

1. The type of NLP application that applies to allocating emails to specific folders based on their content is called text classification or email classification. Text classification involves assigning predefined categories or labels to text documents based on their content or characteristics.

In the case of allocating emails to folders like work, friends, promotions, and important, you would typically train a text classification model using machine learning techniques. The model would learn from a labelled dataset of emails that have been manually assigned to their respective folders. The model would then generalize from the training data to predict the appropriate folder for new, unseen emails.

To build a text classification model for email folder allocation, you would typically follow these steps:

- **Data collection:** Gather a labelled dataset of emails with their corresponding folder assignments. This dataset should include a variety of emails from different categories.
- **Data preprocessing:** Clean and preprocess the email text data by removing noise, normalizing the text, and applying techniques such as tokenization, stemming/lemmatization, and removing stop words.
- **Feature extraction:** Convert the pre-processed text into numerical feature vectors that can be used as input to the machine learning model. Common approaches for feature extraction include bag-of-words, TF-IDF (Term Frequency-Inverse Document Frequency), or word embeddings such as Word2Vec or GloVe.
- **Model training:** Select a suitable machine learning algorithm such as Naive Bayes, Support Vector Machines (SVM), or deep learning models like recurrent neural networks (RNN) or convolutional neural networks (CNN). Split your labeled dataset into a training set and a validation set, and train the model on the training set.
- **Model evaluation:** Evaluate the performance of the trained model using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score. Tweak the model or try different algorithms if the performance is not satisfactory.
- **Deployment and inference:** Deploy the trained model to make predictions on new, unseen emails. Given the content of an email, the model will predict the most suitable folder for it.

2. The type of NLP application that applies to grading or scoring essays is called automated essay scoring (AES) or automated essay grading (AEG). AES systems are designed to assess and assign grades to essays based on various linguistic and content-related features.

Automated essay scoring systems can be particularly helpful for grading large volumes of essays efficiently and consistently. Here's an overview of the typical approach used in AES:

- **Data collection:** Gather a large dataset of essays with their corresponding human-assigned grades. The dataset should cover a range of topics and writing styles to ensure a representative training set.
 - **Feature extraction:** Extract relevant features from the essays that capture both linguistic aspects (e.g., grammar, vocabulary, sentence structure) and content-related characteristics (e.g., coherence, organization, argumentation). Features can include word frequencies, sentence lengths, syntactic patterns, or semantic representations.
 - **Model training:** Use machine learning techniques to train a model that learns the relationship between the extracted features and the human-assigned grades. Common approaches include regression models, support vector machines (SVM), or deep learning models like recurrent neural networks (RNN) or convolutional neural networks (CNN). Split the dataset into training and validation sets, and train the model on the training set.
 - **Model evaluation:** Evaluate the performance of the trained model using appropriate evaluation metrics such as mean squared error (MSE), Pearson correlation coefficient, or Cohen's kappa. Adjust and fine-tune the model as needed based on the evaluation results.
 - **Deployment and inference:** Deploy the trained model to automatically assign grades to new, unseen essays. The model will take the essay as input and output a predicted grade or score.
3. The type of NLP application that applies to providing assistive technology for doctors to aid in diagnosis is known as clinical decision support systems (CDSS). CDSS systems leverage natural language processing techniques to analyze patient data, answer questions posed by doctors, and provide recommendations or probable diagnoses. CDSS systems aim to enhance the decision-making process by integrating patient-specific information, medical knowledge, and data analysis. Here's an overview of the typical approach used in CDSS:
- **Data integration:** Gather and integrate relevant patient data, such as medical history, symptoms, laboratory results, imaging reports, and demographic information. This data can be obtained from electronic health records (EHRs) or other healthcare systems.
 - **Natural language understanding:** Apply natural language processing techniques to process and understand the questions or queries posed by doctors. This involves parsing and analyzing the text to extract relevant information and identify key clinical concepts.
 - **Knowledge representation:** Represent medical knowledge in a structured format that can be processed by the CDSS. This typically involves creating a knowledge base that contains medical guidelines, research findings, treatment protocols, and diagnostic criteria.
 - **Inference and reasoning:** Utilize the patient data and medical knowledge to perform reasoning and inference to generate probable diagnoses or

recommendations. This may involve applying rule-based systems, machine learning algorithms, or probabilistic models to evaluate the patient's symptoms and provide potential diagnoses or treatment options.

- **Presentation and visualization:** Present the generated information, such as probable diagnoses, relevant clinical guidelines, or treatment options, to the doctor in a user-friendly and interpretable format. This may include visualizations, summaries, or interactive interfaces that aid in understanding and decision-making.
- **Continual learning and improvement:** Continually update and refine the CDSS by incorporating new medical knowledge, updating algorithms, and leveraging feedback from doctors to improve accuracy, relevance, and usability.