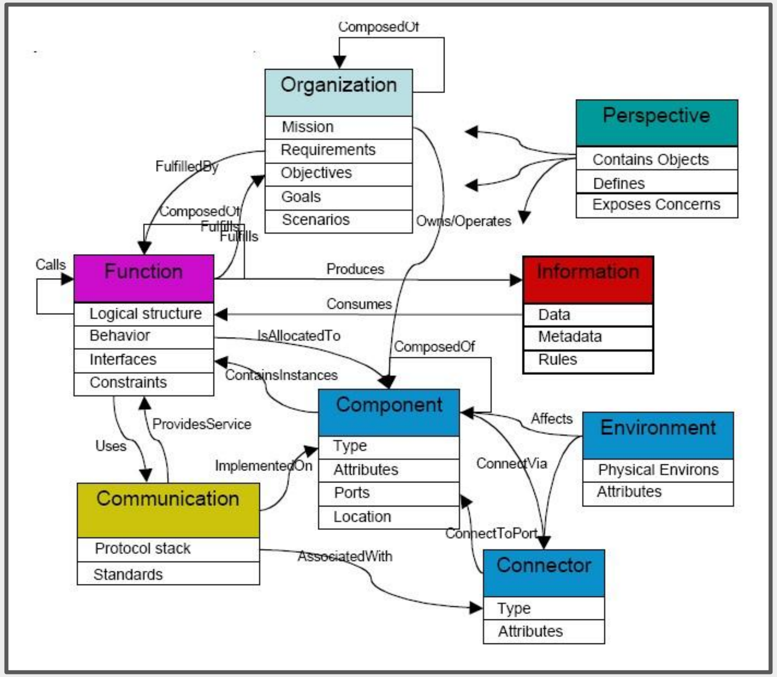
Solution

Why Do We Need Machine Learning?



Solution

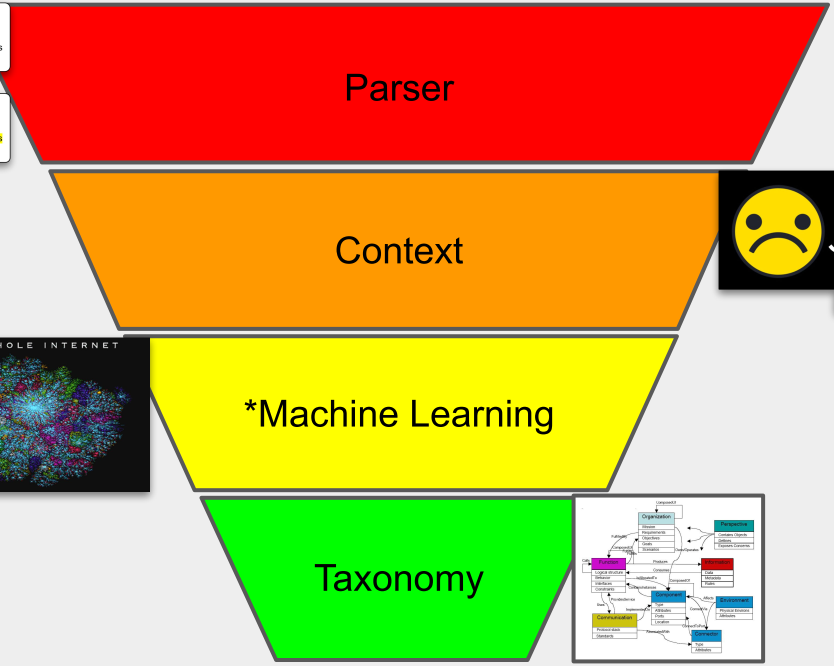
Why Do We Need A Taxonomy?

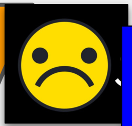


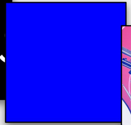
Solution

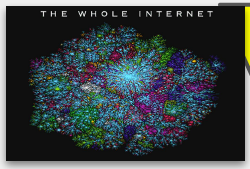
“The quick brown fox jumps over the lazy dog” is a pangram—a sentence that contains all the letters of the alphabet. The phrase is commonly used for touch-typing practice, testing typewriters and computer keyboards, displaying examples of fonts, and other applications involving text where the use of all letters in the alphabet is desired.

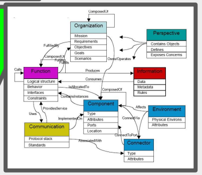
“The quick brown fox jumps over the lazy dog” is a pangram—a sentence that contains all the letters of the alphabet. The phrase is commonly used for touch-typing practice, testing typewriters and computer keyboards, displaying examples of fonts, and other applications involving text where the use of all letters in the alphabet is desired.











