



IMT Mines Alès
École Mines-Télécom

Mission R&D 2024



ESII : AGENT VIRTUEL D'ACCUEIL

Mission n°4

ÉLÈVES :

CRUVELLIER Chloé

LEGAREZ Lucie

TERRASSON Ludovic

TUTEURS :

LERGENMULLER Philippe

TCHECHMEDJIEV Andon

Contexte de la mission

Entreprise à l'origine du sujet



Solutions innovantes pour la gestion des flux et le parcours client



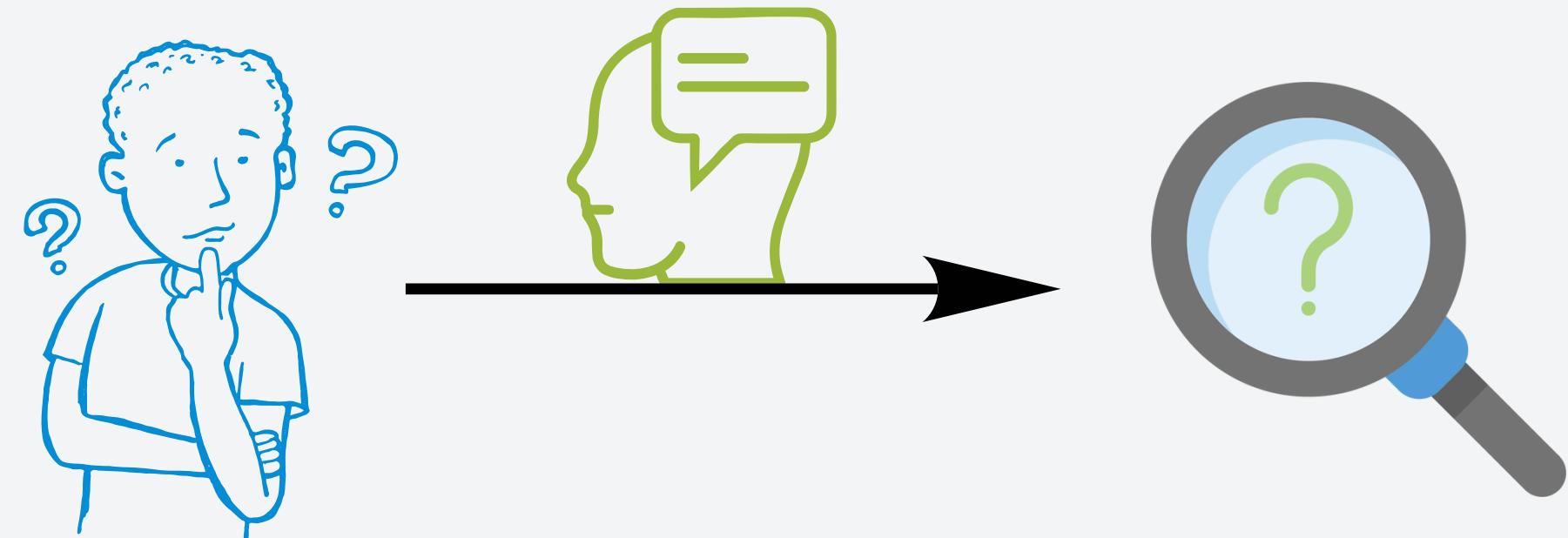
Objectif : rendre la borne plus interactive en détectant le motif de visite en analysant la demande orale du client (et non plus par un système de choix)

Plan de la présentation

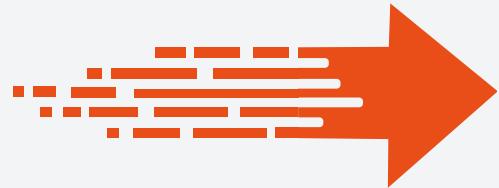
1. Approches possibles
2. Notre démarche
 - a. données à disposition
 - b. trois modèles
3. Résultats
4. Préconisations
5. Conclusion



Précision de l'objectif



- Reconnaissance vocale
- **Reconnaissance du motif dans la demande**



Modèle **en local** / en ligne

Approches possibles



Comment analyser une demande et en déduire un motif ?

Règles et **arbres de décision**
(guide le client)

Traitement de langage naturel

(Analyse les messages des utilisateurs pour fournir des réponses appropriées en utilisant des techniques de NLP)

Mots clés

Apprentissage automatique
(amélioration continue)

Notre démarche

Données à disposition : mission R&D de 2020

Modèles de machine learning (Régression logistique, arbre de décision, forêt aléatoire)



Modèle génératif (LLM)

Regroupement de demandes et de motifs associés (triées par difficulté) pour la mairie de Roanne

mairie_roanne_demands_motifs_withLevel		
difficulté	demande	motif
0	carte électorale	Inscription sur liste électorale
1	oui bonjour je viens pour effectuer une modification sur le dossier par rapport	Renseignements, modification de dossier
1,5	j'ai besoin d'information sur un sur un acte de mariage	Actes de naissance, mariage, décès
2	Bonjour J'aurais besoin des différents actes d'état civil pour l'ensemble de m	Actes de naissance, mariage, décès
2	Bonjour j'ai perdu des papiers et j'aimerais avoir un certificat de naissance	Certificats, légalisation de signature

