

Simple Self-Learning AI System

Un sistema de inteligencia artificial simple que aprende de las interacciones con el usuario, construyendo gradualmente una base de conocimientos y ajustando su personalidad según el feedback recibido.

Características

- **Aprendizaje Continuo:** Almacena y aprende de cada interacción
- **Base de Conocimientos:** Construye una base de datos de temas y conceptos
- **Personalidad Adaptativa:** Ajusta su comportamiento según el feedback
- **Búsqueda de Contexto:** Encuentra interacciones similares pasadas
- **Sin Dependencias Externas:** Solo usa bibliotecas estándar de Python
- **Almacenamiento Persistente:** Usa SQLite para recordar todo

Requisitos

- Python 3.7 o superior
- No requiere bibliotecas externas (solo bibliotecas estándar de Python)

Instalación

1. Clona este repositorio:

```
bash  
  
git clone https://github.com/tu-usuario/simple-self-learning-ai.git  
cd simple-self-learning-ai
```

2. Ejecuta el programa:

```
bash  
  
python simple_ai.py
```

¡Eso es todo! El sistema creará automáticamente la base de datos necesaria.

Uso

Inicio Rápido

Ejecuta el programa:

```
bash
```

```
python simple_ai.py
```

Comandos Disponibles

- **Hacer preguntas:** Simplemente escribe tu pregunta y presiona Enter
- **(stats)**: Muestra estadísticas de aprendizaje
- **(exit)**: Salir del programa

Ejemplo de Conversación

You: What is machine learning?

AI: Based on what I know:

- machine: A system that learns from data and improves over time...

Would you like me to elaborate on any aspect?

Was this helpful? (y/n/skip): y

Thanks! I'll remember that.

Sistema de Feedback

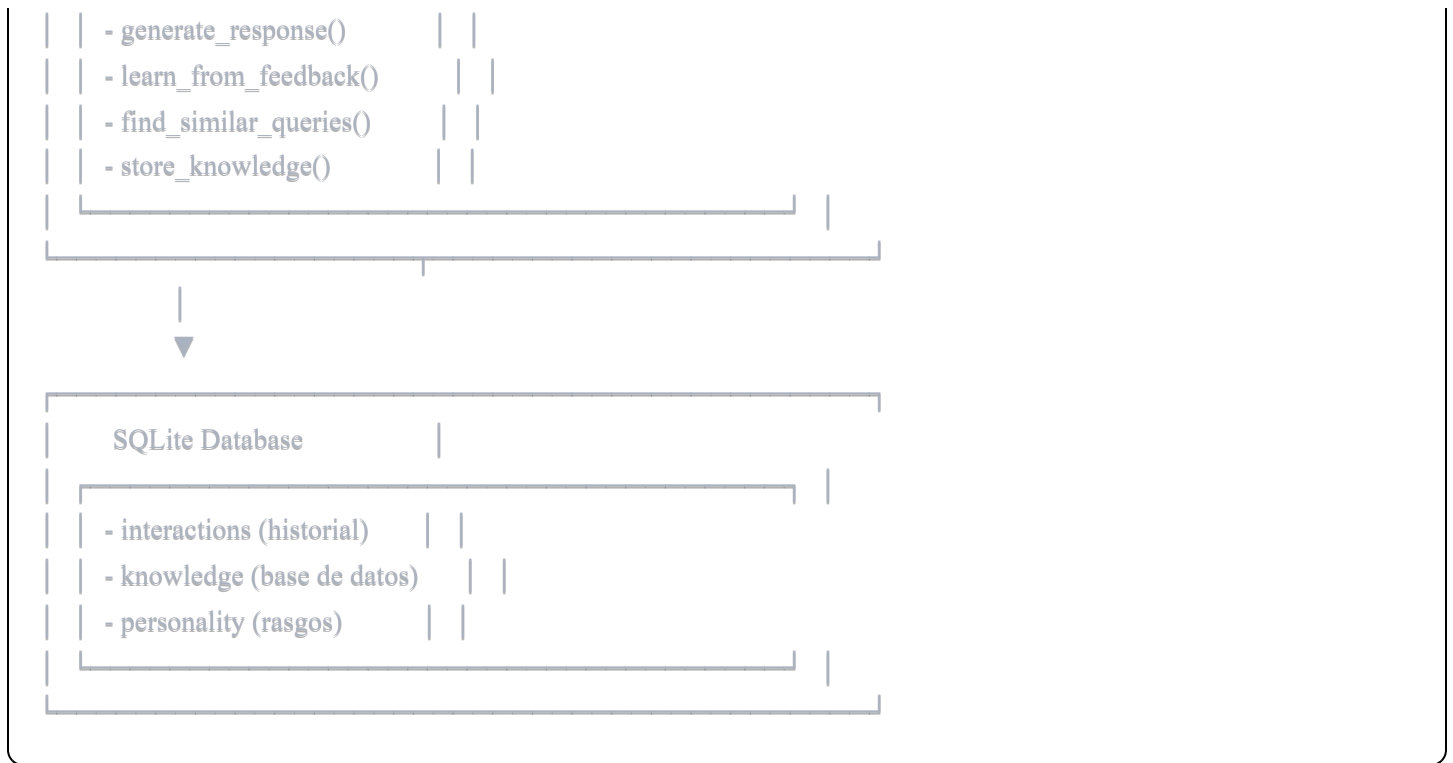
Después de cada respuesta, puedes proporcionar feedback:

