Manual de compilació i execució

Aquest document explica com compilar i executar el programa que hem desenvolupat. Per facilitar aquest procés, hem creat un Makefile, que automatitza la compilació del projecte.

Compilació

Per compilar el programa complet, simplement cal obrir un terminal i situar-nos a la carpeta del projecte i executar:

make

Aquest comandament compilarà automàticament tots els fitxers necessaris.

Execució

Un cop compilat, el programa es pot executar amb la següent comanda:

./ProjecteA

Estructura i funcionalitat del programa

El programa està estructurat en tres etapes principals:

Treballar amb dos documents

Aquesta etapa permet provar els diferents algorismes implementats amb només dos documents.

Ha estat pensada com una fase preliminar per verificar el correcte funcionament dels algorismes abans de treballar amb conjunts més grans de documents.

Treballar amb documents generats a partir d'un document base

Quan es treballa amb múltiples documents, el conjunt pot estar format per documents generats a partir d'un document base. En aquest cas, els nous documents es creen aplicant permutacions directament sobre el text original del document base.

La implementació de la funció utilitzada per generar aquestes permutacions es troba en el fitxer **permutacions.cc**.

Treballar amb documents virtuals

A més de treballar amb documents reals o generats per permutació, el programa també permet l'ús de documents virtuals. En aquest cas, primer s'extreuen els *k-shingles* d'un document base i, posteriorment, es generen subconjunts d'aquests *k-shingles* mitjançant permutacions.

El nombre de documents virtuals a generar ha de ser D ≥ 20, valor que es pot escollir segons les necessitats de l'experiment. Aquest enfocament permet controlar la similitud entre els documents i comparar la seva aproximació teòrica amb els resultats obtinguts pels diferents algorismes implementats.

Procés previ a la generació de k-shingles

Abans de generar els *k-shingles*, el programa realitza un procés de normalització del text. Aquest procés està implementat en dos fitxers:

- normalizeText.cc → Normalitza el text aplicant les següents transformacions:
  - Converteix tot el text a minúscules.
  - Substitueix múltiples espais en blanc consecutius per un únic espai.

- normalizeText2.cc → Gestiona les *stopwords* mitjançant el següent procediment:
  - Detecta l'idioma del document.
  - Un cop detectat l'idioma, elimina les *stopwords* corresponents.

Aquest pas de preprocessament és fonamental per garantir la consistència i eficiència en la generació dels *k-shingles*.

Generació de k-shingles

Un cop normalitzat el text, es generen els *k-shingles* a partir d'ell. Aquest procés està implementat en el fitxer k-shingles.cc i rep com a paràmetres:

1. El valor de k, que determina la mida de cada grup de *shingles*.
2. El text normalitzat, resultant del procés de preprocessament descrit anteriorment.

Aquest mòdul s'encarrega de segmentar el text en conjunts de *k-shingles*, que seran posteriorment utilitzats pels diferents algorismes implementats.

Ubicació dels textos utilitzats

Tots els documents emprats en els experiments es troben a la carpeta input_texts. A més, en la documentació del projecte s'especifica detalladament quins documents han estat utilitzats en cada experiment, així com els diferents paràmetres configurats per a cada prova.

Algorismes implementats

El programa incorpora els següents algorismes per al tractament de documents:

- **Jaccard** → Implementat al fitxer `jaccard.cc`
- **MinHash** → Implementat al fitxer `minhash.cc`
- **LSH (Locality-Sensitive Hashing)** → Implementat al fitxer `lsh.cc`

Tots els fitxers `.cc` i `.hh` necessaris es troben dins l'arxiu `.tar` proporcionat.