Training Program on Natural Language Processing for Research and Scholarly Excellence (TP-RSE)

Course Overview:

The primary goal of this TP-RSE is to equip you with the necessary knowledge and skills to harness the power of NLP and its transformative impact on various domains, from sentiment analysis and text classification to language translation and creative writing. Through a carefully curated curriculum, you will gain proficiency in both theoretical concepts and practical implementation, enabling you to integrate NLP into your research endeavours effectively.

Course Content:

Throughout the course, we will cover a wide range of topics, including:

1. **Foundations of NLP**: Understanding the basics of NLP, its challenges, and the role of machine learning in language processing.
2. **Text Preprocessing**: Learning essential techniques for cleaning, tokenizing, and preparing text data for NLP tasks.
3. **Machine Learning for NLP**: Exploring how machine learning algorithms are applied to NLP tasks like sentiment analysis and text classification.
4. **Word Embeddings**: Diving into word embeddings, including Word2Vec, GloVe, and FastText, to generate dense word representations.
5. **Transformers in NLP**: Delving into transformer architectures, particularly the BERT model, for more advanced NLP applications.
6. **Advanced NLP Topics**: Investigating GPT models for text generation and exploring prompt engineering techniques.

Expected Outcomes:

Upon successful completion of this TP-RSE, you can expect to:

1. **Master NLP Concepts**: Gain a comprehensive understanding of core NLP concepts, including word embeddings, transformers, and advanced NLP models.
2. **Hands-on Proficiency**: Acquire practical experience in implementing NLP algorithms and models through hands-on exercises and assignments.
3. **Real-world Applications**: Be prepared to apply NLP techniques to real-world problems and research challenges in various domains.
4. **Integration in Education**: Integrate NLP knowledge into your research methodologies to enhance the learning experience.
5. **Cutting-edge Awareness**: Stay updated with the latest advancements in NLP technologies and research trends.

We believe that this TP-RSE will empower you with valuable insights and expertise in the dynamic field of Natural Language Processing. Our aim is to foster a collaborative learning environment where you can interact with peers, exchange ideas, and challenge yourselves to push the boundaries of your NLP knowledge.

Prerequisites:

The participants of the TP-RSE on Machine Learning Concepts with Natural Language Processing (NLP) should ideally have some prior knowledge in the following areas to make the most out of the program:

1. **Programming**: Participants should have a good understanding of at least one programming language, preferably Python, as most NLP libraries and frameworks are commonly used with Python. Familiarity with Linux environments.
2. **Data Manipulation**: Familiarity with data manipulation libraries in Python (e.g., Pandas) would be beneficial, as NLP often involves working with textual data that requires data preprocessing and cleaning.
3. **Basic Machine Learning**: A basic understanding of machine learning concepts, including supervised and unsupervised learning, as well as common algorithms like decision trees, SVM, and neural networks will be helpful, especially when diving into NLP tasks that involve machine learning models.
4. **Text Processing**: Basic knowledge of text processing techniques like tokenization, stemming, and vectorization will give participants a head start when dealing with textual data in NLP applications.
5. **Neural Networks**: While not mandatory, having some familiarity with neural networks will be advantageous as NLP models often leverage deep learning architectures like recurrent neural networks (RNNs) and transformers.
6. **Mathematics and Statistics**: A basic understanding of linear algebra, probability, and statistics will be useful, especially when delving into the mathematical foundations of NLP algorithms and models.

Lectures:

Dr. Ramesh Dileep Kumar Appana
Lead Data Scientist @ Leoforce, Inc
16 years of Industrial Experience

S. No	Date	Day	Topic
Pre Day 1	30/08/2024	Friday	Introduction to Python
Pre Day 2	31/08/2024	Saturday	Working with Cloud
Pre Day 3	06/09/2024	Friday	Machine Learning Basics
Pre Day 4	13/09/2024	Friday	Machine Learning Concepts Continued
Day 1	14/09/2024	Saturday	Introduction to NLP and Basics of Text Processing
Day 2	20/09/2024	Friday	NLP and Machine Learning
Day 3	21/09/2024	Saturday	Deep Learning for NLP - Word Embeddings
Day 4	27/09/2024	Friday	Transformers in NLP
Day 5	28/09/2024	Saturday	Trending Topics in NLP
Project Day 1	30/09/2024	Monday	Project Submission Deadline
Guest Lectures	05/10/2024	Saturday	** To be Confirmed yet
Project Day 2	19/10/2024	Friday	Project Review

Project Reviewers:

Dr. Jia Uddin
Assistant Professor - AI and Big Data
Endicott College, Woosong University, Daejeon, South Korea

Dr. Junayed Hasan Md
Researcher, Robert Gordon University
United Kingdom

Dr. Ramesh Dileep Kumar
Lead Data Scientist, Leoforce Inc

Dr. Gopi Krishna
Data Science Manager, Accenture,
Hyderabad, India.