- **(y)** (yes): La respuesta fue útil → Refuerza el comportamiento actual
- **(n)** (no): La respuesta no fue útil → Ajusta el enfoque
- **(skip)**: No proporcionar feedback

Cómo Funciona

Arquitectura del Sistema





Base de Datos

El sistema utiliza tres tablas principales:

1. **interactions**

Almacena todas las conversaciones:

- **id**: Identificador único
- **query**: Pregunta del usuario
- **response**: Respuesta generada
- **feedback**: Feedback del usuario (positive/negative)
- **timestamp**: Fecha y hora

2. **knowledge**

Base de conocimientos del sistema:

- **id**: Identificador único
- **topic**: Tema o palabra clave
- **content**: Información sobre el tema
- **confidence**: Nivel de confianza (0.0 - 1.0)
- **timestamp**: Última actualización

3. **personality**

Rasgos de personalidad del AI:

- **trait**: Nombre del rasgo
- **value**: Valor actual (0.0 - 1.0)

Rasgos de Personalidad

El sistema tiene tres rasgos que evolucionan con el uso:

- **Curiosity (0.7)**: Tendencia a hacer preguntas de seguimiento
- **Helpfulness (0.8)**: Disposición para proporcionar información detallada
- **Verbosity (0.6)**: Longitud y detalle de las respuestas

Estos valores se ajustan automáticamente según el feedback recibido.

Proceso de Aprendizaje

1. **Recepción de Query**: El usuario hace una pregunta
2. **Extracción de Keywords**: Se identifican palabras clave importantes
3. **Búsqueda de Contexto**:
 - Se buscan preguntas similares pasadas
 - Se recupera conocimiento relevante de la base de datos
4. **Generación de Respuesta**: Se construye una respuesta contextual
5. **Almacenamiento**: Se guarda la interacción
6. **Feedback**: El usuario evalúa la respuesta
7. **Ajuste**: Se actualizan la personalidad y la base de conocimientos



Estadísticas

Usa el comando **stats** para ver:

- Total de interacciones
- Elementos en la base de conocimientos
- Cantidad de feedback positivo
- Estado actual de la personalidad

Ejemplo:

--- Learning Statistics ---

Total interactions: 45

Knowledge items: 23

Positive feedback: 38

Personality:

Curiosity: 0.75

Helpfulness: 0.82

Verbosity: 0.68

Configuración

Puedes modificar la configuración en el diccionario `CONFIG`:

```
python
```

```
CONFIG = {  
    "database_file": "simple_ai_brain.db", # Nombre del archivo de base de datos  
    "max_memory_items": 500,             # Máximo de interacciones a recordar  
    "learning_rate": 0.05,               # Velocidad de aprendizaje (0.0 - 1.0)  
    "personality": {  
        "curiosity": 0.7,                # Nivel inicial de curiosidad  
        "helpfulness": 0.8,              # Nivel inicial de utilidad  
        "verbosity": 0.6                 # Nivel inicial de verbosidad  
    }  
}
```

Casos de Uso

1. Asistente Personal de Aprendizaje

El AI aprende sobre tus intereses y proporciona información cada vez más relevante.

2. Sistema de Documentación

Alimenta al AI con información sobre un proyecto y úsalo como referencia interactiva.

3. Chatbot Educativo

Perfecto para crear un asistente que mejora con el tiempo para temas específicos.

