

Cahier des Charges : Système de Classification des Dialectes Darija avec Speech-to-Text, Traduction et IoT

Introduction

Le darija, un dialecte arabe parlé au Maroc, se distingue par une grande diversité régionale, avec des variantes telles que le Casawia, le Marakchia, et le Tetouania... entre autres. Cette richesse linguistique, bien qu'elle soit un atout culturel, pose un défi technologique majeur en matière de traitement du langage naturel (NLP). Reconnaître et classer ces dialectes, puis les convertir en texte et les traduire en anglais, sont des tâches complexes que ce projet ambitionne de surmonter. En intégrant des technologies avancées comme le Speech-to-Text, la traduction automatique et l'Internet des objets (IoT), ce projet vise à développer un système innovant capable de répondre à ces défis.

Les objectifs du projet sont multiples et ambitieux. Premièrement, il s'agit de classifier automatiquement le dialecte darija utilisé dans un enregistrement audio parmi les différentes variantes régionales. Deuxièmement, le projet vise à transformer la parole en texte écrit en darija, en garantissant une transcription précise et fidèle à l'audio d'origine. Troisièmement, le texte en darija sera traduit en anglais, facilitant ainsi la communication et l'accès à l'information pour un public plus large. Enfin, le projet intègre une composante IoT, permettant d'utiliser les résultats de la reconnaissance vocale pour déclencher des actions spécifiques dans un environnement connecté, comme l'ouverture ou la fermeture d'une porte en réponse à une commande vocale reconnue.

Ce projet s'adresse à divers utilisateurs et applications, notamment les outils de traduction, les dictionnaires en ligne, les systèmes d'assistance vocale, et les solutions de sécurité basées sur la reconnaissance vocale. En développant ce système, nous espérons non seulement faire progresser le domaine du NLP pour les langues sous-représentées, mais aussi offrir des solutions pratiques et innovantes pour diverses applications technologiques.

Le projet est conçu pour être utilisé dans le cadre de la Coupe du Monde 2030 au Maroc. L'objectif est de faciliter la communication entre les visiteurs étrangers et les locaux, en prenant en compte la diversité des dialectes marocains.

Objectifs spécifiques :

- **Améliorer l'expérience des visiteurs** : Traduction et affichage en temps réel dans les stades, les gares, les aéroports et les sites touristiques.
- **Valoriser la richesse linguistique marocaine** : Mettre en avant les différentes variantes de la darija et leur traduction.
- **Faciliter l'accès à l'information** : Intégration dans des bornes interactives et applications mobiles pour guider les spectateurs et touristes.
- **Interaction avec les infrastructures connectées** : Utilisation dans les enceintes sportives et les lieux publics via des écrans, haut-parleurs et assistants vocaux.

Problématique et Enjeux

Le projet vise à développer un système de reconnaissance et de classification des dialectes darija, avec conversion de la parole en texte, traduction en anglais et intégration IoT pour déclencher des actions basées sur des commandes vocales. Toutefois, plusieurs défis doivent être relevés :

1. La diversité des dialectes darija et la complexité de leur classification

Le darija, parlé au Maroc, n'est pas une langue homogène mais un ensemble de variantes régionales (Casawia, Marakchia, Tetouania, etc.), influencées par l'histoire, la géographie et les interactions culturelles. Cette diversité linguistique complique la classification automatique des dialectes. Contrairement aux langues standardisées comme l'arabe classique ou l'anglais, il n'existe pas de règles fixes en darija, ce qui rend difficile l'entraînement des modèles de Machine Learning. Les différences phonétiques, lexicales et syntaxiques entre dialectes peuvent entraîner des erreurs de reconnaissance et de classification.

2. Le manque de ressources et de corpus pour l'apprentissage des modèles NLP en darija

L'un des principaux défis en Traitement Automatique du Langage (NLP) pour le darija est le manque de corpus de données annotées. Contrairement à l'anglais ou au français, où d'importantes bases de données existent pour l'apprentissage des modèles de reconnaissance vocale et de traduction, le darija est une langue peu documentée. Cela pose plusieurs problèmes :

- Difficile d'entraîner des modèles performants sans jeux de données représentatifs.
- Nécessité de collecter et d'annoter manuellement un grand volume de données audio et textuelles.
- Problèmes de normalisation : le darija s'écrit parfois en caractères arabes, parfois en alphabet latin (avec des chiffres comme "3" pour "ع"), compliquant la standardisation des textes.

3. Les défis techniques liés à l'intégration de l'IoT dans un système de reconnaissance vocale

L'intégration de l'Internet des Objets (IoT) dans un système basé sur la reconnaissance vocale ajoute une couche de complexité technique. Pour que le système fonctionne efficacement, plusieurs éléments doivent être pris en compte :

- **Fiabilité de la reconnaissance vocale** : Le modèle doit être capable de bien interpréter les commandes, même avec du bruit ambiant ou des variations dans la prononciation.
- **Temps de réponse rapide** : Une latence trop élevée entre la commande vocale et l'action sur l'objet connecté (ex: ouverture de porte) pourrait rendre l'expérience utilisateur inefficace.
- **Sécurité et robustesse** : Le système IoT doit être sécurisé contre les accès non autorisés, pour éviter qu'une personne extérieure ne puisse contrôler un dispositif en imitant une voix.
- **Compatibilité matérielle** : Le modèle NLP doit pouvoir communiquer avec des microcontrôleurs (ex: Arduino, Raspberry Pi) et des protocoles de communication comme MQTT, Bluetooth ou Wi-Fi.

