



IFT3225 (Techno. Web) — Projet 2
Client ConceptNet

Contact :
Philippe Langlais +1 514 343 61 11 ext: 47494
RALI/DIRO felipe@iro.umontreal.ca
Université de Montréal <http://www.iro.umontreal.ca/~felipe/>

■ dernière compilation : 17 février 2022 (07:53)

Contexte

Dans ce projet, vous allez développer une **application côté client** qui offre une interface de consultation du graphe de connaissance [ConceptNet](#) ainsi qu'une interface de jeux manipulant les concepts de ConceptNet. Votre solution sera abritée sur le serveur Web du DIRO sous la forme d'une unique page HTML dynamique.

À faire

1. Prendre en main l'API de ConceptNet afin de poser des requêtes. Votre but est ici de récolter au moins 100 faits de la forme $(start, relation, end)$ où *start* et *end* sont des concepts de ConceptNet (on s'intéressera a priori aux concepts exprimés en français et/ou en anglais), et *relation* est une relation particulière (il existe environ [30 relations](#) dans ConceptNet). Voici quelques faits à titre d'exemple :

start	relation	end
/c/fr/cité universitaire	<u>RelatedTo</u>	/c/fr/étudiant
/c/fr/Un boulanger	<u>UsedFor</u>	/c/fr/faire le pain
/c/en/a radio	<u>UsedFor</u>	/c/en/listening to talk shows
/c/en/very old magazines	<u>AtLocation</u>	/c/fr/the dentist

À l'issue de cette étape, vous disposerez d'au moins 100 faits que vous encoderez sous forme d'une table HTML. Cette structure sera utilisée par défaut pour alimenter les questions posées dans les jeux. Prenez note que votre table doit réunir au moins 40 concepts différents (*start* ou *end*) et au moins 20 *relations* différentes.

2. Écrire une page d'accueil HTML5 qui offre deux choix : consultation de la base ConceptNet et Jeux. Cette page HTML doit contenir la table de faits précédente. Selon l'action choisie par l'utilisateur, une interface de consultation ou une interface de jeux est offerte ; les deux sont maintenant décrites tour à tour.
3. Interface de consultation.

Cette interface permet à l'utilisateur de spécifier un concept et/ou une relation. Si seulement un concept est fourni (ex : /c/fr/dentiste/)

alors tous les faits de ConcepNet où le concept est soit un noeud start, soit un noeud end seront affichés pour autant que les concepts soient en anglais ou en français. Si seulement une relation est fournie, alors toutes les instances retournées par ConcepNet impliquant des concepts en anglais ou en français seront affichées. Si un concept et une relation sont fournis, alors toutes les faits impliquant cette relation et dont un noeud (start ou end) correspond au concept spécifié seront affichés. Là encore, les concepts dans une autre langue que l'anglais et le français seront ignorés.

Les informations rentrées par l'utilisateur (concept et/ou relation) seront transformées en une requête à ConcepNet qui sera faite via AJaX. L'objet JSON retourné sera analysé à la volée pour générer l'affichage des faits ainsi que leur nombre. Vous remarquerez que certaines requêtes amènent un grand nombre de réponses (notamment les requêtes concernant seulement une relation où toutes les instances de cette relation sont des réponses). ConcepNet retourne dans sa réponse un maximum de 1000 faits et indique le cas échéant à l'aide de la propriété **view** de sa réponse un système de pagination. Ce système est expliqué très clairement dans la documentation de ConcepNet, mais voici un exemple où la requête suivante retourne les 1000 premières réponses :

`http://api.conceptnet.io/query?rel=/r/UsedFor&limit=1000`
et offre un accès aux réponses précédentes et suivantes dans le champ **view** de la réponse :

```
{ '@id': '/query?rel=/r/UsedFor&offset=0&limit=1000',  
  '@type': 'PartialCollectionView',  
  'comment': "There are more results. Follow the 'nextPage' link for more.",  
  'firstPage': '/query?rel=/r/UsedFor&offset=0&limit=1000',  
  'nextPage': '/query?rel=/r/UsedFor&offset=1000&limit=1000',  
  'paginatedProperty': 'edges' }
```

Votre interface de consultation affichera un bouton **prev** et **suiv** de façon à calquer ce système de pagination. Le clique d'un de ces boutons relancera (via AJaX) une requête à ConcepNet avec mise à jour des faits retournés.

4. Interface de jeu.

Trois types de jeux sont proposés à l'utilisateur : Oui-Non, Consigne et Qui suis-je ?

