

UNIVERSITE LAVAL



Le 01 mai 2018

## **Questionnements logiques en langage naturel**

*TP4 Conception d'une application de TALN IFT-2003*

**Appoline Bibie Yamatchui Kamaha (111 096 135)**

**Paul Pilote (111 104 263)**

**Éric Cogoluenhès (111 162 638)**

# Sommaire

Sommaire 2

Introduction 3

1. Ensemble de phrases utilisées 3

2. Évaluations des résultats 3

2.1. Test sur analyse de la forme (a) 3

2.2. Test sur analyse de la forme (b) 5

2.3. Test sur analyse de la forme (c) 6

2.4. Bilan de l'expérimentation 8

# Introduction

Le traitement automatique du langage est une discipline de l'intelligence artificielle appliqué à la linguistique.

Dans le cadre de notre travail pratique, il nous a été demandé de résoudre un problème concret de traitement automatique du langage naturel.

Ce rapport présente les démarches permettant la conception d'une mini application du traitement automatique du langage naturel. Ainsi nous allons identifier une liste de cinq phrases en rapport avec les caractéristiques dans ce cadre et avec notre application y donner une réponse.

## 1. Ensemble de phrases utilisées

Ci-dessous se trouvent la liste des phrases en lien avec les données :

Numéro	Phrases
1	Félix est un chat
2	Quentin est un homme
3	Tout chat est un félin
4	Tout homme est un vertébré
5	Est ce que Félix est un félin

## 2. Évaluations des résultats

### 2.1. Test sur analyse de la forme (a)

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.
```

```
?- go.  
|: felix est un chat  
ok  
|:
```

---

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.  
  
?- go.  
|: quentin est un homme  
ok  
|: |
```

Nous pouvons observer sur les captures d'écran ci-dessus que le premier prédicat analyse la phrase saisie et donne la réponse ok pour analyseA comme demandé dans l'énoncé.

```
?- go.  
|: janice est une femme  
false.
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec des mots qui ne sont pas dans notre base de donnée pour analyseA (« janice » et « femme »). Comme attendu, le résultat est 'false'.

```
?- go.  
|: un chat felix est  
false.
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec des mots qui sont dans notre base de donnée, mais avec une structure différente. Comme attendu, le résultat est 'false'.

## 2.2. Test sur analyse de la forme (b)

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.
```

```
?- go.  
|: quentin est un homme  
ok  
|: tout homme est un vertebre  
ok  
|:
```

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.
```

```
?- go.  
|: felix est un chat  
ok  
|: tout chat est un felin  
ok  
|:
```

Nous pouvons observer sur les captures d'écran ci-dessus que les deux premiers prédicats analysent les phrases saisies et donne la réponse ok pour analyseA et analyseB comme demandé dans l'énoncé.

```
?- go.  
|: felix est un chat  
ok  
|: toute femme est un homme  
false.
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec un mot qui n'est pas dans notre base de donnée pour analyseB (« femme »). Comme attendu, le résultat est 'false'.

```
?- go.  
|: felix est un chat  
ok  
|: tout homme, un vertebre est  
false.
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec des mots qui sont dans notre base de donnée, mais avec une structure différente. Comme attendu, le résultat est 'false'.

## 2.3.Test sur analyse de la forme (c)

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.  
  
?- go.  
|: felix est un chat  
ok  
|: tout chat est un felin  
ok  
|: est ce que felix est un felin  
oui  
true |
```

```
?- ['/Users/vivas/Desktop/goFinale.pl'].  
true.  
  
?- go.  
|: quentin est un homme  
ok  
|: tout homme est un vertebre  
ok  
|: est ce que quentin est un vertebre  
oui  
true |
```

Nous pouvons observer sur les captures d'écran ci-dessus que les deux premiers prédicats analysent les phrases saisies et donne la réponse ok pour analyseA et analyseB et le troisième prédicat donne la réponse à la requête pour analyseC comme demandé dans l'énoncé.

For online help and background, visit <http://www.swi-prolog.org>  
For built-in help, use ?- help(Topic). or ?- apropos(Word).

```
% /Users/ericc/Documents/GitHub/Portfolio/GIF - 2003/TP4/TP4.pl compiled 0.00 sec, 39 clauses
?- go.
|: felix est un homme
ok
|: tout homme est un felin
ok
|: est ce que felix est un vertebre
false.

?-
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec un mot qui n'est pas dans notre base de donnée pour analyseC (« vertebre »). Comme attendu, le résultat est 'false'.

```
?- go.
|: felix est un chat
ok
|: tout chat est un felin
ok
|: est ce que felin felix est
false.
```

Sur le test observé sur la capture écran ci-dessus, nous avons construit à l'aide du prédicat une phrase avec des mots qui sont dans notre base de donnée, mais avec une structure différente. Comme attendu, le résultat est 'false'.

```
?- go.
|: felix est un chat
ok
|: tout chat est un felin
ok
|: est-ce que felix est un chat
false.
```

```
?- go.
|: quentin est un homme
ok
|: tout homme est un vertebre
ok
|: est ce que felix est un felin?
false.
```

Pour les tests observés sur les captures d'écran ci-dessus, nous pouvons observer la limite de notre programme. En effet, il ne prend pas en compte les motifs de phrase comme '-' et « ? ».

## **2.4.Bilan de l'expérimentation**

Ce travail pratique nous a permis d'expérimenter sur le traitement des langages naturels. Bien que nos grammaires soient de base, nous avons répondu aux questions selon la base de connaissance initiale.

Cependant notre grammaire ne gère que les motifs de phrase que nous avons choisi (comme par exemple: '-' et '?').

Étant donné qu'il s'agit de modéliser des comportements il serait judicieux de trouver un moyen de rendre plus flexible notre grammaire de manière à donner plus de liberté à son utilisateur.