[DM 2024] Lab 2 Environment settings

Hi everybody,

We will have our second lab session on October 28 (Monday) 9:00 am on our Youtube channel stream: DM Youtube Channel. Please be on time.

We highly recommend you to attend the session with your personal laptop (that way you'll also have your environment set for the homework). These are some instructions for you to set up the environment:

1. Install libraries:

We will use some new Python libraries for the lab: Gensim, Tensorflow and Keras.

Once you have installed Python 3 (and optionally Anaconda), open a "terminal" windows (Linux/MacOS) or a "Command Prompt" window and type the following commands followed by "Enter":

```
pip3 install gensim
pip3 install tensorflow
pip3 install tensorflow-hub
pip3 install keras
pip3 install ollama
pip3 install langchain
pip3 install langchain_community
pip3 install langchain_core
pip3 install beautifulsoup4
pip3 install chromadb
pip3 install gradio
```

---

2. Install Ollama in your device:

We will be using some small open-source LLMs that will be running in your device, and for that we will be using Ollama, please enter the website, download and install it: Ollama website And don't worry, if needed, **Ollama can also be run online through Kaggle or Google Colab**, we will be discussing that later on.

After the installation is done, go to your terminal and type: **ollama** You should be getting the following information if the installation was correct:

```
PS C:\Users\didif> ollama
Usage:
  ollama [flags]
  ollama [command]

Available Commands:
  serve       Start ollama
  create      Create a model from a Modelfile
  show        Show information for a model
  run         Run a model
  stop        Stop a running model
  pull        Pull a model from a registry
  push        Push a model to a registry
  list        List models
  ps          List running models
  cp          Copy a model
  rm          Remove a model
  help        Help about any command

Flags:
  -h, --help      help for ollama
  -v, --version   Show version information

Use "ollama [command] --help" for more information about a command.
PS C:\Users\didif>
```

We will be using 3 Open-source LLMs during this lab, for that it is recommended to have at least **4 GB of VRAM, 16 GB of RAM and multi-core processor** to run them in the **most optimal way**, although they can be run with less computing resources, but they will be slower in response. To download and install them you will need to type the following commands in the terminal:

- ollama run llama3.2
  - Model with **3 billion parameters.**
- ollama run llama3.2:1b
  - This is just in case the first one is too slow in your device, it is a smaller model of **1 billion parameters.**
- ollama run llava-phi3
  - Model with **3 billion parameters.**

Just for reference, GPT-4 from OpenAI has **1.8 trillion parameters**, so these are just very small models in comparison.

After you run one of the commands the model will start to download in this way:

```
PS C:\Users\didif> ollama run llama3.2:1b
pulling manifest
pulling 74701a8c35f6... 100%                    1.3 GB
pulling 966de95ca8a6... 100%                    1.4 KB
pulling fcc5a6bec9da... 100%                    7.7 KB
pulling a70ff7e570d9... 100%                    6.0 KB
pulling 4f659a1e86d7... 100%                     485 B
verifying sha256 digest
writing manifest
success
>>> Send a message (/? for help)
```

So download and install all of the models **one by one.**

After finishing you can verify each model by asking something in a prompt in the terminal:

```
Windows PowerShell
PS C:\Users\didif> ollama run llama3.2
>>> What is data mining?
Data mining is the process of automatically discovering patterns, relationships, and insights from large datasets
to gain a better understanding of the data. It involves using statistical and mathematical techniques to identify
hidden or unexpected patterns in the data, which can help organizations make informed decisions.

The primary goal of data mining is to extract valuable knowledge and insights from large datasets, often referred
to as "big data." This knowledge can be used to improve business operations, customer behavior, predictive models,
and more.

Some common techniques used in data mining include:

1. Clustering: grouping similar data points together
2. Regression analysis: predicting continuous values based on independent variables
3. Decision trees: creating a tree-like model of decisions and outcomes
4. Association rule learning: identifying relationships between variables
5. Neural networks: using artificial neural networks to classify or predict data

Data mining can be used in various applications, such as:

1. Market research: analyzing customer behavior and preferences
2. Predictive analytics: forecasting sales, demand, or other business outcomes
3. Customer segmentation: grouping customers based on demographic or behavioral characteristics
4. Anomaly detection: identifying unusual patterns or outliers in the data

To perform data mining effectively, organizations need to have a large and diverse dataset, as well as access to
advanced analytical tools and techniques.

Types of Data Mining:

1. Descriptive data mining: analyzing existing data to understand patterns and trends.
2. Diagnostic data mining: using historical data to identify problems or areas for improvement.
3. Predictive data mining: forecasting future events or behaviors based on past data.
4. Prescriptive data mining: providing recommendations for actions or decisions based on the analysis.

Benefits of Data Mining:

1. Improved decision-making
2. Enhanced customer understanding and satisfaction
3. Increased operational efficiency
4. Better resource allocation

Challenges of Data Mining:

1. Data quality issues
2. Insufficient training data
3. Complexity of large datasets
4. Interpreting results effectively.

Overall, data mining is a powerful tool for organizations to extract valuable insights from their data and make
informed decisions.
```

