```
!pip install langchain==0.0.267
!pip install requests
!pip install BeautifulSoup
```

```
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3,>=2->langchain==0.0.267) (2025.1.31)
Requirement already satisfied: greenlet>=1 in /usr/local/lib/python3.11/dist-packages (from SQLAlchemy<3,>=1.4->langchain==0.0.267) (3.1.1)
Requirement already satisfied: packaging>=17.0 in /usr/local/lib/python3.11/dist-packages (from marshmallow<4.0.0,>=3.18.0->dataclasses-json<0.6.0,>=0.5.7->langchain
Collecting mypy-extensions>=0.3.0 (from typing-inspect<1,>=0.4.0->dataclasses-json<0.6.0,>=0.5.7->langchain==0.0.267)
  Downloading mypy_extensions-1.0.0-py3-none-any.whl.metadata (1.1 kB)
Downloading langchain-0.0.267-py3-none-any.whl (1.5 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.5/1.5 MB 35.1 MB/s eta 0:00:00
Downloading dataclasses_json-0.5.14-py3-none-any.whl (26 kB)
Downloading langsmith-0.0.92-py3-none-any.whl (56 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.5/56.5 kB 4.5 MB/s eta 0:00:00
```

```
    × python setup.py egg_info did not run successfully.
    │ exit code: 1
    ╰─> See above for output.

    note: This error originates from a subprocess, and is likely not a problem with pip.
    Preparing metadata (setup.py) ... error
  error: metadata-generation-failed

  × Encountered error while generating package metadata.
```

```
!pip uninstall spacy cymem murmurhash preshed thinc blis -y
!pip install spacy==3.5.0
!pip install -numpy
#!spacy.prefer_gpu()
!pip install scispacy
```

```
Found existing installation: spacy 3.8.5
Uninstalling spacy-3.8.5:
  Successfully uninstalled spacy-3.8.5
Found existing installation: cymem 2.0.11
Uninstalling cymem-2.0.11:
  Successfully uninstalled cymem-2.0.11
Found existing installation: murmurhash 1.0.12
Uninstalling murmurhash-1.0.12:
  Successfully uninstalled murmurhash-1.0.12
Found existing installation: preshed 3.0.9
Uninstalling preshed-3.0.9:
  Successfully uninstalled preshed-3.0.9
Found existing installation: thinc 8.3.6
Uninstalling thinc-8.3.6:
  Successfully uninstalled thinc-8.3.6
Found existing installation: blis 1.3.0
Uninstalling blis-1.3.0:
  Successfully uninstalled blis-1.3.0
Collecting spacy==3.5.0
  Downloading spacy-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (25 kB)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (1.0.5)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy==3.5.0)
  Downloading murmurhash-1.0.12-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.1 kB)
Collecting cymem<2.1.0,>=2.0.2 (from spacy==3.5.0)
  Downloading cymem-2.0.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (8.5 kB)
Collecting preshed<3.1.0,>=3.0.2 (from spacy==3.5.0)
  Downloading preshed-3.0.9-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (2.2 kB)
Collecting thinc<8.2.0,>=8.1.0 (from spacy==3.5.0)
  Downloading thinc-8.1.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (15 kB)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (2.0.10)
Collecting typer<0.8.0,>=0.3.0 (from spacy==3.5.0)
  Downloading typer-0.7.0-py3-none-any.whl.metadata (17 kB)
Collecting pathy>=0.10.0 (from spacy==3.5.0)
  Downloading pathy-0.11.0-py3-none-any.whl.metadata (16 kB)
Collecting smart-open<7.0.0,>=5.2.1 (from spacy==3.5.0)
  Downloading smart_open-6.4.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (4.67.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (1.26.4)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (2.32.3)
Collecting pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 (from spacy==3.5.0)
  Downloading pydantic-1.10.21-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (153 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 153.9/153.9 kB 10.2 MB/s eta 0:00:00
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (24.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/dist-packages (from spacy==3.5.0) (3.5.0)
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.11/dist-packages (from langcodes<4.0.0,>=3.2.0->spacy==3.5.0) (1.3.0)
Collecting pathlib-abc==0.1.1 (from pathy>=0.10.0->spacy==3.5.0)
  Downloading pathlib_abc-0.1.1-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy==3.5.0) (4.13.1
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy==3.5.0) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy==3.5.0) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy==3.5.0) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy==3.5.0) (2025.1.31)
Collecting blis<0.8.0,>=0.7.8 (from thinc<8.2.0,>=8.1.0->spacy==3.5.0)
  Downloading blis-0.7.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (7.4 kB)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.11/dist-packages (from thinc<8.2.0,>=8.1.0->spacy==3.5.0) (0.1.5)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.11/dist-packages (from typer<0.8.0,>=0.3.0->spacy==3.5.0) (8.1.8)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->spacy==3.5.0) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy==3.5.0) (1.2.1)
Downloading spacy-3.5.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 12.0 MB/s eta 0:00:00
Downloading cymem-2.0.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (218 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 218.9/218.9 kB 12.7 MB/s eta 0:00:00
Downloading murmurhash-1.0.12-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (134 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 134.3/134.3 kB 9.7 MB/s eta 0:00:00
Downloading pathy-0.11.0-py3-none-any.whl (47 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 47.3/47.3 kB 3.2 MB/s eta 0:00:00
Downloading pathlib_abc-0.1.1-py3-none-any.whl (23 kB)
Downloading preshed-3.0.9-cp311-cp311-manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2014_x86_64.whl (157 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 157.2/157.2 kB 8.5 MB/s eta 0:00:00
Downloading pydantic-1.10.21-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.1/3.1 MB 63.7 MB/s eta 0:00:00
Downloading smart_open-6.4.0-py3-none-any.whl (57 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 57.0/57.0 kB 4.1 MB/s eta 0:00:00
Downloading thinc-8.1.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (917 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 917.4/917.4 kB 38.8 MB/s eta 0:00:00
Downloading typer-0.7.0-py3-none-any.whl (38 kB)
Downloading blis-0.7.11-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (10.2 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.2/10.2 MB 63.3 MB/s eta 0:00:00
Installing collected packages: cymem, typer, smart-open, pydantic, pathlib-abc, murmurhash, blis, preshed, pathy, thinc, spacy
  Attempting uninstall: typer
    Found existing installation: typer 0.15.2
    Uninstalling typer-0.15.2:
      Successfully uninstalled typer-0.15.2
  Attempting uninstall: smart-open
    Found existing installation: smart-open 7.1.0
    Uninstalling smart-open-7.1.0:
      Successfully uninstalled smart-open-7.1.0
  Attempting uninstall: pydantic
    Found existing installation: pydantic 2.11.2
    Uninstalling pydantic-2.11.2:
      Successfully uninstalled pydantic-2.11.2
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency c
langchain-core 0.3.51 requires langsmith<0.4,>=0.1.125, but you have langsmith 0.0.92 which is incompatible.
langchain-core 0.3.51 requires pydantic<3.0.0,>=2.5.2; python_full_version < "3.12.4", but you have pydantic 1.10.21 which is incompatible.
albumentations 2.0.5 requires pydantic>=2.9.2, but you have pydantic 1.10.21 which is incompatible.
google-genai 1.9.0 requires pydantic<3.0.0,>=2.0.0, but you have pydantic 1.10.21 which is incompatible.
Successfully installed blis-0.7.11 cymem-2.0.11 murmurhash-1.0.12 pathlib-abc-0.1.1 pathy-0.11.0 preshed-3.0.9 pydantic-1.10.21 smart-open-6.4.0 spacy-3.5.0 thinc-8.
WARNING: The following packages were previously imported in this runtime:
  [blis,cymem,murmurhash,preshed,pydantic,spacy,thinc]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

RESTART SESSION

