Partie 1 du Projet

Indexation des documents et Recherche d'Information

- **Ce projet est évalué sur 30 points avec une pondération de 30% de la note globale.**
- **Date de remise du fichier de présentation (PDF ou PPT), au plus tard le vendredi 8 décembre 2023, 9h.**
- **Présentation du projet : Vendredi 8 et 15 décembre 2023**
- **Date de remise du rapport final du projet, au plus tard le mercredi 20 décembre 2023**

Tous les envois doivent se faire à cette adresse : inf7546tal@gmail.com

• Objectif du Projet

Le but de ce projet est de réaliser un système de recherche de documents monolingue en utilisant un système RI libre d'accès (*open source)*, en vous familiarisant avec ce qui suit :

(1) une des procédures de base en recherche d'information (RI) – l'indexation des documents;
(2) la recherche d'information en utilisant un des systèmes suivants: Solr, Lucene, Lemur, ou tout autre système RI basé sur le modèle vectoriel;
(3) Amélioration de la recherche d'information monolingue en utilisant une méthode de reformulation ou expansion de requêtes.

• La Collection de Données

Une collection de données (requêtes, documents, jugements de pertinence) en format texte est mise à votre disposition. La collection de données à utiliser (TREC AP88-90) est constituée d'un ensemble d'articles de nouvelles publiées par Associated Press en 1988-1990. Cette collection comporte ce qui suit :

- des documents (AP.tar);
- un ensemble de requêtes en Anglais (ex. topics.1-50.txt, etc.);
  Note : *Dans cette collection, il ya 150 requêtes qui servent à déterminer la précision réelle d'un système RI, entre autres.*
- un ensemble de jugements de pertinence (ex. qrels.1-50.AP8890.text, etc.).

Emplacement de la collection de donnés :
https://drive.google.com/file/d/1cwskOAZ9gHAgHl_nPtKfwxSJQ6lR33KI/view?usp=sharing

- **Description du Projet (partie 1)**

Un système de recherche d'information monolingue doit réaliser deux opérations de base: (1) indexation des documents (et/ou de la requête) et (2) recherche d'information en mode monolingue. Certains choix sont possibles pour ces deux opérations. Pour l'indexation, on peut choisir à utiliser ou non une stop liste, un algorithme de racinisation (stemming) ou de lemmatisation. Pour la recherche, le modèle vectoriel, est choisi afin de pondérer les termes et de comparer la requête avec les documents de la collection.

Note : *L'énoncé de ce projet encourage l'utilisation du système de recherche d'information Solr ou LUCENE mais vous pouvez opter pour un autre choix open source (Lemur, etc.) basé sur le modèle vectoriel.*

1. Indexation des documents et système RI

Dans cette partie du projet, il s'agit d'installer et d'utiliser le système Solr attaché à Lucene ou Lucene, qui et basé sur un modèle vectoriel dans un environnement monolingue (anglais). Tout autre système RI est possible.

1.1. Le modèle à implanter – Solr ou LUCENE,

On vous demande d'implanter le modèle d'indexation et de recherche des documents avec *trois schémas de pondération différents* existants dans le modèle ou programmés par vous.

Vous devez exécuter les tâches suivantes:

1. Installer LUCENE/Solr sur votre ordinateur;
2. Indexer la collection de documents AP avec différentes options :
   a) Sans prétraitement - Ne rien utiliser (approche nommée Baseline).
   b) Avec Prétraitement : Utiliser une stop liste et un stemming (Porter, krovetz, etc) ou une lemmatisation (approche nommée Baseline-pre-process).

1.2. Pondération des termes dans les documents

On vous demande d'utiliser trois schémas de pondérations différents du système RI choisi, ex. pondération par la fréquence normalisée, pondération par tf*idf normalisée, Okapi BM 25, etc…
Au moins trois schémas de pondération devraient être utilisés.

Pour plus d'informations sur l'utilisation des schémas de pondération du système de RI SMART, vous référez à l'article suivant (Inkpen et al.):
https://drive.google.com/file/d/1z0zV8poqgoRlDgaWFrJQ04eHIckrGD7x/view?usp=sharing
Vous devez réaliser le processus de recherche en vous basant sur les différents résultats d'indexation que vous avez produits. Votre programme (via une interface) doit accepter une requête en langue source (anglais), et produire une liste ordonnée de documents dans la même langue source (anglais), comme réponse.

In, out
main.py

1.3.Collection de documents

La collection de test TREC AP88-90 consiste en un ensemble d'articles de nouvelles publié par *Associated Press* en 1988-1990, un ensemble de requêtes (un nombre de 150) et les jugements de pertinences (fichier qrels). Ces requêtes correspondent aux 3 premières années des compétitions TREC.

Pour chaque requête, des jugements de pertinence (fichiers qrels) sont utilisés pour évaluer chaque système RI développé. Dans un fichier qrels, chaque ligne correspond à un jugement de pertinence (0 ou 1) entre une requête et un document.

