Escuela Superior
de Ingeniería y Tecnología
Universidad de La Laguna

Sistemas de recomendación: Modelo basado en contenido

Gestión del conocimiento para las organizaciones

Alejando Martín Linares
alu0101476476@ull.edu.es
Guillermo Emmanuel González Méndez
alu0101466941@ull.edu.es
Joel Saavedra Páez
alu0101437415@ull.edu.es

ÍNDICE

Implementación y desarrollo

Hemos desarrollado la aplicación utilizando el lenguaje C++ debido a que los integrantes del grupo tenemos un conocimiento amplio en dicho lenguaje, además de aprovechar su rápida ejecución.

El programa sigue un estilo POSIX para las entradas y proporciona la salida de los datos mediante consola. **Los parámetros de entrada se especifican en el README del repositorio**.

Ficheros y contenidos

El programa cuenta con 4 archivos principales de código (*main.cc, document.cc, documentManager.cc y tools.cc)* y 2 archivos headers (*recommender-system.h y tools.h)*.

main.cc

Contiene la función principal. Toma los parámetros por línea de comando, llama al constructor de las clases que trataran los documentos individualmente y mediante otra clase se harán las recomendaciones mediante coseno.

document.cc

Implementación de los métodos de la clase ***Document*** que representa y contiene todos los datos relacionados a un documento individual. Su función es contener y agrupar todos los datos del documento y tratar su contenido mediante funciones de lematización y tokenización. Además también se encarga de calcular el *TF* y el *TF normalizado*.

La decisión arquitectónica más significativa es mantener una representación dual del texto mediante originalText_ y simplifiedText_, ambos como vectores bidimensionales de strings donde cada vector interno representa una línea del documento. Esta dualidad permite preservar el contenido original mientras se aplican transformaciones progresivas (limpieza, eliminación de stopwords, lematización) al texto simplificado, facilitando la trazabilidad y comparaciones. El constructor carga el documento línea por línea tokenizando por espacios en blanco, con una estrategia de manejo de errores que termina la ejecución inmediatamente si el archivo no puede abrirse, una decisión simple pero poco flexible que podría mejorarse con excepciones.

Una decisión de diseño fundamental es que el cálculo de TF inicializa TODOS los términos del corpus a 0.0, incluso aquellos que no aparecen en el documento específico, garantizando vectores de dimensión uniforme para todos los documentos del sistema, requisito esencial para comparaciones de similitud. La

normalización vectorial se realiza mediante la norma euclidiana (L2), calculando la longitud del vector como la raíz cuadrada de la suma de cuadrados de los TF, y posteriormente dividiendo cada TF por esta longitud para obtener el TF normalizado. El cálculo de índices de términos almacena únicamente la primera aparición de cada palabra usando un map<string, pair<int, int>>, con el valor centinela (-1, -1) para indicar términos del corpus no encontrados en el documento.

Sin embargo, presenta un acoplamiento intencional con el concepto de corpus: el documento requiere conocer el conjunto completo de palabras del corpus (allWordsInCorpus_) establecido mediante un setter después de la construcción, lo que separa la creación del objeto de la configuración del contexto pero crea una dependencia explícita del vocabulario global.

documentManager.cc

La clase ***DocumentManager*** actúa como orquestador del sistema de recomendación, gestionando una colección de documentos y coordinando todos los cálculos necesarios. El constructor realiza una carga secuencial en tres fases: primero carga las stop words en un set para búsquedas eficientes, luego parsea el archivo JSON de lematización mediante un parser manual implementado desde cero (evitando dependencias externas), y finalmente procesa cada documento aplicando la pipeline de transformaciones (limpieza, lematización, eliminación de stop words) mientras construye simultáneamente el vocabulario del corpus. Una decisión arquitectónica clave es el procesamiento en dos pasadas: primero se procesan todos los documentos individualmente, y luego se establece el corpus completo en cada documento mediante setters, garantizando que todos compartan el mismo espacio vectorial.

El método Recommend() implementa la secuencia de cálculos del modelo TF-IDF y similitud coseno, ejecutando en orden los cálculos de índices de términos, TF, longitud vectorial, TF normalizado, IDF y finalmente la matriz de similitud. CountDocumentsOccurrences() utiliza un set para contar en cuántos documentos aparece cada término sin duplicados. La matriz de similitud coseno se calcula mediante producto escalar de vectores TF normalizados, aprovechando que estos ya están normalizados por la norma L2, lo que simplifica el cálculo al eliminar la necesidad de dividir por las magnitudes de los vectores.

