

# Rapport TAL : Chatbot

Kévin TEBIB

September 11, 2017

## 1 Introduction

Dans le cadre de la création du jeu demandé en EVIJV, nous avons initialement prévu de munir certains PNJ de la possibilité de communiquer à l'aide du langage naturel textuel. Il devait pouvoir à partir de phrases entrées par le joueur, les analyser et y répondre. On nomme un tel logiciel, un chatbot. Malheureusement, celui-ci ne put être intégré à l'intérieur du jeu à cause d'un manque de temps. Cependant un chatbot à belle et bien était en coder en *C#*. Le but de ce rapport est de présenter son fonctionnement. Nous allons tout d'abord exposer les différentes hypothèses que nous avons considérées lors de la création du chatbot. Puis nous présenterons l'algorithme suivi des résultats obtenus. Et après avoir conclu, nous présenterons quelques pistes d'amélioration du chatbot.

## 2 Hypothèses

Comme dit précédemment le chatbot est codé en *C#*. Ce choix est dû à l'intention initiale de l'intégrer à l'intérieur du jeu qui codé sous Unity 3D qui a la particularité d'utiliser de script en *C#*. Bien qu'il soit aussi possible d'intégrer des scripts JavaScript à Unity. Mon désir de réellement comprendre comment fonctionne un chatbot m'a poussé à commencer le code de zéro. De plus, le jeu étant défini comme étant en français dans le cahier des charges, m'empêche d'utiliser les dictionnaires fournis vu qu'ils sont en anglais. Par conséquent, ce projet aura accès à des dictionnaires extrêmement limités.

Tous les traitements effectués par le chatbot seront effectué hors contexte. En effet, le chatbot n'aura accès à une représentation du contexte à aucun moment. Il ne pourra pas optimiser chacune des différentes phases de l'algorithme.

## 3 Algorithme

Avant de lancer le chatbot, nous présumons que la fonction *initialiseDictionnaire()* est initialisé les différents dictionnaires. En entrée, l'algorithme prend une phrase qu'il traitera.

### 3.1 Segmentation de la phrase

La segmentation est le premier pré-traitement que devra subir la phrase donnée en entrée. Il consiste tout d'abord à récupérer le signe de ponctuation de la phrase. Pour cela il se focalisera sur le signe de ponctuation de celle-ci. Évidemment le signe de ponctuation sera différent en fonction du type de la phrase. Voici pour chaque type de phrase le signe de ponctuation considéré :

**Déclarative** → .

**Exclamative** → !

**Interrogative** → ?

Ensuite nous diviserons la phrase au niveau des caractères virgules pour obtenir les différentes propositions de la phrase. Une fois les propositions obtenues, nous séparons tous les mots de chacune d'entre elles grâce aux caractères espaces.

### 3.2 Lemmatisation des mots

Pour chaque mot de la phrase quelle que soit sa proposition, nous vérifierons dans nos dictionnaires s'il existe une forme canonique. Si c'est bien le cas alors nous remplacerons ce mot par sa forme canonique. Sinon nous laisserons le mot initial dans la phrase.

### 3.3 Étiquetage morpho-syntaxique des mots

L'étiquetage morpho-syntaxique consiste à associer chaque mot de la phrase à sa forme syntaxique. Nous considérons ici 10 formes syntaxiques possibles : Adjectif, Adverbe, Conjonction, Déterminant, Interjection, Nom, Préposition, Pronom, Verbe et Inconnu. Il existe évidemment bien plus de forme syntaxique mais l'objectif étant de développer l'algorithme. Nous n'avons pas pris le temps de toutes les prendre en compte.

### 3.4 Analyse syntaxique

Le but de cette analyse est de vérifier si la syntaxe de la phrase est correcte. Pour ce faire, nous devons nous assurer que la phrase correspond à une grammaire prédéfinie.

### 3.5 Analyse sémantique

Si l'analyse syntaxique conclut que la phrase a une grammaire connue, alors nous pouvons commencer à associer un mot-clef à chaque token de la phrase. Et ainsi commencer à donner un sens à la phrase indépendamment du contexte.

### 3.6 Création de la réponse

A partir de la liste des mots-clefs, l'algorithme va le remplacer par un des mots qui lui est associés. Puis nous rechercherons la forme syntaxique de ces nouveaux mots. Ensuite nous récupérerons une syntaxe de phrase incluant la forme syntaxique de tous les nouveaux mots-clefs. Avant d'envoyer la réponse, nous devons la constituer à l'aide des nouveaux mots-clefs qui seront disposés dans le même ordre que la grammaire de phrase choisie précédemment. S'il manque des mots pour constituer la grammaire de la phrase, alors nous les récupérerons le dictionnaire morpho-syntaxique qui associe des mots à des formes syntaxiques.