Oui-Non Dans ce jeu, une question à réponse booléenne est formulée et un temps maximum est décrémenté (disons 1 minute) pendant lequel le joueur peut répondre. Une fois sa réponse entrée (ou que le temps est écoulé), un message indiquera au joueur s'il a gagné ou pas. Les formes des questions colleront à la syntaxe de Concept-Net. Un exemple de question correspondant au deuxième fait dans la table précédente serait : *Un boulanger UsedFor ? faire le pain*. Vous noterez que vous pouvez poser des questions à réponse négative, en remplaçant par exemple le concept *end* par un concept quelconque, comme dans : *Un boulanger UsedFor ? couper du bois*.

Consigne Dans ce jeu, le joueur reçoit une consigne à laquelle il doit en un temps limité fournir des réponses. Un exemple de consigne serait par exemple de demander de lister des synonymes d'un concept particulier. Là encore, la syntaxe d'une consigne suivra celle de ConceptNet, et la consigne "donnez des synonymes de fruit", sera donnée sous la forme : *fruit Synonym ?*. À l'issue du temps maximum autorisé pour répondre, les réponses proposées qui sont déjà connues de ConceptNet donneront chacune lieu à un point et seront visuellement distinguées des autres.

Qui suis-je ? Dans ce jeu, le joueur doit identifier un concept à l'aide d'indices qui lui sont proposés. Toutes les 20 secondes, un indice est posé au joueur jusqu'à épuisement du temps de réponse (2 minutes) ou des indices disponibles. Si la réponse de l'utilisateur est la bonne alors il reçoit 8 points moins le nombre d'indices proposés. Voici un exemple d'interaction possible pour le concept *dentist*.

0 ??? CapableOf cause you to have great pain

20 a receptionist AtLocation ???

40 ??? ReceivesAction paid to interact with you

60 old magazines AtLocation ???

80 ??? UsedFor clean your teeth

Le premier indice (proposé à la seconde 0) peut se lire : *le concept à trouver est capable de vous causer beaucoup de souffrance*. À la seconde 20 un deuxième indice est affiché qui peut se lire : *Il y a un réceptionniste chez le concept à trouver*. À la seconde 80, un cinquième indice est proposé qui permet au joueur de trouver la bonne réponse (*dentist*). Le joueur reçoit alors $8-5 = 3$ points.

D'où viennent les questions qui sont posées au joueur ? Par défaut les faits de la table HTML décrite plus haut seront sollicités. Cependant au fil de l'exploration de ConceptNet via l'interface de consultation, vous pourriez ajouter des concepts ou des relations d'intérêt à la table de façon à augmenter l'étendue des jeux. Tout fait peut être transformé en une question pour le jeu Oui-Non. En revanche, pour que cela ait de l'intérêt, le jeu consigne devrait impliquer des relations associant plusieurs réponses à un concept. Enfin, le jeu Qui suis-je ? requiert d'accumuler un nombre critique de faits sur un concept qui serviront d'indices.

Une grande liberté vous est donnée quant à l'apparence de votre page Web. La notation tiendra cependant compte de la bonne ergonomie de votre solution. Nous vous encourageons notamment à faire usage de bootstrap et à réaliser vos requêtes AJaX à l'aide de jQuery. Les interfaces de consultation et de jeux offertes doivent être a priori greffées à la page d'accueil de sorte qu'il ne devrait y avoir qu'une seule page HTML (dynamique) dans votre solution. À tout moment, votre page devrait indiquer des informations sur votre base de faits (votre table HTML encodant des faits), dont : le nombre de faits mémorisés (100 au moins au départ, plus au gré des explorations), le nombre de concept différents (40 au moins au départ), et le nombre de relations différentes (20 minimum au départ).

À remettre

- Un rapport au format pdf `readme.pdf` d'au plus 2 pages qui indique :
 - a) l'identité des personnes du groupe, b) l'url de la page HTML abritée au DIRO donnant accès au jeu (vous prendrez soin tout comme dans le projet 1 de ne pas exposer cette url afin de minimiser les risques de plagiat), et c) tout détail que vous jugez pertinent de souligner.
- Une archive (zip, tar, etc.) de toutes les ressources de votre projet

La remise est à faire sur Studium sous le libellé **projet2**. Vous devez remettre votre code et votre rapport (format pdf) dans une archive (gzip, tar, tar.gz) dont le nom est préfixé de **projet2-noms**, où **noms** est à remplacer par l'identité des personnes (**prénom_nom**) impliquées dans le projet. Donc si j'avais à remettre seul ma solution au projet2, je le ferais sous le nom **projet2-philippe_langlais.tar.gz**. Le projet est à remettre en groupe d'au plus deux personnes **au plus tard dimanche 13 mars 2022 à 23h59**.

Notation

Ce projet compte pour 25 points répartis comme suit :

- Interface de consultation : 10 points (prev, suiv, justesse, ergonomie)
- Interface de jeu : 10 points (bonne gestion du temps, ergonomie)
- Mise à jour de la base de faits interne : 5 points

Questions

Posez vos questions via le slack du cours sur le canal **demo**.