

PyTextRank

In [2]: `!pip install pytextrank`

Collecting pytextrank

Downloading pytextrank-3.3.0-py3-none-any.whl.metadata (12 kB)

Requirement already satisfied: GitPython>=3.1 in c:\users\vaish\anaconda3\lib\site-packages (from pytextrank) (3.1.37)

Collecting graphviz>=0.13 (from pytextrank)

Downloading graphviz-0.20.3-py3-none-any.whl.metadata (12 kB)

Collecting icecream>=2.1 (from pytextrank)

Downloading icecream-2.1.4-py3-none-any.whl.metadata (1.3 kB)

Requirement already satisfied: networkx>=2.6 in c:\users\vaish\anaconda3\lib\site-packages (from networkx[default]>=2.6->pytextrank) (3.2.1)

Requirement already satisfied: pygments>=2.7.4 in c:\users\vaish\anaconda3\lib\site-packages (from pytextrank) (2.15.1)

Requirement already satisfied: scipy>=1.7 in c:\users\vaish\anaconda3\lib\site-packages (from pytextrank) (1.13.1)

Requirement already satisfied: spacy>=3.0 in c:\users\vaish\anaconda3\lib\site-packages (from pytextrank) (3.8.4)

Requirement already satisfied: gitdb<5,>=4.0.1 in c:\users\vaish\anaconda3\lib\site-packages (from GitPython>=3.1->pytextrank) (4.0.7)

Requirement already satisfied: colorama>=0.3.9 in c:\users\vaish\anaconda3\lib\site-packages (from icecream>=2.1->pytextrank) (0.4.6)

Collecting executing>=2.1.0 (from icecream>=2.1->pytextrank)

Downloading executing-2.2.0-py2.py3-none-any.whl.metadata (8.9 kB)

Requirement already satisfied: asttokens>=2.0.1 in c:\users\vaish\anaconda3\lib\site-packages (from icecream>=2.1->pytextrank) (2.0.5)

Requirement already satisfied: numpy>=1.22 in c:\users\vaish\anaconda3\lib\site-packages (from networkx[default]>=2.6->pytextrank) (1.26.4)

Requirement already satisfied: matplotlib>=3.5 in c:\users\vaish\anaconda3\lib\site-packages (from networkx[default]>=2.6->pytextrank) (3.8.4)

Requirement already satisfied: pandas>=1.4 in c:\users\vaish\anaconda3\lib\site-packages (from networkx[default]>=2.6->pytextrank) (2.2.2)

Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (3.0.12)

Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (1.0.5)

Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (1.0.11)

Requirement already satisfied: cymem<2.1.0,>=2.0.2 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (2.0.10)

Requirement already satisfied: preshed<3.1.0,>=3.0.2 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (3.0.9)

Requirement already satisfied: thinc<8.4.0,>=8.3.4 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (8.3.4)

Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (1.1.3)

Requirement already satisfied: srsly<3.0.0,>=2.4.3 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (2.5.0)

Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (2.0.10)

Requirement already satisfied: weasel<0.5.0,>=0.1.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (0.4.1)

Requirement already satisfied: typer<1.0.0,>=0.3.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (0.15.1)

Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (4.66.4)

Requirement already satisfied: requests<3.0.0,>=2.13.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (2.32.2)

Requirement already satisfied: pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (2.10.6)

Requirement already satisfied: jinja2 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (3.1.4)

Requirement already satisfied: setuptools in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (75.1.0)

Requirement already satisfied: packaging>=20.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (23.2)

Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in c:\users\vaish\anaconda3\lib\site-packages (from spacy>=3.0->pytextrank) (3.5.0)

Requirement already satisfied: six in c:\users\vaish\anaconda3\lib\site-packages (from asttokens>=2.0.1->icecream>=2.1->pytextrank) (1.16.0)

Requirement already satisfied: smmap<5,>=3.0.1 in c:\users\vaish\anaconda3\lib\site-packages (from gitdb<5,>=4.0.1->GitPython>=3.1->pytextrank) (4.0.0)

Requirement already satisfied: language-data>=1.2 in c:\users\vaish\anaconda3\lib\site-packages (from langcodes<4.0.0,>=3.2.0->spacy>=3.0->pytextrank) (1.3.0)

Requirement already satisfied: contourpy>=1.0.1 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (1.2.0)

Requirement already satisfied: cyclor>=0.10 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (4.51.0)

Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (1.4.4)

Requirement already satisfied: pillow>=8 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (10.3.0)

Requirement already satisfied: pyparsing>=2.3.1 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (3.0.9)

Requirement already satisfied: python-dateutil>=2.7 in c:\users\vaish\anaconda3\lib\site-packages (from matplotlib>=3.5->networkx[default]>=2.6->pytextrank) (2.9.0.post0)

Requirement already satisfied: pytz>=2020.1 in c:\users\vaish\anaconda3\lib\site-packages (from pandas>=1.4->networkx[default]>=2.6->pytextrank) (2024.1)

Requirement already satisfied: tzdata>=2022.7 in c:\users\vaish\anaconda3\lib\site-packages (from pandas>=1.4->networkx[default]>=2.6->pytextrank) (2023.3)

