Progetti Information Retrieval – A.A. 2019 – 2020

Giorgio Gambosi – Danilo Croce

1. **Completamento esercizi di laboratorio.** Sviluppare un sistema di IR basato su Vector Space Model a partire dal materiale visto durante i laboratori
   - Sistema di indicizzazione non posizionale ma con *query* complesse
   - Sistema di indicizzazione posizionale senza *query* complesse
   - Sistema di ranking basato su *Vector Space Model*

   Definire un unico sistema in grado di combinare i contenuti dei 3 laboratori rispondendo agli esercizi proposti durante le lezioni (vedi *python book*).
   Valutare il sistema su benchmark fornito dal docente (Cranfield).
   Difficolta 1/3, Gruppo max: 1
2. **Sviluppo di un Sistema di IR per la ricerca documentale.** Implementare un sistema di Retrieval per la ricerca documentale, come ad esempio:
   - Liber Liber (https://www.liberliber.it)
   - Wikipedia (selezionando un sottoinsieme di pagine concordate con il docente)
   - **Materiale didattico dei docenti del Corso di Studi di Informatica**

   Il sistema deve essere completo di interfaccia grafica per eseguire richieste e visualizzare l'elenco (dovutamente paginato) dei testi recuperati.
   Possibili implementazioni:
   - Python: Whoosh
   - Java: Lucene / SOLR

   Difficolta 2/3, Gruppo max: 2
3. **Implementazione di Algoritmi per IR in ambienti distribuiti.** Implementazione in ambiente Spark di algoritmi distribuiti per il Retrieval
   - Calcolo del Word Count (visto a lezione)
   - Implementazione di Singular Value Decomposition
   - …

   Difficolta 3/3, Gruppo max: 2
4. **Implementare un sistema di Retrieval per la aggregazione di statistiche su Twitter.** Il sistema deve essere completo di interfaccia grafica per eseguire *query* per il recupero dei messaggi. Inoltre l'applicativo deve consentire la creazione di report che sintetizzino le informazioni nei messaggi, come ad esempio:
   - *Timeline* dei *tweet* che soddisfano una certa *query*
   - *Timeline* dei *topic* (hashtag) più discussi
   - Utenti più citati

   Possibili implementazioni:
   - Python: Whoosh
   - Java: Lucene / SOLR

   Opzionale: addestrare un classificatore in grado di identificare il "Sentiment" del singolo messaggio, da usare come meta-dato per arricchire la collezione di messaggi
   Difficolta 3/3, Gruppo max: 2 (Gruppo massimo opzionale: 3)

- Si accettano proposte di progetto.

Durante la discussione del progetto è richiesta una presentazione di al più 15 slide.