El parser JSON personalizado en LoadLemmatizationRules() recorre el archivo carácter por carácter identificando pares clave-valor con evaluación exhaustiva que termina el programa ante cualquier malformación del JSON.. El operador << sobrecargado genera una tabla formateada con información completa de TF, IDF, TF-IDF y posiciones de términos para cada documento, seguida de la matriz de similitud, proporcionando una salida comprensiva para análisis del sistema de recomendación.

tools.cc

El archivo tools.cc implementa la interfaz de línea de comandos del sistema, siguiendo un patrón de validación estricta con mensajes de error descriptivos y ayuda detallada. La función CheckArguments() retorna una estructura CommandLineArgs que encapsula los parámetros validados, usando un parser que reconoce tres opciones obligatorias (-d, -s, -l) con validación de duplicados, existencia de archivos y completitud de argumentos requeridos.

El diseño del parser permite múltiples documentos después de -d mediante un bucle que consume argumentos hasta encontrar otro flag (detectado por el carácter -), mientras que -s y -l esperan exactamente un archivo cada uno. Cualquier error en la validación (opciones desconocidas, archivos inexistentes, parámetros faltantes) resulta en una llamada a ErrorOutput() que muestra el uso correcto y termina la ejecución, mientras que los flags --help o -h invocan HelpOutput() con documentación completa del programa.

La estrategia de salida inmediata ante errores mediante exit() en lugar de excepciones mantiene consistencia con el resto del sistema pero limita la testabilidad y reutilización del código. Las funciones de ayuda y error están diseñadas para nunca retornar, clarificando el flujo de control y evitando estados inconsistentes. La verificación de apertura de archivos durante el parsing previene errores posteriores, implementando validación temprana de precondiciones.

1. Análisis de los resultados para todos los documentos de ejemplo (document-01.txt hasta document-10.txt)

Resultado de la ejecución

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	Doc 9	Doc 10
Doc 1	1.000000	0.670136	0.655077	0.685978	0.701553	0.708060	0.677567	0.638804	0.679728	0.656513
Doc 2	0.670136	1.000000	0.641121	0.653204	0.687237	0.700249	0.654785	0.660759	0.646779	0.633017
Doc 3	0.655077	0.641121	1.000000	0.678387	0.706171	0.705951	0.689920	0.676126	0.683954	0.681578
Doc 4	0.685978	0.653204	0.678387	1.000000	0.719201	0.735106	0.713111	0.723936	0.712314	0.666965
Doc 5	0.701553	0.687237	0.706171	0.719201	1.000000	0.749090	0.725626	0.703316	0.726314	0.697514
Doc 6	0.708060	0.700249	0.705951	0.735106	0.749090	1.000000	0.724556	0.702727	0.726726	0.714575
Doc 7	0.677567	0.654785	0.689920	0.713111	0.725626	0.724556	1.000000	0.725567	0.749394	0.703019

Doc 8	0.638804	0.660759	0.676126	0.723936	0.703316	0.702727	0.725567	1.000000	0.717825	0.666666
Doc 9	0.679728	0.646779	0.683954	0.712314	0.726314	0.726726	0.749394	0.717825	1.000000	0.705021
Doc 10	0.656513	0.633017	0.681578	0.666965	0.697514	0.714575	0.703019	0.666666	0.705021	1.000000

Procesamiento

El sistema procesó todos los documentos resultando en el vocabulario del corpus con todos los términos únicos encontrados en los 10 documentos, inicializando cada documento con este vocabulario completo para garantizar vectores de dimensión uniforme necesarios para las comparaciones.

Matriz de Similitud Coseno

La matriz de similitud muestra valores de coseno entre 0 y 1, donde 1 indica documentos idénticos. Los resultados revelan que Doc 6 y Doc 5 tienen la mayor similitud con 0.749090, seguidos de Doc 7 y Doc 9 con 0.749394, y Doc 4 y Doc 6 con 0.735106. Esto indica que estos pares de documentos comparten vocabulario y temática muy similar. Por otro lado, Doc 2 y Doc 10 presentan la menor similitud con 0.633017, sugiriendo contenido más divergente. Todos los documentos muestran similitudes moderadas-altas (entre 0.63 y 0.75), lo que indica que comparten un dominio temático común pero con variaciones en el contenido específico.