Requirement already satisfied: annotated-types>=0.6.0 in c:\users\vaish\anaconda3\lib\site-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy>=3.0->pytextrank) (0.6.0)

Requirement already satisfied: pydantic-core==2.27.2 in c:\users\vaish\anaconda3\lib\site-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy>=3.0->pytextrank) (2.27.2)

Requirement already satisfied: typing-extensions>=4.12.2 in c:\users\vaish\anaconda3\lib\site-packages (from pydantic!=1.8,!1.8.1,<3.0.0,>=1.7.4->spacy>=3.0->pytextrank) (4.12.2)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\vaish\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0->pytextrank) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\users\vaish\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0->pytextrank) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\vaish\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0->pytextrank) (2.2.2)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\vaish\anaconda3\lib\site-packages (from requests<3.0.0,>=2.13.0->spacy>=3.0->pytextrank) (2024.8.30)

Requirement already satisfied: blis<1.3.0,>=1.2.0 in c:\users\vaish\anaconda3\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy>=3.0->pytextrank) (1.2.0)

Requirement already satisfied: confection<1.0.0,>=0.0.1 in c:\users\vaish\anaconda3\lib\site-packages (from thinc<8.4.0,>=8.3.4->spacy>=3.0->pytextrank) (0.1.5)

```

Requirement already satisfied: click>=8.0.0 in c:\users\vaish\anaconda3\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy>=3.0->pytextrank) (8.1.7)
Requirement already satisfied: shellingham>=1.3.0 in c:\users\vaish\anaconda3\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy>=3.0->pytextrank) (1.5.4)
Requirement already satisfied: rich>=10.11.0 in c:\users\vaish\anaconda3\lib\site-packages (from typer<1.0.0,>=0.3.0->spacy>=3.0->pytextrank) (13.3.5)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in c:\users\vaish\anaconda3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy>=3.0->pytextrank) (0.20.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in c:\users\vaish\anaconda3\lib\site-packages (from weasel<0.5.0,>=0.1.0->spacy>=3.0->pytextrank) (5.2.1)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\vaish\anaconda3\lib\site-packages (from jinja2->spacy>=3.0->pytextrank) (2.1.3)
Requirement already satisfied: marisa-trie>=1.1.0 in c:\users\vaish\anaconda3\lib\site-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy>=3.0->pytextrank) (1.2.1)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\users\vaish\anaconda3\lib\site-packages (from rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy>=3.0->pytextrank) (2.2.0)
Requirement already satisfied: mdurl~=0.1 in c:\users\vaish\anaconda3\lib\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich>=10.11.0->typer<1.0.0,>=0.3.0->spacy>=3.0->pytextrank) (0.1.0)
Downloading pytextrank-3.3.0-py3-none-any.whl (26 kB)
Downloading graphviz-0.20.3-py3-none-any.whl (47 kB)
----- 0.0/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 10.2/47.1 kB ? eta -:--:--
----- 47.1/47.1 kB 69.3 kB/s eta 0:00:00
Downloading icecream-2.1.4-py3-none-any.whl (14 kB)
Downloading executing-2.2.0-py2.py3-none-any.whl (26 kB)
Installing collected packages: graphviz, executing, icecream, pytextrank
  Attempting uninstall: executing
    Found existing installation: executing 0.8.3
    Uninstalling executing-0.8.3:
      Successfully uninstalled executing-0.8.3
Successfully installed executing-2.2.0 graphviz-0.20.3 icecream-2.1.4 pytextrank-3.3.0

