PRECOG RECRUITMENT TASK

Tasks are centered around three central themes

- NLP
- Computer Vision
- Graphs

Applicants must choose **any one** theme and complete the task listed under that theme. Task consists of two parts - a programming task and a paper reading task. You must attempt both the tasks.

You are encouraged to rely on the existing literature, blogs etc to inform your design choices; we only recommend you to write and justify your choices / assumptions. Note that you are absolutely not required to use the architectures provided in the task descriptions, if any. Develop your own! Make modifications, try something new! Whatever you try and fail at, write it in your report/presentation.

For some tasks there are bonus tasks - to demonstrate that you can go the extra mile (which is an important characteristic of being in our group)! Do not worry if you are not able to do all of the non-bonus tasks. Try to do as much as possible and at least detail what you tried for the tasks you could not complete in the report/presentation.

To overcome compute constraints:

- You can use Google Colab, Kaggle for compute intensive jobs.
- Take a subset of data if the dataset is huge - ex. 25% / 50% / 75% of train data

You must attempt this task on your own. Please make sure you cite any external resources that you may use. We will also check your submission for plagiarism, including ChatGPT :)

Submission:

- Put all your code/notebooks in a GitHub repository. Maintain a README.md explaining your codebase, the directory structure, commands to run your project, the dependency libraries used and the approach you followed.
- The link to the GitHub repository will be asked for during the interview.
- Presentation / Report:
    - Programming Task : Document the process and compile a presentation / report summarizing your findings, methodologies, and any insights gained from the analysis in a presentation/report.
    - Paper Reading: Answer the following questions in a presentation / report
        a. Summarize the paper in 3 slides.
        b. What are the three major strengths of the paper ?
        c. What are the three major weaknesses of the paper ?
        d. Suggest three improvements to the paper, that would improve the paper?

Evaluation:

- You will be evaluated based on how you approach the problem, and not so much the performance measures like accuracy etc.
- How do you handle and sample from large real-world datasets
- How interesting and creative your insights are
- How you present your code and findings. You should justify all the choices you make regarding data, model, hyperparameters that you use. You should also be able to demonstrate theoretical understanding of the approaches used.

For any doubts regarding this task please directly email prashant.kodali@research.iiit.ac.in.

1. Representations for Words, Phrases, Sentences

NLP encompasses various tasks - Regression, Classification, Generation. A common denominator across all these tasks is the question of "how do we convert text into numbers" so that machines can process them. One way to measure a machine's ability to “understand” text is Semantic Similarity i.e one should be able to tell you how similar or dissimilar are a given pair of inputs - and this is the central theme of this task. You are expected to come up with solutions to the problems listed below. For all the tasks below, you can assess how well your solution is working based on some quantitative measures - you are expected to choose a metric that is suitable and justify the same. In addition to quantitative measures, you are expected to come up with 3 interesting takeaways - that could be error analysis, dataset analysis, etc - SURPRISE US!!.

a. **Word Similarity Scores:** Given a pair of words, predict their similarity score. The focus is how do you convert a word to its numerical representation, on which learning algorithms (like Regression, classification etc) can be applied. Download the dataset from this link. You have to come up with an unsupervised / semi supervised method to achieve the task. Assume that you don't have any supervised training data at your disposal. The whole dataset will be used as a test set. Choose an appropriate metric that is suitable to assess the task and report the results. You have to come up with a solution for the following conditions:
    i. Constraints on Data Resources: You can only use the following resources (any one or all) to solve the problem:
        - any monolingual English corpus - Maximum 1 million tokens.
        - any curated/structured knowledge-bases / ontologies

    ii. Unconstrained : Consider that the constraints above are removed and you are allowed to use any data or model.

b. **Phrase and Sentence Similarity :** In question (1) you would have come up with a method to get numerical/vector representation given a word. Now you have to come up with a mechanism to get representations for phrases and sentences. How do you aggregate individual word representations to get phrase or sentence embedding? You can use any pretrained static word embeddings like word2vec, GLOVE, FASTTEXT etc, or create your own.
    i. Phrase Similarity : Given a pair of phrases classify whether or not they are similar. Dataset can be found here. Dataset has train/dev/test split. You have to report results on the test set, and use train/dev sets as needed.

ii. Sentence Similarity : Given a pair of sentences classify whether or not they are similar. Dataset can be found here. Dataset has train/dev/test split. You have to report results on the test set. , and use train/dev sets as needed.