## 4 Résultats Obtenus

Comme mes dictionnaires sont très limités, je suis obligé d'utiliser un exemple jouet composé de trois phrases pour tester mon programme. Ces phrases sont les suivantes :

- Salut!
- Tu as mangé quoi au petit déjeuner?
- Le nutella, c'est trop bon!

Tous les résultats que je présenterai dans cette partie peuvent être directement obtenu en lançant le programme en effet les affichages pour était prévu pour qu'on puisse suivre facilement ce qui se passe.

### 4.1 Segmentation de la phrase

Ce premier pré-traitement a permis de récupérer le signe de ponctuation de chacune des phrases donnant le résultat suivant :

**Phrase 1 :** .

**Phrase 2 :** ?

**Phrase 3 :** !

Puis il a bien séparé chaque phrase en préposition et en liste de mots de la manière suivante :

**Phrase 1 :** Proposition 0 : 'Salut'

**Phrase 2 :** Proposition 0 : 'Tu' 'as' 'mangé' 'quoi' 'au' 'petit' 'déjeuner'

**Phrase 3 :** Proposition 0 : 'Le' 'nutella'  
Proposition 1 : 'c'est' 'trop' 'bon'

séparer les deux premières phrases en liste de mots. Nous obtenons donc le résultat suivant pour les deux premières qui ne contiennent pas de préposition

## 4.2 Lemmatisation des mots

A cause des dictionnaires limités, je ne remis sous forme canonique que quatre mots dans les trois phrases. Ces mots sont : "Salut", "as", "mangé", "c'est". Le dernier est moyennement cohérent mais je n'ai pas eu le temps de m'occuper de la séparation par le caractère "". Je considère donc le mot "c'est" comme étant une forme du verbe "être". Voici les résultats obtenus :

**Phrase 1 :** Proposition 0 : 'bonjour'

**Phrase 2 :** Proposition 0 : 'Tu' 'avoir' 'manger' 'quoi' 'au' 'petit' 'déjeuner'

**Phrase 3 :** Proposition 0 : 'Le' 'nutella'

Proposition 1 : 'être' 'trop' 'bon'

## 4.3 Étiquetage morpho-syntaxique des mots

A l'issue de l'étiquetage, nous obtenons le résultat suivant :

**Phrase 1 :** Proposition 0 : 'bonjour'[INTER]

**Phrase 2 :** Proposition 0 : 'Tu'[PRO] 'avoir'[VERB] 'manger'[VERB] 'quoi'[INCO]  
'au'[DET] 'petit'[ADJ] 'déjeuner'[NOM]

**Phrase 3 :** Proposition 0 : 'Le'[DET] 'nutella'[NOM]

Proposition 1 : 'être'[VERB] 'trop'[ADV] 'bon'[ADJ]

C'est résultat correspond au dictionnaire que j'ai mis en place. Cela dit ayant rempli les dictionnaires rapidement, je ne garantis pas la justesse ceux-ci.

## 4.4 Analyse syntaxique

Toutes les phrases ont réussi à passer le test de l'analyse syntaxique comme prévu. En effet les seules grammaires de phrases disponibles dans le dictionnaire sont celles qui correspondent aux phrases de test.

## 4.5 Analyse sémantique

Voici les mots-clefs qui ont été associé à chaque phrase :

**Phrase 1 :** "Salutation"

**Phrase 2 :** "Manger", "Matin"

**Phrase 3 :** "Sucrerie", "Aimer"

Ça correspond bien au résultat attendu si on prend en compte que les dictionnaires sont incomplets.

## 4.6 Création de la réponse

Malgré le fait que le programme réussisse à effectuer correctement les opérations de la création de la réponse. Il faut admettre que le résultat n'est pas réellement probant. Par exemple, la réponse de la phrase "*Le nutella, c'est trop bon*" serait la suivante : "Tu avoir avoir quoi au bon nutella."

La procédure de la création du message ne semble pas être complète pour obtenir une réponse cohérente.

## 5 Conclusion

Nous avons donc obtenu un chatbot qui effectue bon nombre d'opération nécessaire à la création d'une application de ce genre. Bien que la réponse de la création de réponse ne soit pas vraiment cohérente pour son interlocuteur. Il n'empêche que son algorithme n'est pas si intéressant.

## 6 Perspective

Les améliorations qu'on peut apporter à ce travail sont les suivantes :

- Remplissage des différents dictionnaires
- Amélioration de la création de la réponse
- Ajouter une gestion du contexte au chatbot