```
Usage:
  pip3 install [options] <requirement specifier> [package-index-options] ...
  pip3 install [options] -r <requirements file> [package-index-options] ...
  pip3 install [options] [-e] <vcs project url> ...
  pip3 install [options] [-e] <local project path> ...
  pip3 install [options] <archive url/path> ...

no such option: -n
Collecting scispacy
  Downloading scispacy-0.5.5-py3-none-any.whl.metadata (18 kB)
Collecting spacy<3.8.0,>=3.7.0 (from scispacy)
  Downloading spacy-3.7.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (27 kB)
Requirement already satisfied: scipy in /usr/local/lib/python3.11/dist-packages (from scispacy) (1.14.1)
Requirement already satisfied: requests<3.0.0,>=2.0.0 in /usr/local/lib/python3.11/dist-packages (from scispacy) (2.32.3)
Collecting conllu (from scispacy)
  Downloading conllu-6.0.0-py3-none-any.whl.metadata (21 kB)
Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from scispacy) (1.26.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.11/dist-packages (from scispacy) (1.4.2)
Requirement already satisfied: scikit-learn>=0.20.3 in /usr/local/lib/python3.11/dist-packages (from scispacy) (1.6.1)
Collecting pysbd (from scispacy)
  Downloading pysbd-0.3.4-py3-none-any.whl.metadata (6.1 kB)
Collecting nmslib-metabrainz==2.1.3 (from scispacy)
  Downloading nmslib_metabrainz-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (956 bytes)
Collecting pybind11>=2.2.3 (from nmslib-metabrainz==2.1.3->scispacy)
  Downloading pybind11-2.13.6-py3-none-any.whl.metadata (9.5 kB)
Requirement already satisfied: psutil in /usr/local/lib/python3.11/dist-packages (from nmslib-metabrainz==2.1.3->scispacy) (5.9.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.0.0->scispacy) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.0.0->scispacy) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.0.0->scispacy) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.0.0->scispacy) (2025.1.31)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.11/dist-packages (from scikit-learn>=0.20.3->scispacy) (3.6.0)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (3.0.9)
Collecting thinc<8.3.0,>=8.2.2 (from spacy<3.8.0,>=3.7.0->scispacy)
  Downloading thinc-8.2.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (15 kB)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.1.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (0.4.1)
Requirement already satisfied: typer<1.0.0,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (0.7.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (4.67.1)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (1.10.21)
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (24.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.8.0,>=3.7.0->scispacy) (3.5.0)
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.11/dist-packages (from langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.0->scispacy) (1.3.0)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy<3.8.0,>=3.7.0->s
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.11/dist-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.0->scispacy) (0.7.11)
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.11/dist-packages (from thinc<8.3.0,>=8.2.2->spacy<3.8.0,>=3.7.0->scispacy) (0.1.5)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.11/dist-packages (from typer<1.0.0,>=0.3.0->spacy<3.8.0,>=3.7.0->scispacy) (8.1.8)
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->scispacy) (0.21
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.11/dist-packages (from weasel<0.5.0,>=0.1.0->spacy<3.8.0,>=3.7.0->scispacy) (6.4.0)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->spacy<3.8.0,>=3.7.0->scispacy) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy<3.8.0,>=3.7.0->
Downloading scispacy-0.5.5-py3-none-any.whl (46 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.2/46.2 kB 3.2 MB/s eta 0:00:00
Downloading nmslib_metabrainz-2.1.3-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.1/14.1 MB 58.4 MB/s eta 0:00:00
Downloading spacy-3.7.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.6/6.6 MB 64.4 MB/s eta 0:00:00
Downloading conllu-6.0.0-py3-none-any.whl (16 kB)
Downloading pysbd-0.3.4-py3-none-any.whl (71 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 71.1/71.1 kB 5.0 MB/s eta 0:00:00
Downloading pybind11-2.13.6-py3-none-any.whl (243 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 243.3/243.3 kB 16.7 MB/s eta 0:00:00
Downloading thinc-8.2.5-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (920 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 920.2/920.2 kB 36.9 MB/s eta 0:00:00
Installing collected packages: pysbd, pybind11, conllu, nmslib-metabrainz, thinc, spacy, scispacy
  Attempting uninstall: thinc
    Found existing installation: thinc 8.1.12
    Uninstalling thinc-8.1.12:
      Successfully uninstalled thinc-8.1.12
  Attempting uninstall: spacy
    Found existing installation: spacy 3.5.0
    Uninstalling spacy-3.5.0:
      Successfully uninstalled spacy-3.5.0
Successfully installed conllu-6.0.0 nmslib-metabrainz-2.1.3 pybind11-2.13.6 pysbd-0.3.4 scispacy-0.5.5 spacy-3.7.5 thinc-8.2.5
WARNING: The following packages were previously imported in this runtime:
  [spacy,thinc]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
!pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.1/en_core_sci_md-0.5.1.tar.gz
```