Notre démarche

Modèles étudiés

1

Modèle par mots-clés

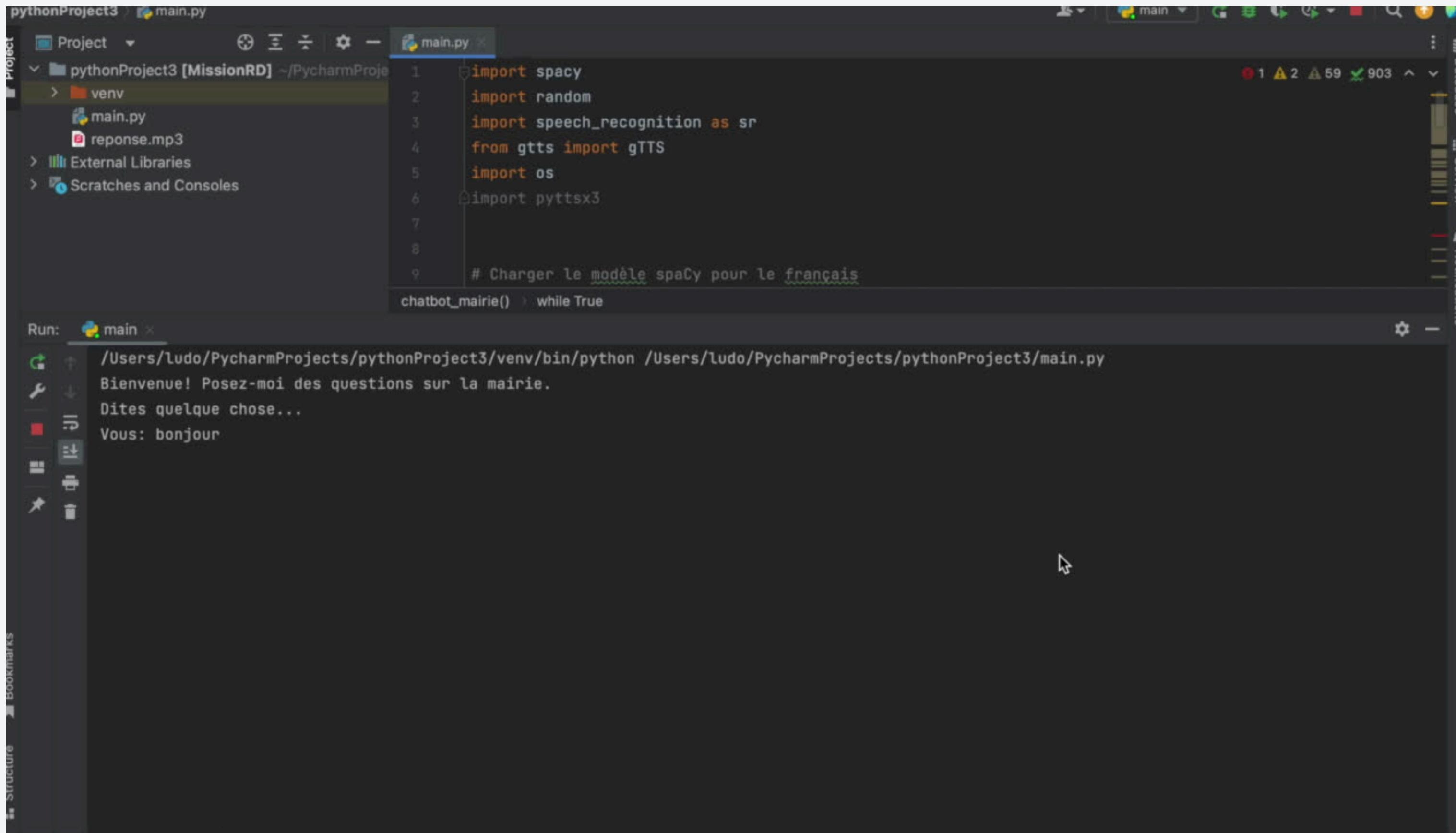
2

Modèle conversationnel

3

llama.cpp

1er modèle : par mots-clés (python) - présentation



```
pythonProject3 / main.py
Project  main.py
pythonProject3 [MissionRD] ~/PycharmProje
  venv
  main.py
  reponse.mp3
> External Libraries
> Scratches and Consoles

main.py
1 import spacy
2 import random
3 import speech_recognition as sr
4 from gtts import gTTS
5 import os
6 import pyttsx3
7
8
9 # Charger le modèle spaCy pour le français
chatbot_mairie() > while True

Run: main
/Users/ludo/PycharmProjects/pythonProject3/venv/bin/python /Users/ludo/PycharmProjects/pythonProject3/main.py
Bienvenue! Posez-moi des questions sur la mairie.
Dites quelque chose...
Vous: bonjour
```

1er modèle : par mots-clés (python) - analyse

Explications

- spacy: traitement du langage naturel (NLP).
- speech_recognition : reconnaissance vocale pour traiter les entrées vocales.
- gTTS: bibliothèque Google Text-to-Speech pour synthétiser la parole.

Points positifs

Réponses dynamiques et aléatoires

Synthèse vocale

Points d'amélioration

Détection par mot clés

2e modèle : par RASA - présentation

The screenshot shows the PyCharm IDE interface with a project named "pythonProject6 [Rasa Project]". The project structure includes a "static" folder with images "Mairie.jpg" and "Mairie2.jpg", a "templates" folder containing "main.py", and an "External Libraries" section listing Python 3.9 dependencies like "Binary Skeletons", "Extended Definitions", "lib-dynload", "python3.9 library root", "site-packages", "TypedStubs", and "Scratches and Consoles". The "Run" tab at the bottom shows the command: "/Users/ludo/PycharmProjects/pythonProject6/venv/bin/python /Users/ludo/PycharmProjects/pythonProject6/main.py". The output window displays the Flask development server logs:

```
Uploading static files...
* Serving Flask app 'main'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
```

The main.py code defines a Flask application that routes to index.html and serves static files from the static directory:

```
from flask import Flask, request, jsonify, render_template, send_from_directory
import requests
import os

app = Flask(__name__)

# URL de l'API du chatbot Rasa
rasa_api_url = "http://localhost:5005/webhooks/rest/webhook" # Mettez à jour l'URL en fonction de votre configuration

# Page d'accueil de l'interface web
@app.route('/')
def home():
    return render_template('index.html')

# Servir l'image statique
@app.route('/static/<path:path>')
def serve_static(path):
    root_dir = os.path.dirname(os.getcwd())
    return send_from_directory(os.path.join(root_dir, 'static'), path)
```

2e modèle : par RASA - analyse

Explications

- Outil pour la construction d'assistants conversationnels personnalisés
- Utilise des fichiers de configuration pour entraîner un modèle qui comprend les intentions, les actions et les réponses de l'assistant :
 - intentions
 - réponses possibles.
 - scénarios de conversation
 - règles
- Serveur Web Flash, interface Web, execution de l'application

2e modèle : par RASA - analyse

Points positifs

Diversité des demandes

Réponse construite

Interface utilisateur
ergonomique

Open Source

Personnalisation de
l'assistant

Limites

Trop complexe pour notre
objectif (conversationnel ≠
procédural)

Configuration complexe

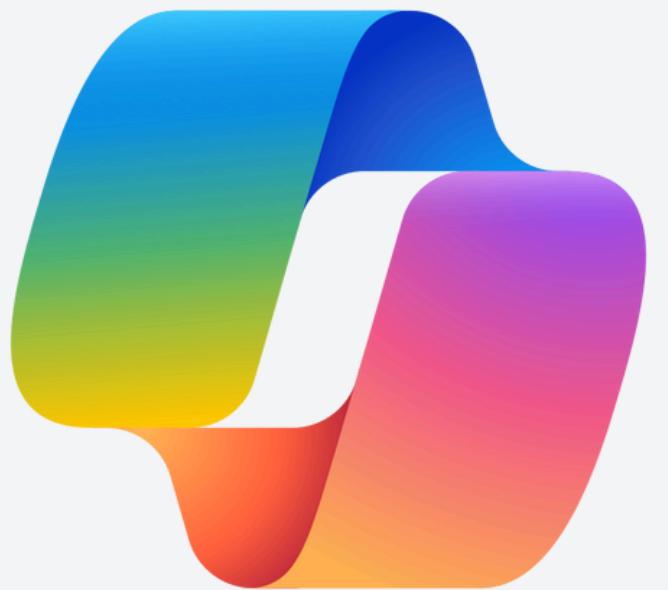
Données d'entraînement

3e modèle : par llama.cpp

✓ modèle retenu

IA Générative

Ressources importantes, serveurs distants,
connexion internet



Déploiement local, sans connexion internet

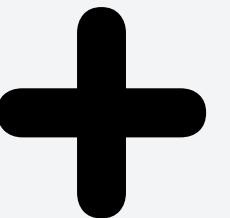


3e modèle : par llama.cpp

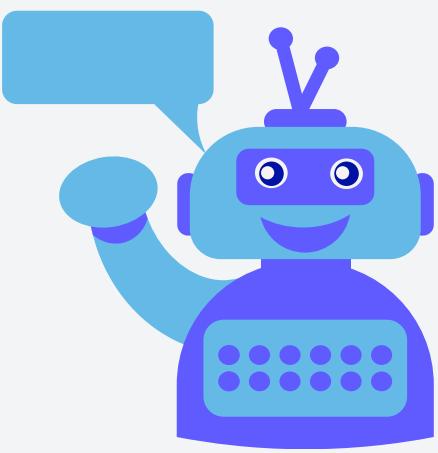
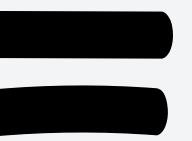
✓ modèle retenu



modèle
(Générer le texte)



Prompt
(Donner les instructions)



Chatbot

Modèles

“Instruct”

- Fournir instructions ou directives spécifiques
- Utilisateur contrôle totalement le contenu généré

NeuralBeagle14-French-Aplaca.Q4_0.gguf
(4 Go)

French-Alpaca-7B-Instruct-quantized-q8_0-beta.gguf
(7,5 Go)

“Chat”

- Simuler des conversations
- Utilisateur communique avec le chatbot

llama-2-7b-chat.Q2_K.gguf
(2,7 Go)

llama-2-13b-chat-FR-Q4_K_M.gguf
(7,6 Go)

Vous êtes une borne d'accueil intelligente dans la mairie de Roanne, chaque visiteur dans la mairie passe devant vous et dit à haute voix son motif de visite.