Frontend

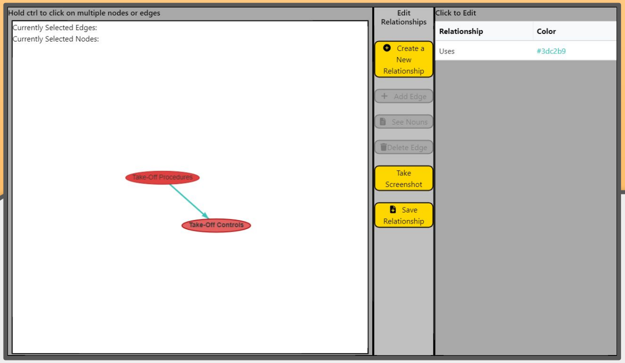
Use Case

John at Boeing

John is a test pilot for Boeing. He would like to use the 737-flight manual as a reference, but he does not want to read through the entire 200-page document to find the information he needs.

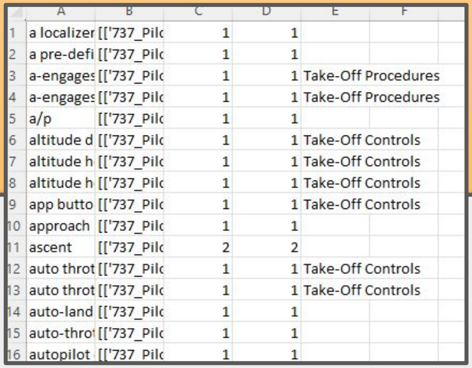
1. Running our tool first finds key terms from the document (the vocabulary).
2. He then organizes the vocabulary into categories (as an expert himself, he knows which categories to create).
3. One of the categories he creates is “Take-Off Procedures”, and another is “Take-Off Controls”. He finishes by creating a “uses” relationships between “Take-Off Procedures” and “Take-Off Controls”.

Now, if John wants to find information on take-off procedures, he can search for that category. If he wants to find information on take-off controls, then he would check the related categories and find the “Take-Off Controls” category.


Hold ctrl to click on multiple nodes or edges

Currently Selected Edges:
Currently Selected Nodes:

Relationship	Color
Uses	#3dc2b9



	A	B	C	D	E	F
1	a localizer	[['737_Pilc	1	1		
2	a pre-defi	[['737_Pilc	1	1		
3	a-engages	[['737_Pilc	1	1	Take-Off Procedures	
4	a-engages	[['737_Pilc	1	1	Take-Off Procedures	
5	a/p	[['737_Pilc	1	1		
6	altitude d	[['737_Pilc	1	1	Take-Off Controls	
7	altitude h	[['737_Pilc	1	1	Take-Off Controls	
8	altitude h	[['737_Pilc	1	1	Take-Off Controls	
9	app butto	[['737_Pilc	1	1	Take-Off Controls	
10	approach	[['737_Pilc	1	1		
11	ascent	[['737_Pilc	2	2		
12	auto throt	[['737_Pilc	1	1	Take-Off Controls	
13	auto throt	[['737_Pilc	1	1	Take-Off Controls	
14	auto-land	[['737_Pilc	1	1		
15	auto-throt	[['737_Pilc	1	1		
16	autopilot	[['737_Pilc	1	1		