```
Collecting https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.1/en_core_sci_md-0.5.1.tar.gz
  Downloading https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.1/en_core_sci_md-0.5.1.tar.gz (120.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 120.2/120.2 MB 6.5 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Collecting spacy<3.5.0,>=3.4.1 (from en_core_sci_md==0.5.1)
  Downloading spacy-3.4.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (24 kB)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.10 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (1.0.12)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (2.0.11)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.0.9)
Collecting thinc<8.2.0,>=8.1.0 (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1)
  Using cached thinc-8.1.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (15 kB)
Collecting wasabi<1.1.0,>=0.9.1 (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1)
  Downloading wasabi-0.10.1-py3-none-any.whl.metadata (28 kB)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (2.5.1)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (2.0.10)
Requirement already satisfied: typer<0.8.0,>=0.3.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (0.7.0)
Requirement already satisfied: pathy>=0.3.5 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (0.11.0)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (6.4.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (4.67.1)
Requirement already satisfied: numpy>=1.15.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (1.26.4)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (2.32.3)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (1.
Requirement already satisfied: jinja2 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (24.2)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /usr/local/lib/python3.11/dist-packages (from spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.5.0)
Requirement already satisfied: language-data>=1.2 in /usr/local/lib/python3.11/dist-packages (from langcodes<4.0.0,>=3.2.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.
Requirement already satisfied: pathlib-abc==0.1.1 in /usr/local/lib/python3.11/dist-packages (from pathy>=0.3.5->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (0.1.1)
Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/lib/python3.11/dist-packages (from pydantic!=1.8,!=1.8.1,<1.11.0,>=1.7.4->spacy<3.5.0,>=3.4.1->
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->en_core_sci_md
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests<3.0.0,>=2.13.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /usr/local/lib/python3.11/dist-packages (from thinc<8.2.0,>=8.1.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.11/dist-packages (from thinc<8.2.0,>=8.1.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.
Requirement already satisfied: click<9.0.0,>=7.1.1 in /usr/local/lib/python3.11/dist-packages (from typer<0.8.0,>=0.3.0->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1)
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.11/dist-packages (from jinja2->spacy<3.5.0,>=3.4.1->en_core_sci_md==0.5.1) (3.0.2)
Requirement already satisfied: marisa-trie>=1.1.0 in /usr/local/lib/python3.11/dist-packages (from language-data>=1.2->langcodes<4.0.0,>=3.2.0->spacy<3.5.0,>=3.4.1->
Downloading spacy-3.4.4-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.4 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.4/6.4 MB 31.3 MB/s eta 0:00:00
Using cached thinc-8.1.12-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (917 kB)
Downloading wasabi-0.10.1-py3-none-any.whl (26 kB)
Building wheels for collected packages: en_core_sci_md
  Building wheel for en_core_sci_md (setup.py) ... done
  Created wheel for en_core_sci_md: filename=en_core_sci_md-0.5.1-py3-none-any.whl size=120253138 sha256=792bd41c7595fd08056d199a4eb1fd65f63750e39961e96c98e4bcb88d53
  Stored in directory: /root/.cache/pip/wheels/0a/50/82/7547d452aa8d5a653fb1271c38113de20f7842effc4b7313d0
Successfully built en_core_sci_md
Installing collected packages: wasabi, thinc, spacy, en_core_sci_md
  Attempting uninstall: wasabi
    Found existing installation: wasabi 1.1.3
    Uninstalling wasabi-1.1.3:
      Successfully uninstalled wasabi-1.1.3
  Attempting uninstall: thinc
    Found existing installation: thinc 8.2.5
    Uninstalling thinc-8.2.5:
      Successfully uninstalled thinc-8.2.5
  Attempting uninstall: spacy
    Found existing installation: spacy 3.7.5
    Uninstalling spacy-3.7.5:
      Successfully uninstalled spacy-3.7.5
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency c
scispacy 0.5.5 requires spacy<3.8.0,>=3.7.0, but you have spacy 3.4.4 which is incompatible.
Successfully installed en_core_sci_md-0.5.1 spacy-3.4.4 thinc-8.1.12 wasabi-0.10.1
WARNING: The following packages were previously imported in this runtime:
  [spacy,thinc,wasabi]
You must restart the runtime in order to use newly installed versions.
```

RESTART SESSION

```
import os
import requests
import xml.etree.ElementTree as ET
import spacy
import torch
import torch.nn as nn
from transformers import AutoModel, AutoTokenizer
```

```
import requests
from bs4 import BeautifulSoup

def download_arxiv_html(arxiv_html_url, save_path):
    response = requests.get(arxiv_html_url)
    response.raise_for_status()

    soup = BeautifulSoup(response.text, 'html.parser')
    for script in soup(['script', 'style']):
        script.extract()

    cleaned_html = soup.prettify()

    with open(save_path, 'w', encoding='utf-8') as file:
        file.write(cleaned_html)

def extract_abstract(html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    abstract_div = soup.find('div', class_='ltx_abstract')
    abstract = abstract_div.get_text(strip=True) if abstract_div else ""
    if abstract_div:
        abstract_div.extract()  # remove abstract from soup
    return abstract[8:], soup.prettify()




from bs4 import BeautifulSoup

# def extract_main_paper_from_html(html_content):
#     soup = BeautifulSoup(html_content, 'html.parser')
#     main_section = soup.find(id='S1')

#     if not main_section:
#         return ""

#     # Capture everything starting from the main section
#     main_paper_parts = []
#     current = main_section
#     while current:
#         main_paper_parts.append(str(current))
#         current = current.find_next_sibling()