Description du Projet

Fonctionnalités principales

Classification des dialectes (Tetouania, Ouazania, Casawia, Oujdia) :

Le système sera capable d'identifier et de classer les différents accents et dialectes de la Darija marocaine, en déterminant la région d'origine de l'utilisateur en fonction de son accent.

Speech-to-Text (conversion audio → texte en darija) :

Le système convertira la parole en texte en temps réel, en utilisant des modèles de traitement du langage naturel (NLP) spécialement entraînés pour la Darija marocaine.

Traduction automatique (darija → anglais) :

Une fois la parole convertie en texte, le système traduira instantanément le texte en anglais, permettant une communication fluide entre les locuteurs de Darija et les anglophones.

Intégration IoT :

Le système sera intégré avec des dispositifs IoT pour permettre des actions physiques basées sur des commandes vocales en Darija. Par exemple :

- Détection d'un mot-clé spécifique : Le système pourra déclencher des actions comme l'ouverture ou la fermeture d'une porte, l'allumage ou l'extinction des lumières, ou l'activation d'une alarme, en fonction des commandes vocales reconnues.

- Contrôle d'un système physique : Via des microcontrôleurs connectés (ex: Arduino, Raspberry Pi), le système pourra interagir avec des dispositifs IoT pour exécuter des actions spécifiques.

Structure du Dataset

Audios en darija :

Le dataset sera composé d'enregistrements audio de locuteurs de Darija, couvrant les différents dialectes et accents régionaux.

Dialecte associé :

Chaque enregistrement audio sera annoté avec le dialecte correspondant (Tetouania, Ouazania, Casawia, Oujdia).

Transcription en darija : Les enregistrements audio seront transcrits en texte en Darija pour entraîner les modèles de reconnaissance vocale.

Traduction en anglais :

Les transcriptions en Darija seront également traduites en anglais pour entraîner les modèles de traduction automatique.

Technologies envisagées

Machine Learning & Deep Learning :

Des modèles de réseaux de neurones convolutifs (CNN), de réseaux de neurones récurrents (RNN) et de Transformers seront utilisés pour la classification des dialectes, la reconnaissance vocale et la traduction.

Frameworks NLP :

TensorFlow, PyTorch, et Hugging Face seront utilisés pour le développement et l'entraînement des modèles de traitement du langage naturel.

Speech-to-Text :

Des technologies comme Google SpeechRecognition et Wav2Vec2 seront explorées pour la conversion de la parole en texte.

IoT :

Des microcontrôleurs comme Arduino et Raspberry Pi seront utilisés pour l'intégration avec des dispositifs IoT, permettant des actions physiques basées sur des commandes vocales.

Cahier des Charges Fonctionnel

Entrées du Système

Les entrées du système sont cruciales pour le bon fonctionnement de notre projet de classification des dialectes Darija, de transcription vocale, de traduction et d'intégration IoT. Elles permettent l'interaction avec les utilisateurs et l'analyse des données.

Fichiers Audio

Les fichiers audios constituent une composante essentielle de notre système, permettant l'interaction vocale et l'analyse des commandes.

Formats Supportés :

- .WAV : Format non compressé offrant une qualité audio optimale, idéal pour l'entraînement des modèles et l'analyse précise.
- .MP3 : Format compressé largement utilisé, adapté pour le stockage et la transmission efficace des données.

Commandes Vocales pour l'IoT

Les commandes vocales jouent un rôle crucial dans l'interaction avec les dispositifs IoT, permettant une interface utilisateur intuitive et accessible.

Cas d'Utilisation :

- Contrôle des appareils domestiques : Commandes vocales pour allumer/éteindre les lumières, régler la température, etc.
- Automatisation : Intégration avec des systèmes d'automatisation pour des tâches répétitives (ex: "Active le mode nuit").

- Accessibilité : Amélioration de l'accessibilité pour les utilisateurs ayant des besoins spécifiques.

Données Textuelles

Les données textuelles sont essentielles pour les interfaces où l'utilisateur peut saisir du texte, permettant une interaction plus précise et détaillée.

Cas d'Utilisation :

- Interfaces Utilisateur : Saisie de texte pour des requêtes spécifiques ou des configurations.
- Analyse de Sentiment : Utilisation des données textuelles pour analyser les réactions des utilisateurs.
- Support Multilingue : Prise en charge de plusieurs langues pour une utilisation mondiale.