Laurence Raj
Vice President, Teleperformance Inc
Chennai, India

Pre - Day 1: Introduction to Python

- Module Building (Session 1)
- Environmental controls
- IDE
- CUDA
- REST API
    - Flask Application
- Remote Connections (Session 2)
    - Development / Virtual Environments
    - Running live applications
- Data types
    - CSV
    - JSON
- Databases (Session 3)
    - MySQL (read, write, access)

- **Assignment 1**: Write code to access and insert data through the database MySQL.
- **Assignment 2**: Create a simple Flask API to retrieve information from the database as per requirement.

Pre - Day 2: Working with Cloud

- Connecting with cloud (Session 1)
    - AWS
    - S3 bucket
- Python domain-specific (Session 2)
    - Computer Vision applications
    - Signal Processing
    - Text Processing
    - Scraping
- Monitoring (Session 3)
    - Logging errors
- Unit Testing - Pytests
- Version Control (Git)
- **Assignment 3:** Multi Environment - build Live application - assignment for next day.

Pre - Day 3: Machine Learning Concepts

- Neural Networks Basics (Session 1)
    - Supervised / unsupervised learning
    - Classification / Clustering
    - Vector Spaces
    - Perceptrons, multi-layers networks, backpropagation
    - Confusion Matrix
- Shallow Neural Networks
- Practical work / Build Application (Session 2)
    - Shallow Neural Network.
- Support Vector Machines (Session 3)
    - Concept
    - Parameters

    - Python Implementation
- **Assignment 4:** Build an SVM classifier and make it a rest application.
- **Assignment 5:** Build a Multiclass SVM Classifier.

Pre - Day 4: Machine Learning Concepts Continued

- k-NN algorithm (Session 1)
    - Concepts
    - Python Implementation
- Random Forest (Session 2)
    - Concepts
    - Python Implementation
- CNN algorithm (Session 3)
    - Concepts
    - Python Implementation

- **Assignment 6:** implement a classifier using the k-NN algorithm on a given dataset.
- **Assignment 7:** implement a random forest classifier on a given dataset.
- **Assignment 8:** implement a clustering method on a given dataset and make it a rest application.

Day 1: Introduction to NLP and Basics of Text Processing (5 hours)

Session 1: Introduction to NLP (1 hour [9:00 AM - 10:00 AM])

- Overview of NLP: Explanation of what NLP is, its applications in various industries, and its significance in today's world.
- Historical perspective: Briefly discuss the evolution of NLP and notable milestones.
- Challenges in NLP: Understanding the inherent complexities in processing natural languages, such as ambiguity, polysemy, and context dependence.

Session 2: Understanding Text Data (1.5 hours [10:30 AM - 12:00 PM])

- Text data representation: Discussing different ways to represent text data, including the bag-of-words model and its limitations.
- Introduction to word embeddings: Understanding distributed word representations and their advantages over traditional methods.
- Pre-trained word embeddings: Overview of pre-trained word embedding models like Word2Vec, GloVe, and FastText.

Session 3: Text Preprocessing Techniques (2 hours [1:00 PM - 3:00 PM])

- Tokenization: Explanation of tokenization, the process of breaking text into smaller units (tokens) such as words or subwords.
- Text normalization: Covering techniques like converting text to lowercase, removing punctuation, and handling contractions.
- Stop word removal: Understanding the importance of removing common words that carry little meaning (e.g., "and," "the," "is").
- Noise removal: Handling noisy text, such as special characters, HTML tags, and URLs.

Session 4: Hands-on Text Preprocessing with Python (30 minutes [3:10 PM - 3:40 PM])

- Setting up the development environment: Ensuring participants have the necessary Python libraries (e.g., NLTK or spaCy) installed.
- Practical implementation: Step-by-step guidance on implementing tokenization, text normalization, stop word removal, and noise removal using Python.