Interpretación para Recomendaciones

El sistema permite identificar documentos similares: si un usuario está interesado en Doc 5, el sistema recomendaría Doc 6, Doc 4, Doc 7 y Doc 9 como los más relacionados (similitudes > 0.72). Los valores en la diagonal de la matriz son 1.0 (cada documento consigo mismo), y la matriz es simétrica reflejando que la similitud entre Doc A y Doc B es igual que entre Doc B y Doc A. Los términos con IDF alto (cercano a 1.0) son discriminatorios ya que aparecen en pocos documentos, mientras que términos con IDF bajo (cercano a 0.0) aparecen en muchos documentos y aportan menos a la diferenciación entre textos.

2. Análisis de los resultados para los documentos añadidos en español (document-01.txt hasta document-10.txt)

Resultado de la ejecución

	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	Doc 9	Doc 10
Doc 1	1.000000	0.335152	0.319940	0.294087	0.346309	0.291842	0.176012	0.275057	0.194175	0.186263
Doc 2	0.335152	1.000000	0.285330	0.316863	0.302307	0.225450	0.168763	0.258265	0.219496	0.182800
Doc 3	0.319940	0.285330	1.000000	0.276814	0.303579	0.239005	0.167896	0.236917	0.188774	0.179437
Doc 4	0.294087	0.316863	0.276814	1.000000	0.331839	0.216960	0.184145	0.266451	0.196112	0.156831
Doc 5	0.346309	0.302307	0.303579	0.331839	1.000000	0.269150	0.188895	0.235346	0.196240	0.173472
Doc 6	0.291842	0.225450	0.239005	0.216960	0.269150	1.000000	0.158765	0.231977	0.208131	0.203033
Doc 7	0.176012	0.168763	0.167896	0.184145	0.188895	0.158765	1.000000	0.156439	0.186902	0.156326

Doc 8	0.275057	0.258265	0.236917	0.266451	0.235346	0.231977	0.156439	1.000000	0.225685	0.182964
Doc 9	0.194175	0.219496	0.188774	0.196112	0.196240	0.208131	0.186902	0.225685	1.000000	0.173486
Doc 10	0.186263	0.182800	0.179437	0.156831	0.173472	0.203033	0.156326	0.182964	0.173486	1.000000

El resultado muestra la ejecución del sistema de recomendación basado en contenido aplicado a 10 documentos en español (esp-01.txt a esp-10.txt), utilizando el corpus de lematización español (corpus-es.json) y las palabras vacías en español (stop-words-es.txt).

Características de los Documentos
Los documentos en español presentan vocabularios más ricos comparados con los documentos en inglés del análisis anterior. Por ejemplo, el documento esp-01.txt contiene términos característicos como "axolotl", "lago", "cuaderno", "escribir", "sendero", "árboles", que sugieren una narrativa literaria o descriptiva. Los valores TF-IDF más altos indican términos distintivos de cada documento: términos con IDF=1.000000 son únicos de un documento, mientras que términos con IDF más bajo (como "a", "de", "que", "y") aparecen en todos los documentos.

Matriz de Similitud Coseno
La matriz 10×10 revela las similitudes semánticas entre pares de documentos. Los valores van desde 0.156326 (Doc 7-Doc 10, menor similitud) hasta 1.000000 (diagonal, cada documento consigo mismo). Las similitudes más altas fuera de la diagonal son:

- Doc 1-Doc 5: 0.346309 (mayor similitud entre documentos diferentes)
- Doc 1-Doc 2: 0.335152
- Doc 4-Doc 5: 0.331839
- Doc 2-Doc 4: 0.316863

Interpretación del Sistema de Recomendación
El Documento 1 es el más versátil: tiene similitudes relativamente altas con múltiples documentos (esp. Doc 5, 2, 3), sugiriendo contenido temático amplio. El Documento 7 muestra el mayor aislamiento temático: sus similitudes con otros documentos son consistentemente las más bajas (rango 0.156-0.189), indicando que trata temas distintivos no compartidos con el resto del corpus.

Para un sistema de recomendación basado en contenido, si un usuario lee el Documento 1, el sistema recomendaría en orden: Doc 5 (34.6% similar), Doc 2 (33.5%), Doc 5 (30.4%), o Doc 4 (29.4%). Si lee el Doc 7, las recomendaciones serían más débiles debido a su baja similitud global con todos los demás documentos.