

L'objet du Hackaton IWCS 2017 est d'extraire à partir de textes des relations sémantiques entre paires de termes. Par exemple :

"Dans le ciel livide, on devinait la lune pleine, derrière de grands nuages, des haillons noirs"

```
ciel ; r_carac ; livide
deviner ; r_patient ; lune
lune ; r_carac ; pleine
nuages ; r_carac ; grands
haillons ; r_carac ; noirs
```

Début : lundi 18/09 à 14h

Fin : mercredi 20/09 à 8h

Durée : 42 h

Modalités : Les participants peuvent arriver en équipe de 1 à 3 personnes. Ils peuvent utiliser pour réaliser leur programme n'importe quels packages, logiciels et ressources librement disponibles, et devront les fournir à la fin. Les logiciels propriétaires, et les données non libres de droit sont donc exclus. Tous les langages de programmation sont autorisés.

Au début du hackathon, un corpus de textes d'où devront être extraites les relations sera fourni (sous forme d'une liste d'url). Durant les 42h, les participants enverront une à une les relations découvertes par leur logiciel, sous la forme d'une requête à un serveur HTTP dont l'adresse sera donnée au début du hackathon. Les arguments de la requête seront le premier mot, le nom de la relation, le second mot et le segment de texte dont il est extrait avec quelques caractères avant et après.

Par exemple : ciel ; r_carac ; livide ; Dans le ciel livide, on devinait

Les productions des participants seront évaluées via une activité de crowdsourcing durant les 42h. Une relation proposée doit être juste, par rapport au segment de texte. On comptera 1 par relation juste et -1 point par relation jugée fautive. En cas de désaccord entre les validateurs, la moyenne des notes sera retenue (par exemple, 2 pour et 1 contre donne 1/3 point). À l'issue des 42h, les équipes remettront leur logiciel ainsi que les packages et ressources ayant été utilisées. Les résultats doivent être déterministes et reproductibles. Un guide sous forme de fichier texte doit décrire la procédure (aussi simple que possible) pour faire fonctionner le logiciel de l'équipe.

Guide du contributeur

Les contributions se font, soit en utilisant manuellement le formulaire, soit via un accès par programme (en envoyant une requête get ou post). Le format de contribution est le suivant :

```
mot 1 ; nom de relation ; mot 2 ; contexte
```

Par exemple : chat;r_has_carac;tigré;j'ai aperçu un joli chat tigré dans le jardin.

Le contexte est un segment de texte d'où on peut extraire la relation. Ce segment de texte doit être issu des textes fournis. Une fois envoyée, la relation ne peut pas être resoumise, même avec un contexte différent. Il est possible que la relation proposée soit une inférence issue du texte. Par exemple :

```
félin;r_has_carac;tigré;j'ai aperçu un joli chat tigré dans le
jardin.
```

Le contexte ne doit pas contenir de point-virgule.

Guide de l'évaluateur

Un évaluateur peut voter pour les propositions de relations (les soumissions). Après s'être logué, il pourra afficher les relations pour lesquelles voter. Un vote peut être révisé. La relation proposée doit impérativement être déductible du contexte donné. Même si la relation est correcte, si elle ne peut être déduite du contexte, alors la proposition sera considérée comme invalide.

Exemple de proposition appelant un vote contre :

`félin;r_has_carac;tigré;j'aime les haribos.`

Autre cas faux : `éplucher;r_has_patient;pommes;il épluche des pommes de terre`

Par contre, les propositions suivantes sont correctes :

`éplucher;r_has_patient;pomme de terre;il épluche des pommes de terre`

`éplucher;r_has_patient;légumes;il épluche des pommes de terre`

`éplucher;r_has_patient;légume;il épluche des pommes de terre`

Les hyperonymes proches (génériques, comme 'légume' pour 'pomme de terre') sont donc acceptés, mais pas les spécifiques (par exemple, 'chat de goutière' pour 'chat').

Parmi les pièges à éviter :

- Les inversions sujet-verbe, comme "Le matin tôt passait le laitier". C'est bien sûr le laitier qui passe... le laitier n'est donc pas le patient, mais l'agent de passer.

Liste des 10 relations à identifier

`r_is_a` - Il est demandé d'énumérer les générique/hyperonymes du terme. Par exemple, 'animal' et 'mammifère' sont des génériques de 'chat'. Syntaxe : 'chat' ; `r_is_a` ; 'félin'

`r_has_part` - Il faut donner des parties/constituants/éléments (a pour méronymes) du mot cible. Par exemple, 'voiture' a comme parties : 'porte', 'roue', 'moteur', ... Syntaxe : 'chat' ;

`r_has_part` ; 'queue'

`r_has_carac` - Pour un terme donné, souvent un objet, il est demandé d'en énumérer les CARACTéristiques (adjectifs) possibles/typiques. Par exemple, 'liquide', 'froide', 'chaude', pour 'eau'. Syntaxe : 'eau' ; `r_has_carac` ; 'froide'

`r_lieu` - Il est demandé d'énumérer les lieux typiques où peut se trouver le terme/objet en question. Syntaxe : 'cheval' ; `r_lieu` ; 'champ de courses'

`r_own` - Que POSSÈDE le terme suivant ? (un soldat possède un fusil, une cavalière des bottes, ...) Syntaxe : 'soldat' ; `r_own` ; 'lance'

`r_has_agent` - L'agent (qu'on appelle aussi le sujet) est l'entité qui effectue l'action, OU la subit pour des formes passives ou des verbes d'état. Par exemple, dans - Le chat mange la souris -, l'agent est le chat. Des agents typiques de 'courir' peuvent être 'sportif', 'enfant',... Syntaxe : 'manger' ; `r_has_agent` ; 'chat'

`r_has_patient` - Le patient (qu'on appelle aussi l'objet) est l'entité qui subit l'action. Par exemple dans - Le chat mange la souris -, le patient de manger est la souris. Des patients typiques de manger peuvent être 'viande', 'légume', 'pain', ... Syntaxe : 'manger' ;

`r_has_patient` ; 'souris'

`r_has_instrument` - l'instrument utilisé pour effectuer l'action. Par exemple, il abat un arbre à la hache - abattre à pour instrument hache. Syntaxe : 'abattre' ; `r_has_instrument` ; 'hache'

r_has_consequence - la conséquence associé au terme. Par exemple : 'prendre un médicament ; has_consequence ; ' aller mieux' r_make_use_of - ce que peut utiliser ou consommer le terme. Par exemple : automobile ; r_make_use_of ; essence r_make_use_of - ce que peut utiliser ou consommer le terme. Par exemple : automobile ; r_make_use_of ; essence