```

```
In [4]: !python -m spacy download en_core_web_sm
```

Collecting en-core-web-sm==3.8.0

Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any.whl (12.8 MB)

```
----- 0.0/12.8 MB ? eta -:-:--
----- 0.0/12.8 MB ? eta -:-:--
----- 0.0/12.8 MB ? eta -:-:--
----- 0.0/12.8 MB ? eta -:-:--
----- 0.0/12.8 MB 217.9 kB/s eta 0:00:59
----- 0.1/12.8 MB 272.3 kB/s eta 0:00:47
----- 0.1/12.8 MB 272.3 kB/s eta 0:00:47
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.1/12.8 MB 448.2 kB/s eta 0:00:29
----- 0.2/12.8 MB 197.6 kB/s eta 0:01:04
----- 0.2/12.8 MB 197.6 kB/s eta 0:01:04
----- 0.3/12.8 MB 321.8 kB/s eta 0:00:39
----- 0.3/12.8 MB 321.8 kB/s eta 0:00:39
----- 0.3/12.8 MB 321.8 kB/s eta 0:00:39
----- 0.4/12.8 MB 401.0 kB/s eta 0:00:31
----- 0.4/12.8 MB 401.0 kB/s eta 0:00:31
----- 0.7/12.8 MB 565.4 kB/s eta 0:00:22
----- 0.8/12.8 MB 716.5 kB/s eta 0:00:17
----- 0.8/12.8 MB 716.5 kB/s eta 0:00:17
----- 1.0/12.8 MB 827.4 kB/s eta 0:00:15
----- 1.2/12.8 MB 900.0 kB/s eta 0:00:13
----- 1.2/12.8 MB 901.5 kB/s eta 0:00:13
----- 1.4/12.8 MB 975.6 kB/s eta 0:00:12
----- 1.5/12.8 MB 1.1 MB/s eta 0:00:11
----- 1.5/12.8 MB 1.0 MB/s eta 0:00:11
----- 1.7/12.8 MB 1.1 MB/s eta 0:00:10
----- 1.7/12.8 MB 1.1 MB/s eta 0:00:10
----- 1.7/12.8 MB 1.1 MB/s eta 0:00:10
----- 1.8/12.8 MB 1.1 MB/s eta 0:00:10
----- 1.9/12.8 MB 1.1 MB/s eta 0:00:10
----- 2.0/12.8 MB 1.1 MB/s eta 0:00:10
----- 2.0/12.8 MB 1.2 MB/s eta 0:00:10
----- 2.2/12.8 MB 1.2 MB/s eta 0:00:09
----- 2.2/12.8 MB 1.2 MB/s eta 0:00:09
----- 2.4/12.8 MB 1.3 MB/s eta 0:00:09
----- 2.5/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.5/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.5/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.5/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.8/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.9/12.8 MB 1.3 MB/s eta 0:00:08
----- 2.9/12.8 MB 1.3 MB/s eta 0:00:08
----- 3.0/12.8 MB 1.4 MB/s eta 0:00:08
----- 3.2/12.8 MB 1.4 MB/s eta 0:00:07
```