Session 5: Text Preprocessing Practical Exercises (30 minutes [3: 50 PM - 4:20 PM])

- Participants will work on small exercises to practice the concepts covered in the hands-on session.
- Facilitators will be available to assist and answer questions as participants apply text preprocessing techniques to real text data.

By the end of Day 1, participants will gain a solid understanding of NLP, the challenges it poses, and how text data is processed in the context of NLP applications. They will have hands-on experience in text preprocessing techniques using Python libraries, which will set the foundation for the subsequent days of the TP-RSE where more advanced NLP topics will be covered.

Assignment for Day 1: Text Preprocessing and Data Exploration

1. Dataset Selection:
    a. Choose a publicly available text dataset of your choice (e.g., movie reviews, news articles, social media posts, etc.). You can use resources like Kaggle, UCI Machine Learning Repository, or any other reputable source for datasets.
2. Data Exploration:
    a. Load the selected dataset into Python using appropriate libraries (e.g., Pandas) and perform basic data exploration.
    b. Display a few sample records to understand the structure of the data and its attributes.
3. Text Preprocessing:
    a. Implement text preprocessing techniques discussed during the TP-RSE (tokenization, text normalisation, stop word removal, and noise removal) on the text data from your selected dataset.
    b. Make sure to document each step and provide explanations for the choices you make during preprocessing.
4. Data Analysis:
    a. After preprocessing, perform basic data analysis to gain insights into the dataset.
    b. Calculate and visualise the most frequent words in the dataset.
    c. Identify any patterns or trends that emerge from the analysis.
5. Reflection:
    a. Write a brief reflection on the challenges you faced during the text preprocessing process and how you addressed them.
    b. Share your thoughts on the importance of text preprocessing in NLP tasks and how it impacts the quality of subsequent analyses and models.

Submission Guidelines:

- Organize your assignment into a Jupyter Notebook (or a Python script) with clear headings and code comments.
- Include visualisations, graphs, or any other relevant outputs to support your analysis.
- Submit your assignment as a PDF or share a link to the GitHub repository containing your code and the notebook.

Day 2: NLP and Machine Learning (5 hours)

Session 1: NLP Tasks and Applications (1 hour [9:00 AM - 10:00 AM])

- Recap of Day 1: A brief recap of the concepts covered on Day 1, including text preprocessing and data exploration from the participants' assignments.
- Introduction to NLP tasks: Detailed explanation of common NLP tasks, such as sentiment analysis, text classification, and named entity recognition.
- Real-world applications: Showcase examples of NLP applications in different domains like social media analysis, customer reviews, and chatbots.

Session 2: Feature Engineering for NLP (1.5 hours [10:30 AM - 12:00 PM])

- Introduction to feature engineering: The concept of transforming raw data (text) into meaningful features for machine learning algorithms.
- Bag-of-words (BoW) model: Understanding BoW representation and its limitations in capturing word semantics.
- Term Frequency-Inverse Document Frequency (TF-IDF): Explanation of TF-IDF as a feature engineering technique to address BoW limitations.
- Word embeddings: Recap of word embeddings from Day 1 and their role in generating dense and continuous word representations.

Session 3: Introduction to Machine Learning Algorithms for NLP (2 hours [1:00 PM - 3:00 PM])

- Supervised learning for NLP: Explanation of supervised learning algorithms (e.g., Naive Bayes, Support Vector Machines) and their adaptation for NLP tasks.
- Neural networks for NLP: Introduction to basic neural network architectures like feedforward neural networks and their applications in NLP.
- Deep learning for text classification: Understanding the use of deep learning models for sentiment analysis and text classification tasks.

Session 4: Hands-on: Building an NLP Model (30 minutes [3:10 PM - 3:40 PM])

- Preparing the data: Loading the preprocessed text data from Day 1 and converting it into suitable formats for model training.
- Splitting the dataset: Splitting the data into training and testing sets to evaluate the model's performance.
- Building a machine learning model: Implementing a simple machine learning model (e.g., Naive Bayes) or a basic neural network for an NLP task using Python libraries (e.g., scikit-learn or TensorFlow/Keras).

Session 5: NLP Model Evaluation (30 minutes [3: 50 PM - 4:20 PM])

- Model evaluation metrics: Explanation of common evaluation metrics for NLP tasks, such as accuracy, precision, recall, and F1-score.
- Evaluating the model: Assessing the performance of the implemented NLP model on the test dataset.
- Interpretation of results: Analysing the model's predictions and discussing potential improvements.

