Intelligence Artificielle pour le Jeu Vidéo
Support de cours n°3

TAL

Contact : Agnes Delaborde – agnes.delaborde@gmail.com

Dates
Dimanche 10 avril : envoi du code et du rapport (par mail) **Mercredi 13 avril** : envoi de la présentation (par mail) **Vendredi 15 avril** : soutenance

Objectifs

Reconnaissance de patterns simples, initiation aux problématiques de bases en TAL, modélisation psycholinguistique / sociolinguistique.

Exercice 1 – Je sais qui tu es.

Nous allons doter Gus d'un mécanisme lui permettant de reconnaître automatiquement le profil du PNJ avec lequel il interagit, à partir de la façon dont le PNJ s'exprime.

a. Dans data\dialogues.lua, vous avez listé toutes les phrases des PNJ. Vous les avez a priori classées en fonction des étiquettes de profil disponibles dans la simulation (si elles ne sont pas présentées sous forme de tableau associatif, il est peut-être temps de faire un peu de rangement).

   1. Pour toutes les phrases du dialogue ask_for_name dans lesquelles le PNJ accepte de donner son nom, pour chaque profil, relevez un mot (et juste **un** mot, pas des suites de mots) qui soit discriminant, et représentatif du profil. Au besoin, modifiez vos phrases si elles ne sont pas suffisamment caricaturales.

      Par exemple, si pour le profil « Charmeur » vous avez la phrase « Moi c'est %s. Ravie de te rencontrer mon chou ! », le mot pertinent pourra être « chou ».

   2. Dans constantes.lua, créez des listes de lexique pour chaque profil. Dans ces listes, rangez vos quatre mots.

      En reprenant l'exemple précédent, vous obtiendriez :

      ```
      NPC_LEXICON = {
         [NPC_PROFILE_CHARMER] = {"chou",},
         etc.
      }
      ```

   3. Remplissez maintenant votre lexique de façon exhaustive, en vous appuyant sur toutes les énonciations que vous avez créées jusqu'à présent, pour tous les dialogues.

b. Pour chaque phrase prononcée par le PNJ, nous souhaitons que cette phrase soit parsée en mots, afin que Gus puisse rechercher des mots-clés et déterminer le profil du PNJ.

   1. Créez une fonction Gus.interaction.guessProfile() prenant comme paramètre d'entrée la phrase du PNJ, et qui renverra la valeur de profil détectée. Cette fonction devra être appelée à chaque fois qu'un PNJ a prononcé une phrase (et la valeur de retour devra être stockée dans la mémoire de Gus). Ne codez rien à l'intérieur de cette fonction, mais choisissez dès maintenant la façon la plus optimisée de réaliser cet appel.

2. Dans cette fonction, réalisez un appel à une fonction que vous nommerez parseSentence(), qui prendra en entrée une phrase, et renverra un tableau contenant les mots de la phrase. La fonction parseSentence peut tout à fait être stockée dans tools.lua.

c. Éditons maintenant la fonction parseSentence afin qu'elle renvoie la liste des mots de la phrase.

1. Je vous conseille d'utiliser l'itérateur string.gmatch, et d'adapter cet exemple (extrait de http://www.lua.org/pil/20.3.html) :

```
words = {}
for w in string.gmatch(s, "(%a)") do
  table.insert(words, w)
end
```

Un peu d'aide pour les patterns en Lua :

http://lua-users.org/wiki/PatternsTutorial (introduction)

http://www.lua.org/pil/20.2.html (patterns)

http://www.lua.org/pil/20.3.html (captures)

http://www.lua.org/manual/5.2/manual.html#6.4.1 (character class, patterns)

Problèmes avec les accents ? Consultez la section Encodage à la fin du support de cours.