[illegible]

```
----- 5.0/12.8 MB 1.1 MB/s eta 0:00:08
----- 5.0/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.0/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.0/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.0/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 1.0 MB/s eta 0:00:08
----- 5.1/12.8 MB 977.0 kB/s eta 0:00:08
----- 5.1/12.8 MB 970.2 kB/s eta 0:00:08
----- 5.1/12.8 MB 970.2 kB/s eta 0:00:08
----- 5.1/12.8 MB 970.2 kB/s eta 0:00:08
----- 5.1/12.8 MB 970.2 kB/s eta 0:00:08
----- 5.1/12.8 MB 970.2 kB/s eta 0:00:08
----- 5.1/12.8 MB 929.7 kB/s eta 0:00:09
----- 5.1/12.8 MB 931.6 kB/s eta 0:00:09
----- 5.1/12.8 MB 931.6 kB/s eta 0:00:09
----- 5.1/12.8 MB 917.1 kB/s eta 0:00:09
----- 5.1/12.8 MB 917.1 kB/s eta 0:00:09
----- 5.1/12.8 MB 917.1 kB/s eta 0:00:09
----- 5.2/12.8 MB 903.0 kB/s eta 0:00:09
----- 5.2/12.8 MB 903.0 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 888.9 kB/s eta 0:00:09
----- 5.2/12.8 MB 839.0 kB/s eta 0:00:10
----- 5.2/12.8 MB 839.0 kB/s eta 0:00:10
----- 5.2/12.8 MB 829.7 kB/s eta 0:00:10
----- 5.2/12.8 MB 826.8 kB/s eta 0:00:10
----- 5.3/12.8 MB 820.4 kB/s eta 0:00:10
----- 5.3/12.8 MB 820.4 kB/s eta 0:00:10
----- 5.3/12.8 MB 811.6 kB/s eta 0:00:10
----- 5.3/12.8 MB 811.6 kB/s eta 0:00:10
----- 5.3/12.8 MB 811.6 kB/s eta 0:00:10
----- 5.3/12.8 MB 801.6 kB/s eta 0:00:10
----- 5.3/12.8 MB 795.3 kB/s eta 0:00:10
----- 5.3/12.8 MB 792.7 kB/s eta 0:00:10
----- 5.3/12.8 MB 790.6 kB/s eta 0:00:10
----- 5.4/12.8 MB 788.2 kB/s eta 0:00:10
----- 5.4/12.8 MB 784.2 kB/s eta 0:00:10
----- 5.4/12.8 MB 783.7 kB/s eta 0:00:10
----- 5.4/12.8 MB 779.6 kB/s eta 0:00:10
----- 5.4/12.8 MB 777.5 kB/s eta 0:00:10
----- 5.4/12.8 MB 777.0 kB/s eta 0:00:10
----- 5.5/12.8 MB 774.5 kB/s eta 0:00:10
----- 5.5/12.8 MB 773.9 kB/s eta 0:00:10
----- 5.5/12.8 MB 766.9 kB/s eta 0:00:10
----- 5.5/12.8 MB 766.2 kB/s eta 0:00:10
```

```
----- 5.6/12.8 MB 767.4 kB/s eta 0:00:10
----- 5.6/12.8 MB 765.0 kB/s eta 0:00:10
----- 5.6/12.8 MB 763.1 kB/s eta 0:00:10
----- 5.6/12.8 MB 762.7 kB/s eta 0:00:10
----- 5.6/12.8 MB 762.7 kB/s eta 0:00:10
----- 5.6/12.8 MB 755.8 kB/s eta 0:00:10
----- 5.6/12.8 MB 750.9 kB/s eta 0:00:10
----- 5.6/12.8 MB 750.9 kB/s eta 0:00:10
----- 5.7/12.8 MB 747.4 kB/s eta 0:00:10
----- 5.7/12.8 MB 747.4 kB/s eta 0:00:10
----- 5.7/12.8 MB 747.4 kB/s eta 0:00:10
----- 5.7/12.8 MB 747.4 kB/s eta 0:00:10
----- 5.7/12.8 MB 729.2 kB/s eta 0:00:10
----- 5.7/12.8 MB 727.4 kB/s eta 0:00:10
----- 5.7/12.8 MB 727.1 kB/s eta 0:00:10
----- 5.7/12.8 MB 721.3 kB/s eta 0:00:10
----- 5.7/12.8 MB 721.3 kB/s eta 0:00:10
----- 5.7/12.8 MB 721.3 kB/s eta 0:00:10
----- 5.8/12.8 MB 713.9 kB/s eta 0:00:10
----- 5.8/12.8 MB 713.9 kB/s eta 0:00:10
----- 5.8/12.8 MB 713.9 kB/s eta 0:00:10
----- 5.8/12.8 MB 704.1 kB/s eta 0:00:10
----- 5.8/12.8 MB 704.1 kB/s eta 0:00:10
----- 5.8/12.8 MB 699.9 kB/s eta 0:00:11
----- 5.8/12.8 MB 698.5 kB/s eta 0:00:11
----- 5.8/12.8 MB 697.1 kB/s eta 0:00:11
----- 5.8/12.8 MB 694.4 kB/s eta 0:00:11
----- 5.8/12.8 MB 694.4 kB/s eta 0:00:11
----- 5.8/12.8 MB 694.4 kB/s eta 0:00:11
----- 5.9/12.8 MB 687.9 kB/s eta 0:00:11
----- 5.9/12.8 MB 685.2 kB/s eta 0:00:11
----- 5.9/12.8 MB 685.2 kB/s eta 0:00:11
----- 5.9/12.8 MB 680.2 kB/s eta 0:00:11
----- 5.9/12.8 MB 678.9 kB/s eta 0:00:11
----- 5.9/12.8 MB 676.4 kB/s eta 0:00:11
----- 5.9/12.8 MB 676.4 kB/s eta 0:00:11
----- 6.0/12.8 MB 675.1 kB/s eta 0:00:11
----- 6.0/12.8 MB 671.4 kB/s eta 0:00:11
----- 6.0/12.8 MB 671.4 kB/s eta 0:00:11
----- 6.0/12.8 MB 666.7 kB/s eta 0:00:11
----- 6.0/12.8 MB 666.7 kB/s eta 0:00:11
----- 6.0/12.8 MB 662.1 kB/s eta 0:00:11
----- 6.0/12.8 MB 662.1 kB/s eta 0:00:11
----- 6.0/12.8 MB 662.1 kB/s eta 0:00:11
----- 6.1/12.8 MB 657.6 kB/s eta 0:00:11
----- 6.1/12.8 MB 654.2 kB/s eta 0:00:11
----- 6.1/12.8 MB 654.2 kB/s eta 0:00:11
----- 6.1/12.8 MB 652.1 kB/s eta 0:00:11
----- 6.1/12.8 MB 652.1 kB/s eta 0:00:11
----- 6.1/12.8 MB 648.8 kB/s eta 0:00:11
----- 6.1/12.8 MB 648.8 kB/s eta 0:00:11
----- 6.2/12.8 MB 646.7 kB/s eta 0:00:11
----- 6.2/12.8 MB 644.6 kB/s eta 0:00:11
----- 6.2/12.8 MB 644.7 kB/s eta 0:00:11
----- 6.2/12.8 MB 643.7 kB/s eta 0:00:11
----- 6.2/12.8 MB 642.6 kB/s eta 0:00:11
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

----- - 12.4/12.8 MB 115.2 kB/s eta 0:00:04
----- - 12.4/12.8 MB 114.1 kB/s eta 0:00:04
----- - 12.4/12.8 MB 114.1 kB/s eta 0:00:04
----- - 12.4/12.8 MB 114.1 kB/s eta 0:00:04
----- - 12.5/12.8 MB 114.7 kB/s eta 0:00:04
----- - 12.5/12.8 MB 114.7 kB/s eta 0:00:04
----- - 12.5/12.8 MB 116.1 kB/s eta 0:00:03
----- - 12.5/12.8 MB 116.1 kB/s eta 0:00:03
----- - 12.6/12.8 MB 118.6 kB/s eta 0:00:02
----- - 12.6/12.8 MB 119.4 kB/s eta 0:00:02
----- - 12.6/12.8 MB 119.4 kB/s eta 0:00:02
----- - 12.6/12.8 MB 119.4 kB/s eta 0:00:02
----- - 12.6/12.8 MB 119.4 kB/s eta 0:00:02
----- - 12.6/12.8 MB 119.4 kB/s eta 0:00:02
----- - 12.7/12.8 MB 120.6 kB/s eta 0:00:01
----- - 12.7/12.8 MB 120.6 kB/s eta 0:00:01
----- - 12.7/12.8 MB 120.6 kB/s eta 0:00:01
----- - 12.7/12.8 MB 120.6 kB/s eta 0:00:01
----- - 12.7/12.8 MB 120.6 kB/s eta 0:00:01
----- - 12.8/12.8 MB 122.9 kB/s eta 0:00:01
----- - 12.8/12.8 MB 122.9 kB/s eta 0:00:00