1.4. Les étapes à suivre et analyse des résultats

Vous devez réaliser les traitements expliqués en haut, et compléter avec les étapes suivantes:

- construction de deux jeux de requêtes en anglais de la façon suivante :
  - requêtes courtes : avec le champs <title> des topics
  - requêtes longues : avec le champs <title> + <desc> des topics

courtes
longues

- réalisation des étapes de prétraitement sur ces requêtes et documents (anglais) : tokénization, élimination des stop words (stop list), lemmatisation ou racinisation, etc.
- indexation des documents en utilisant les trois (au moins) schémas de pondération du système RI. Vous pouvez utiliser la structure de données de votre choix pour une recherche rapide (fichier, table hash, listes liées, base de données, etc.).
- recherche d'information avec les requêtes (en anglais), documents et le système RI. Une liste ordonnée des documents obtenus comme réponse doit être réalisée.
- analyse des résultats.

rapport

L'analyse des résultats est une partie importante de l'expérimentation afin de comprendre mieux les problèmes et ainsi pouvoir proposer des solutions adéquates.
Dans ce projet, on tente de comprendre en premier lieu, l'effet du pré-traitement en utilisant la racinisation ou la lemmatisation ainsi qu'une stop liste ; et en deuxième lieu, l'utilisation de différents schémas de pondération.

map 1000 doc.

Pour simplifier la comparaison, vous allez vous fier à la mesure MAP (moyenne des précisions moyennes) aux 1000 documents (c'est-à-dire de calculer MAP pour une liste de réponses de 1000 documents). C'est la mesure classique utilisée dans TREC. Cependant, d'autres mesures peuvent aussi donner des indications intéressantes. Par exemple, P@10 peut indiquer si une méthode est performante pour les premières réponses (utilisée surtout pour une recherche orientée précision).

Ainsi, les étapes suivantes devraient être suivies :

1. Comparer avec les jugements de pertinence en utilisant le programme ou script trec_eval que vous trouverez à cette adresse : http://trec.nist.gov/trec_eval/

2. Inclure les résultats sous forme de tableau comparatif, en utilisant la MAP (moyenne des précisions moyennes) et les trois schémas de pondérations de votre système RI. Vous devez aussi inclure les résultats en terme de Précision-Rappel et la courbe Rappel-Précision pour chaque méthode utilisée (les différentes courbes en un seul graphe).
3. Discuter les résultats - *C'est important de bien discuter les résultats d'après le style des publications présentées sur le site du cours.*
4. Voici un exemple d'un tableau à présenter (ci-bas). Ce tableau devrait être répété pour chaque expérimentation comme suit :
    - Ne rien utiliser en pré-traitement (Baseline)
    - Utiliser une stopliste et un stemming (Porter, krovetz, etc) ou une lemmatisation, suite à la stop list (Baseline-pre-process).

Types de Requêtes	**Schéma de pondération**	**MAP** *(exemples)*
Courtes	Tf*idf	0.44
Courtes	BM25	0.65
Courtes	autre schéma	0.22
Longues	Tf*idf	…
Longues	BM25	…
Longues	autre schéma	…

2. Amélioration du système RI

L'affichage de la partie 2 du projet qui consiste à améliorer le système RI développé, se fera le vendredi 3 novembre 2023.

3. Rédaction du Rapport

Le rapport final (un seul pour les deux parties du projet) doit suivre le style de la conférence NAACL 2016. Un modèle en Word et LaTeX se trouve sur le site web suivant : http://naacl.org/naacl-pubs/

Il doit respecter **une limite de 5 pages au minimum et 8 pages au maximum** (en plus des pages incluant les références), en français ou en anglais et inclure les points suivants :

a) Titre que vous donnerez à votre projet
b) Résumé de votre projet (quelques lignes seulement)
c) Mots clés - trois à cinq mots clé liés à votre cas d'étude
d) Introduction et définition du problème
e) État de l'art (au moins sept références)
f) Méthodologie - vous devez vous concentrer sur le problème que vous traitez.

g) Description des données et ressources utilisées.
h) Évaluations et résultats – doit inclure les résultats de base (baseline), taux des erreurs, analyse des erreurs (pas moins de 30 requêtes examinées), évaluation contre la partie baseline, la(es) méthode(s) présentée(s) dans le projet et autres comparaisons. Aussi, une partie discussion détaillée.
i) Conclusion et perspectives.
j) Bibliographie - articles cités dans le rapport (au moins sept références).

Références

- Lucene (full-featured text search engine library written entirely in Java) - https://lucene.apache.org
- Solr (search engine based on Lucene) - https://solr.apache.org
- Terrier (system developed in Java at the University of Glasgow) - http://terrier.org/
- Lemur (http://www.lemurproject.org/),
- https://sourceforge.net/p/lemur/wiki/Scored%20Query%20Evaluation/
- **Lucy / Zettair** (http://www.seg.rmit.edu.au/zettair/) – très vieux
- Smart (system developed in C at Cornell University in the 1960s) : ftp://ftp.cs.cornell.edu/pub/smart/
- Okapi BM-25 : https://en.wikipedia.org/wiki/Okapi_BM25
- Autres liens sur Lucene et Solr:
  - https://dzone.com/refcardz/lucene
  - https://lucidworks.com/post/flexible-ranking-in-lucene-4/
  - http://lucenetutorial.com/lucene-in-5-minutes.html
  - http://www.lucenetutorial.com/your-first-project.html
  - http://opensourceconnections.com/blog/2015/10/16/bm25-the-next-generation-of-lucene-relevation/
  - http://ipl.cs.aueb.gr/stougiannis/bm25_2.html
  - http://www.solrtutorial.com/

Livres:

- Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schutze, Cambridge University Press, 2008 (online version available)
- Information Retrieval. D. Grossman and O. Frieder, Springer, 2004 (second edition).
- Information Retrieval, by C. J. van Rijsbergen (1979)
- Modern Information Retrieval. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, 1999. https://web.cs.ucla.edu/~miodrag/cs259-security/baeza-yates99modern.pdf

Bon travail !