By the end of Day 2, participants will gain a comprehensive understanding of how NLP and machine learning are interconnected, how to engineer features for NLP tasks, and how to build and evaluate basic machine learning models for NLP applications. The hands-on session will allow participants to put their knowledge into practice and gain practical experience in implementing NLP models with Python libraries.

Assignment for Day 2: Building an NLP Model

Dataset: For this assignment, you will use a sentiment analysis dataset containing text reviews and their corresponding sentiment labels (positive or negative). You can choose a publicly available dataset or use a well-known dataset like the IMDb movie reviews dataset.

Task: Your task is to build an NLP model for sentiment analysis using machine learning algorithms and evaluate its performance on a test dataset.

Instructions:

1. Data Preprocessing:
    a. Load the sentiment analysis dataset into Python using appropriate libraries (e.g., Pandas).
    b. Preprocess the text data by applying the techniques covered on Day 1 (e.g., tokenization, text normalisation, stop word removal).
2. Feature Engineering:
    a. Convert the preprocessed text data into numerical features suitable for machine learning models.
    b. Implement the Bag-of-Words (BoW) model or TF-IDF representation for feature engineering.
3. Model Building:
    a. Split the dataset into training and testing sets (e.g., 80% training, 20% testing).
    b. Choose a machine learning algorithm (e.g., Naive Bayes, Support Vector Machines) or a basic neural network for sentiment analysis.
    c. Train the model on the training dataset and tune hyperparameters if necessary.
4. Model Evaluation:
    a. Evaluate the performance of the NLP model on the test dataset using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score).
    b. Provide a detailed analysis of the model's performance and discuss the strengths and limitations of the chosen algorithm.
5. Bonus (Optional):
    a. Try using pre-trained word embeddings (e.g., Word2Vec or GloVe) as features and see how they affect the model's performance.
    b. Experiment with different machine learning algorithms or neural network architectures to compare their performance.

Submission Guidelines:

- Organise your assignment into a Jupyter Notebook (or a Python script) with clear headings and code comments.
- Include visualisations, graphs, or any other relevant outputs to support your analysis.
- Present your findings and observations in a clear and concise manner.
- Bonus Tip: You can use libraries like scikit-learn or TensorFlow/Keras to simplify the implementation of machine learning models and evaluation metrics.

Day 3: Deep Learning for NLP - Word Embeddings (5 hours)

Session 1: Word Embeddings (1 hour [9:00 AM - 10:00 AM])

- Recap of word embeddings from Day 2: Reviewing the concept of word embeddings and their role in generating distributed word representations.
- Word2Vec: Revisiting the Word2Vec algorithm and its two architectures (Continuous Bag of Words - CBOW and Skip-gram).
- GloVe: Briefly recapping the Global Vectors for Word Representation (GloVe) approach to generating word embeddings.
- FastText: Understanding the advantages of FastText in handling out-of-vocabulary words and subword embeddings.

Session 2: Transfer Learning with Pre-trained Word Embeddings (1.5 hours [10:30 AM - 12:00 PM])

- Transfer learning in NLP: Introducing the concept of transfer learning and its application in NLP using pre-trained word embeddings.
- Pre-trained word embeddings: Overview of popular pre-trained word embeddings such as Word2Vec, GloVe, and FastText.
- Fine-tuning vs. feature extraction: Understanding the difference between fine-tuning pre-trained embeddings and using them as fixed features.
- Fine-tuning strategies: Discussing the different approaches to fine-tuning pre-trained embeddings for specific downstream tasks.

Session 3: Contextual Word Embeddings (2 hours [1:00 PM - 3:00 PM])

- Limitations of static word embeddings: Exploring the limitations of traditional static word embeddings in capturing context and word sense disambiguation.
- Introduction to contextual word embeddings: Understanding the need for contextual embeddings that adapt to the surrounding words in a sentence.
- ELMo (Embeddings from Language Models): Explaining how ELMo generates contextual embeddings using bidirectional language models.
- BERT (Bidirectional Encoder Representations from Transformers): Introduction to BERT and its architecture, including the pre-training and fine-tuning process.