2. Si vous ne l'avez pas déjà fait, mettez tous les mots du tableau en minuscule (string.lower()).
3. Testez votre parseur sur ces phrases :
   - « Aujourd'hui, je suis d'une humeur exceptionnelle ! »
   - « Amène-moi cette fiole. »
   - « J'ai pris rendez-vous chez le médecin. »
   - « J'suis trop content ! »
4. Vous avez vu que certains mots sont mal parsés. Améliorez votre algorithme pour que les mots « aujourd'hui » et « rendez-vous » soient considérés chacun comme un seul mot. Prévoyez un algorithme suffisamment généralisable pour pouvoir traiter facilement d'autres mots composés que l'on pourrait ultérieurement ajouter aux dialogues de la simulation.
5. Maintenant testez votre nouvel algorithme sur la phrase « Vous, là, rendez-vous, vous êtes cernés ! ». Vous constatez normalement un problème. Voyez-vous comment nous pourrions le résoudre ? **(ne codez rien)**
6. Éditez la fonction guessProfile() afin que pour chaque phrase, l'étiquette profil correspondante soit renvoyée aussitôt que nous avons trouvé un mot-clé dans la phrase. Prévoyez le cas où aucune étiquette n'aurait été trouvée.

   Note : Dans tools.lua, il existe une fonction table.contains(t, item) qui pourrait vous être utile.

Exercice 2 – Je sais qui tu es. Le retour.

a. Jusque là, vous avez créé vos règles en utilisant des mots-clés qui étaient propres à chaque PNJ. Vous allez améliorer vos règles de prédiction du profil en incluant également des mots non discriminants, c'est-à-dire : 1) qui peuvent être employés par plusieurs PNJ, 2) qui seraient un indice pour déterminer le profil (par élimination), mais qui 3) ne permettent pas de déterminer, à eux seuls, le profil d'un PNJ.

1. Considérez ces nouvelles règles, adaptées de théories sociolinguistiques, et vérifiez si elles conviennent à votre propre set de phrases (**ne codez rien, et ne modifiez pas vos propres phrases**) :
   - L'utilisation du pronom personnel « je » montre que le personnage aime parler de lui-même.

     → arrogant ~~charmeur~~ ~~timide~~ ~~agressif~~
   - L'utilisation du pronom personnel « je » montre que le personnage s'implique dans la discussion, il ne reste pas en retrait.

     → arrogant charmeur ~~timide~~ agressif

- L'utilisation du pronom personnel « vous » montre que le personnage met des distances avec Gus.

  → arrogant ~~charmeur~~ timide agressif

- L'utilisation du pronom personnel « tu » montre que le personnage se veut familier avec Gus.

  → ~~arrogant~~ charmeur ~~timide~~ agressif

Vous constatez obligatoirement que ces règles ne marchent pas à 100% (par ex. dans mon propre set de phrases, le timide utilise quand même « je ») : ces règles ne peuvent pas être appliquées seules (hors de tout contexte, et sans règles supplémentaires). Elles fournissent néanmoins des indices supplémentaires pour déterminer efficacement le profil.

2. Pour chaque phrase, nous allons calculer un score, en fonction de la valeur de profil associée à chacun des mots de la phrase.

   Partons de cette phrase : « Houla mon chéri, je donne pas mon nom au premier venu ! ». Dans le tableau ci-dessous, j'ai ajouté les étiquettes associées (j'ai étendu les règles du pronom personnel « je » aux adjectifs possessifs « mon, ma, mes ») :

Mot	Houla	mon	chéri	je	donne	pas	mon	nom	au	premier	venu
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
Étiquette associée	**Ø**	ARR. CHAR AGR.	CHAR	ARR. CHAR AGR.	**Ø**	**Ø**	ARR. CHAR AGR.	**Ø**	**Ø**	**Ø**	**Ø**

   Vous saisissez bien d'après cet exemple que l'étiquette « Charmeur » apparaît 4 fois, mais que « Arrogant » et « Agressif » n'apparaissent que 3 fois. L'étiquette gagnante sera donc « Charmeur ».