```

[+] Download and installation successful

You can now load the package via `spacy.load('en_core_web_sm')`

```
In [5]: import spacy
import pytextrank
```

C:\Users\Vaish\anaconda3\Lib\site-packages

```
In [6]: document = """Not only did it only confirm that the film would be unfunny and gener
plot point, every joke is told in the trailer."""
```

```
In [14]: en_nlp = spacy.load("en_core_web_sm")
en_nlp.add_pipe("textrank")
doc = en_nlp(document)
```

tr=doc._.textrank print(tr.elapsed_time)

```
In [16]: for combination in doc._.phrases:
print(combination.text,combination.rank,combination.count)
```

```
ENTIRE 0.13514348101679782 1
the ENTIRE movie 0.09548608913294183 1
every
```

```
plot point 0.07067668581298282 1
every joke 0.05936552514177136 1
the film 0.05423292745389326 1
the trailer 0.04834919915077192 1
I 0.0 1
it 0.0 2
```

```
In [18]: from bs4 import BeautifulSoup
from urllib.request import urlopen
```

```
In [20]: def get_only_text(url):
        page = urlopen(url)
        soup = BeautifulSoup(page)
        text = '\t'.join(map(lambda p: p.text, soup.find_all('p')))
        print(text)
        return soup.title.text, text
```

```
In [24]: # Mention the Wikipedia url
        url="https://en.wikipedia.org/wiki/Natural_language_processing"
        # Call the function created above
        text = get_only_text(url)
```


Natural language processing (NLP) is a subfield of computer science and especially artificial intelligence. It is primarily concerned with providing computers with the ability to process data encoded in natural language and is thus closely related to information retrieval, knowledge representation and computational linguistics, a subfield of linguistics. Typically data is collected in text corpora, using either rule-based, statistical or neural-based approaches in machine learning and deep learning.

Major tasks in natural language processing are speech recognition, text classification, natural-language understanding, and natural-language generation.

Natural language processing has its roots in the 1950s.[1] Already in 1950, Alan Turing published an article titled "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence, though at the time that was not articulated as a problem separate from artificial intelligence. The proposed test includes a task that involves the automated interpretation and generation of natural language.

The premise of symbolic NLP is well-summarized by John Searle's Chinese room experiment: Given a collection of rules (e.g., a Chinese phrasebook, with questions and matching answers), the computer emulates natural language understanding (or other NLP tasks) by applying those rules to the data it confronts.

Up until the 1980s, most natural language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language processing. This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language processing.[8]

Symbolic approach, i.e., the hand-coding of a set of rules for manipulating symbols, coupled with a dictionary lookup, was historically the first approach used both by AI in general and by NLP in particular:[18][19] such as by writing grammars or devising heuristic rules for stemming.

Machine learning approaches, which include both statistical and neural networks, on the other hand, have many advantages over the symbolic approach:

Although rule-based systems for manipulating symbols were still in use in 2020, they have become mostly obsolete with the advance of LLMs in 2023.

Before that they were commonly used:

In the late 1980s and mid-1990s, the statistical approach ended a period of AI winter, which was caused by the inefficiencies of the rule-based approaches.[20][21]

The earliest decision trees, producing systems of hard if-then rules, were still very similar to the old rule-based approaches.

Only the introduction of hidden Markov models, applied to part-of-speech tagging, announced the end of the old rule-based approach.

A major drawback of statistical methods is that they require elaborate feature engineering. Since 2015,[22] the statistical approach has been replaced by the neural networks approach, using semantic networks[23] and word embeddings to capture semantic properties of words.

Intermediate tasks (e.g., part-of-speech tagging and dependency parsing) are not needed anymore.

Neural machine translation, based on then-newly invented sequence-to-sequence transformations, made obsolete the intermediate steps, such as word alignment, previously necessary for statistical machine translation.

The following is a list of some of the most commonly researched tasks in natural language processing. Some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks.

Though natural language processing tasks are closely intertwined, they can be

e subdivided into categories for convenience. A coarse division is given below.

Based on long-standing trends in the field, it is possible to extrapolate future directions of NLP. As of 2020, three trends among the topics of the long-standing series of CoNLL Shared Tasks can be observed:[46]

Most higher-level NLP applications involve aspects that emulate intelligent behaviour and apparent comprehension of natural language. More broadly speaking, the technical operationalization of increasingly advanced aspects of cognitive behaviour represents one of the developmental trajectories of NLP (see trends among CoNLL shared tasks above).

Cognition refers to "the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses." [47] Cognitive science is the interdisciplinary, scientific study of the mind and its processes. [48] Cognitive linguistics is an interdisciplinary branch of linguistics, combining knowledge and research from both psychology and linguistics. [49] Especially during the age of symbolic NLP, the area of computational linguistics maintained strong ties with cognitive studies.

