Reading and Lab: Application of AI Across Industries

Bookmark this page

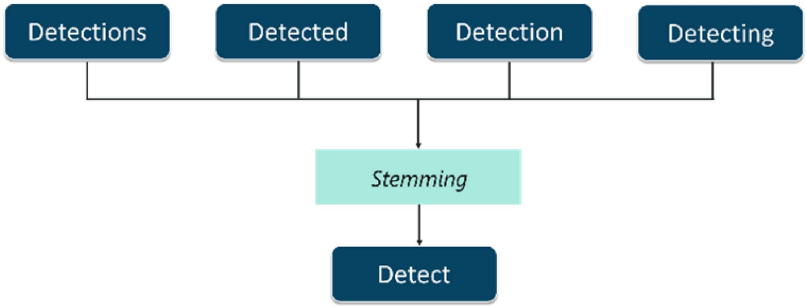
Like humans, AI machines must have structures in place that allows them to acquire, build, and understand language. To do this, programs use operations like:

Text analytics, which is the automated process of understanding and classifying unstructured text—also called text mining. An example is a word cloud. Programs that do this: Microsoft Azure, Monkey Learn, IBM Watson, Aylien.

Word Frequency counts the number of occurrences of any word in a document without regard to the weight or meaning of the words. Tf-idf- has two parts, term frequency is how often a word appears in a document—the more frequently a term appears in a specific document, the more important it is and inverse document frequency which is how important a word is in a document or corpus when weighted and measured with a numerical value.

Together these two components ensure that frequently used words like “the” or “an” are not given importance in a document or corpus, but frequently used words that will provide insight and meaning are selected. This function is used for things like ranking results in a search engine, building chatbots, and text mining.

Stemming is the algorithmic process of reducing a word to its stem, root, or base form. A stemming algorithm usually employs the simplistic process of lopping off the endings of a word. For example, a stemming algorithm might reduce the words "dogs," "dogfish," and "dogfood" to the word "dog." But reducing the words "organizing," "organizational," and "organization" to the root word "organ" would be a step too far from the correct morpheme, so a lemmatization algorithm, which uses natural language processing, would be in order to employ a morphological analysis of the word to parse the correct root. Stemming is used by web search engines, domain analysis, and information retrieval. A common stemming algorithm is Porter’s algorithm, though many others are also used.



Sentiment Analysis is the analysis and classification of text that reveals subjective information like emotion and opinion about specific topics. Sentiment analysis most often examines whether a text is positive, negative, or neutral. These tools are used to mine text to gauge public opinion, monitor brand reputation, conduct market research, study customer experiences, and monitor social media. A challenge for sentiment analysis is detecting sarcasm in otherwise serious text. For example, in a customer review of a product, there can be a complex and diametrically opposed subtext like the sample below. The reviewer wrote a very tongue-in-cheek literary criticism for a children’s book that is very humorous but could be categorized as a negative review if not analyzed closely.

In this lab, you will be using a stemming algorithm to create a frequency count of words in several sentences. You will be using some of the advanced packages in this demo. You will learn about these packages later in the course.

Jupyter Notebook - Stemming Demo Complete

Step 1: Download the following Jupyter Notebook on your computer:

StemmingDemo_complete.ipynb - This is the completed notebook that you can use if you need help.

StemmingDemo_blank.ipynb - This notebook is blank and you will enter the code below.

Step 1: Start Anaconda Navigator and select Jupyter Notebook and then upload the StemmingDemo_blank.ipynb file to your notebook.

Step 2: Copy and paste the following code into the 11 **Code** cells.

Cell #1

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer
import re
```

Cell #2

```
texts = [
    "Penny bought bright blue fishes.",
    "Penny bought a bright blue and orange fish.",
    "The fish fished fish.",
    "I'm fishing fish.",
    "I hate blue bugs",
```

```
    "A blue bug ate a fish",
    "fish"
]
```

Cell #3

```
vec = CountVectorizer()
```

Cell #4

```
matrix = vec.fit_transform(texts)
matrix.toarray()
```

Cell #5

```
vec.get_feature_names()
```

Cell #6

```
results = pd.DataFrame(matrix.toarray(), columns=vec.get_feature_names())
results
```

Cell #7

```
vec = CountVectorizer()
matrix = vec.fit_transform(texts)
results = pd.DataFrame(matrix.toarray(), columns=vec.get_feature_names())
results
```

In today's modern age of disruption, SkillUp Online is your ideal learning platform that enables you to upskill to the most in-demand technology skills like Data Science, Big Data, Artificial Intelligence, Cloud, Front-End Development, DevOps & many more. In your journey of evolution as a technologist, SkillUp Online helps you work smarter, get to your career goals faster and create an exciting technology led future.

Corporate

- Home
- About Us
- Enterprise
- Blog
- Press

Support

- Contact us
- Terms of Service
- Privacy Policy