**Problem Statement:**

Design a recommendation system for **ServiceNow support** that automatically suggests relevant **Knowledge Base (KB) articles** for resolving user-reported issues. The dataset consists of multilingual user descriptions, ServiceNow cases, and associated knowledge base entries. The system will aim to reduce human intervention by automatically mapping case descriptions to the most relevant KB articles, improving both efficiency and resolution rates.

**Key Challenges:**

1. **Multilingual Descriptions**: User descriptions are written in multiple languages, requiring translation for consistent processing.
2. **Unstructured Text**: The short descriptions and full case details are noisy and varied in format, complicating text analysis.
3. **High Dimensionality**: Each case is associated with multiple fields like service offering, brand, and task, adding complexity to the data.
4. **Resolution Rates**: Analyzing which KB articles result in successful resolutions will guide the recommendation system’s training.

**Feel Free to try your own approach too or any opensource pretrained model.**

**Example Solution Approach.**

**Data Preprocessing**:

* + **Translation**: Use a translation API (e.g., Azure Translator) to standardize all case descriptions into a common language (like English).
  + **Text Cleaning**: Remove noise, special characters, and unnecessary data from the short descriptions and case details.
  + **Feature Engineering**: Use methods like **TF-IDF** or **Word Embeddings** (BERT, Sentence Transformers) to convert text into vectors.

1. **Clustering for Problem Grouping**:
   * Implement **unsupervised clustering techniques** like **K-means**, **Agglomerative Clustering**, or **DBSCAN** to group cases based on similarity in their descriptions and other metadata. These groups will represent common issues users face frequently.
   * **Topic Modeling**: Apply **LDA** or **BERT-LDA** models to identify latent topics from the case descriptions. This will help understand the types of problems that frequently occur and how they relate to KB articles.
   * You may also consider **Non-Negative Matrix Factorization (NMF)** or hybrid approaches if short-text data is challenging to process using traditional LDA models.
2. **Recommendation System**:
   * **Mapping KB Articles**: Use the clustering results and resolution rates to understand which KB articles were most successful in resolving certain types of cases. This forms the basis for recommending KB articles.
   * Use **cosine similarity** or **nearest neighbors** to match a new case’s description to previously resolved clusters and suggest KB articles that led to successful resolutions (i.e., resolution rate = 1).
   * Leverage **pretrained models** like **BERT for semantic similarity** to improve article matching accuracy.
3. **Evaluation and Feedback**:
   * **Performance Evaluation**: Use metrics like precision, recall, and F1-score to evaluate the accuracy of the recommendations. A/B testing with user feedback can help refine the model.
   * **Fine-tuning**: The system should be designed to learn from new cases and continuously update clusters and recommendations based on feedback and new data.

**Hints for Implementation:**

* **Multilingual Data**: Use **machine translation** APIs for consistency but monitor accuracy as certain nuances might be lost during translation.
* **Dimensionality Reduction**: For better clustering performance, use techniques like **PCA**, **t-SNE**, or **UMAP** to reduce the dimensionality of vectorized text data before clustering.
* **Hybrid Topic Modeling**: A combined **BERT-LDA** approach may yield better results than LDA alone, especially for short descriptions.
* **Pretrained Models**: Explore pretrained models such as **BERT**, **GPT-3** or **Universal Sentence Encoder** for semantic understanding and similarity comparisons.
* **Clustering**: Apply **elbow method** or **silhouette scores** to determine the optimal number of clusters for K-means. Hierarchical clustering can be an alternative if the elbow method fails to provide clear results.

This approach will not only group cases by common issues but also provide a framework for automating the recommendation of KB articles based on historical resolution patterns.