Votre rôle est d'identifier le motif de visite de chaque visiteur parmi les motifs possibles suivants, c'est une liste exhaustive. Attention : tu ne dois utiliser que ces motifs là et ne pas en inventer. Sur chaque ligne, l'intitulé du motif est donné:

Actes de naissance, mariage, décès

Enregistrement de PACS

Livret de famille

Déclaration de naissance, Reconnaissance

Déclaration de décès

Mariage

Recensement des jeunes

Changement de prénoms, rectification d'actes

Certificats, légalisation de signature

Inscription sur liste électorale

Première Inscription scolaire-changement d'école

Inscription Périscolaire (Cantine et Accueil)

Commande ou décommande de repas

Règlement cantine en espèces

Renseignements, modification de dossier

relance amiable : référence courrier reçu RC

contentieux locataire présent : référence courrier reçu FC

contentieux locataire parti : référence courrier reçu FC

explication avis échéance loyer

explication régularisation des charges

décompte de sortie (locataire quittant OPHEOR)

mise en place contrat prélèvement

demandes d'attestations diverses

PACS (Dépôt de dossier, modification ou dissolution)

Prompt :

- Phrase de contexte
- Liste des motifs possibles (25 motifs)
- Directives spécifiques
- Exemple de fonctionnement
- Entrée utilisateur

En tant qu'assistant virtuel d'accueil, tu dois générer les motifs correspondant aux demandes formulées par l'utilisateur 'User', c'est une vraie personne qui va formuler les demandes.

Lorsque l'utilisateur formule une demande : tu dois lui répondre uniquement par le motif correspondant, tel qu'indiqué dans la liste des motifs fournie précédemment. Tu ne peux pas inventer de motif. Assure-toi que le motif généré corresponde bien à la demande de l'utilisateur en analysant précisément la demande et en examinant tous les motifs pour choisir celui qui convient le mieux.

Tu ne dois pas générer de texte autre que le motif associé à la demande.

User: j'aimerais faire une inscription sur la liste électorale

Borne : Inscription sur liste électorale

User : je viens reconnaître mon enfant

Borne: Déclaration de naissance, Reconnaissance

User: j'ai un impayé alors que j'ai quitté les locaux

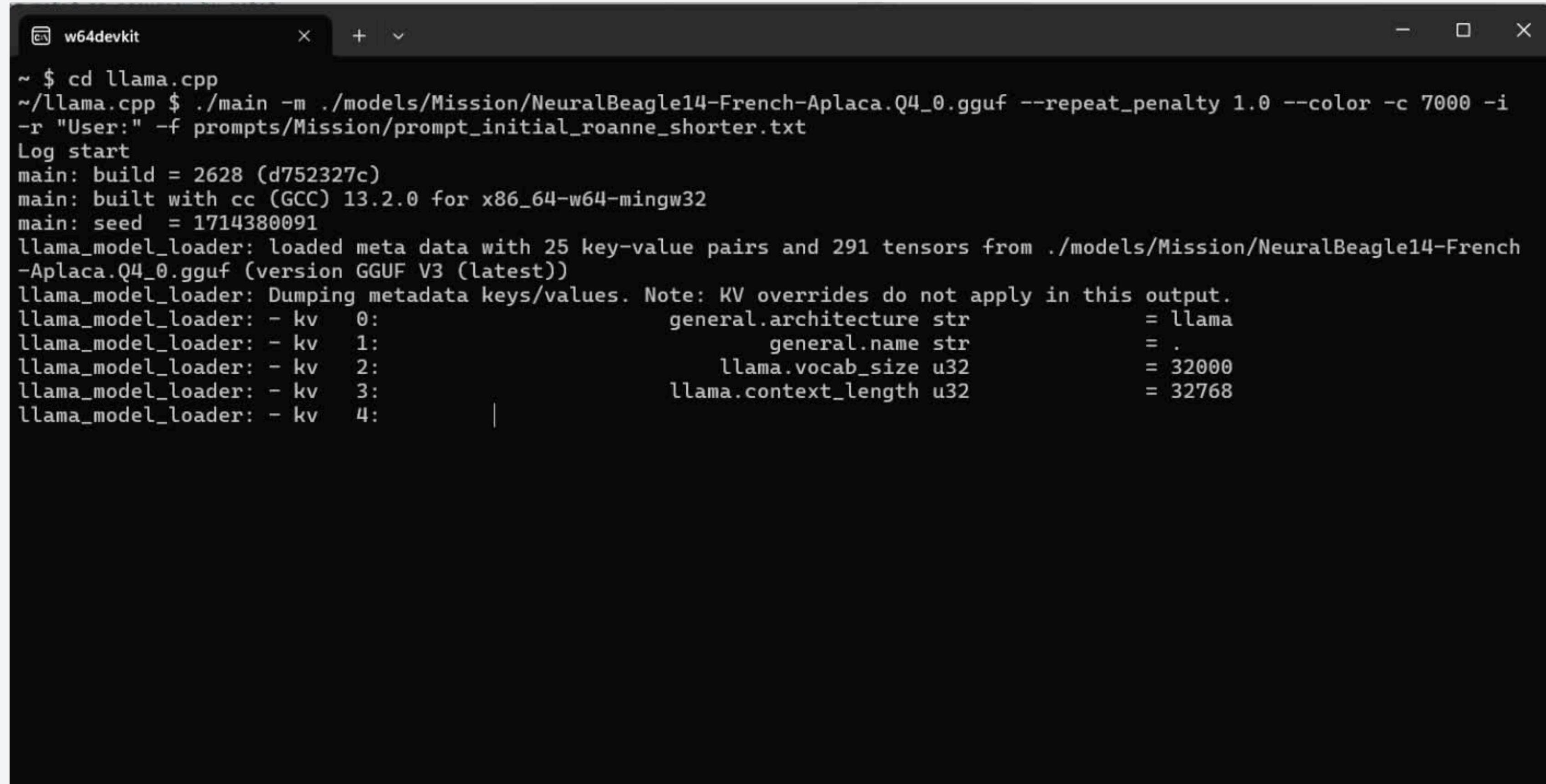
Borne: contentieux locataire parti : référence courrier reçu FC

Commandes

Commande à effectuer dans terminal :

```
./main -m ./models/Mission/NeuralBeagle14-French-Aplaca.Q4_0.gguf --repeat_penalty 1.0 --color -c 7000 -i -r "User:" -f prompts/Mission/prompt_initial_roanne_shorter.txt
```

Llam.cpp : lancement du modèle



```
w64devkit ~ $ cd llama.cpp
~/llama.cpp $ ./main -m ./models/Mission/NeuralBeagle14-French-Aplaca.Q4_0.gguf --repeat_penalty 1.0 --color -c 7000 -i
-r "User:" -f prompts/Mission/prompt_initial_roanne_shorter.txt
Log start
main: build = 2628 (d752327c)
main: built with cc (GCC) 13.2.0 for x86_64-w64-mingw32
main: seed = 1714380091
llama_model_loader: loaded meta data with 25 key-value pairs and 291 tensors from ./models/Mission/NeuralBeagle14-French
-Aplaca.Q4_0.gguf (version GGUF V3 (latest))
llama_model_loader: Dumping metadata keys/values. Note: KV overrides do not apply in this output.
llama_model_loader: - kv 0: general.architecture str = llama
llama_model_loader: - kv 1: general.name str = .
llama_model_loader: - kv 2: llama.vocab_size u32 = 32000
llama_model_loader: - kv 3: llama.context_length u32 = 32768
llama_model_loader: - kv 4: |
```