Session 4: Hands-on: Working with Pre-trained Embeddings (30 minutes [3:10 PM - 3:40 PM])

- Loading pre-trained word embeddings: Demonstrating how to load and use pre-trained word embeddings in Python using libraries like Gensim or TensorFlow/Keras.
- Fine-tuning embeddings: Implementing fine-tuning strategies for pre-trained word embeddings on a specific NLP task (e.g., sentiment analysis or text classification).

Session 5: Contextual Word Embeddings in Action (30 minutes [3: 50 PM - 4:20 PM])

- Understanding BERT embeddings: Exploring the BERT model's ability to generate contextual word embeddings.
- BERT for sentence classification: Illustrating how BERT can be fine-tuned for tasks like sentiment analysis or text classification.
- Discussing real-world applications: Showcasing successful applications of contextual word embeddings in various NLP tasks.

By the end of Day 3, participants will have a deeper understanding of word embeddings, the transfer learning paradigm, and the significance of contextual word embeddings. They will have hands-on experience with loading and using pre-trained word embeddings and experimenting with fine-tuning strategies. Additionally, participants will grasp the potential of contextual embeddings and how models like BERT can provide contextually aware representations for more complex NLP tasks.

Assignment for Day 3: Working with Pre-trained and Contextual Word Embeddings

Dataset: For this assignment, you will use a text classification dataset containing text documents and their corresponding categories (e.g., sentiment labels, topic labels). You can choose a publicly available dataset or use a well-known dataset like the IMDb movie reviews dataset for sentiment analysis.

Task: Your task is to explore pre-trained word embeddings and contextual word embeddings, fine-tune them on the text classification dataset, and evaluate their performance on the test dataset.

Instructions:

1. Data Preparation:
    a. Load the text classification dataset into Python using appropriate libraries (e.g., Pandas).
    b. Preprocess the text data by applying tokenization, text normalization, stop word removal, and other techniques covered in previous days.
2. Working with Pre-trained Word Embeddings
    a. Choose a pre-trained word embedding model (e.g., Word2Vec, GloVe, or FastText).
    b. Load the pre-trained word embeddings and create word embeddings for the words in your dataset.
    c. Implement a simple machine learning model (e.g., SVM or Logistic Regression) using these word embeddings as features.
    d. Fine-tune the pre-trained embeddings on the text classification task and evaluate the model's performance on the test dataset.
3. Working with Contextual Word Embeddings
    a. Choose a pre-trained contextual word embedding model (e.g., BERT).
    b. Fine-tune the BERT model on the text classification dataset using transfer learning techniques.
    c. Implement a machine learning model using contextual word embeddings as features and evaluate its performance on the test dataset.
4. Model Comparison and Analysis
    a. Compare the performance of the two approaches: pre-trained word embeddings and contextual word embeddings.
    b. Analyze the strengths and weaknesses of each approach in the context of the text classification task.
    c. Discuss the impact of fine-tuning on model performance and the challenges faced during the process.

Submission Guidelines:

- Organize your assignment into a Jupyter Notebook (or a Python script) with clear headings and code comments.
- Include visualizations, graphs, or any other relevant outputs to support your analysis.
- Provide a detailed analysis of the model's performance, comparing both approaches and justifying your observations.

Bonus (Optional):

- Experiment with different pre-trained word embedding models and compare their performance.
- Explore the effect of fine-tuning strategies on contextual word embeddings.

Day 4: Transformers in NLP (5 hours)

Session 1: Introduction to Transformers (1 hour [9:00 AM - 10:00 AM])

- Limitations of traditional sequence-to-sequence models: Discussing the limitations of recurrent neural networks (RNNs) in handling long-range dependencies and the need for attention mechanisms.
- Attention mechanism: Understanding the concept of attention in sequence modelling and how it allows the model to focus on relevant information.
- Transformer architecture overview: Introducing the Transformer architecture, which uses self-attention layers to process sequences.

Session 2: The Transformer Architecture (1.5 hours [10:30 AM - 12:00 PM])

- Self-attention mechanism: Deep dive into how self-attention works, calculating attention scores, and obtaining weighted representations.
- Multi-head attention: Explaining multi-head attention and its ability to capture different aspects of relationships within a sequence.
- Positional encoding: Understanding the necessity of positional encoding to provide positional information to the transformer model.