   - Complétez votre liste de lexique avec les pronoms personnels « je, vous » et les adjectifs possessifs « mon, ma, mes, votre, vos », et modifiez votre algorithme afin de déterminer l'étiquette gagnante.
   - Prenez une décision quant à la façon de traiter les ex aequo, ainsi que les phrases où aucune étiquette n'aura été trouvée.

b. Améliorez votre algorithme pour pouvoir prendre en compte certaines expressions figées significatives (avec 2 mots **contigus**) : « mon chou », « mon loulou », « casse-toi ».

c. Mettons qu'un PNJ arrogant utilisera, même dans un dialogue oral, la formulation « ne…pas ». Il dira « Je **ne** mange pas » plutôt que d'utiliser la formulation plus familière « Je mange pas ». Si nous devions améliorer l'algorithme pour pouvoir repérer l'utilisation de la structure « ne…pas », voyez-vous les limitations que nous aurions dans l'état actuel de notre code ?

Projet – Propositions relatives à ce cours

Proposition 1 : Le diagnostic de Gus quant au profil du PNJ n'est pas toujours bon (par exemple si la phrase ne contenait pas de mot-clé, ou si les mots listés ne permettent pas une discrimination). Gus va alors tenter de deviner le profil du PNJ à chaque fois que ce dernier parle, et stocker sa prédiction dans un historique. La valeur de profil qu'il gardera en mémoire pour ce PNJ sera le mode de toutes les prédictions réalisées jusque là.

Proposition 2 : Implémentez de nouvelles règles de détermination du profil par l'analyse, non plus des mots, mais des ponctuations et de la taille des phrases/mots. Exemples : un PNJ arrogant utilisera des mots d'une longueur moyenne supérieure aux autres PNJ, car, lui, il exprime indubitablement des mots excessivement et significativement supérieurs à la moyenne. La charmeuse est pleine de vie, elle s'exprime avec beaucoup d'exclamations ! Le timide hésite beaucoup, et… je crois… que ça se sent… Les mots et les phrases de l'agressif sont courts, nets, tranchants.

Proposition 3 : Vous vous découvrez férus de linguistique-informatique. Réalisez un étiquetage morphosyntaxique manuel sur les phrases de vos dialogues (pas nécessairement exhaustif), et réalisez des mini-transducteurs afin de repérer certaines constructions pertinentes pour l'analyse du profil (« ne…pas » est

un exemple), et de quantifier par exemple l'utilisation des adverbes (si nous partons du postulat que le PNJ arrogant tendra à utiliser beaucoup d'adverbes).

Problèmes d'encodage ?

- **Mauvais affichage des caractères accentués ?**

- Assurez-vous que toute la chaîne d'affichage est en UTF-8.
  - Au sein de votre code

  Vérifiez que votre éditeur de code est en UTF-8.
  Sur Notepad++, menu Encodage : vérifiez que « Encoder en UTF-8 » est sélectionné. Sinon, cliquez sur « Convertir en UTF-8 ».

  - A la console

  Si vous utilisez NppExec, Menu Compléments>NppExec>Console output

- **Les accents ne sont pas pris en compte dans la classe %w ni dans les groupes de type [aA-zZ] ?**

- Des informations ici : http://www.fhug.org.uk/wiki/wiki/doku.php?id=plugins:understanding_lua_patterns
- S'il vous faut la liste des accents du français pour compléter votre pattern, la voici (copiez-la depuis la version pdf de ce document sur le dossier partagé) : ÀÂÄÇÉÈÊËÎÏÔÖÙÛÜŸÆŒàâäçéèêëîïôöùûüÿæœ
- Selon certaines sources, modifier les locales pourrait permettre de régler des problèmes de gestion des caractères accentués. Modifier les locales : http://lua-users.org/wiki/LuaLocales. Le code de localisation est dépendant de votre configuration. Deux exemples, testez avec « fr_FR » ou « fra_fra ».