As an example, George Lakoff offers a methodology to build natural language processing (NLP) algorithms through the perspective of cognitive science, along with the findings of cognitive linguistics, [50] with two defining aspects:

Ties with cognitive linguistics are part of the historical heritage of NLP, but they have been less frequently addressed since the statistical turn during the 1990s. Nevertheless, approaches to develop cognitive models towards technically operationalizable frameworks have been pursued in the context of various frameworks, e.g., of cognitive grammar, [53] functional grammar, [54] construction grammar, [55] computational psycholinguistics and cognitive neuroscience (e.g., ACT-R), however, with limited uptake in mainstream NLP (as measured by presence on major conferences [56] of the ACL). More recently, ideas of cognitive NLP have been revived as an approach to achieve explainability, e.g., under the notion of "cognitive AI". [57] Likewise, ideas of cognitive NLP are inherent to neural models multimodal NLP (although rarely made explicit) [58] and developments in artificial intelligence, specifically tools and technologies using large language model approaches [59] and new directions in artificial general intelligence based on the free energy principle [60] by British neuroscientist and theoretician at University College London Karl J. Friston.

```
In [27]: len(''.join(text))
```

```
Out[27]: 6587
```

```
In [29]: text[:1000]
```

Out[29]: ('Natural language processing - Wikipedia',
'Natural language processing (NLP) is a subfield of computer science and especially artificial intelligence. It is primarily concerned with providing computers with the ability to process data encoded in natural language and is thus closely related to information retrieval, knowledge representation and computational linguistics, a subfield of linguistics. Typically data is collected in text corpora, using either rule-based, statistical or neural-based approaches in machine learning and deep learning.\n\tMajor tasks in natural language processing are speech recognition, text classification, natural-language understanding, and natural-language generation.\n\tNatural language processing has its roots in the 1950s.[1] Already in 1950, Alan Turing published an article titled "Computing Machinery and Intelligence" which proposed what is now called the Turing test as a criterion of intelligence, though at the time that was not articulated as a problem separate from artificial intelligence. The proposed test includes a task that involves the automated interpretation and generation of natural language.\n\tThe premise of symbolic NLP is well-summarized by John Searle's Chinese room experiment: Given a collection of rules (e.g., a Chinese phrasebook, with questions and matching answers), the computer emulates natural language understanding (or other NLP tasks) by applying those rules to the data it confronts.\n\tUp until the 1980s, most natural language processing systems were based on complex sets of hand-written rules. Starting in the late 1980s, however, there was a revolution in natural language processing with the introduction of machine learning algorithms for language processing. This was due to both the steady increase in computational power (see Moore's law) and the gradual lessening of the dominance of Chomskyan theories of linguistics (e.g. transformational grammar), whose theoretical underpinnings discouraged the sort of corpus linguistics that underlies the machine-learning approach to language processing.[8]\n\tSymbolic approach, i.e., the hand-coding of a set of rules for manipulating symbols, coupled with a dictionary lookup, was historically the first approach used both by AI in general and by NLP in particular:[18][19] such as by writing grammars or devising heuristic rules for stemming.\n\tMachine learning approaches, which include both statistical and neural networks, on the other hand, have many advantages over the symbolic approach: \n\tAlthough rule-based systems for manipulating symbols were still in use in 2020, they have become mostly obsolete with the advance of LLMs in 2023. \n\tBefore that they were commonly used:\n\tIn the late 1980s and mid-1990s, the statistical approach ended a period of AI winter, which was caused by the inefficiencies of the rule-based approaches.[20][21]\n\tThe earliest decision trees, producing systems of hard if-then rules, were still very similar to the old rule-based approaches.\n\tOnly the introduction of hidden Markov models, applied to part-of-speech tagging, announced the end of the old rule-based approach.\n\tA major drawback of statistical methods is that they require elaborate feature engineering. Since 2015,[22] the statistical approach has been replaced by the neural networks approach, using semantic networks[23] and word embeddings to capture semantic properties of words. \n\tIntermediate tasks (e.g., part-of-speech tagging and dependency parsing) are not needed anymore. \n\tNeural machine translation, based on then-newly invented sequence-to-sequence transformations, made obsolete the intermediate steps, such as word alignment, previously necessary for statistical machine translation.\n\tThe following is a list of some of the most commonly researched tasks in natural language processing. Some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks.\n\tThough natural language processing tasks are closely intertwined, they can be subdivided into categories for convenience. A coarse division is given below.\n\tBased on long-standing trends in the field, it is possible to extrapolate future directions of NLP. As of 2020, three trends among the topics of the long-standing series of CoNLL Shared Tasks can be observed:[46]\n\tMost higher-level NLP applications involve aspects that emulate intelligent behaviour and apparent comprehension of natural language. More broadly speaking, the technical oper

ationalization of increasingly advanced aspects of cognitive behaviour represents one of the developmental trajectories of NLP (see trends among CoNLL shared tasks above).