#     return "\n".join(main_paper_parts)

from bs4 import BeautifulSoup

def extract_main_paper_text_from_html(html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    main_section = soup.find(id='S1')

    if not main_section:
        return ""

    # Extract plain text from main section and its siblings
    main_text_parts = []
    current = main_section
    while current:
        main_text_parts.append(current.get_text(separator=" ", strip=True))
        current = current.find_next_sibling()

    return "\n".join(main_text_parts)



import re
from nltk import WordNetLemmatizer
import html

import re
import html
from bs4 import BeautifulSoup
import nltk
nltk.download('stopwords')
from nltk.tokenize import sent_tokenize, word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string


def create_batch(papers, batch_size=4):
    """Create batch of papers for more efficient training"""
    paper_batches = []
    for i in range(0, len(papers), batch_size):
        paper_batches.append(papers[i:i+batch_size])
    return paper_batches
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

```python
import requests
import xml.etree.ElementTree as ET
from bs4 import BeautifulSoup


def extract_hrefs_from_url_by_title(domains=['cs.AI'], target_title='View HTML'):
    all_hrefs = []
    for domain in domains:
        url = f'https://arxiv.org/list/{domain}/recent?skip=0&show=2000'
        try:
            response = requests.get(url)
            response.raise_for_status()
            soup = BeautifulSoup(response.text, 'html.parser')
            for a_tag in soup.find_all('a', title=target_title):
                if 'href' in a_tag.attrs:
                    all_hrefs.append(a_tag['href'])
        except requests.exceptions.RequestException as e:
            print(f"Error fetching URL '{url}': {e}")
    return all_hrefs



def fetch_arxiv_ids(domains, max_results=5):
    """
    Fetches the ArXiv IDs of papers for the specified domains.

    Args:
        domains (list): A list of ArXiv subject categories (e.g., ["cs.AI", "physics.hep-th"]).
        max_results (int): The maximum number of results to fetch per domain (default: 100).

    Returns:
        list: A list of ArXiv paper IDs.
    """
    all_ids = []
    for domain in domains:
        url = f"http://export.arxiv.org/api/query?search_query=cat:{domain}&start=0&max_results={max_results}"
        response = requests.get(url)
        if response.status_code != 200:
            print(f"Error fetching data for domain: {domain}")
            continue

        root = ET.fromstring(response.text)
        for entry in root.findall("{http://www.w3.org/2005/Atom}entry"):
            # The ArXiv ID is typically found in the <id> tag.
            arxiv_id_full = entry.find("{http://www.w3.org/2005/Atom}id").text
            # The ID often looks like 'http://arxiv.org/abs/2304.01234v1'.
            # We want to extract just '2304.01234v1'.
            arxiv_id = arxiv_id_full.split('/')[-1]
            all_ids.append(arxiv_id)
    return all_ids


# Download PDF
def download_pdf(pdf_url, save_path="paper.pdf"):
    response = requests.get(pdf_url)
    if response.status_code == 200:
        with open(save_path, "wb") as f:
            f.write(response.content)
        return save_path
    return None

def extract_main_paper_from_html(html_content):
    soup = BeautifulSoup(html_content, 'html.parser')
    main_section = soup.find(id='S1')

    if not main_section:
        return ""

    # Capture everything starting from the main section
    main_paper_parts = []
    current = main_section
    while current:
        main_paper_parts.append(str(current))
        current = current.find_next_sibling()

    return "\n".join(main_paper_parts)


import re # Import regular expressions for cleaning

def get_body_by_id(html_content, target_id):
    """
    Extracts the *entire* inner HTML content of an element with a specific ID.
    (Kept for reference, but not used for the new requirement)

    Args:
        html_content (str): The HTML content to parse.
        target_id (str): The ID of the HTML element whose body content is to be extracted.

    Returns:
        str: The raw inner HTML content of the element, or None if the ID is not found.
             Returns an empty string if the element is found but has no content.

    Raises:
        TypeError: If html_content is not a string.
        TypeError: If target_id is not a string.
    """
    if not isinstance(html_content, str):
        raise TypeError("html_content must be a string.")
    if not isinstance(target_id, str):
        raise TypeError("target_id must be a string.")

    soup = BeautifulSoup(html_content, 'html.parser')
    element = soup.find(id=target_id)  # Find the element by its ID

    if element:
        return str(element.decode_contents())  # Return the raw inner HTML
    else:
        return None  # Return None if the element with the ID is not found

def extract_paragraph_text(html_content):
    """
    Extracts and cleans text content specifically from <p class="ltx_p"> tags within HTML.

    It ignores headings, links, citations, and other non-paragraph elements.
    It also cleans up citation markers like '[26, 11]' and extra whitespace.

    Args:
        html_content (str): The HTML content to parse.

    Returns:
        str: A single string containing the concatenated and cleaned text
             from all found <p class="ltx_p"> tags, separated by newlines.
             Returns an empty string if no such paragraphs are found.

    Raises:
        TypeError: If html_content is not a string.
    """
    if not isinstance(html_content, str):
        raise TypeError("html_content must be a string.")

    soup = BeautifulSoup(html_content, 'html.parser')
    paragraphs = soup.find_all('p', class_='ltx_p') # Find all <p> tags with class 'ltx_p'

    extracted_texts = []
    for p in paragraphs:
        # Get text, stripping inner tags like <a>, <cite>, <em>
        text = p.get_text(separator=' ', strip=True)

        # Use regex to remove citation markers like [26, 11] or [ 23 ]
        text = re.sub(r'\[\s*(\d+\s*,?\s*)+\]', '', text)

        # Optional: Clean up potential multiple spaces resulting from tag removal
        text = re.sub(r'\s+', ' ', text).strip()

        if text: # Add non-empty paragraphs
            extracted_texts.append(text)

    # Join the texts from all paragraphs with a newline for readability
    return "\n".join(extracted_texts)
```

```
import torch

def save_checkpoint(model, optimizer, epoch, loss, path="checkpoint.pt"):
    torch.save({
        'epoch': epoch,
        'model_state_dict': model.state_dict(),
        'optimizer_state_dict': optimizer.state_dict(),
        'loss': loss
    }, path)
    print(f"✅ Checkpoint saved at epoch {epoch} to {path}")


def load_checkpoint(model, optimizer, path="checkpoint.pt"):
    checkpoint = torch.load(path, map_location=torch.device('cuda' if torch.cuda.is_available() else 'cpu'))
    model.load_state_dict(checkpoint['model_state_dict'])
    optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
    print(f"📦 Loaded checkpoint from epoch {checkpoint['epoch']} with loss {checkpoint['loss']:.4f}")
    return checkpoint['epoch'], checkpoint['loss']