BONUS TASK: Transformers are all the rage right now (backbone of most of the LLMs you might have used). Can you use transformer based models to solve Phrase and Sentence Similarity Tasks? How does this compare with static word embeddings? You are free to use any resource out there.

c. **Paper Reading Task : BERTSCORE: EVALUATING TEXT GENERATION WITH BERT**

2. Analyzing hateful memes

This task involves a technical exploration of a dataset comprising hateful memes. You are expected to focus on object detection within the image. Additionally, assess whether the overlaid captions are a hindrance to the object detection process and explore methods to mitigate this issue. You are allowed to use any off the shelf models available online. However, for the classification system, you are not allowed to use models that are pre-trained for the classification task that you are performing. The dataset to be used is at: **https://hatefulmemeschallenge.com/**

We neither expect nor forbid you to make your own models for tasks a and b. Search around and try to find pre-existing models that can perform this task. Record the challenges encountered, especially those related to the impact of captions on object detection and classification.

For the classification task these resources may help:

https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

https://medium.com/bitgrit-data-science-publication/building-an-image-classification-model-with-pytorch-from-scratch-f10452073212

a. Object Detection:
   - Goal: Utilize computer vision techniques to detect and identify objects within the images of the memes.
   - Tasks:
     - Apply object detection algorithms to identify various elements within the meme images.
     - Catalog the types of objects detected and analyze their frequency and distribution across the dataset.

b. Caption Impact Assessment:
   - Goal: Assess the effect of overlaid captions on the accuracy and effectiveness of object detection.
   - Tasks:
     - Determine how text overlays influence the object detection process.
     - If necessary, develop and implement methods to minimize the impact of captions, such as using image processing techniques to filter out text. (You are not expected to make the model for this, try to find models that can do this for you)

c. Classification System Development:
   - Goal: Develop a system to classify the images based on something non-trivial. Suggestion: You could try classifying whether the image is a meme or not. Dataset for this is readily available as the positive class set

is the dataset given, and you can easily source non-memes from other sources. You may freely choose any other classification task as well, but keep in mind that sourcing labeled data for the same might not be as easy. It is imperative that your classification task involves the provided dataset in part or as a whole. Properly report your methodologies, findings and performance of the model.

d. Paper Reading Task: https://arxiv.org/pdf/2305.15913.pdf

3. Analyzing citation networks

This task involves exploring the High-energy physics citation network. Arxiv HEP-PH (high energy physics phenomenology) citation graph is from the e-print arXiv and covers all the citations within a dataset of 34,546 papers with 421,578 edges. If a paper i cites paper j, the graph contains a directed edge from i to j. If a paper cites, or is cited by, a paper outside the dataset, the graph does not contain any information about this. This dataset is temporal, which means the structure of the network changes over time as new academic papers are published.

Data: http://snap.stanford.edu/data/cit-HepPh.html

a. Task 1 : The first task is a graph exploration task. Build out a graph from the dataset given, and record how the graph and its properties change over time. You are expected to perform this task on at least 5 properties, and report interesting insights. Few simple properties include centrality, density, and diameter. This task is focused on exploratory data analysis, and you are expected to show plots and metrics to support your findings.
b. Task 2: Community detection or clustering is an important analysis for graphs. In the study of complex networks, a network is said to have community structure if the nodes of the network can be easily grouped into disjoint sets of nodes such that each set of nodes is densely connected internally, and sparsely between different communities. In this task, you are required to perform community detection on the graph. This is a well studied problem, and various static algorithms as well as machine learning methods exist for community detection. You are required to:
    1. Implement any two algorithms/ ML methods for community detection on the full final graph
    2. Analyze the communities (Can you build an understanding of why the algorithm chose the communities it did?)
    3. Perform temporal community detection, through which you can study how communities evolve over time as new papers are added. Report interesting insights using various plots and metrics
c. Task 3 [Bonus]: Link Prediction is a task in graph and network analysis where the goal is to predict missing or future connections between nodes in a network. As before, multiple algorithms exist for this task. You are required to implement any algorithm/ ML method of your choice on the citation network at any time T, remove edges from the network randomly, and run the algorithm to see if it can predict the missing edges. You will be evaluated on how well the algorithm/method can predict these edges.
d. Paper Reading Task: https://arxiv.org/abs/2310.02859

Resources:

- Networkx Python Library