John at Boeing

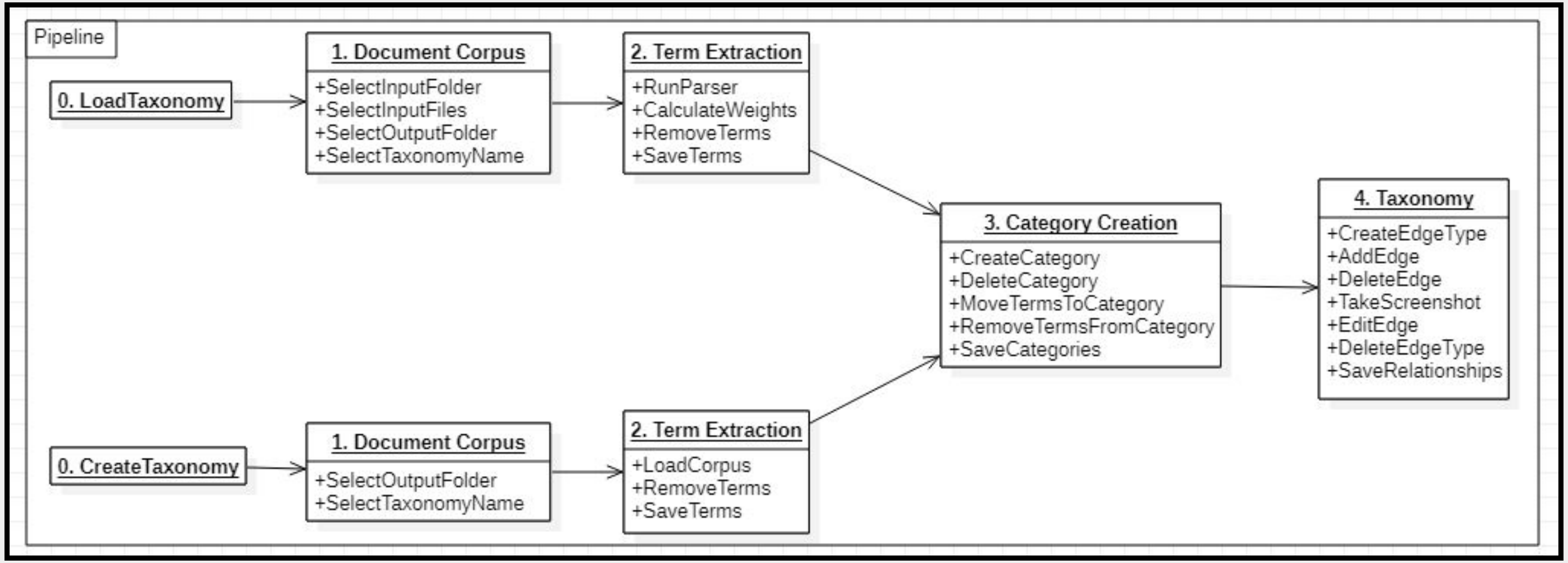
John is a test pilot for Boeing. He would like to use the 737-flight manual as a reference, but he does not want to read through the entire 200-page document to find the information he needs.

1. Running our tool first finds key terms from the document (the vocabulary).
2. He then organizes the vocabulary into categories (as an expert himself, he knows which categories to create).
3. One of the categories he creates is “Take-Off Procedures”, and another is “Take-Off Controls”. He finishes by creating a “uses” relationships between “Take-Off Procedures” and “Take-Off Controls”.

Now, if John wants to find information on take-off procedures, he can search for that category. If he wants to find information on take-off controls, then he would check the related categories and find the “Take-Off Controls” category.

Software Design

Pipeline Architecture



Tools and Technology

Dev/Communications:

Discord/Webex	Communications
GitHub	Version Control
Git Bash	Shell Environment
VS Code	Main Editor
Docker*	Containerization

Middle/ Backend:

Flask	Front-to-back
SpaCy	NLP library
Pdfplummer	PDF library

Frontend:

React	Frontend Framework
Bootstrap	Frontend Framework
Font Awesome	Buttons
Download	Download Screenshots
Html2Canvas	Screenshots
React-vis	Graph visualization
React-step-progress-bar	Progress Bar

Languages:

Python

JavaScript

HTML

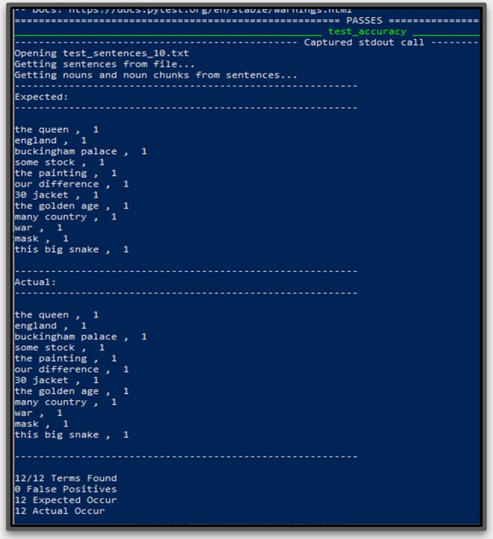
CSS

Testing

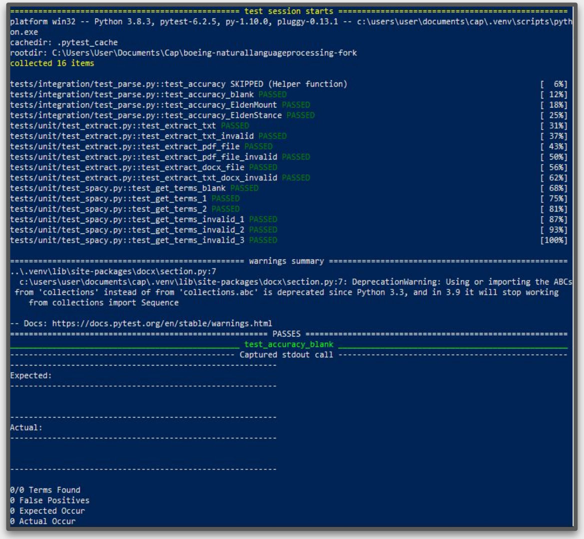
1. Unit and integration tests with Pytest. (Text extraction accuracy, Performance, etc.)

