

Examen

Exercice 1 : (N.B. : Les questions sont indépendantes)

1. Citer un exemple de problème indécidable.
2. Expliquez comment un agent d'intelligence artificielle perçoit et agit sur l'environnement en utilisant des composants spécifiques.
3. Un système de trading automatisé effectue des transactions boursières sur un marché financier.
 - (a) Identifiez l'agent dans ce système.
 - (b) Identifiez l'environnement dans lequel l'agent opère.
 - (c) Classez cet environnement en trois catégories distinctes.
4. Donner la forme clausale de la formule suivante : $((P \rightarrow (Q \rightarrow R)) \rightarrow ((P \wedge S) \rightarrow R))$.
5. Montrer que $P \wedge Q \wedge \neg(P \vee Q)$ est inconsistante en utilisant la table de vérité. *Ce toujours faux*
6. La Formule Bien Formée $P \wedge \neg P$ est-elle valide ? Pourquoi ? *invalid*

Exercice 2 :

On considère les connaissances suivantes :

" Si la Licorne est mythique, alors elle est immortelle. Si elle n'est pas mythique, alors c'est un mammifère mortel. Si la Licorne est immortelle ou est un mammifère, alors elle a une corne. Si elle a une corne, alors elle est magique."

1. Proposer une formulation des prédicats liés à ces connaissances.
2. Formuler les règles d'inférences associées.
3. Énoncer la loi de Morgan.
4. Pour des termes P, Q et R , démontrer que :

- (a) La règle $(P \rightarrow (Q \wedge R))$ peut être remplacée par les deux règles $(P \rightarrow Q)$ et $(P \rightarrow R)$.
- (b) La règle $((P \vee Q) \rightarrow R)$ peut être remplacée par les deux règles $(P \rightarrow R)$ et $(Q \rightarrow R)$.
5. Reformuler alors de cette manière les règles du problème de la licorne, quand c'est nécessaire.
6. Transformer les règles en des formes clausales.
7. Prouver que la Licorne est magique

Exercice 3 : (N.B. : Les questions sont indépendantes)

1. Dans une matrice de cooccurrence, que représente l'élément $M[i][j]$? (Sélectionnez une seule réponse)
- A) La fréquence de la paire de mots (i, j)
 - B) La fréquence du mot i suivi par le mot j
 - C) La fréquence du mot j suivi par le mot i
 - ☒ D) Le poids TF-IDF entre les mots i et j
2. Quel est le rôle principal de techniques d'embedding de mots comme Word2Vec ? (Sélectionnez une seule réponse)
- A) Classification de texte
 - B) Réduction de la dimensionnalité du texte.
 - ☒ C) Conversion de texte en vecteurs numériques
 - ~~D) Suppression des mots vides (stopwords)~~
3. (a) Expliquer les concepts de stemming et de lemmatisation en traitement du langage naturel.
- (b) Énumérer les principales différences entre stemming et lemmatisation.
- (c) Fournir un exemple pour illustrer chacune de ces techniques. Décrivez comment un mot donné serait transformé par chaque méthode.
- (d) Identifier la méthode qui, bien que plus précise, est généralement plus lente à exécuter. Justifiez votre réponse.
- (e) Écrire un fragment de code Python pour effectuer du stemming et de la lemmatisation en utilisant le package "nltk". Assurez-vous que le code est bien commenté.

4. Supposons que vous ayez une liste de commentaires appelée `comments`, où chaque commentaire a été pré-traité et est représenté sous forme de liste de mots. De plus, chaque commentaire est associé à une étiquette de sentiment (`ys`), qui est soit 0 (négatif) soit 1 (positif).

(a) Expliquez le rôle de la fonction `build_freqs` dans le code ci-dessous.

(b) Quelle est la structure de données de sortie (`freqs`) et que contient-elle?

(c) Que fait la ligne `yslist = np.squeeze(ys).tolist()`?

(d) Expliquez le rôle du bloc `for` imbriqué dans la fonction.

```
def build_freqs(comments, ys):
    """ Build frequencies.
    Input:
        comments: a list of comments from social media
        ys: an n x 1 array with the sentiment label of each comment (either 0 or 1)
    Output:
        freqs: a dictionary mapping each (word, sentiment) pair to its frequency
    """
    yslist = np.squeeze(ys).tolist()
    freqs = {}
    for y, comment in zip(yslist, comments):
        for word in comment:
            pair = (word, y)
            if pair in freqs:
                freqs[pair] += 1
            else:
                freqs[pair] = 1
    return freqs
```