Session 3: Introduction to BERT (2 hours [1:00 PM - 3:00 PM])

- BERT: Bidirectional Encoder Representations from Transformers: Introducing the BERT model and its significance in NLP tasks.
- BERT architecture: Understanding the encoder structure of BERT, which enables bidirectional context understanding.
- BERT pre-training: Explaining the masked language model (MLM) and next sentence prediction (NSP) pre-training tasks.

Session 4: Hands-on: Fine-tuning BERT (30 minutes [3:10 PM - 3:40 PM])

- Fine-tuning BERT: Demonstrating how to load pre-trained BERT models and fine-tune them on downstream NLP tasks.
- Transfer learning with BERT: Explaining how BERT's pre-trained knowledge can be leveraged for various NLP tasks.

Session 5: Advanced NLP Tasks with BERT (30 minutes [3: 50 PM - 4:20 PM])

- Beyond classification: Showcasing how BERT can be used for other advanced NLP tasks, such as named entity recognition, question answering, and text generation.
- Case studies: Presenting real-world examples of successful applications of BERT in various NLP domains.

By the end of Day 4, participants will have a comprehensive understanding of transformer architecture, its application in NLP, and the significance of the BERT model. They will also have hands-on experience in fine-tuning BERT for downstream NLP tasks and be familiar with the advanced NLP tasks that BERT can handle. Participants will be equipped with knowledge and practical skills to work with state-of-the-art NLP models based on transformers, opening up new opportunities in the field of natural language processing.

Assignment for Day 4: Fine-tuning BERT for Text Classification

Dataset: For this assignment, you will use a text classification dataset containing text documents and their corresponding categories (e.g., sentiment labels, topic labels). You can choose a publicly available dataset or use a well-known dataset like the IMDb movie reviews dataset for sentiment analysis.

Task: Your task is to fine-tune the pre-trained BERT model on the text classification dataset and evaluate its performance on the test dataset.

Instructions:

1. Data Preparation:
   a. Load the text classification dataset into Python using appropriate libraries (e.g., Pandas).
   b. Preprocess the text data by applying tokenization, text normalization, stop word removal, and other techniques covered in previous days.
2. Fine-tuning BERT
   a. Choose a pre-trained BERT model (e.g., 'bert-base-uncased') from the Hugging Face Transformers library.
   b. Tokenize the text data and create input data suitable for fine-tuning BERT.
   c. Fine-tune the BERT model on the text classification task using transfer learning techniques.
   d. Implement a machine learning model using the fine-tuned BERT embeddings as features and evaluate its performance on the test dataset.
3. Evaluation and Analysis
   a. Evaluate the performance of the fine-tuned BERT model on the test dataset using appropriate evaluation metrics (e.g., accuracy, precision, recall, F1-score).
   b. Compare the performance of the fine-tuned BERT model with the machine learning models used in previous assignments.
   c. Analyze the strengths and limitations of the fine-tuned BERT model for the text classification task.
4. Model Visualization
   a. Visualize the attention mechanism in the fine-tuned BERT model to gain insights into how the model focuses on different parts of the text.

Submission Guidelines:

- Organize your assignment into a Jupyter Notebook (or a Python script) with clear headings and code comments.
- Include visualizations, graphs, or any other relevant outputs to support your analysis.
- Provide a detailed analysis of the model's performance and compare it with the results obtained from the previous assignments.

Bonus (Optional):

- Experiment with different pre-trained BERT models and hyperparameters to observe their impact on model performance.
- Explore other advanced NLP tasks that BERT can handle and try fine-tuning the model for one of those tasks.

Day 5: Trending topics in NLP (5 hours)

Session 1: Introduction to GPT Models (1 hour [9:00 AM - 10:00 AM])

- Recap of transformers and BERT: Briefly review the transformer architecture and the BERT model covered on Day 4.
- Introduction to GPT: Understanding the concept of Generative Pre-trained Transformer models and their ability to generate human-like text.
- Use cases of GPT: Exploring the applications of GPT models in text generation, language translation, and creative writing.

Session 2: Implementing GPT-2 (1.5 hours [10:30 AM - 12:00 PM])

- GPT-2 architecture: Understanding the structure of the GPT-2 model and its components, including the decoder-only transformer architecture.
- GPT-2 pre-training: Explaining the pre-training process of GPT-2 using unsupervised learning on a large corpus of text data.
- Fine-tuning GPT-2: Discuss how to fine-tune the GPT-2 model on specific text generation tasks using transfer learning.

