ENSAM – Casablanca **Année Universitaire 2025-2026**
Département Génie Informatique **Pr. Soukaina Bouhsissin**

Intelligence Artificielle

TP 3 : Chatbot à Règles
Intentions + Mémoire de Contexte

Objectifs du TP

- Implémenter un agent symbolique (sans apprentissage automatique)
- Détecter des intentions à partir de mots-clés
- Gérer une mémoire de contexte (dernier sujet abordé)
- Concevoir des modules distincts : détection, mémoire, génération de réponses
- Comprendre les limitations des systèmes à règles

Scénario

Développer un chatbot capable de :

- Répondre aux salutations
- Donner l'heure actuelle
- Fournir une météo factice pour différentes villes
- Se présenter
- Dire au revoir

Le chatbot reconnaît des intentions (salutation, météo, heure, identité, aurevoir) et se souvient du dernier sujet pour interpréter des questions ambiguës comme « Et demain ? » après une question météo.

Partie 1 : Intentions et Mots-Clés

Dans cette première partie, vous allez définir les intentions que votre chatbot peut reconnaître et les mots-clés associés à chaque intention.

Question 1 : Définition du dictionnaire INTENTS

Complétez le dictionnaire INTENTS avec au moins 2 synonymes ou variantes par intention. Vous devez couvrir les intentions suivantes :

- **salutation** : mots pour dire bonjour
- **meteo** : mots liés à la météo et aux villes
- **heure** : mots pour demander l'heure
- **identite** : questions sur l'identité du bot
- **aurevoir** : mots pour dire au revoir

Exemple de structure :

```
INTENTS = {
   "salutation": ["bonjour", "salut", ...],
   "meteo": ["meteo", "temps", ...],
   ...
}
```

Votre code Python :

Partie 2 : Normalisation du Texte

La normalisation du texte est une étape cruciale pour améliorer la robustesse de la détection d'intentions. Elle permet de traiter uniformément les variations d'écriture.

Question 2 : Implémentation de la fonction normalize

Implémentez la fonction normalize(text) qui doit :

1. Convertir tout le texte en minuscules
2. Retirer les accents (approximation : é→e, è→e, à→a, etc.)
3. Supprimer la ponctuation (remplacer par des espaces)
4. Supprimer les espaces multiples et les espaces en début/fin

Exemples de tests :

normalize("Bonjour!!") → "bonjour"

normalize("QUELLE heure ?") → "quelle heure"

normalize("Météo à Casablanca") → "meteo a casablanca"

Votre code Python :

Partie 3 : Détection d'Intention

Maintenant que vous avez défini les intentions et la normalisation, vous allez implémenter la fonction qui détecte l'intention d'un message utilisateur.

Question 3 : Implémentation de detect_intent

Implémentez la fonction detect_intent(message) qui :

1. Normalise le message avec la fonction normalize
2. Parcourt le dictionnaire INTENTS
3. Pour chaque intention, teste si un mot-clé apparaît dans le message
4. Retourne le nom de l'intention trouvée, ou "inconnu" si aucune correspondance

Exemples de tests :

detect_intent("Salut") → "salutation"

detect_intent("Quelle est la meteo de Rabat ?") → "meteo"

detect_intent("Quelle heure svp") → "heure"

detect_intent("Qui es-tu ?") → "identite"

Votre code Python :

Partie 4 : Génération de Réponses + Mémoire

C'est la partie la plus importante du TP ! Vous allez implémenter la fonction qui génère les réponses en fonction de l'intention détectée et qui gère la **mémoire de contexte**.

Question 4 : Implémentation de respond avec mémoire

Implémentez la fonction respond(intent, message, memory) qui doit :

A. Réponses de base

- **salutation** : Retourner une salutation aléatoire parmi plusieurs options
- **heure** : Retourner l'heure actuelle (format HH:MM)
- **identite** : Se présenter comme un agent IA
- **aurevoir** : Dire au revoir

B. Météo avec extraction de ville

Pour l'intention "meteo" :

5. Extraire la ville mentionnée dans le message (casablanca, rabat, marrakech)
6. Si aucune ville n'est mentionnée, utiliser "casablanca" par défaut
7. Mémoriser last_intent = "meteo" et last_city = ville
8. Retourner la météo pour cette ville

Données météo factices :

```
FAKE_WEATHER = {"casablanca": 23, "rabat": 22, "marrakech": 26}
```

C. Gestion de la mémoire

Si l'intention est "inconnu" ET que last_intent dans la mémoire vaut "meteo" :

- Vérifier si le message contient des mots comme "demain", "apres demain", "ensuite"
- Si oui, utiliser last_city de la mémoire pour donner la prévision

Exemple de dialogue avec mémoire :

Vous: La meteo a Marrakech svp

Bot: Météo à Marrakech : 26°C

Vous: Et demain ?

Bot: Prévision à Marrakech demain : ensoleillé, ~26°C

Votre code Python (fonction respond complète) :

Partie 5 : Boucle de Dialogue Interactive

Maintenant que toutes les fonctions sont implémentées, vous allez créer une boucle interactive permettant de discuter avec le chatbot.

Question 5 : Implémentation de chat_loop

Implémentez la fonction chat_loop() qui :

1. Initialise un dictionnaire memory vide
2. Affiche un message de bienvenue
3. Entre dans une boucle infinie qui lit les messages de l'utilisateur
4. Détecte l'intention du message
5. Génère et affiche la réponse
6. Sort de la boucle si l'utilisateur tape 'exit', 'quit' ou 'bye'

Votre code Python :

Testez votre chatbot avec le dialogue suivant :

Vous: Bonjour

Vous: Quelle heure est-il ?

Vous: La météo à Casablanca

Vous: Et demain ?

Vous: Qui es-tu ?

Vous: Au revoir

Partie 6 : Questions de Réflexion

Cette dernière partie vous invite à réfléchir sur les concepts abordés et les limitations de l'approche par règles.

Question 6.1 : Différence intention vs réponse

Expliquez la différence entre une intention et une réponse dans le contexte d'un chatbot. Pourquoi sépare-t-on ces deux concepts ?

Question 6.2 : Utilité de la normalisation

Pourquoi la normalisation du texte aide-t-elle à améliorer la détection d'intentions ? Donnez un exemple concret où la normalisation fait la différence entre détecter ou non une intention.

Question 6.3 : Exemple d'utilisation de la mémoire

Donnez un exemple de dialogue où la mémoire de contexte change la réponse du chatbot. Expliquez ce qui est mémorisé et comment cela affecte la réponse.

Question 6.4 : Limites de l'approche par règles

Identifiez deux limites majeures de l'approche par règles et mots-clés pour créer un chatbot. Pour chaque limite, proposez une amélioration possible (sans nécessairement l'implémenter).

Limite 1 :

Amélioration proposée :

Limite 2 :

Amélioration proposée :

Question 6.5 : Extension - Intention "blague"

Ajoutez une nouvelle intention "blague" qui répond par une blague aléatoire parmi 3 options. Montrez le code et testez-le avec des exemples.

Code de l'extension (ajouts dans INTENTS et respond) :

Exemples de test :