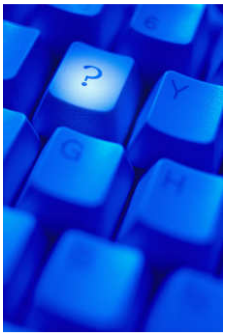
# TP : Conception d'une application de TALN



# (activité d'équipe de 2 à 4 personnes)

Faculté de sciences et de génie
Département d'informatique et de génie logiciel
IFT-2003 Intelligence artificielle I
Laurence Capus

# Travail pratique : questions/réponses

*Ce travail compte pour 5% de la note finale.*
*Échéance : le 21 avril décembre 2013 (23h59)*

**Objectifs d'apprentissage :**

À l'issue de ce travail, un étudiant devrait être capable de :

- résoudre un problème concret de traitement automatique du langage naturel;
- identifier les enjeux du traitement du langage naturel sur un problème concret;
- analyser les limites d'une solution à un problème concret de traitement du langage naturel.

**Description du contexte du travail pratique :**

On vous demande de développer un module de TALN pour un système de questions/réponses en langage naturel permettant à un utilisateur d'interagir avec un système à base de connaissances, par exemple, à propos d'un texte de plusieurs phrases écrit en langage naturel. Le système pourra répondre à des questions du type :

<utilisateur> Qu'aime Michel?
<système> Michel aime le cinéma et les randonnées.

**Travail à réaliser :**

**1- (10 %)** Proposer un texte de plusieurs phrases que le système pourra interpréter. Constituer une base de connaissances qui représentera les connaissances contenues dans ces phrases. Décrire les patrons des interprétations proposées. La taille du texte devrait être d'au moins une trentaine de mots. Par exemple, avec l'exemple de l'énoncé, le patron aime(X,Y) signifie que X aime Y et la base de connaissances contiendrait les faits : « aime(michel, cinema). » et « aime(michel, randonnees) ». Implanter la base de connaissances en Prolog.

**2- (10 %)** Proposer un ensemble de questions (au moins 5), en langage naturel, auxquelles le système pourra répondre en se basant sur les connaissances contenues dans le texte présenté en (**1**). Noter qu'une même interprétation sémantique peut correspondre à plusieurs questions en langage naturel. Décrire les patrons des interprétations sémantiques de chaque question proposée.

**3- (25 %)** Donner la grammaire capable d'analyser les questions indiquées en (**2**). Implémenter cette grammaire en langage Prolog à l'aide du prédicat repondre/2 dont le 1$^{e}$ argument est la question (liste de mots en lettres minuscules) et le 2 argument un fait trouvée dans la base de connaissances et permettant de répondre à la question posée.

4- **(25 %)** Écrire une grammaire pour générer les réponses du robot à partir de sa base de connaissances. Implémenter cette grammaire en Prolog à l'aide du prédicat ecrire/2 dont le 1e argument est un fait et le 2e argument est la phrase générée à partir de ce fait.

5- **(5 %)** Ajouter un prédicat « lancer/0 » qui permet de démarrer l'interprétation des questions et utilise les prédicats repondre/2 et ecrire/2. L'utilisateur saisit une question en langage naturel et l'interpréteur lui donne une réponse en langage naturel (voir aide en annexe 2).

6- **(15%)** Donner les résultats.
- Faire des tests pour montrer le niveau d'interprétation des questions (à placer dans le rapport).
- Discuter les résultats obtenus : Êtes-vous satisfait de votre travail ? Comment pourriez-vous l'améliorer ? Est-ce que de nouvelles questions pourraient-elles facilement être interprétées ?

Les 10% restants sont dédiés à l'appréciation globale du travail remis (présentation et expression écrite).

**NB**. : Ne pas oublier de valider votre travail à l'aide de la grille d'autoévaluation fournie à l'annexe 1.

**Éléments à remettre :**

Partie logicielle (tous les éléments de la partie logicielle doivent être documentés) (en 1 seul fichier .PL) :
- la base de connaissances;
- le prédicat repondre/2;
- le prédicat ecrire/2;
- le prédicat lancer/0 (seul prédicat d'interface).

Partie rapport (en .PDF) :
- la base de connaissances avec les patrons utilisés;
- les questions choisies;
- la grammaire qui permet d'analyser les questions;
- la grammaire qui permet de générer les réponses;
- la partie Résultats.

**Modalités de remise de ce travail :** L'ensemble du travail est à remettre en format électronique via le Portail des cours. Les travaux remis en retard ne seront pas corrigés.