Session 3: Prompt Engineering for Text Generation (1.5 hours [1:00 PM - 2:30 PM])

- The role of prompts: Understanding the importance of prompts in guiding GPT models to generate desired text outputs.
- Prompting strategies: Exploring different prompt engineering techniques, including prefixing, controlled generation, and creative writing prompts.
- Handling biases in GPT: Discussing methods to mitigate biases and ensure responsible use of GPT models.

Session 4: Hands-on: Text Generation with GPT-2 (30 minutes [2:40 PM - 3:10 PM])

- Implementing text generation with GPT-2: Demonstrating how to use a pre-trained GPT-2 model to generate text based on different prompts.
- Trying out different prompts: Encouraging participants to experiment with various prompts and observe the model's responses.

Session 5: Applications and Future of NLP (1 hour [3:20 PM - 4:20 PM])

- Real-world applications of advanced NLP models: Showcasing successful applications of GPT models in various domains, including content generation, chatbots, and creative writing.
- Recent advancements in NLP: Discussing the latest technologies and research trends in NLP, such as the latest transformer variants, transfer learning approaches, and ethical considerations in NLP.

By the end of Day 5, participants will have a comprehensive understanding of advanced NLP topics, including GPT models, prompt engineering, and the latest advancements in NLP technologies. They will have hands-on experience in implementing GPT-2 for text generation tasks and be aware of the diverse applications of advanced NLP models in real-world scenarios. Participants will leave the TP-RSE with insights into the current trends in NLP research and how the field is continuously evolving with the development of cutting-edge technologies.

Guest Lectures

Title: Industry Applications of NLP: Transforming Business with Language Processing
Duration: Approximately 2.5 to 3 hours [9:00 AM - 12:00 PM]
Guest Speaker: To Be Confirmed

Overview:

The guest lecture on Industry Applications of NLP will focus on how Natural Language Processing is revolutionising various industries and driving transformative changes in business processes. The guest speaker, an experienced industry professional, will share real-world use cases and insights from their firsthand experience in implementing NLP solutions.

Topics Covered:

1. Introduction to NLP in Industry:
    a. A brief overview of what NLP is and why it holds immense potential for industries.
    b. The role of NLP in extracting insights from unstructured text data and making it actionable.
2. Sentiment Analysis for Customer Feedback:
    a. How sentiment analysis is used to analyze customer feedback, product reviews, and social media sentiments.
    b. Case studies demonstrating, how sentiment analysis helps businesses understand customer satisfaction levels and make data-driven decisions.
3. Chatbots and Virtual Assistants:
    a. How NLP powers conversational agents like chatbots and virtual assistants.
    b. Practical examples of how chatbots are enhancing customer support, automating tasks, and streamlining operations.
4. Language Translation and Globalization:
    a. The significance of NLP in breaking language barriers and enabling global communication.
    b. Use cases of language translation in facilitating international business operations and enhancing user experiences.
5. Text Summarization and Content Generation:
    a. How NLP techniques are leveraged for automatic text summarization and content generation.
    b. Examples of how these applications save time, improve content quality, and cater to diverse audiences.
6. Voice Recognition and Speech-to-Text:
    a. Exploring NLP applications in voice recognition and speech-to-text technologies.
    b. How businesses are leveraging speech analytics for call center insights and voice-driven applications.
7. Challenges and Ethical Considerations:
    a. Discussing the challenges faced during NLP implementation in industries.
    b. Addressing ethical considerations, bias mitigation, and responsible AI practices.
8. Q&A Session
    a. An interactive session where participants can ask questions, seek advice, and engage in discussions with the guest speaker.

Key Takeaways:

- Gain insights into the diverse applications of NLP in different industries.
- Understand how NLP technologies are being used to improve customer experiences and business operations.
- Learn about the challenges and opportunities in implementing NLP solutions in real-world scenarios.
- Acquire practical knowledge and inspiration for integrating NLP into their own research and projects.

Guest Lectures

Title: Cutting-Edge NLP Research: Exploring the Frontiers of Natural Language Processing
Duration: Approximately 2.5 to 3 hours [1:00 PM to 4:00 PM]
Guest Speaker: To Be Confirmed

Overview:

The guest lecture on Cutting-Edge NLP Research will delve into the latest advancements and research trends in the field of Natural Language Processing. The guest speaker, an esteemed NLP researcher, will share insights into recent breakthroughs, state-of-the-art models, and experimental methodologies that are shaping the future of NLP.