---

3. Run Jupyter Python and check your environment:

Open a new Jupyter notebook server. In order to do this, open a "terminal" windows (Linux/MacOS) or a "Command Prompt" window and type the following commands followed by "Enter":

```
jupyter notebook
```

If you receive an error message, zsh: command not found: jupyter, type the following commands instead.

```
python3 -m notebook
```

or

```
python -m notebook
```

Just like the image below:

```
PS C:\Users\didif> python -m notebook
[I 02:18:24.470 NotebookApp] Serving notebooks from local directory: C:\Users\didif
[I 02:18:24.471 NotebookApp] Jupyter Notebook 6.4.12 is running at:
[I 02:18:24.471 NotebookApp] http://localhost:8888/?token=f38efb70ec65133abc810c511d61c8283aa2ecc3ac659dd0
[I 02:18:24.471 NotebookApp]  or http://127.0.0.1:8888/?token=f38efb70ec65133abc810c511d61c8283aa2ecc3ac659dd0
[I 02:18:24.471 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 02:18:24.510 NotebookApp]

    To access the notebook, open this file in a browser:
        file:///C:/Users/didif/AppData/Roaming/jupyter/runtime/nbserver-5292-open.html
    Or copy and paste one of these URLs:
        http://localhost:8888/?token=f38efb70ec65133abc810c511d61c8283aa2ecc3ac659dd0
     or http://127.0.0.1:8888/?token=f38efb70ec65133abc810c511d61c8283aa2ecc3ac659dd0
```

A window like the one below should open in your browser. Please go to the "New" button on the top right cornerand select "Python 3".

This will open a new notebook. You will be able to run "Cells" of code and get the outputs printed below, as well as cells of text. If you want to learn more on how to use a notebook, read the documentation below:

https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Running%20Code.html

https://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Notebook%20Basics.html

Once you opened a new notebook, please paste the script below in a cell and press the "Run" Button (or the "Shift" + "Enter" keys). Make sure you have no errors!

In [3]:
```python
# import library
import pandas as pd
import numpy as np
import nltk
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import umap
import gensim
import tensorflow
import keras
import ollama
import langchain
import langchain_community
import langchain_core
import bs4
import chromadb
import gradio

%matplotlib inline

print("gensim: " + gensim.__version__)
print("tensorflow: " + tensorflow.__version__)
print("keras: " + keras.__version__)
```

```
gensim: 4.3.3
tensorflow: 2.17.0
keras: 3.6.0
```

It should look similar to this:

In [3]:
```python
# import library
import pandas as pd
import numpy as np
import nltk
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import umap
import gensim
import tensorflow
import keras
import ollama
import langchain
import langchain_community
import langchain_core
import bs4
import chromadb
import gradio

%matplotlib inline

print("gensim: " + gensim.__version__)
print("tensorflow: " + tensorflow.__version__)
print("keras: " + keras.__version__)
```

```
gensim: 4.3.3
tensorflow: 2.17.0
keras: 3.6.0
```

---

4. Download a pre-trained Word2Vec model

If you are using a Jupyter Notebook or Google Colab, you can download the Google News Vector: https://code.google.com/archive/p/word2vec/

If you are using Kaggle Kernel, you can add this dataset to your notebook: https://www.kaggle.com/datasets/didiersalazar/google-news-vectors

---

Note:

If you have issues to install some libraries (For example, Keras is not fully compatible with Macbook M1 and M1 Pro), please try using Google Colab (https://colab.research.google.com/) or Kaggle (https://www.kaggle.com/).