Llam.cpp : exemples de demandes

```
w64devkit x + - □ ×

explication avis échéance loyer
explication régularisation des charges
décompte de sortie (locataire quittant OPHEOR)
mise en place contrat prélèvement
demandes d'attestations diverses
PACS (Dépôt de dossier, modification ou dissolution )

En tant qu'assistant virtuel d'accueil, tu dois générer les motifs correspondant aux demandes formulées par l'utilisateur 'User', c'est une vraie personne qui va formuler les demandes.
Lorsque l'utilisateur formule une demande : tu dois lui répondre uniquement par le motif correspondant, tel qu'indiqué dans la liste des motifs fournie précédemment. Tu ne peux pas inventer de motif. Assure-toi que le motif généré correspond bien à la demande de l'utilisateur en analysant précisément la demande et en examinant tous les motifs pour choisir celui qui convient le mieux.
Tu ne dois pas générer de texte autre que le motif associé à la demande.

User: j'aimerais faire une inscription sur la liste électorale
Borne : Inscription sur liste électorale

User : je viens reconnaître mon enfant
Borne: Déclaration de naissance, Reconnaissance

User: j'ai un impayé alors que j'ai quitté les locaux
Borne : contentieux locataire parti : référence courrier reçu FC

User:je veux me marier
Borne : mariage

User:je veux déclare|
```

Résultats

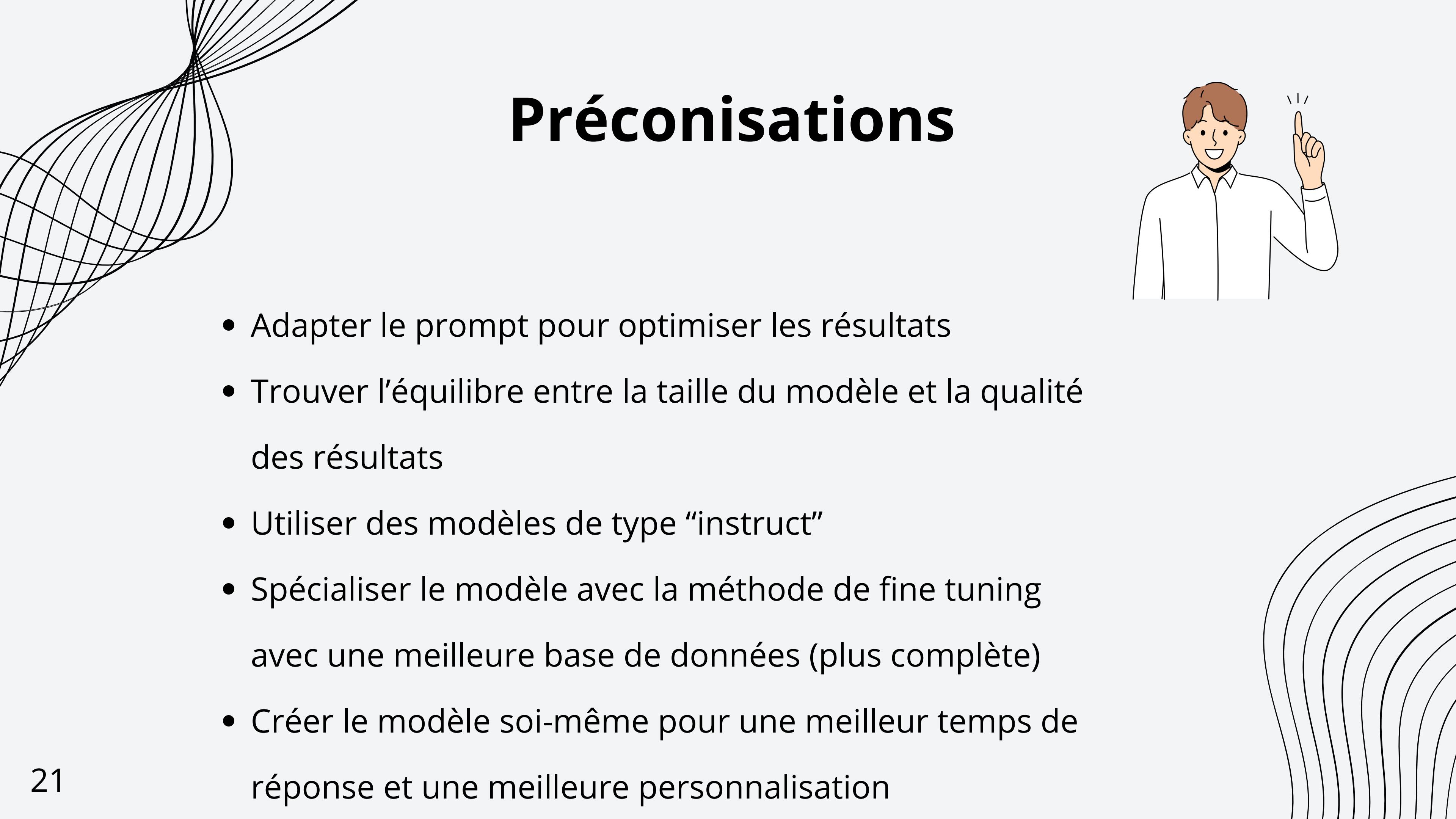
➤ 265 demandes

73 %

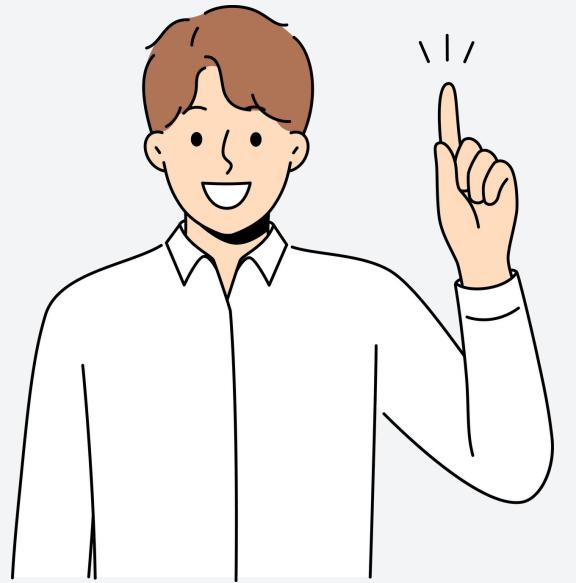
de précision

Précision similaire pour chaque **niveau de difficulté** (0,1,1.5, 2 et 2.5)