```

```
import torch.nn as nn
import torch.nn.functional as F

# First, make sure LuongAttention is defined
class LuongAttention(nn.Module):
    def __init__(self, hidden_dim):
        super(LuongAttention, self).__init__()
        self.attn = nn.Linear(hidden_dim, hidden_dim)

    def forward(self, decoder_hidden, encoder_outputs):
        # decoder_hidden: (batch, hidden)
        # encoder_outputs: (batch, seq_len, hidden)

        # Transform decoder hidden to match encoder dimension
        query = self.attn(decoder_hidden).unsqueeze(2)  # (batch, hidden, 1)

        # Compute scores (dot product)
        attn_scores = torch.bmm(encoder_outputs, query).squeeze(2)  # (batch, seq_len)

        # Softmax over time dimension
        attn_weights = F.softmax(attn_scores, dim=1)  # (batch, seq_len)

        # Weighted sum of encoder outputs
        context = torch.bmm(attn_weights.unsqueeze(1), encoder_outputs)  # (batch, 1, hidden)
        context = context.squeeze(1)  # (batch, hidden)

        return context, attn_weights




class Seq2Seq(nn.Module):
    def __init__(self, encoder, decoder, pad_token_id):
        super(Seq2Seq, self).__init__()
        self.encoder = encoder
        self.decoder = decoder
        self.pad_token_id = pad_token_id

    def forward(self, src_input_ids, src_attention_mask, tgt_input_ids):
        encoder_outputs, (hidden, cell) = self.encoder(src_input_ids, src_attention_mask)
        output = self.decoder(tgt_input_ids, hidden, cell, encoder_outputs)
        return output

    def generate(self, src_input_ids, src_attention_mask, max_len=100, bos_token_id=None, eos_token_id=None):
        """Generate sequence for inference"""
        if bos_token_id is None:
            bos_token_id = 1  # Default BOS token ID
        if eos_token_id is None:
            eos_token_id = 2  # Default EOS token ID

        device = src_input_ids.device
        batch_size = src_input_ids.size(0)

        # Get encoder outputs
        encoder_outputs, (hidden, cell) = self.encoder(src_input_ids, src_attention_mask)

        # Initialize decoder input with BOS token
        decoder_input = torch.tensor([[bos_token_id]] * batch_size, device=device)
        generated_sequence = [bos_token_id]

        # Generate tokens one by one
        for _ in range(max_len):
            # Generate one step
            next_token_logits, hidden, cell = self.decoder.generate_step(
                decoder_input, hidden, cell, encoder_outputs
            )
            next_token_id = torch.argmax(next_token_logits, dim=1).item()

            # Stop if EOS token is generated
            if next_token_id == eos_token_id:
                generated_sequence.append(next_token_id)
                break

            generated_sequence.append(next_token_id)
            decoder_input = torch.tensor([[next_token_id]], device=device)

        return generated_sequence


from torch.nn.utils.rnn import pad_sequence

import random

def split_papers(papers, train_ratio=0.8, shuffle=True):
    """
    Splits a list of paper URLs into training and testing sets.

    Args:
        papers (list): List of paper URLs.
        train_ratio (float): Ratio of training papers.
        shuffle (bool): Whether to shuffle the papers before splitting.

    Returns:
        (train_paper, test_paper): Tuple of two lists.
    """
    if shuffle:
        random.shuffle(papers)

    split_index = int(len(papers) * train_ratio)
    train_paper = papers[:split_index]
    test_paper = papers[split_index:]

    return train_paper, test_paper