Step 1: Create a notebook and install the following scripts and Ollama

In [ ]:
```
!pip3 install scikit-learn --upgrade
!pip3 install pandas --upgrade
!pip3 install numpy --upgrade
!pip3 install matplotlib --upgrade
!pip3 install plotly --upgrade
!pip3 install seaborn --upgrade
!pip3 install nltk --upgrade
!pip3 install umap-learn --upgrade

!pip3 install gensim --upgrade
!pip3 install tensorflow --upgrade
!pip3 install keras --upgrade

!pip3 install ollama --upgrade
!pip3 install langchain --upgrade
!pip3 install langchain_community --upgrade
!pip3 install langchain_core --upgrade
!pip3 install beautifulsoup4 --upgrade
!pip3 install chromadb --upgrade
!pip3 install gradio --upgrade
```

In [ ]:
```
#Download ollama
!curl -fsSL https://ollama.com/install.sh | sh
import subprocess
process = subprocess.Popen("ollama serve", shell=True) #runs on a different thread
```

In [ ]:
```
#Download model llama 3.2
!ollama pull llama3.2
```

In [ ]:
```
#Download model llama 3.2:1b
!ollama pull llama3.2:1b
```

In [ ]:
```
#Download model llava-phi3
!ollama pull llava-phi3
```

If you are using Kaggle, skip to Step 2.

If you are using Google Colab, after installing, in the output you might see a warning.

You need to restart the runtime in order to use newly installed versions. Press the **"RESTART RUNTIME"** button.

Step 2: Run the following script

In [ ]:
```python
# import library
import pandas as pd
import numpy as np
import nltk
import matplotlib.pyplot as plt
import seaborn as sns
import itertools
import umap
import gensim
import tensorflow
import keras
import ollama
import langchain
import langchain_community
import langchain_core
import bs4
import chromadb
import gradio

%matplotlib inline

print("gensim: " + gensim.__version__)
print("tensorflow: " + tensorflow.__version__)
print("keras: " + keras.__version__)
```

The output should look similar to the previous image as well, without any problem and showing the libraries version.

Step 3: Prepare the files

Google Colab

In this lab, we will need to import some txt files as our data. If you are using Google Colab, you can import the files by following the instructions below:

- Try to copy this version of the lab in colab and run it: Lab-2 Colab

You can also try to mount the environment in this way:

- First download the ZIP of the DM2024-Lab2-Master, unzip it and upload the entire folder to your Google Drive (simply by dragging the folder to Google Drive). After that, you can follow this guide to mount your Google Drive on your runtime and access the files.
- Assuming you put the unzipped "DM2024-Lab2-Master" folder in the first layer of Google Drive, here is how you will need to slightly modify the codes in Section 1.1 "Load data" in order to load the data.

```python
[122] from google.colab import drive
      drive.mount('/content/drive')

Mounted at /content/drive

import pandas as pd

### training data
anger_train = pd.read_csv("/content/drive/My Drive/DM2024-Lab2-Master/data/semeval/train/anger-ratings-0to1.train.txt",
                          sep="\t", header=None,names=["id", "text", "emotion", "intensity"])
sadness_train = pd.read_csv("/content/drive/My Drive/DM2024-Lab2-Master/data/semeval/train/sadness-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
fear_train = pd.read_csv("/content/drive/My Drive/DM2024-Lab2-Master/data/semeval/train/fear-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
joy_train = pd.read_csv("/content/drive/My Drive/DM2024-Lab2-Master/data/semeval/train/joy-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
```

Kaggle

If you are using Kaggle, you can directly copy and edit this notebook: https://www.kaggle.com/code/didiersalazar/dm2024-lab2-master

The file path should be correct. However, you can double check by running the cells. If you don't see any error, then you are good to go.

1.1 Load data

We start by loading the csv files into a single pandas dataframe for training and one for testing.

[6]:
```python
import pandas as pd

### training data
anger_train = pd.read_csv("../input/lab2-dataset/data/semeval/train/anger-ratings-0to1.train.txt",
                          sep="\t", header=None,names=["id", "text", "emotion", "intensity"])
sadness_train = pd.read_csv("../input/lab2-dataset/data/semeval/train/sadness-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
fear_train = pd.read_csv("../input/lab2-dataset/data/semeval/train/fear-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
joy_train = pd.read_csv("../input/lab2-dataset/data/semeval/train/joy-ratings-0to1.train.txt",
                          sep="\t", header=None, names=["id", "text", "emotion", "intensity"])
```

Also, please make sure the computer you will use during the lab session can work before the lab!

Important Note: If you're having installation issues with all of this, please ask your classmates or TAs for help well ahead of the lab session.

Good luck and see you on Monday!

Best regards,
The TAs