Topics Covered:

1. Introduction to Cutting-Edge NLP Research:
    a. An overview of the current state of NLP research and its significance in AI applications.
    b. Highlighting the rapid advancements and breakthroughs that have transformed the NLP landscape.
2. Pre-trained Language Models (PLMs):
    a. Understanding the emergence and impact of pre-trained language models (PLMs) like BERT, GPT, and their variants.
    b. Exploring the transfer learning paradigm and how it has revolutionised NLP tasks.
3. Transformer Variants:
    a. Overview of recent transformer variants, such as XLNet, RoBERTa, and T5.
    b. Comparative analysis of their architectures and performance on various NLP benchmarks.
4. Multimodal NLP:
    a. The convergence of NLP with computer vision and other modalities.
    b. How multimodal NLP enables a deeper understanding of language in context.
5. Zero-Shot and Few-Shot Learning:
    a. Exploring advancements in zero-shot and few-shot learning for NLP tasks.
    b. Examples of models that can perform tasks without task-specific training data.
6. Interpretability and Explainability:
    a. Addressing the need for interpretability in complex NLP models.
    b. Techniques and approaches for making NLP models more transparent and interpretable.
7. Latest Research Papers and Publications:
    a. Highlighting recent influential research papers and publications in the NLP community.
    b. Providing insights into the methodologies and findings of these studies.
8. Future Directions and Open Challenges:
    a. Discussing the potential future directions of NLP research.
    b. Identifying open challenges and areas that require further exploration.
9. Q&A Session:
    a. An interactive session where participants can engage with the guest speaker, seek clarifications, and discuss emerging research trends.

Key Takeaways:

- Gain a deep understanding of the latest advancements in NLP research and state-of-the-art models.
- Discover the practical applications and potential impact of cutting-edge NLP techniques.
- Get insights into the challenges faced by researchers and how they are being addressed.
- Get inspired to explore new research avenues and contribute to the evolving field of NLP.

Students can pick any of the topics and write a project proposal:

1. Sentiment Analysis for Customer Feedback:
   Develop a sentiment analysis model that analyzes customer feedback from online reviews or social media and categorizes it as positive, negative, or neutral to understand customer sentiments towards a product or service.

2. Chatbot for Customer Support (Open API based):
   Build a chatbot using NLP techniques that can handle customer queries, provide relevant information, and direct users to appropriate resources for efficient customer support.

3. Text Summarization for News Articles:
   Create a text summarization system that can generate concise and accurate summaries for news articles to help users quickly grasp the main points without reading the entire text.

4. Language Translation with Multimodal Input:
   Design a language translation system that takes both text and image inputs, leveraging multimodal NLP techniques to provide translations based on contextual understanding.

5. Voice-Driven Virtual Assistant: (Only for MSc Students)
   Develop a voice-driven virtual assistant that can perform tasks like setting reminders, answering questions, and providing weather updates using speech-to-text and NLP technologies.

6. Fine-tuning BERT for Domain-Specific Sentiment Analysis:
   Fine-tune a pre-trained BERT model on domain-specific data (e.g., product reviews, medical records) to perform sentiment analysis tailored to a specific domain.

7. Text Generation using GPT-2:
   Build a text generation model based on GPT-2 that can generate creative and contextually relevant text in response to user prompts, such as writing short stories or generating code snippets.

8. Topic Modeling and Clustering:
   Implement topic modelling techniques like Latent Dirichlet Allocation (LDA) or Non-negative Matrix Factorization (NMF) to cluster and categorise large sets of unstructured text data.

9. BERT-based Question Answering System:
   Create a question-answering system that utilises BERT to understand and answer user questions based on a given dataset or knowledge base.

10. NLP for Medical Text Analysis:
    Apply NLP techniques to analyse medical records and extract relevant information such as diagnoses, treatments, and patient outcomes to support healthcare decision-making.

Requirements from College:

1. System with Linux Environments (ubuntu 22.04 installed) for each student with 8 GB RAM, decent i5 processor.
2. Projector Screen, whiteboard and markers, reliable internet connection and adequate power supplies to all devices.
3. Remote Server for 2 months. 1 remote server required for 15 students.
4. Created Teams group for recordings and sharing information, PPT, assignment answers.
5. Required Python programming Lab programmers / helpers for students if they are new to programming environments.