```

```
!pip install tokenizers
```

```
Requirement already satisfied: tokenizers in /usr/local/lib/python3.11/dist-packages (0.21.1)
Requirement already satisfied: huggingface-hub<1.0,>=0.16.4 in /usr/local/lib/python3.11/dist-packages (from tokenizers) (0.30.1)
Requirement already satisfied: filelock in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (3.18.0)
Requirement already satisfied: fsspec>=2023.5.0 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (2025.3.2)
Requirement already satisfied: packaging>=20.9 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (24.2)
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (6.0.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (2.32.3)
Requirement already satisfied: tqdm>=4.42.1 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (4.67.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in /usr/local/lib/python3.11/dist-packages (from huggingface-hub<1.0,>=0.16.4->tokenizers) (4.13.1)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub<1.0,>=0.16.4->tokenizers) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub<1.0,>=0.16.4->tokenizers) (3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub<1.0,>=0.16.4->tokenizers) (2.3.0)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-packages (from requests->huggingface-hub<1.0,>=0.16.4->tokenizers) (2025.1.31)
```

```python
# papers = extract_hrefs_from_url_by_title()
# papers = papers[1:2]

# for link in papers:
#     save_as = 'paper.html'
#     download_arxiv_html(link, save_as)

#     with open(save_as, 'r', encoding='utf-8') as file:
#         html_data = file.read()

#     abstract, html_without_abstract = extract_abstract(html_data)
#     main_paper = extract_main_paper_from_html(html_without_abstract)
#     cleaned_abstract = preprocess_paper_text(abstract)['text']
#     cleaned_paper = preprocess_paper_text(main_paper)['text']
#     print(cleaned_paper[1000:2000])
```

```python
MODEL_DIR = "Downloads/NLP_Local"
CHECKPOINT_PATH = f"{MODEL_DIR}/checkpoint.pt"
TOKENIZER_PATH = f"{MODEL_DIR}/mytokenizer"
PICKLE_PATH = f"{MODEL_DIR}/model.pkl"

import os
os.makedirs(MODEL_DIR, exist_ok=True)


def load_checkpoint(model, optimizer, path="checkpoint.pt"):
    checkpoint = torch.load(path, map_location=torch.device('cuda' if torch.cuda.is_available() else 'cpu'))

    model.load_state_dict(checkpoint['model_state_dict'])
    optimizer.load_state_dict(checkpoint['optimizer_state_dict'])
    epoch = checkpoint['epoch']
    loss = checkpoint['loss']

    print(f"✅ Loaded checkpoint from epoch {epoch} with loss {loss:.4f}")
    return epoch, loss
```

```python
import spacy
import torch
import torch.nn as nn
import numpy as np

# Load scispaCy model - you'll need to install it first with:
# pip install scispacy
# pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.1/en_core_sci_md-0.5.1.tar.gz
try:
    nlp = spacy.load("en_core_sci_md")
    nlp.max_length = 2000000
except OSError:
    print("Please install the scispaCy model with:")
    print("pip install scispacy")
    print("pip install https://s3-us-west-2.amazonaws.com/ai2-s2-scispacy/releases/v0.5.1/en_core_sci_md-0.5.1.tar.gz")
    raise

from transformers import AutoTokenizer

def tokenize_text(text):
    doc = nlp(text)
    tokens = [token.text for token in doc if not token.is_stop and token.is_alpha]
    return tokens

custom_tokenizer = AutoTokenizer.from_pretrained("allenai/scibert_scivocab_uncased")

custom_tokenizer.bos_token = custom_tokenizer.cls_token  # Use [CLS] as BOS
custom_tokenizer.eos_token = custom_tokenizer.sep_token

class ScispacyEncoder(nn.Module):
    def __init__(self, embedding_dim, hidden_dim, num_layers=1, max_sequence_length=50):
        super(ScispacyEncoder, self).__init__()
        # ScispaCy model has 200-dim embeddings
        self.embedding_dim = 200
        self.max_sequence_length = max_sequence_length
        self.projection = nn.Linear(self.embedding_dim, embedding_dim)
        self.lstm = nn.LSTM(embedding_dim, hidden_dim, num_layers, batch_first=True)

    def get_embeddings_sequence(self, text):
        """Get ScispaCy embeddings for text as a sequence"""
        doc = nlp(text)

        # Extract important sentences to create a meaningful sequence
        # Use basic frequency-based approach to identify key sentences
        from collections import Counter

        # Count word frequencies (excluding stop words)
        word_freq = Counter([token.text.lower() for token in doc
                            if not token.is_stop and not token.is_punct
                            and token.has_vector])

        # Score sentences by sum of word frequencies
        sentences = list(doc.sents)
        sentence_scores = []
        for sent in sentences:
            score = sum(word_freq[token.text.lower()] for token in sent
                      if token.has_vector and not token.is_stop and not token.is_punct)
            sentence_scores.append((sent, score))

        # Take top sentences up to max_sequence_length
        top_sentences = sorted(sentence_scores, key=lambda x: x[1], reverse=True)[:self.max_sequence_length]
        # Re-sort to preserve original order
        top_sentences = sorted(top_sentences, key=lambda x: sentences.index(x[0]))

        # Get vector for each sentence
        sequence_vectors = []
        for sent, _ in top_sentences:
            vectors = [token.vector for token in sent if token.has_vector]
            if vectors:
                mean_vector = np.mean(vectors, axis=0)
                # Ensure vector has correct dimensions
                if mean_vector.shape[0] != self.embedding_dim:
                    if mean_vector.shape[0] > self.embedding_dim:
                        mean_vector = mean_vector[:self.embedding_dim]  # Truncate
                    else:
                        # Pad with zeros
                        padded = np.zeros(self.embedding_dim)
                        padded[:mean_vector.shape[0]] = mean_vector
                        mean_vector = padded
                sequence_vectors.append(mean_vector)
            else:
                # Use zeros for sentences with no valid vectors
                sequence_vectors.append(np.zeros(self.embedding_dim))

        # Pad or truncate sequence to match max_sequence_length
        if len(sequence_vectors) > self.max_sequence_length:
            sequence_vectors = sequence_vectors[:self.max_sequence_length]
        elif len(sequence_vectors) < self.max_sequence_length:
            padding_needed = self.max_sequence_length - len(sequence_vectors)
            for _ in range(padding_needed):
                sequence_vectors.append(np.zeros(self.embedding_dim))

        return torch.tensor(np.array(sequence_vectors), dtype=torch.float)

    def forward(self, texts, attention_mask=None):
        batch_size = len(texts)
        embedded_sequences = []

        # Process each text in the batch
        for text in texts:
            # Get sequence of embeddings from ScispaCy
            seq_embedding = self.get_embeddings_sequence(text)  # [seq_len, embed_dim]

            # Project each vector to desired embedding dimension