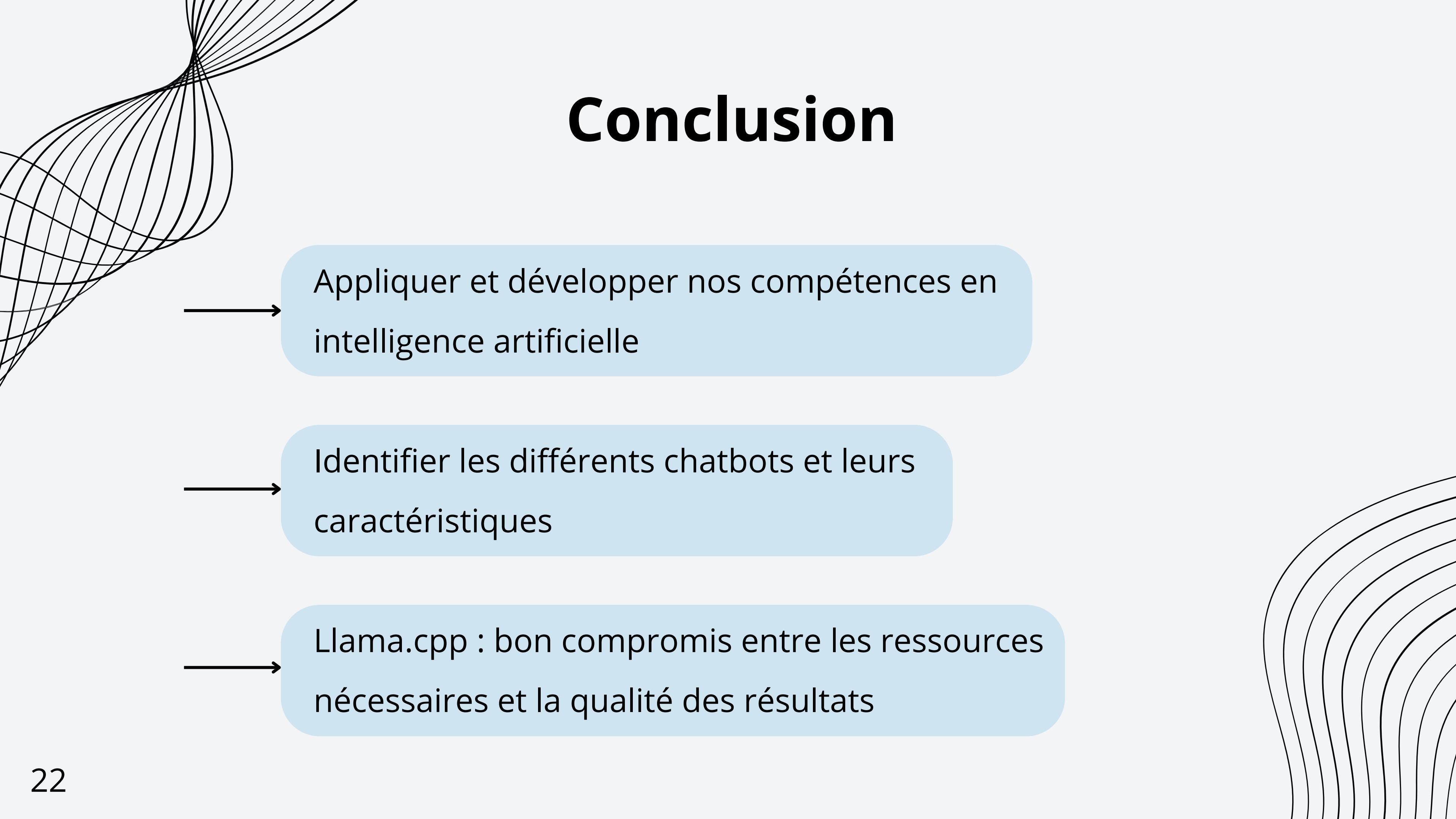
Motifs plus ou moins bien détectés



Préconisations



- Adapter le prompt pour optimiser les résultats
- Trouver l'équilibre entre la taille du modèle et la qualité des résultats
- Utiliser des modèles de type “instruct”
- Spécialiser le modèle avec la méthode de fine tuning avec une meilleure base de données (plus complète)
- Créer le modèle soi-même pour une meilleur temps de réponse et une meilleure personnalisation



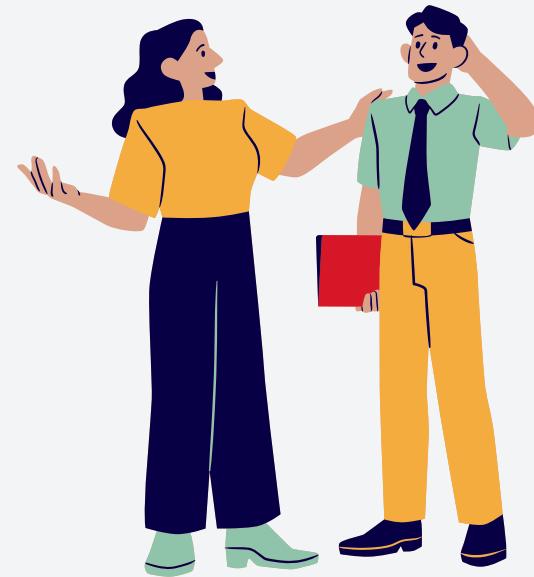
Conclusion

→ Appliquer et développer nos compétences en intelligence artificielle

→ Identifier les différents chatbots et leurs caractéristiques

→ Llama.cpp : bon compromis entre les ressources nécessaires et la qualité des résultats

Remerciements



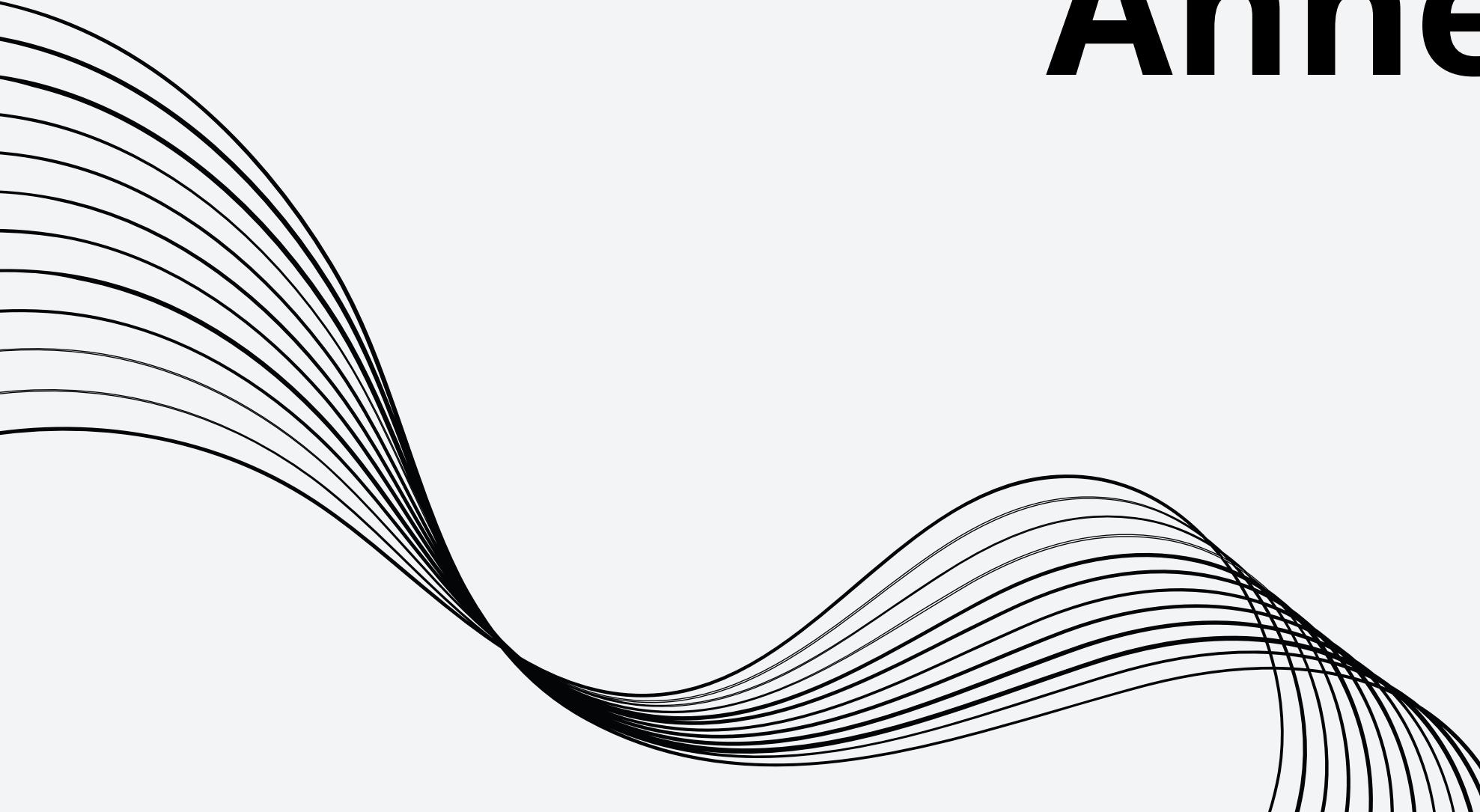
- Tuteur entreprise : LERGENMULLER Philippe
- Tuteur école : TCHECHMEDJIEV Andon
- Les documentalistes

Merci à vous

Avez-vous des questions ?



