

Sistemas de recomendación: Modelo basado en contenido

Gestión del conocimiento para las
organizaciones

Alejandro 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	2
Ficheros y contenidos	2
main.cc	2
document.cc	2
documentManager.cc	3
tools.cc	3
1. Análisis de los resultados para todos los documentos de ejemplo (document-01.txt hasta document-10.txt)	4
2. Análisis de los resultados para los documentos añadidos en español (document-01.txt hasta document-10.txt)	6



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 tratarán 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*.

Se mantiene la representación original del texto y la simplificada mediante *originalText_* y *simplifiedText_*, ambos como vectores bidimensionales de strings donde cada vector interno representa una línea del documento. Esto permite preservar el contenido original mientras se aplican transformaciones progresivas (limpieza, eliminación de stopwords, lematización) al texto simplificado.

Es importante que el map de TF(término-TF) almacene todos los términos del corpus, incluso aquellos que no aparecen en el documento específico, esto es para poder comparar los documentos. Este vector se normaliza 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.



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 las transformaciones (limpieza, lematización, eliminación de stopwords), además construye el vocabulario del corpus.

El método Recommend() implementa la secuencia de cálculos del modelo TF-IDF y similaridad coseno. Para cada documento se ejecutan los siguientes cálculos: índices de términos, TF, longitud del vector y TF normalizado, a nivel global se calcula 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.

El parser JSON personalizado en LoadLemmatizationRules() recorre el archivo carácter por carácter identificando pares clave-valor, termina el programa ante cualquier error 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.

tools.cc

El archivo tools.cc contiene la comprobación y el manejo de errores de la entrada del programa. La función CheckArguments() es la encargada de tratar los argumentos introducidos por el usuario por línea de comandos y asegurarse que se usan las opciones obligatorias (-d, -s, -l). Retorna una estructura CommandLineArgs que encapsula los parámetros validados(documentos, documento de stopword y documento de lematización).

Se permiten múltiples documentos después de -d mediante un bucle que consume argumentos hasta encontrar otro flag (-s o -l), 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.



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

[Enlace al resultado](#)

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	1.0000 00	0.6701 36	0.655 077	0.685 978	0.701 553	0.708 060	0.6775 67	0.638 804	0.679 728	0.656 513
Doc 2	0.6701 36	1.0000 00	0.6411 21	0.653 204	0.687 237	0.700 249	0.6547 85	0.660 759	0.646 779	0.633 017
Doc 3	0.6550 77	0.6411 21	1.000 000	0.678 387	0.706 171	0.705 951	0.6899 20	0.676 126	0.683 954	0.681 578
Doc 4	0.6859 78	0.6532 04	0.678 387	1.000 000	0.719 201	0.735 106	0.7131 11	0.723 936	0.712 314	0.666 965
Doc 5	0.7015 53	0.6872 37	0.706 171	0.719 201	1.000 000	0.749 090	0.7256 26	0.703 316	0.726 314	0.697 514
Doc 6	0.7080 60	0.7002 49	0.705 951	0.735 106	0.749 090	1.000 000	0.7245 56	0.702 727	0.726 726	0.714 575
Doc 7	0.6775 67	0.6547 85	0.689 920	0.713 111	0.725 626	0.724 556	1.0000 00	0.725 567	0.749 394	0.703 019



Doc 8	0.6388 04	0.6607 59	0.676 126	0.723 936	0.703 316	0.702 727	0.7255 67	1.000 000	0.717 825	0.666 666
Doc 9	0.6797 28	0.6467 79	0.683 954	0.712 314	0.726 314	0.726 726	0.7493 94	0.717 825	1.000 000	0.705 021
Doc 10	0.6565 13	0.6330 17	0.681 578	0.666 965	0.697 514	0.714 575	0.7030 19	0.666 666	0.705 021	1.000 000

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 7 y Doc 9 con 0.749394 tienen la mayor similitud seguidos de Doc 6 y Doc 5 con 0.749090 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 9, Doc 7 y Doc 4 como los más relacionados ($\text{similitudes} > 0.72$). 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 (cerca de 1.0) aparecen en pocos documentos, mientras que términos con IDF bajo (cerca de 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)

[Enlace al resultado](#)

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	1.0000 00	0.335 152	0.319 940	0.294 087	0.3463 09	0.291 842	0.176 012	0.275 057	0.1941 75	0.1862 63
Doc 2	0.3351 52	1.000 000	0.285 330	0.316 863	0.3023 07	0.225 450	0.168 763	0.258 265	0.2194 96	0.1828 00
Doc 3	0.3199 40	0.285 330	1.000 000	0.276 814	0.3035 79	0.239 005	0.167 896	0.236 917	0.1887 74	0.1794 37
Doc 4	0.2940 87	0.316 863	0.276 814	1.000 000	0.3318 39	0.216 960	0.184 145	0.266 451	0.1961 12	0.1568 31
Doc 5	0.3463 09	0.302 307	0.303 579	0.331 839	1.0000 00	0.269 150	0.188 895	0.235 346	0.1962 40	0.1734 72
Doc 6	0.2918 42	0.225 450	0.239 005	0.216 960	0.2691 50	1.000 000	0.158 765	0.231 977	0.2081 31	0.2030 33



Doc 7	0.1760 12	0.168 763	0.167 896	0.184 145	0.1888 95	0.158 765	1.000 000	0.156 439	0.1869 02	0.1563 26
Doc 8	0.2750 57	0.258 265	0.236 917	0.266 451	0.2353 46	0.231 977	0.156 439	1.000 000	0.2256 85	0.1829 64
Doc 9	0.1941 75	0.219 496	0.188 774	0.196 112	0.1962 40	0.208 131	0.186 902	0.225 685	1.0000 00	0.1734 86
Doc 10	0.1862 63	0.182 800	0.179 437	0.156 831	0.1734 72	0.203 033	0.156 326	0.182 964	0.1734 86	1.0000 00

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. 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 1-Doc 3: 0.319940

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%) y Doc 3 (31.9%). Si lee el Doc 7, las recomendaciones serían más débiles debido a su baja similitud global con todos los demás documentos.