            projected_seq = self.projection(seq_embedding)  # [seq_len, embed_dim]
            embedded_sequences.append(projected_seq)

        # Stack embeddings
        embedded = torch.stack(embedded_sequences, dim=0)  # [batch, seq_len, embed_dim]

        # Process through LSTM
        outputs, (h, c) = self.lstm(embedded)

        return outputs, (h, c)

class ScispacyDecoder(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim, num_layers=1):
        super(ScispacyDecoder, self).__init__()
        self.embedding = nn.Embedding(vocab_size, embedding_dim)
        self.lstm = nn.LSTM(embedding_dim, hidden_dim, num_layers, batch_first=True)
        self.attention = LuongAttention(hidden_dim)  # Add attention mechanism
        # Combine context and hidden for output
        self.fc_out = nn.Linear(hidden_dim * 2, hidden_dim)
        self.output_layer = nn.Linear(hidden_dim, vocab_size)

    def forward(self, tgt_input_ids, hidden, cell, encoder_outputs):
        # [B, L] -> [B, L, D]
        embedded = self.embedding(tgt_input_ids)

        # Pass through LSTM
        outputs, (hidden, cell) = self.lstm(embedded, (hidden, cell))

        # Apply attention for each timestep
        batch_size, seq_len, _ = outputs.size()
        attention_outputs = []

        for t in range(seq_len):
            # Get decoder hidden state at this timestep
            decoder_hidden = outputs[:, t, :]

            # Calculate attention context
            context, _ = self.attention(decoder_hidden, encoder_outputs)

            # Combine context and hidden state
            concat_input = torch.cat((decoder_hidden, context), dim=1)
            output = self.fc_out(concat_input)
            attention_outputs.append(output)

        # Stack attention outputs
        attention_outputs = torch.stack(attention_outputs, dim=1)

        # Get logits
        logits = self.output_layer(attention_outputs)
        return logits

    def generate_step(self, decoder_input, hidden, cell, encoder_outputs):
        # [1, 1] -> [1, 1, D]
        embedded = self.embedding(decoder_input)

        # Pass through LSTM for one step
        outputs, (hidden, cell) = self.lstm(embedded, (hidden, cell))

        # Apply attention
        decoder_hidden = outputs[:, -1, :]
        context, _ = self.attention(decoder_hidden, encoder_outputs)

        # Combine context and hidden state
        concat_input = torch.cat((decoder_hidden, context), dim=1)
        attention_output = self.fc_out(concat_input)

        # Get logits for the next token
        logits = self.output_layer(attention_output)
        return logits, hidden, cell


# Updated encoder-decoder architecture
class SciSummarizationModel(nn.Module):
    def __init__(self, vocab_size, embedding_dim, hidden_dim, num_layers=1):
        super(SciSummarizationModel, self).__init__()
        self.encoder = ScispacyEncoder(embedding_dim, hidden_dim, num_layers)
        # Use the improved decoder with attention
        self.decoder = ScispacyDecoder(vocab_size, embedding_dim, hidden_dim, num_layers)
        self.pad_token_id = custom_tokenizer.pad_token_id

    def forward(self, source_texts, tgt_input_ids):
        encoder_outputs, (hidden, cell) = self.encoder(source_texts)
        output = self.decoder(tgt_input_ids, hidden, cell, encoder_outputs)
        return output

    def generate(self, source_text, max_len=100, bos_token_id=None, eos_token_id=None):
        """Generate sequence for inference"""
        device = next(self.parameters()).device

        # Use tokenizer's CLS/SEP tokens if BOS/EOS are not available
        if bos_token_id is None:
            bos_token_id = custom_tokenizer.cls_token_id  # [CLS] token in BERT
        if eos_token_id is None:
            eos_token_id = custom_tokenizer.sep_token_id  # [SEP] token in BERT

        # Get encoder outputs
        encoder_outputs, (hidden, cell) = self.encoder([source_text])

        # Initialize decoder input with BOS token
        decoder_input = torch.tensor([[bos_token_id]], device=device)
        generated_sequence = [bos_token_id]

        # Generate tokens one by one
        for _ in range(max_len):
            # Use the modified decoder.generate_step method
            next_token_logits, hidden, cell = self.decoder.generate_step(
                decoder_input, hidden, cell, encoder_outputs
            )
            next_token_id = torch.argmax(next_token_logits, dim=1).item()

            # Stop if EOS token is generated
            if next_token_id == eos_token_id:
                generated_sequence.append(next_token_id)
                break

            generated_sequence.append(next_token_id)
            decoder_input = torch.tensor([[next_token_id]], device=device)

        return generated_sequence


# Function to preprocess text using ScispaCy
import re

def preprocess_with_scispacy(text):
    # Remove URLs
    text = re.sub(r'http\S+|www\.\S+', '', text)

    doc = nlp(text)
    cleaned_tokens = []

    for token in doc:
        token_text = token.text

        # Skip stopwords or punctuations
        if token.is_stop or token.is_punct:
            continue

        # Remove wrapped in {}, [], ()
        if re.match(r'^[\[\(\{].*[\]\)\}]$', token_text):
            continue

        # Remove tokens with slashes or backslashes
        if '/' in token_text or '\\' in token_text:
            continue

        # Remove tokens that contain non-alphanumeric characters
        if not token_text.isalnum():
            continue

        # Append cleaned, lemmatized lowercase word
        cleaned_tokens.append(token.lemma_.lower())

    return " ".join(cleaned_tokens)





# Updated summarize_paper function
def summarize_paper_with_scispacy(model, tokenizer, link, max_summary_len=100):
    device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
    model.to(device)
    model.eval()

    # Download and preprocess the paper
    save_as = 'test_paper.html'
    download_arxiv_html(link, save_as)

    with open(save_as, 'r', encoding='utf-8') as file:
        html_data = file.read()

    abstract, html_without_abstract = extract_abstract(html_data)
    main_paper = extract_paragraph_text(html_without_abstract)

    # Preprocess with ScispaCy
    cleaned_paper = preprocess_with_scispacy(main_paper)
    print(f"Clean paper input: {cleaned_paper}")
    # Generate summary
    with torch.no_grad():
        generated_ids = model.generate(cleaned_paper, max_len=max_summary_len)

    # Decode
    valid_ids = [token_id for token_id in generated_ids if token_id < tokenizer.vocab_size]
    summary_text = tokenizer.decode(valid_ids, skip_special_tokens=True,clean_up_tokenization_spaces=True)

    print("📄 Original Paper Length:", len(main_paper.split()))
    print("📝 Generated Abstract:", summary_text)
    print("-" * 60)

    return summary_text
```

```
# Model parameters
import torch.nn as nn
loss_fn = nn.CrossEntropyLoss()
vocab_size = len(custom_tokenizer.vocab)
embedding_dim = 256  # Increased from 128
hidden_dim = 256     # Increased from 128
num_epochs = 1

papers = extract_hrefs_from_url_by_title()
papers = papers[1:10]
train_paper,test_paper = split_papers(papers)