**Annexe 1 : Grille d'auto-évaluation (meilleure évaluation à gauche et pire évaluation à droite)**

| **Conception** | | | | |
|---|---|---|---|---|
| *Base de connaissances (10%)* | Une base de connaissances est implantée et tous les patrons utilisés sont expliqués. | Une base de connaissances est implantée mais tous les patrons utilisés ne sont pas expliqués. | Seulement les patrons utilisés sont expliqués. | Pas de base de connaissances implantée et aucun patron utilisé n'est expliqué. |
| *Identification des questions (10%)* | Au moins 5 questions sont identifiées et utilisés par le programme. | Seulement 4 questions sont identifiées et utilisés par le programme. | Seulement 2 questions sont identifiées et utilisés par le programme. | Aucune question n'a été identifiée. |
| *Analyse des questions (25%)* | Une grammaire attribuée est proposée et permet d'analyser les questions. | Une grammaire attribuée est proposée mais ne permet pas d'analyser toutes les questions. | Une grammaire attribuée est proposée mais ne permet d'analyser aucune des questions. | Aucune grammaire n'est proposée pour analyser les questions. |
| | La grammaire est implantée et fonctionne sans erreur. | La grammaire est implantée mais fonctionne avec erreur. | La grammaire est implantée mais ne fonctionne pas du tout. | La grammaire n'est pas implantée. |
| *Génération les réponses (25%)* | Une grammaire attribuée est proposée et permet de générer les réponses. | Une grammaire attribuée est proposée mais ne permet pas de générer toutes les réponses. | Une grammaire attribuée est proposée mais ne permet de générer aucune des réponses. | Aucune grammaire n'est proposée pour générer les réponses. |
| | La grammaire est implantée et fonctionne sans erreur. | La grammaire est implantée mais fonctionne avec erreur. | La grammaire est implantée mais ne fonctionne pas du tout. | La grammaire n'est pas implantée. |
| *Prédicat lancer/0 (5%)* | Le prédicat est implanté correctement. | Le prédicat est implanté avec des erreurs. | Le prédicat n'est pas du tout implanté. | |

**Annexe 1 : Grille d'auto-évaluation (meilleure évaluation à gauche et pire évaluation à droite)**

| | | | | |
|---|---|---|---|---|
| **Résultats (15%)** | | | | |
| *Exemples de questions /réponses* | Plusieurs exemples sont présentés et les résultats sont discutés. | Plusieurs exemples sont présentés mais les résultats ne sont pas discutés. | Un seul exemple est présenté et discuté. | Pas d'exemple ou un seul exemple est présenté mais non discuté. |
| *Améliorations possibles* | Plusieurs améliorations possibles sont proposées et illustrées par des exemples de questions/réponses ou autres. | Plusieurs améliorations possibles sont proposées mais plus ou moins illustrées. | Une seule amélioration est proposée et plus ou moins illustrée. | Aucune amélioration n'est proposée. |
| **Appréciation globale (10%)** | | | | |
| *Expression écrite* | Le rapport ne contient aucune faute (vocabulaire, grammaire, syntaxe, etc.). | Le rapport ne contient pas plus d'une dizaine de fautes (vocabulaire, grammaire, syntaxe, etc.). | Le rapport ne contient pas plus de 5 fautes par page (vocabulaire, grammaire, syntaxe, etc.). | Le rapport contient plus de 5 fautes par page (vocabulaire, grammaire, syntaxe, etc.). |
| *Présentation du rapport* | Tous les éléments demandés sont présents et le rapport est paginé et contient une page de garde. | Tous les éléments demandés sont présents, mais le rapport n'est pas paginé ou ne contient pas de page de garde. | Il manque 1 élément demandé. | Il y a au moins 2 éléments demandés manquants. |

**Annexe 2 :**

La définition en Prolog du prédicat lire/2 qui permet de traduire une phrase en une liste de mots est la suivante :

```
% Le prédicat lire/2 lit une chaîne de caractères Chaine entre apostrophes
% et terminée par un point.
% Resultat correspond à la liste des mots contenus dans la phrase.
% Les signes de ponctuation ne sont pas gérés.

lire(Chaine,Resultat):- write('Entrer la phrase '), read(Chaine),
                        name(Chaine, Temp), chaine_liste(Temp, Resultat),!.

% Prédicat de transformation de chaîne en liste

chaine_liste([],[]).
chaine_liste(Liste,[Mot|Reste]):- separer(Liste,32,A,B), name(Mot,A),
chaine_liste(B,Reste).

% Sépare une liste par rapport à un élément

separer([],X,[],[]):-!.
separer([X|R],X,[],R):-!.
separer([A|R],X,[A|Av],Ap):- X\==A, !, separer(R,X,Av,Ap).
```