2. End-to-End tests with TestCafé.(Adding or deleting categories, Displaying terms in tables, etc.)

3. Automated CI/CD pipeline using GitHub



```
-- Docs: https://docs.pytest.org/en/stable/warnings.html
======================================================== PASSES ===============
_____________________________________________________ test_accuracy ___________
---------------------------------------------------- Captured stdout call --------
Opening test_sentences_10.txt
Getting sentences from file...
Getting nouns and noun chunks from sentences...
------------------------------------------------------------
Expected:
------------------------------------------------------------

the queen ,  1
england ,  1
buckingham palace ,  1
some stock ,  1
the painting ,  1
our difference ,  1
30 jacket ,  1
the golden age ,  1
many country ,  1
war ,  1
mask ,  1
this big snake ,  1

------------------------------------------------------------
Actual:
------------------------------------------------------------

the queen ,  1
england ,  1
buckingham palace ,  1
some stock ,  1
the painting ,  1
our difference ,  1
30 jacket ,  1
the golden age ,  1
many country ,  1
war ,  1
mask ,  1
this big snake ,  1

------------------------------------------------------------

12/12 Terms Found
0 False Positives
12 Expected Occur
12 Actual Occur
```



```
============================================ test session starts ============================================
platform win32 -- Python 3.8.3, pytest-6.2.5, py-1.10.0, pluggy-0.13.1 -- c:\users\user\documents\cap\.venv\scripts\pyth
on.exe
cachedir: .pytest_cache
rootdir: C:\Users\User\Documents\Cap\boeing-naturallanguageprocessing-fork
collected 16 items

tests/integration/test_parse.py::test_accuracy SKIPPED (Helper function)                                [  6%]
tests/integration/test_parse.py::test_accuracy_blank PASSED                                             [ 12%]
tests/integration/test_parse.py::test_accuracy_EldenMount PASSED                                        [ 18%]
tests/integration/test_parse.py::test_accuracy_EldenStance PASSED                                       [ 25%]
tests/unit/test_extract.py::test_extract_txt PASSED                                                     [ 31%]
tests/unit/test_extract.py::test_extract_txt_invalid PASSED                                             [ 37%]
tests/unit/test_extract.py::test_extract_pdf_file PASSED                                                [ 43%]
tests/unit/test_extract.py::test_extract_pdf_file_invalid PASSED                                        [ 50%]
tests/unit/test_extract.py::test_extract_docx_file PASSED                                               [ 56%]
tests/unit/test_extract.py::test_extract_txt_docx_invalid PASSED                                        [ 62%]
tests/unit/test_spacy.py::test_get_terms_blank PASSED                                                   [ 68%]
tests/unit/test_spacy.py::test_get_terms_1 PASSED                                                       [ 75%]
tests/unit/test_spacy.py::test_get_terms_2 PASSED                                                       [ 81%]
tests/unit/test_spacy.py::test_get_terms_invalid_1 PASSED                                               [ 87%]
tests/unit/test_spacy.py::test_get_terms_invalid_2 PASSED                                               [ 93%]
tests/unit/test_spacy.py::test_get_terms_invalid_3 PASSED                                               [100%]

============================================ warnings summary ============================================
..\.venv\lib\site-packages\docx\section.py:7
  c:\users\user\documents\cap\.venv\lib\site-packages\docx\section.py:7: DeprecationWarning: Using or importing the ABCs
 from 'collections' instead of from 'collections.abc' is deprecated since Python 3.3, and in 3.9 it will stop working
    from collections import Sequence

-- Docs: https://docs.pytest.org/en/stable/warnings.html
================================================= PASSES =================================================
________________________________________ test_accuracy_blank _________________________________________
---------------------------------------- Captured stdout call ----------------------------------------
------------------------------------------------------------
Expected:
------------------------------------------------------------


------------------------------------------------------------
Actual:
------------------------------------------------------------


------------------------------------------------------------


0/0 Terms Found
0 False Positives
0 Expected Occur
0 Actual Occur
```

Next Steps

- Machine Learning
  - Automation

- Frontend Tweeks
  - Search functionality

- Parser Accuracy
  - Optical Search

Thanks!

Thank you Don, Rocky, and AJ!

[WSUCptSCapstone-Fall2021Spring2022/boeing-naturallanguageprocessing (github.com)]