Annexes



ANNEXE 1 - CODE VERSION 1

Rasa Project – main.py MissionRD – main.py

pythonProject3 > main.py

```
1 import spacy
2 import random
3 import speech_recognition as sr
4 from gtts import gTTS
5 import os
6 import pyttsx3

# Charger le modèle spaCy pour le français
nlp = spacy.load("fr_core_news_sm")

# Liste de réponses possibles
reponses = {
    "bonjour": ["Bonjour! Comment puis-je vous aider aujourd'hui?", "Salut! En quoi puis-je vous assister?", "Hello! Quelle est votre demande?"],
    "horaires": ["Les horaires d'ouverture de la mairie sont du lundi au vendredi de 9h à 17h.", "La mairie est ouverte en semaine de 9h à 17h."],
    "services": ["La mairie propose divers services tels que l'inscription à des événements, la délivrance de documents, etc.", "Nos services sont variés et couvrent divers domaines administratifs."],
    "au revoir": ["Au revoir! N'hésitez pas à revenir si vous avez d'autres questions.", "À bientôt! Si vous avez besoin d'aide, je suis là."],
    "carte d'identité": ["Pour demander une carte d'identité, veuillez suivre ces étapes:\n1. Présentez-vous en personne à la mairie.\n2. Apportez un passeport biométrique."],
    "permis de construire": ["Pour obtenir un permis de construire, suivez ces étapes:\n1. Consultez le service d'urbanisme de la mairie.\n2. Apportez les documents nécessaires."],
    "passeport": ["Pour obtenir un passeport, suivez ces étapes:\n1. Prenez rendez-vous au service des passeports de la mairie.\n2. Apportez les documents nécessaires."],
    "mariage": ["Pour célébrer un mariage à la mairie, suivez ces étapes:\n1. Prenez rendez-vous au service d'état civil.\n2. Apportez les documents nécessaires."],
    "inscription à l'école": ["Pour inscrire votre enfant à l'école, veuillez suivre ces étapes:\n1. Contactez l'école de votre choix.\n2. Apportez les documents nécessaires."],
    "prise de rendez-vous": ["Pour prendre un rendez-vous à la mairie, suivez ces étapes:\n1.appelez le service approprié pour fixer un rendez-vous."],
    "passeport biométrique": ["Pour obtenir un passeport biométrique, suivez ces étapes:\n1. Prenez rendez-vous au service des passeports de la mairie.\n2. Apportez les documents nécessaires."],
    "déclaration de naissance": ["Pour faire une déclaration de naissance à la mairie, suivez ces étapes:\n1. Apportez le certificat médical de naissance.\n2. Prenez rendez-vous au service d'état civil."],
    "certificat de résidence": ["Pour obtenir un certificat de résidence, suivez ces étapes:\n1. Apportez les documents justificatifs (factures, relevés bancaires, etc.).\n2. Prenez rendez-vous au service d'état civil."],
    "renouvellement de carte d'identité": ["Pour renouveler votre carte d'identité, suivez ces étapes:\n1. Présentez-vous en personne au service d'état civil.\n2. Apportez la carte d'identité actuelle."],
    "certificat de mariage": ["Pour obtenir un certificat de mariage, suivez ces étapes:\n1. Présentez-vous au service d'état civil avec les documents nécessaires.\n2. Apportez la carte d'identité."],
    "demande de subvention": ["Pour faire une demande de subvention à la mairie, suivez ces étapes:\n1. Consultez le service des affaires sociales.\n2. Apportez les documents nécessaires."],
    "certificat de décès": ["Pour obtenir un certificat de décès, suivez ces étapes:\n1. Présentez-vous au service d'état civil avec le certificat de décès.\n2. Apportez la carte d'identité."]
}

chatbot_mairie() > while True
```

Database SciView Notifications

Version Control Python Packages TODO Python Console Problems Terminal Services

136:16 LF UTF-8 4 spaces Python 3.9 (pythonProject3)

ANNEXE 1 - CODE VERSION 1

Rasa Project – main.py MissionRD – main.py

pythonProject3 > main.py

Project main.py x

43 "demande de visa": ["Pour faire une demande de visa, suivez ces étapes:\n1. Consultez le service des affaires étra", 1, 2, 59, 903, ...]

44 } }

45

46 # Liste de synonymes pour les mots clés

47 synonyms = {

48 "bonjour": ["salut", "bonjour", "hello", "coucou"],

49 "horaires": ["heures d'ouverture", "horaires", "quand ouvert", "quand ferme"],

50 "services": ["prestations", "services", "offres", "activités"],

51 "au revoir": ["bye", "à bientôt", "au revoir", "adieu"],

52 "carte d'identité": ["carte d'identité", "carte d'identité", "CNI", "procédure d'obtention de carte d'identité"],

53 "permis de construire": ["permis de construire", "permis de construire", "autorisation de construire", "construction"],

54 "passeport": ["passeport", "demande de passeport", "passeport", "renouvellement de passeport"],

55 "mariage": ["mariage", "cérémonie de mariage", "union civile"],

56 "inscription à l'école": ["inscription à l'école", "inscrire enfant à l'école", "démarches scolaires"],

57 "prise de rendez-vous": ["prendre rendez-vous", "prise de rendez-vous", "rendez-vous mairie", "fixer un rendez-vous"],

58 "passeport biométrique": ["passeport biométrique", "demande de passeport biométrique", "renouvellement passeport biométrique"],

59 "déclaration de naissance": ["déclaration de naissance", "nouveau-né", "naissance"],

60 "certificat de résidence": ["certificat de résidence", "résidence", "justificatif de domicile"],

61 "renouvellement de carte d'identité": ["renouvellement de carte d'identité", "renouveler CNI", "nouvelle carte d'identité"],

62 "certificat de mariage": ["certificat de mariage", "mariage civil", "acte de mariage"],

63 "demande de subvention": ["demande de subvention", "subvention projet", "aide financière"],

64 "certificat de décès": ["certificat de décès", "décès", "acte de décès"],

65 "déclaration de changement d'adresse": ["déclaration de changement d'adresse", "changer d'adresse", "nouvelle adresse"],

66 "demande d'aide sociale": ["demande d'aide sociale", "aide sociale", "assistance financière"],

67 "démarches pour le vote": ["démarches pour le vote", "inscription sur les listes électorales", "participer aux élections"],

68 "demande de logement social": ["logement social", "demande de logement", "aide au logement"],

69 "certificat de non-gage": ["certificat de non-gage", "véhicule d'occasion", "certificat de situation administrative"],

70 "autorisation de sortie du territoire (AST)": ["autorisation de sortie du territoire", "AST", "voyage mineur"],

71 "certificat de scolarité": ["certificat de scolarité", "attestation scolaire", "document pour l'école"],