Cognition refers to "the mental action or process of acquiring knowledge and understanding through thought, experience, and the senses." [47] Cognitive science is the interdisciplinary, scientific study of the mind and its processes. [48] Cognitive linguistics is an interdisciplinary branch of linguistics, combining knowledge and research from both psychology and linguistics. [49] Especially during the age of symbolic NLP, the area of computational linguistics maintained strong ties with cognitive studies.

As an example, George Lakoff offers a methodology to build natural language processing (NLP) algorithms through the perspective of cognitive science, along with the findings of cognitive linguistics, [50] with two defining aspects:

Ties with cognitive linguistics are part of the historical heritage of NLP, but they have been less frequently addressed since the statistical turn during the 1990s. Nevertheless, approaches to develop cognitive models towards technically operationalizable frameworks have been pursued in the context of various frameworks, e.g., of cognitive grammar, [53] functional grammar, [54] construction grammar, [55] computational psycholinguistics and cognitive neuroscience (e.g., ACT-R), however, with limited uptake in mainstream NLP (as measured by presence on major conferences [56] of the ACL). More recently, ideas of cognitive NLP have been revived as an approach to achieve explainability, e.g., under the notion of "cognitive AI". [57] Likewise, ideas of cognitive NLP are inherent to neural models multi-modal NLP (although rarely made explicit) [58] and developments in artificial intelligence, specifically tools and technologies using large language model approaches [59] and new directions in artificial general intelligence based on the free energy principle [60] by British neuroscientist and theoretician at University College London Karl J. Friston.

feature based text summarization

```
In [35]: !pip install sumy
```

Requirement already satisfied: sumy in c:\users\vaish\anaconda3\lib\site-packages (0.11.0)

Requirement already satisfied: docopt<0.7,>=0.6.1 in c:\users\vaish\anaconda3\lib\site-packages (from sumy) (0.6.2)

Requirement already satisfied: breadability>=0.1.20 in c:\users\vaish\anaconda3\lib\site-packages (from sumy) (0.1.20)

Requirement already satisfied: requests>=2.7.0 in c:\users\vaish\anaconda3\lib\site-packages (from sumy) (2.32.2)

Requirement already satisfied: pycountry>=18.2.23 in c:\users\vaish\anaconda3\lib\site-packages (from sumy) (24.6.1)

Requirement already satisfied: nltk>=3.0.2 in c:\users\vaish\anaconda3\lib\site-packages (from sumy) (3.9.1)

Requirement already satisfied: chardet in c:\users\vaish\anaconda3\lib\site-packages (from breadability>=0.1.20->sumy) (4.0.0)

Requirement already satisfied: lxml>=2.0 in c:\users\vaish\anaconda3\lib\site-packages (from breadability>=0.1.20->sumy) (5.2.1)

Requirement already satisfied: click in c:\users\vaish\anaconda3\lib\site-packages (from nltk>=3.0.2->sumy) (8.1.7)

Requirement already satisfied: joblib in c:\users\vaish\anaconda3\lib\site-packages (from nltk>=3.0.2->sumy) (1.4.2)

Requirement already satisfied: regex>=2021.8.3 in c:\users\vaish\anaconda3\lib\site-packages (from nltk>=3.0.2->sumy) (2023.10.3)

Requirement already satisfied: tqdm in c:\users\vaish\anaconda3\lib\site-packages (from nltk>=3.0.2->sumy) (4.66.4)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\vaish\anaconda3\lib\site-packages (from requests>=2.7.0->sumy) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\users\vaish\anaconda3\lib\site-packages (from requests>=2.7.0->sumy) (3.7)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\vaish\anaconda3\lib\site-packages (from requests>=2.7.0->sumy) (2.2.2)

Requirement already satisfied: certifi>=2017.4.17 in c:\users\vaish\anaconda3\lib\site-packages (from requests>=2.7.0->sumy) (2024.8.30)

Requirement already satisfied: colorama in c:\users\vaish\anaconda3\lib\site-packages (from click->nltk>=3.0.2->sumy) (0.4.6)

In [42]: `!pip install lxml_html_clean`

Collecting lxml_html_clean

Downloading lxml_html_clean-0.4.1-py3-none-any.whl.metadata (2.4 kB)

Requirement already satisfied: lxml in c:\users\vaish\anaconda3\lib\site-packages (from lxml_html_clean) (5.2.1)

Downloading lxml_html_clean-0.4.1-py3-none-any.whl (14 kB)

Installing collected packages: lxml_html_clean

Successfully installed lxml_html_clean-0.4.1

In [43]: `# Import the packages`

```
from sumy.parsers.html import HtmlParser

from sumy.parsers.plaintext import PlaintextParser

from sumy.nlp.tokenizers import Tokenizer

from sumy.summarizers.lsa import LsaSummarizer

from sumy.nlp.stemmers import Stemmer
```

```
from sumy.utils import get_stop_words

from sumy.summarizers.luhn import LuhnSummarizer
```

```
In [46]: # Extracting and summarizing

LANGUAGE = "english"
SENTENCES_COUNT = 10
url="https://en.wikipedia.org/wiki/Natural_language_processing"
parser = HtmlParser.from_url(url, Tokenizer(LANGUAGE))
summarizer = LsaSummarizer()
summarizer = LsaSummarizer(Stemmer(LANGUAGE))
summarizer.stop_words = get_stop_words(LANGUAGE)
for sentence in summarizer(parser.document, SENTENCES_COUNT):
    print(sentence)
```

[2] However, real progress was much slower, and after the ALPAC report in 1966, which found that ten years of research had failed to fulfill the expectations, funding for machine translation was dramatically reduced.

