Project Report: Task Extraction and Categorization System

---

1. Introduction

The **Task Extraction and Categorization System** aims to automate the process of extracting and categorizing tasks from natural language text. This system can be applied in various domains such as personal productivity, project management, and task scheduling. By analyzing text input, the system identifies tasks, their associated deadlines, and categorizes them into predefined categories such as **Personal**, **Academic**, **Work**, or **Household**. The project uses natural language processing (NLP) techniques, including **spaCy** for text parsing and custom keyword-based rules for task extraction and categorization.

2. Methodology

The approach for developing the Task Extraction and Categorization System is divided into the following main stages:

1. **Text Preprocessing:**
    - **spaCy** is used for tokenization and part-of-speech tagging to process the input text. This allows us to identify subjects, verbs, and other important components of each sentence, which are crucial for extracting tasks and deadlines.
2. **Task Extraction:**
    - The core of task extraction is based on detecting action words that signal the presence of a task. These keywords include phrases like **"has to," "needs to," "must," "should,"** and **"is required to."**
    - Once the keyword is identified, the system extracts the task description following the keyword and also looks for any deadlines or time references.
    - Time patterns, such as specific times (e.g., **“by 5 pm”**), time-related words (e.g., **“tomorrow”**), or relative time (e.g., **“in 3 hours”**), are extracted using regular expressions.
3. **Task Categorization:**
    - After extracting the task and its deadline, the system categorizes the task based on predefined categories. The categories include **Personal**, **Academic**, **Work**, and **Household**. The categorization is done by comparing the task description against a set of predefined keywords.
    - For example, tasks involving submission of assignments or exams are categorized under **Academic**, while tasks like **“buy snacks”** are categorized as **Personal**.
4. **Output Generation:**
    - The extracted tasks are then formatted into a structured JSON output, which includes the task description, the person involved, the deadline, and the category. This structured data format allows for easy consumption and further processing.

3. Insights and Challenges

Throughout the development of this system, several insights and challenges emerged:

- **Insights:**

    1. **Effectiveness of Keyword-Based Extraction:** The system efficiently extracts tasks from structured text with clear action keywords and deadlines. Using a combination of **action keywords** and **time patterns**, the task extraction process is both accurate and easy to implement for simple use cases.

    2. **Task Categorization Success:** The keyword-based categorization worked well for many common task types. It effectively grouped tasks such as **"submit assignment"** or **"buy groceries"** into the correct categories.

- **Challenges:**

    1. **Ambiguity in Text:** One of the key challenges faced was handling ambiguous or complex sentences. For instance, sentences like **"He has to complete the task tomorrow"** may not clearly specify what task is being referred to. In such cases, the system struggles to extract the correct task without further contextual understanding. Improving this part of the system could involve using more advanced NLP techniques like **named entity recognition (NER)** or **dependency parsing**.

    2. **Handling Multiple Tasks in One Sentence:** In cases where a sentence contains multiple tasks, such as **"He has to buy snacks and submit the assignment by 5 pm"**, the system may struggle to extract both tasks separately. Currently, the system processes sentences one at a time, but it could be improved to handle multiple tasks per sentence.

    3. **Complex Time Patterns:** Time extraction based on regular expressions works well for simple cases (e.g., **"by 5 pm"**), but handling more complex time expressions (e.g., **"in 2 days and 3 hours"**) can be difficult. Expanding the system to understand more natural time expressions could improve its performance in such cases.

- **Future Improvements:**

    - **Advanced NLP Models:** Moving beyond keyword-based extraction to more advanced models like **BERT** or **GPT-3** could improve the understanding of complex tasks and enhance context interpretation.

    - **Contextual Task Detection:** By leveraging contextual information and deep learning, the system could be enhanced to extract tasks even when they are less explicitly stated or when the sentences are complex.

4. Conclusion

The Task Extraction and Categorization System successfully automates the process of extracting and organizing tasks from text. While it works well for structured and straightforward text, improvements can be made to handle ambiguous, complex, or multiple-task sentences. This project demonstrated the power of **spaCy** for NLP tasks and established a foundation for developing more advanced task extraction and categorization systems.