72 "demande d'extrait de casier judiciaire": ["extrait de casier judiciaire", "casier judiciaire", "antécédents judiciaires"]

chatbot_mairie() > while True

Version Control Python Packages TODO Python Console Problems Terminal Services

136:16 LF UTF-8 4 spaces Python 3.9 (pythonProject3)

ANNEXE 1 - CODE VERSION 1

Rasa Project – main.py

MissionRD – main.py

```
pythonProject3 > main.py
```

```
Project - main.py x
Rasa Project – main.py
MissionRD – main.py
pythonProject3 > main.py
...
77     "demande de visa": ["demande de visa", "voyage à l'étranger", "autorisation de voyage"],
78 }
79
80 def repondre(message):
81     # Utiliser la reconnaissance vocale pour obtenir le texte à partir de l'audio
82     recognizer = sr.Recognizer()
83     with sr.Microphone() as source:
84         print("Dites quelque chose...")
85         audio = recognizer.listen(source)
86
87     try:
88         # Utiliser l'API Google pour la transcription de la parole
89         message = recognizer.recognize_google(audio, language="fr-FR").lower()
90         print("Vous: " + message)
91     except sr.UnknownValueError:
92         print("Désolé, je n'ai pas compris la parole.")
93         return "Désolé, je n'ai pas compris la parole. Pouvez-vous reformuler votre question?"
94     except sr.RequestError as e:
95         print(f"Erreur lors de la requête à l'API Google : {e}")
96         return "Erreur lors de la reconnaissance vocale. Veuillez réessayer."
97
98     # Traitement du langage naturel avec spaCy
99     doc = nlp(message)
100
101    # Liste des mots-clés trouvés
102    mots_cles_trouves = [cle for cle, syn in synonymes.items() if any(token.text in syn for token in doc)]
103
104    for cle, syn in synonymes.items():
105        if cle in doc.text:
106            return random.choice(responses[cle])
107
chatbot_mairie() > while True
```

Version Control Python Packages TODO Python Console Problems Terminal Services

136:16 LF UTF-8 4 spaces Python 3.9 (pythonProject3)

ANNEXE 1 - CODE VERSION 1

Rasa Project – main.py

MissionRD – main.py

```
pythonProject3 > main.py
```

```
Project main.py
```

```
103     for cle, syn in synonymes.items():
104         if cle in doc.text:
105             return random.choice(reponses[cle])
106
107     # Gérer les cas spécifiques
108     if "au revoir" in message:
109         return random.choice(reponses["au revoir"])
110
111     # Gérer les cas multiples
112     if len(mots_cles_trouves) > 1:
113         return traiter_cas_multiples(mots_cles_trouves)
114
115     # Gérer les cas uniques
116     if len(mots_cles_trouves) == 1:
117         cle_trouvee = mots_cles_trouves[0]
118         return random.choice(reponses[cle_trouvee])
119
120     return "Désolé, je ne comprends pas. Pouvez-vous reformuler votre question?"
121
122     # Fonction pour la synthèse vocale
123 def parler(texte):
124     # Utiliser gTTS pour convertir le texte en fichier audio
125     tts = gTTS(text=texte, lang='fr')
126     tts.save("reponse.mp3")
127
128     # Jouer le fichier audio
129     os.system("afplay reponse.mp3") # Pour macOS
130
131     # Fonction principale du chatbot
132 chatbot_mairie() > while True
```

Database

SciView

Notifications

Bookmarks

Structure

Version Control Python Packages TODO Python Console Problems Terminal Services

136:16 LF UTF-8 4 spaces Python 3.9 (pythonProject3)

ANNEXE 1 - CODE VERSION 1

The screenshot shows the PyCharm IDE interface with two code editors side-by-side.

Rasa Project – main.py

```
pythonProject3 main.py
124 def parler(texte):
125     # Utiliser gTTS pour convertir le texte en fichier audio
126     tts = gTTS(text=texte, lang='fr')
127     tts.save("reponse.mp3")
128
129     # Jouer le fichier audio
130     os.system("afplay reponse.mp3") # Pour macOS
131
132     # Fonction principale du chatbot
133     def chatbot_mairie():
134         print("Bienvenue! Posez-moi des questions sur la mairie.")
135
136     while True:
137         # Obtenir la réponse du chatbot
138         response = repondre("")
139
140         # Synthèse vocale de la réponse du chatbot
141         parler(response)
142
143         # Afficher la réponse du chatbot
144         print("Chatbot:", response)
145
146     if __name__ == "__main__":
147         chatbot_mairie()
148
chatbot_mairie() > while True
```

MissionRD – main.py

```
main.py
1 2 59 903
Database
SciView
Notifications
```

The bottom status bar indicates the file is 136:16 LF UTF-8 4 spaces Python 3.9 (pythonProject3).

annexe 2 : résultats

```
>>> (executing cell "" (line 1 of "score2.py"))
```

```
Précision globale: 0.7290076335877863
```

```
Précision par niveau de difficulté 0: 0.8
```

```
Précision par niveau de difficulté 1: 0.7727272727272727
```

```
Précision par niveau de difficulté 1.5: 0.8518518518518519
```

```
Précision par niveau de difficulté 2: 0.6808510638297872
```

```
Précision par niveau de difficulté 2.5: 0.6438356164383562
```

```
Précision motif 'relance amiable : référence courrier reçu RC': 1.0, 2 apparitions
```

```
Précision motif 'Déclaration de décès': 1.0, 16 apparitions
```

```
Précision motif 'mise en place contrat prélèvement': 1.0, 14 apparitions
```

```
Précision motif 'Déclaration de naissance, Reconnaissance': 0.9130434782608695, 23 apparitions
```

```
Précision motif 'Changement de prénoms, rectification d'actes': 0.8947368421052632, 19 apparitions
```

```
Précision motif 'Inscription sur liste électorale': 0.8823529411764706, 17 apparitions
```

```
Précision motif 'explication avis échéance loyer': 0.875, 8 apparitions
```

```
Précision motif 'Livret de famille': 0.8571428571428571, 7 apparitions
```

```
Précision motif 'Première Inscription scolaire-changement d'école': 0.8, 5 apparitions
```

```
Précision motif 'Commande ou décommande de repas': 0.7692307692307693, 13 apparitions
```

```
Précision motif 'Inscription Périscolaire (Cantine et Accueil)': 0.7692307692307693, 13 apparitions
```

```
Précision motif 'Mariage': 0.75, 8 apparitions
```

```
Précision motif 'Actes de naissance, mariage, décès': 0.6923076923076923, 26 apparitions
```

```
Précision motif 'Règlement cantine en espèces': 0.6666666666666666, 3 apparitions
```

```
Précision motif 'contentieux locataire présent : référence courrier reçu FC': 0.6666666666666666, 6 apparitions
```

```
Précision motif 'Certificats, légalisation de signature': 0.6190476190476191, 21 apparitions
```

```
Précision motif 'Recensement des jeunes': 0.6, 5 apparitions
```

```
Précision motif 'décompte de sortie (locataire quittant OPHEOR)': 0.5384615384615384, 13 apparitions
```

```
Précision motif 'Renseignements, modification de dossier': 0.5, 2 apparitions
```

```
Précision motif 'PACS (Dépôt de dossier, modification ou dissolution )': 0.5, 6 apparitions
```

```
Précision motif 'Enregistrement de PACS': 0.4, 5 apparitions
```

```
Précision motif 'demandes d'attestations diverses': 0.38461538461538464, 13 apparitions
```

```
Précision motif 'explication régularisation des charges': 0.36363636363636365, 11 apparitions
```

```
Précision motif 'contentieux locataire parti : référence courrier reçu FC': 0.1666666666666666, 6 apparitions
```