However, there is an enormous amount of non-annotated data available (including, among other things, the entire content of the World Wide Web), which can often make up for the worse efficiency if the algorithm used has a low enough time complexity to be practical.

[14] This is increasingly important in medicine and healthcare, where NLP helps analyze notes and text in electronic health records that would otherwise be inaccessible for study when seeking to improve care[16] or protect patient privacy.

the larger such a (probabilistic) language model is, the more accurate it becomes, in contrast to rule-based systems that can gain accuracy only by increasing the amount and complexity of the rules leading to intractability problems.

[34][35][36] As far as orthography, morphology, syntax and certain aspects of semantics are concerned, and due to the development of powerful neural language models such as GPT-2, this can now (2019) be considered a largely solved problem and is being marketed in various commercial applications.

Increasing interest in multilinguality, and, potentially, multimodality (English since 1999; Spanish, Dutch since 2002; German since 2003; Bulgarian, Danish, Japanese, Portuguese, Slovenian, Swedish, Turkish since 2006; Basque, Catalan, Chinese, Greek, Hungarian, Italian, Turkish since 2007; Czech since 2009; Arabic since 2012; 2017: 40+ languages; 2018: 60+/100+ languages) Elimination of symbolic representations (rule-based over supervised towards weakly supervised methods, representation learning and end-to-end systems)

More broadly speaking, the technical operationalization of increasingly advanced aspects of cognitive behaviour represents one of the developmental trajectories of NLP (see trends among CoNLL shared tasks above).

PMID 33736486.^ Lee, Jennifer; Yang, Samuel; Holland-Hall, Cynthia; Sezgin, Emre; Gill, Manjot; Linwood, Simon; Huang, Yungui; Hoffman, Jeffrey (2022-06-10).

Retrieved 5 December 2021.^ Yi, Chucai; Tian, Yingli(2012), "Assistive Text Reading from Complex Background for Blind Persons", Camera-Based Document Analysis and Recognition, Lecture Notes in Computer Science, vol.

Advances in Neural Information Processing Systems.^ Kariampuzha, William; Alyea, Gioconda; Qu, Sue; Sanjak, Jaleal; Mathé, Ewy; Sid, Eric; Chatelaine, Haley; Yadaw, Arjun; Xu, Yanji; Zhu, Qian (2023).

```
In [ ]: text="""A vaccine for the coronavirus will likely be ready by early 2021 but rollin
India, which is host to some of the front-runner vaccine clinical trials, currently
```

```
The timing of the vaccine is a contentious subject around the world. In the U.S., P
"""
```

```
In [50]: # Load Packages
from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
```

```
In [52]: # For Strings
parser = PlaintextParser.from_string(text,Tokenizer("english"))
```

```
In [58]: from sumy.parsers.plaintext import PlaintextParser
from sumy.nlp.tokenizers import Tokenizer
```

```
In [64]: parser = PlaintextParser.from_string(text,Tokenizer("english"))
```

```
In [71]: from sumy.summarizers.lex_rank import LexRankSummarizer
from sumy.utils import get_stop_words
# Summarize using sumy LexRank
summarizer_lex.stop_words = get_stop_words("english")
summary= summarizer_lex(parser.document, 5)
lex_summary=""
for sentence in summary:
    lex_summary+=str(sentence)
print(lex_summary)
```

('Natural language processing - Wikipedia', 'Natural language processing (NLP) is a subfield of computer science and especially artificial intelligence.[8]\n\nSymbolic approach, i.e., the hand-coding of a set of rules for manipulating symbols, coupled with a dictionary lookup, was historically the first approach used both by AI in general and by NLP in particular:[18][19] such as by writing grammars or devising heuristic rules for stemming.\n\nMachine learning approaches, which include both statistical and neural networks, on the other hand, have many advantages over the symbolic approach: \n\nAlthough rule-based systems for manipulating symbols were still in use in 2020, they have become mostly obsolete with the advance of LLMs in 2023.As of 2020, three trends among the topics of the long-standing series of CoNLL Shared Tasks can be observed:[46]\n\nMost higher-level NLP applications involve aspects that emulate intelligent behaviour and apparent comprehension of natural language.[48] Cognitive linguistics is an interdisciplinary branch of linguistics, combining knowledge and research from both psychology and linguistics.[49] Especially during the age of symbolic NLP, the area of computational linguistics maintained strong ties with cognitive studies.\n\nAs an example, George Lakoff offers a methodology to build natural language processing (NLP) algorithms through the perspective of cognitive science, along with the findings of cognitive linguistics,[50] with two defining aspects:\n\nTies with cognitive linguistics are part of the historical heritage of NLP, but they have been less frequently addressed since the statistical turn during the 1990s.

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```