

FEUILLE DE ROUTE

Chasse aux triangles dans jeux de mots

FACULTÉ DES SCIENCES
UNIVERSITÉ DE MONTPELLIER

Groupe :

Error!404

Participants :

TRAN Thi Tra My
NGUYEN Hoai Nam
NGUYEN Tran Tuan Nam
NGUYEN Huu Khang

Encadrant :

Mme. Anne LAURENT
M. Mathieu LAFOURCADE
M. Arnaud CASTELLTORT

10 Février 2020

1. Liste des documents à lire

- [1] Cheap, Fast and Good! Voting Games with a Purpose - **Karën Fort, Mathieu Lafourcade, Nathalie Le Brun** - ref: hal.archives-ouvertes.fr/hal-01790614
- [2] Si beaucoup d'oiseaux volent, peut-on en déduire que tous les oiseaux volent ? Production endogène de règles déductives dans un réseau lexico-sémantique - **Mathieu Lafourcade, Alain Joubert** - ref: hal.archives-ouvertes.fr/hal-02269036
- [3] Inférence de règles déductives par abduction - **Mathieu Lafourcade, Manel Zarrouk, Alain Joubert** - ref: <https://hal-lirmm.ccsd.cnrs.fr/lirmm-00830445>
- [4] Évaluation et consolidation d'un réseau lexical grâce à un assistant ludique pour le "mot sur le bout de la langue" - **Alain Joubert, Mathieu Lafourcade, Didier Schwab, Michael Zock** - ref: <https://hal.archives-ouvertes.fr/hal-00661659>

2. Outils de stockage choisis

Nous avons à disposition 2 type de stockage les données. Ce sont NoSql et SQL.

Le tableau suivant compare ces deux types de bases de données.

	SQL	NoSQL
Flexibilité et évolutivité	Il est difficile de plus en plus difficile d'ajouter ou d'enlever les relations chez les données relationnels	Il y a la possibilité d'ajouter les relations et les nodes sur une graphe existant
Point importance	A utiliser quand il est important les données sont correctes	A utiliser quand c'est mieux d'obtenir les résultats rapidement
Utilisation	C'est l'idéale pour les requêtes complexes	Ce n'est pas un bon choix pour faire les requêtes complexes
Schéma	La base de données doit être prédéfini	La base de données est non-structuré et fonctionne dynamiquement
Outils	Oracle, Postgres, and MySQL	MongoDB, Redis, , Neo4j, Cassandra, Hbase, OrientDB.

Enfin, on a choisi le type NoSQL, plus précisément, la forme graphique de Neo4j.

Dans le contexte de ce TER, Neo4j présente plus d'avantages sur les autres en ce que, nos données se concentrent sur les relations entre plusieurs entités. Les méthodes de stockage les donnée du type SQL ou quelque autres formes du type

de NoSQL(key-value, column family, document,...) se connectent par créer des keys/foreign links vers d'autres enregistrements/documents et joindre des tables ou vice versa est complexe, inefficace et sujet aux erreurs, surtout si les relations sont complexes.

Les avantages de Neo4j sont évidemment. Sa façon d'interroger les données se concentrent uniquement sur une subdivision de graphique avec des données connexes, de sorte que les performances ne changent pas beaucoup quand il y a de plus en plus de données. Par contre, il est facile à ajouter des nœuds, des relations, des sous-graphiques sans affecter les anciennes données, les anciennes requêtes.

Neo4j contient les limites de stockage de données distribuées. Il n'affecte pas néanmoins notre modèle. D'ailleurs, Neo4j ne possède pas les indices. Cela peut être une problématique pour trier les nodes mais dans le cadre de ce TER, cela ne devrait pas nous causer un problème.

Finalement, cet outil est conseillé dans le sujet de TER et nous sommes motivé pour travailler sur un nouvel outil. Cela va nous permettre d'avoir les connaissances sur cet outil.

3. Les tâches

Les tâches seront divisés en sous tâches et seront partagés au cours du développement. Il est possible que le délai des tâches ne corresponde pas exactement pendant la période du développement.

Concernant la liste des tâches, nous allons les diviser en 3 phases principales.

- Convertir les données en format texte vers les données de Neo4j
- L'étude sur l'algorithme permet de fournir un triangle
- Implémentation sur un serveur

Ensuite, si le temps permet, nous allons chercher à optimiser le passage des requêtes sur le serveur et ajouter les extensions possibles.

4. La diagramme de gantt prévisionnel

Le tableau suivant présente notre itinéraire prévu. Le projet commence au 28/01/2020 et il finira au 03/04/2020.

	Tâche	Durée (Jour)	Début	Fin
1	Durée de projet	120	28/01/20	26/05/20
2	1er rendez-vous	1	28/01/20	28/01/20
3	L'étude sur les données fournis	8	29/01/20	05/02/20
3.1	Etudier la structure de données de Jeux de mots	2	29/01/20	30/01/20
3.2	Réaliser les scripts permettant de récupérer les mots	3	31/01/20	02/02/20
3.3	Réaliser les scripts permettant de récupérer les relations	3	03/02/20	05/02/20
4	Réalisation la feuille de route	4	06/02/20	09/02/20
5	Installation Neo4j	1	10/02/20	10/02/20
6	2e rendez-vous	1	11/02/20	11/02/20
7	Traduire les données du textes vers données en graphes	3	12/02/20	14/02/20
8	Réaliser l'algorithme permettant de retrouver les triangles	7	15/02/20	21/02/20
9	3e rendez-vous	1	25/02/20	25/02/20
10	Mettre en place une serveur web	7	22/02/20	28/02/20
11	Tester le serveur et corriger les erreurs	4	29/02/20	03/03/20
12	Réalisation d'un bilan mi-parcours	3	04/03/20	06/03/20
13	Début sur la rédaction du rapport	3	07/03/20	09/03/20
14	4e rendez-vous	1	10/03/20	10/03/20
15	Optimiser le passage des requêtes sur le serveur	6	11/03/20	17/03/20
16	Réaliser les algorithmes pour le graphe d'inférences	6	18/03/20	23/03/20
17	5e rendez-vous	1	24/03/20	24/03/20
18	Installation de l'algorithme permettant d'enrichir le graphe	6	25/03/20	30/03/20
19	La suite de la rédaction du rapport	4	31/03/20	03/04/20
24				
25				
26				