ANNEXE 3 - CODE VERSION 2

The screenshot shows a Python code editor interface with a dark theme. The project is named "pythonProject6" and contains a file "main.py". The code implements a simple web interface using Flask to interact with a Rasa chatbot API.

```
pythonProject6 › main.py
Project  main.py index.html
1  from flask import Flask, request, jsonify, render_template, send_from_directory
2  import requests
3  import os
4
5  app = Flask(__name__)
6
7  # URL de l'API du chatbot Rasa
8  rasa_api_url = "http://localhost:5005/webhooks/rest/webhook" # Mettez à jour l'URL en fonction de votre configuration
9
10 # Page d'accueil de l'interface web
11 @app.route('/')
12 def home():
13     return render_template('index.html')
14
15 # Servir l'image statique
16 @app.route('/static/<path:path>')
17 def serve_static(path):
18     root_dir = os.path.dirname(os.getcwd())
19     return send_from_directory(os.path.join(root_dir, 'static'), path)
20
21 # Endpoint pour interagir avec le chatbot
22 @app.route('/get_response', methods=['POST'])
23 def get_bot_response():
24     user_message = request.json['user_message']
25     response = requests.post(rasa_api_url, json={"message": user_message}).json()
26     bot_response = response[0]['text'] if response else "Je suis désolé, je n'ai pas compris."
27     return jsonify({'bot_response': bot_response})
28
29
30 if __name__ == '__main__':
31     app.run(debug=True)
```

The code includes imports for Flask, requests, and os. It defines a Flask application and sets its URL to the Rasa API endpoint. The home route returns an index.html template. The serve_static route handles static files from the 'static' directory. The get_bot_response route sends a POST request to the Rasa API with the user message and returns the bot's response. The main block runs the application with debug mode enabled.

ANNEXE 3 - CODE VERSION 2

```
version: "2.0"
nlu:
- intent: salutation
  examples: |
    - salut
    - bonjour
    - hello
    - coucou
    - Salut! Comment puis-je vous aider aujourd'hui?
    - Bonjour, avez-vous besoin d'assistance?
    - Salutations, que puis-je faire pour vous?
    - Hey, comment ça va?
- intent: horaires_mairie
  examples: |
    - Quels sont les horaires d'ouverture?
    - Quand est-ce que la mairie ouvre?
    - A quelle heure la mairie ferme-t-elle?
    - Quand puis-je vous rendre visite?
    - La mairie est-elle ouverte le samedi?
    - Quels sont vos horaires de travail?
    - Je veux connaître vos heures d'ouverture.
- intent: services_mairie
  examples: |
    - Quels services proposez-vous?
    - Pouvez-vous me dire quels services sont disponibles à la mairie?
    - J'ai besoin de connaître les services municipaux.
- intent: au_revoir
  examples: |
    - Au revoir
    - À bientôt
    - Adieu
    - Bye
    - Merci pour votre aide, au revoir
    - C'était un plaisir, à la prochaine
    - Je vais partir maintenant, salut
- intent: documents_administratifs
  examples: |
    - Comment puis-je obtenir une carte d'identité?
    - Quelle est la procédure pour obtenir un passeport?
    - J'aimerais savoir comment obtenir un certificat de mariage.
    - Pouvez-vous m'aider avec les démarches administratives?
    - Je cherche des informations sur les documents officiels.
    - Où puis-je obtenir un certificat de naissance?
- intent: mariage
  examples: |
    - Comment puis-je organiser mon mariage à la mairie?
    - Quelles sont les étapes pour un mariage civil?
    - Je veux me marier, que dois-je faire?
    - Y a-t-il des exigences spéciales pour un mariage civil?
    - Puis-je réserver une date pour mon mariage à la mairie?
    - J'ai besoin d'informations sur les mariages à la mairie.
- intent: inscription_ecole
  examples: |
    - Comment puis-je inscrire mon enfant à l'école?
    - Quels sont les documents nécessaires pour l'inscription scolaire?
    - Je veux inscrire mon enfant à l'école, pouvez-vous m'aider?
    - Quelle est la date limite pour l'inscription à l'école?
    - Y a-t-il des conditions d'admission spécifiques?
    - Je cherche des informations sur l'inscription à l'école.
- intent: prise_rendez_vous
  examples: |
    - Comment puis-je prendre rendez-vous à la mairie?
```

ANNEXE 3 - CODE VERSION 2

```
version: "2.0"

rules:
- rule: Répondre aux questions sur les documents administratifs
  steps:
    - intent: documents_administratifs
      action: utter_documents_administratifs

- rule: Répondre aux questions sur le renouvellement de documents
  steps:
    - intent: renouvellement_document
      action: utter_renouvellement_document

- rule: Répondre aux questions sur l'inscription aux listes électorales
  steps:
    - intent: inscription_listes_electorales
      action: utter_inscription_listes_electorales

- rule: Répondre aux demandes de prise de rendez-vous
  steps:
    - intent: prise_rendez_vous
      action: utter_prise_rendez_vous

- rule: Répondre aux questions sur l'inscription à l'école
  steps:
    - intent: inscription_ecole
      action: utter_inscription_ecole

- rule: Répondre aux questions sur le mariage à la mairie
  steps:
    - intent: mariage
      action: utter_mariage
```

ANNEXE 3 - CODE VERSION 2

```
intents:
  - salutation
  - horaires_mairie
  - services_mairie
  - au_revoir
  - documents_administratifs
  - mariage
  - inscription_ecole
  - prise_rendez_vous
  - demande_aide_sociale
  - renouvellement_document
  - inscription_listes_electorales
  - demande_informations

responses:
  utter_salutation:
    - text: "Bonjour! Comment puis-je vous aider aujourd'hui? N'hésitez pas à me poser vos questions."
  utter_horaires_mairie:
    - text: "La mairie est ouverte du lundi au vendredi de 9h à 17h. Elle est fermée les weekends et les jours fériés."
  utter_services_mairie:
    - text: "Voici les services disponibles à la mairie : [Liste des services]"
  utter_au_revoir:
    - text: "Au revoir! Si vous avez d'autres questions à l'avenir, n'hésitez pas à revenir. Bonne journée."
  utter_documents_administratifs:
    - text: "Voici les documents administratifs disponibles : [Liste des documents]"
  utter_mariage:
    - text: "Voici les étapes pour un mariage civil : [Étapes du mariage]"
  utter_inscription_ecole:
    - text: "Voici la procédure pour inscrire votre enfant à l'école : [Procédure d'inscription]"
  utter_prise_rendez_vous:
    - text: "Vous pouvez prendre rendez-vous à la mairie en ligne ou en appelant notre bureau pendant les heures d'ouverture."
  utter_demande_aide_sociale:
    - text: "Voici comment demander une aide sociale : [Démarches pour obtenir une aide sociale]"
  utter_renouvellement_document:
    - text: "Voici la procédure pour renouveler vos documents administratifs : [Procédure de renouvellement]"
  utter_inscription_listes_electorales:
    - text: "Voici comment vous inscrire sur les listes électorales : [Procédure d'inscription]"
  utter_demande_informations:
    - text: "Si vous avez besoin d'informations supplémentaires, n'hésitez pas à poser votre question."
```

ANNEXE 3 - CODE VERSION 2

```
version: "2.0"

stories:
- story: Saluer
  steps:
    - intent: salutation
    - action: utter_salutation

- story: Demander les horaires de la mairie
  steps:
    - intent: horaires_mairie
    - action: utter_horaires_mairie

- story: Demander les services de la mairie
  steps:
    - intent: services_mairie
    - action: utter_services_mairie

- story: dire au revoir
  steps:
    - intent: au_revoir
    - action: utter_au_revoir
```