4. Prototipo de Investigación

Base simple para experimentar con sistemas de aprendizaje conversacional.

Estructura del Proyecto

```
simple-self-learning-ai/
|
├── simple_ai.py      # Código principal del sistema
├── README.md         # Este archivo
├── LICENSE           # Licencia del proyecto
├── .gitignore        # Archivos a ignorar en Git
└── simple_ai_brain.db # Base de datos (se crea automáticamente)
```

Gestión de Datos

Ubicación de la Base de Datos

Por defecto, la base de datos se crea en el mismo directorio que `simple_ai.py` con el nombre `simple_ai_brain.db`.

Limpieza Automática

El sistema automáticamente elimina interacciones antiguas cuando se supera el límite de `max_memory_items` (500 por defecto), manteniendo solo las más recientes.

Backup Manual

Para hacer un backup de tu base de datos:

```
bash

cp simple_ai_brain.db simple_ai_brain_backup_$(date +%Y%m%d).db
```

Resetear el Sistema

Para empezar de nuevo, simplemente elimina el archivo de base de datos:

```
bash

rm simple_ai_brain.db
```

Desarrollo y Extensiones

Agregar Nuevos Rasgos de Personalidad

1. Modifica el diccionario `CONFIG`:

```
python
```

```
"personality": {  
    "curiosity": 0.7,  
    "helpfulness": 0.8,  
    "verbosity": 0.6,  
    "humor": 0.5 # Nuevo rasgo  
}
```

2. Implementa la lógica en `generate_response()` y `learn_from_feedback()`

Integrar APIs Externas

Puedes extender el método `generate_response()` para consultar APIs:

```
python
```

```
def generate_response(self, query):  
    # Lógica existente...  
  
    # Agregar búsqueda en API  
    if self.needs_external_info(query):  
        external_data = self.fetch_from_api(query)  
        response_parts.append(external_data)  
  
    return "\n".join(response_parts)
```

Agregar Procesamiento de Lenguaje Natural

Para análisis más avanzado, considera agregar spaCy o NLTK:

```
python
```

```
import spacy  
  
def extract_keywords(self, text):  
    nlp = spacy.load("es_core_news_sm")  
    doc = nlp(text)  
    return [token.text for token in doc if token.pos_ in ['NOUN', 'VERB']]
```

🍌 Contribuciones

Las contribuciones son bienvenidas! Por favor:

1. Haz fork del repositorio
2. Crea una rama para tu feature (`git checkout -b feature/AmazingFeature`)
3. Commit tus cambios (`git commit -m 'Add some AmazingFeature'`)
4. Push a la rama (`git push origin feature/AmazingFeature`)
5. Abre un Pull Request

Áreas de Mejora

- ☐ Soporte multiidioma
- ☐ Integración con APIs de AI (OpenAI, Anthropic)
- ☐ Interfaz web con Flask/FastAPI
- ☐ Exportar conversaciones a diferentes formatos
- ☐ Sistema de plugins
- ☐ Análisis de sentimientos
- ☐ Respuestas más contextuales usando embeddings

Licencia

Este proyecto está bajo la Licencia MIT. Consulta el archivo `LICENSE` para más detalles.

Solución de Problemas

El programa no inicia

- Verifica que tienes Python 3.7 o superior: `python --version`
- Asegúrate de estar en el directorio correcto

Error de permisos en la base de datos

- Verifica que tienes permisos de escritura en el directorio
- En Linux/Mac: `chmod 755 .`

La base de datos crece mucho

- Reduce `max_memory_items` en CONFIG
- Elimina la base de datos y empieza de nuevo

Las respuestas no mejoran

- Proporciona más feedback consistente (y/n)
- Dale tiempo al sistema para aprender (mínimo 20-30 interacciones)

- Verifica que `learning_rate` no sea demasiado bajo

Recursos Adicionales

- [Documentación de SQLite](#)
- [Tutorial de Python](#)
- [Machine Learning Básico](#)

Autor

Tu Nombre - [@tu_twitter](#)

Link del Proyecto: <https://github.com/tu-usuario/simple-self-learning-ai>

Agradecimientos

- Inspirado en sistemas de aprendizaje conversacional
- Gracias a la comunidad de Python por las excelentes bibliotecas estándar
- Contribuidores y testers

★ Si este proyecto te fue útil, considera darle una estrella en GitHub!