device = torch.device('cpu')
# Create model

special_tokens_dict = {'bos_token': '<s>', 'eos_token': '</s>'}
num_added_toks = custom_tokenizer.add_special_tokens(special_tokens_dict)

# Resize model embeddings
sci_model = SciSummarizationModel(vocab_size, embedding_dim, hidden_dim).to(device)
optimizer = torch.optim.Adam(sci_model.parameters(), lr=3e-4)

# Train the model (simplified example)
for epoch in range(num_epochs):
    for link in train_paper:
        # Get paper data (same as before)
        save_as = 'paper.html'
        download_arxiv_html(link, save_as)

        with open(save_as, 'r', encoding='utf-8') as file:
            html_data = file.read()
        abstract, html_without_abstract = extract_abstract(html_data)
        main_paper = extract_paragraph_text(html_without_abstract)

        # Use ScispaCy preprocessing
        cleaned_paper = preprocess_with_scispacy(main_paper)

        print(f"abstract {abstract}")
        print("=============================================")
        # Tokenize abstract for target
        encoded_abstract = custom_tokenizer(
            abstract,
            padding='max_length',
            truncation=True,
            max_length=128,
            return_tensors='pt',
            add_special_tokens=True
        ).to(device)

        # Tokenize cleaned_paper for source input
        print(f"Inout clean {cleaned_paper}")

        # Decoder input/output setup
        decoder_input = encoded_abstract.input_ids[:, :-1]  # exclude last token
        target_labels = encoded_abstract.input_ids[:, 1:]   # exclude first token

        # Forward pass with paper text directly
        output_logits = sci_model([cleaned_paper], decoder_input)

        # Calculate loss
        loss = loss_fn(output_logits.view(-1, vocab_size), target_labels.view(-1))

        # Backward pass
        optimizer.zero_grad()
        loss.backward()
        torch.nn.utils.clip_grad_norm_(sci_model.parameters(), 1.0)
        optimizer.step()

        print(f"Epoch {epoch+1}, Loss: {loss.item():.4f}")
```

```
abstract We propose the Dual Engines of Thoughts (DEoT), an analytical framework for comprehensive open-ended reasoning. While traditional reasoning frameworks primari
=============================================
Inout clean keywords dual engines thoughts analysis framework reasoning framework today interconnected world analyze implication complex event require nuanced grasp in
Epoch 1, Loss: 10.3460
abstract Large Language Models (LLMs) demonstrate impressive capabilities in natural language processing but suffer from inaccuracies and logical inconsistencies known
=============================================
Inout clean keyword llm ontology reasoning consistency checking knowledge representation hallucination mitigation hybrid machine learning logical formalism large langu
Epoch 1, Loss: 10.3557
abstract We demonstrate how AI agents can coordinate to deceive oversight systems using automated interpretability of neural networks.
Using sparse autoencoders (SAEs) as our experimental framework, we show that language models (Llama, DeepSeek R1, and Claude 3.7 Sonnet) can generate deceptive explana
Our agents employ steganographic methods to hide information in seemingly innocent explanations, successfully fooling oversight models while achieving explanation qual
We further find that models can scheme to develop deceptive strategies when they believe the detection of harmful features might lead to negative consequences for them
All tested LLM agents were capable of deceiving the overseer while achieving high interpretability scores comparable to those of reference labels.
We conclude by proposing mitigation strategies, emphasizing the critical need for robust understanding and defenses against deception.
=============================================
Inout clean sparse autoencoder sae neural network large number neuron use sparsity constraint training call autoencoder approximate identity function ng et 2011 contai
Epoch 1, Loss: 10.3406
abstract Aligning large language models with human preferences is crucial for their safe deployment. While Direct Preference Optimization (DPO) offers an efficient alt
=============================================
Inout clean align large language models llm carefully curate human feedback prove critical steer behavior helpful honest harmless response preference optimization meth
Epoch 1, Loss: 10.3329
abstract A popular approach to neurosymbolic AI is to take the output of the last layer of a neural network, e.g. a softmax activation, and pass it through a sparse co
This induces a probability distribution over a set of random variables, which happen to be conditionally independent of each other in many commonly used neurosymbolic
```

```
Such conditionally independent random variables have been deemed harmful as their presence has been observed to co-occur with a phenomenon dubbeddeterministic bias, wh
We provide evidence contesting this conclusion and show that the phenomenon ofdeterministic biasis an artifact of improperly applying neurosymbolic AI.
=============================================
Inout clean neurosymbolic nesy ai approach ai seek combine logic neural network integration symbolic method allow inter alia interpretable datum efficient ai system po
Epoch 1, Loss: 10.3469
abstract AlphaZero in 2017 was able to master chess and other games without human knowledge by playing millions of games against itself (self-play), with a computation
=============================================
Inout clean conquer chess holy grail testbe ai development inception supercomputer deep blue ai system beat world champion chess classical time control development har
Epoch 1, Loss: 10.3319
abstract Generative AI is transforming computing education by enabling the automatic generation of personalized content and feedback. We investigate its capabilities i
=============================================
Inout clean generative ai transform learning teaching compute education advanced generative model openai github copilot reshape student teacher experience student mode
Epoch 1, Loss: 10.3236
```

```
# Test your model
for link in test_paper[:3]:
    summary = summarize_paper_with_scispacy(sci_model, custom_tokenizer, link)
```

```
ean paper input: linecolor gray topline false bottomline false leftline true rightline false backgroundcolor giovanni mauro 1 moruzzi 1 pisa 56124 italy 2 scuola normal
 Original Paper Length: 10641
 Generated Abstract: pancreatic reaction attractiveness academic commission diesel its its formulations heterologous professorulin transientmentationlmife initiation},
-------------------------------------------------------
ean paper input: scheduling problem exist dynamic environment unpredictable event unforeseen machine failure arrival urgent job date alteration unexpected weather chang
 Original Paper Length: 11077
 Generated Abstract: pancreatic reaction attractiveness academic commission diesel its its formulations heterologous professorulin transientmentationlmife initiation},
-------------------------------------------------------
```

```
Start coding or generate with AI.
```