Assignment 6: Clustering and Topic Modeling

In this assignment, you'll need to use the following dataset:

- text_train.json: This file contains a list of documents. It's used for training models
- text_test.json: This file contains a list of document and labels of each document. It's used for testing performance. This file is in the format shown below. Note, each document has a list of labels.

Text	**Labels**
faa issues fire warning for lithium ...	[T1, T3]
rescuers pull from flooded coal mine ...	[T1]
....	...

Q1: K-Mean Clustering

Define a function **cluster_kmean()** as follows:

- Takes two file name strings as inputs: $train\_file$ is the file path of text_train.json, and $test\_file$ is the file path of text_test.json
- Uses **KMeans** to cluster documents in both $train\_file$ and $test\_file$ into 3 clusters by **cosine similarity**
- Tests the clustering model performance using $test\_file$:
    - Let's only use the **first label** in the label list of each test document as the ground_truth label, e.g. the first document in the table above will have the ground_truth label "T1".
    - Apply **majority vote** rule to map the clusters to the labels in $test\_file$, i.e., T1, T2, T3
    - Calculate **precision/recall/f-score** for each label
    - Check centroids/samples in each cluster to interpret it, and give a **meaningful name** (instead of T1, T2, T3) to it.
- This function has no return. Print out precision/recall/f-score. Write down the meaningful cluster names in a document. Also find one document sample from $train\_file$ for each cluster in the doucment.

Q2: LDA Clustering

Define a function **cluster_lda()** as follows:

- Takes two file name strings as inputs: $train\_file$ is the file path of text_train.json, and $test\_file$ is the file path of text_test.json
- Uses **LDA** to train a topic model with documents in $train\_file$ and the number of topics $K = 3$
- Predicts the topic distribution of each document in $test\_file$, and selects only the **top one topic** (i.e. the topic with highest probability)
- Evaluates the topic model performance using topic prediction from documents in $test\_file$:
  - Let's use the **first label** in the label list of each test document as the ground_truth label, e.g. the first document in the table above will have the ground_truth label "T1".
  - Apply **majority vote rule** to map the topics to the labels in $test\_file$, i.e., T1, T2, T3
  - Calculate **precision/recall/f-score** for each label
  - Based on the **word distribution of each topic**, give the topic a **meaningful name** (instead of T1, T2, T3).
- This function has no return. Print out precision/recall/f-score. Also, provide a document which contains:
  - the meaningful topic names
  - one document sample from $train\_file$ for each topic
  - performance comparison between Q1 and Q2.

Q3 (Bonus): LDA Parameter Tunning

Define a function **tune_lda()** as follows:

- Takes two file name strings as inputs: $train\_file$ is the file path of text_train.json, and $test\_file$ is the file path of text_test.json
- Fits **LDA** models (from gensim package) using documents from $train\_file$ with different parameter values:
  - **Number of clusters** (K) from 2 to 6
  - **Topic distribution prior** (i.e. α): 'symmetric' (i.e. $\alpha = [1, 1, 1, \ldots]$), 'asymmetric' (i.e. $\alpha = [1/K, 1/K, 1/K, \ldots]$), and 'auto' (i.e. the prior is calculated based on word frequency)
- With all parameter combinations, in total, you'll train **15** LDA models. When fitting each model, set the maximum number of iterations to 40 to make sure your model converges. Note, it may take a few minutes to train all models.
- For each model, calculate **topic coherence** using 'u_mass' formula. The details of coherence can be found at [https://radimrehurek.com/gensim/models/coherencemodel.html (https://radimrehurek.com/gensim/models/coherencemodel.html)](https://radimrehurek.com/gensim/models/coherencemodel.html). Read the paper referenced in the link to make sure you understand the meaning of topic coherence. Note, 'c_v' instead of 'u_mass' is recommended to evaluate topic coherence. For simplicity, let's use 'u_mass' here. However, if you can figure out how to use 'c_v', that's even better.
- Create a plot to show how topic coherence changes as the K increases under different α values (i.e., a line for each α value).
- Based on the plot, determine best K and α values in terms of topic coherence
- This function does not have a return. Write a document to show:
  - best parameter combination in terms of topic coherence
  - do you think topic coherence is a good metric for you to choose K?

In [ ]:

```

```