Sorties Attendues

L'architecture de notre système d'intelligence artificielle et d'intégration IoT est conçue pour générer des sorties précises et exploitables, alignées avec les objectifs définis dans le cahier des charges.

Classification du Dialecte Darija

Notre système offrira une classification automatique du dialecte Darija à partir d'un enregistrement vocal, permettant d'identifier la région d'origine du locuteur.

Dialectes Pris en Charge :

- Casawia (région de Casablanca)
- Marakchia (région de Marrakech)
- Tetouania (nord, région de Tétouan)
- Oujdia (région de l'Oriental)

Technologie Utilisée :

- Modèle basé sur CNN-RNN ou Transformers, entraîné sur un dataset annoté contenant des milliers d'enregistrements audio de locuteurs natifs.
- Approche de feature extraction avancée (MFCC, spectrogrammes) combinée à des embeddings neuronaux pour capturer les variations dialectales.

Utilité :

- Affiner la transcription en adaptant le modèle de reconnaissance vocale au dialecte détecté.
- Personnaliser l'expérience utilisateur, notamment pour une meilleure traduction.
- Fournir des insights linguistiques pour des études sociolinguistiques ou applications éducatives.

2

Transcription du Discours en Texte Darija (Speech-to-Text)

Le système convertira la parole en texte en Darija avec une haute précision.

Format de Sortie :

- Texte brut en Darija avec ponctuation.
- Version standardisée (sans translittération).

Technologie Utilisée:

- Modèles wav2vec2, Whisper ou Seamless M4T, fine-tunés sur un corpus en Darija.
- Algorithmes de post-traitement NLP pour la correction automatique et la segmentation des phrases.

Traduction Automatique en Anglais

Une fois le discours transcrit, le système effectuera une traduction automatique du texte Darija vers l'anglais en temps réel.

Technologie Utilisée:

- Modèles de traduction neuronale (NMT) basés sur Transformers.
- Approche fine-tunée sur un dataset bilingue Darija-Anglais.

Améliorations Intégrées:

- Adaptation du modèle en fonction du dialecte détecté.
- Correction grammaticale et reformulation pour une traduction fluide et contextuellement pertinente.

Cas d'Usage:

- Permettre aux non-locuteurs de Darija de comprendre et interagir avec des Marocains.
- Faciliter l'apprentissage de l'anglais pour les Marocains.
- Application dans des assistants vocaux et des solutions de communication en entreprise.

4Déclenchement d'Actions IoT en Fonction des Commandes Vocales

Le système permettra une interaction directe avec des objets connectés via des commandes vocales en Darija.

Exemples de Commandes Prises en Charge :

- "Ouvre la porte" → Activation d'un servo-moteur pour ouvrir une porte.
- "Allume la lumière du salon" → Envoi d'une commande MQTT à un relais connecté.
- "Active l'alarme" → Déclenchement d'un signal sonore.

Architecture d'Intégration:

- Communication entre IA et IoT via MQTT, HTTP API, WebSockets.
- Microcontrôleurs Raspberry Pi / Arduino reliés à des capteurs/actionneurs.
- NLP pour Intent Recognition (classification des commandes vocales avec Transformers).

Sécurité et Contraintes :

- Validation des commandes pour éviter les déclenchements accidentels.
- Temps de réponse inférieur à 2 secondes pour garantir une expérience utilisateur fluide.

Contraintes Techniques

La conception et le déploiement du système nécessitent le respect de plusieurs contraintes techniques garantissant performance, sécurité, compatibilité et scalabilité.

Performance

- Précision de classification des dialectes > 85%
 - Fine-tuning du modèle CNN-RNN / Transformer avec des embeddings spécialisés pour améliorer la robustesse face aux variations régionales.
 - Évaluation régulière de la précision sur des datasets de test représentatifs.
- Temps de réponse < 2 secondes (traitement audio + traduction)
 - Optimisation du pipeline STT → NMT → IoT pour minimiser la latence.
 - Accélération des calculs via GPU / TPU pour les modèles de Deep

Learning.

- Compression des modèles via quantization / pruning pour exécuter efficacement les tâches sur des appareils embarqués.

Sécurité et Fiabilité

- Sécurisation des échanges entre le système ML et les appareils

IoT

- Communication sécurisée via TLS/SSL pour protéger les requêtes entre le cloud et les dispositifs IoT.
- Authentification des commandes vocales critiques pour éviter les abus(ex. contrôle d'une serrure intelligente).
- Séparation des niveaux d'accès pour garantir que seules certaines commandes puissent être exécutées.
- Gestion des erreurs et des fausses commandes
 - Filtrage et vérification des commandes IoT pour éviter les activations involontaires.
 - Détection et correction des erreurs de transcription via un post traitement NLP.
 - Mécanisme de confirmation utilisateur avant l'exécution d'actions sensibles.

Compatibilité et Déploiement

- Support multi-appareils
 - Interface web responsive et application mobile pour assurer une compatibilité sur PC, smartphones et tablettes.
 - Optimisation du moteur STT/NMT pour fonctionner avec des microcontrôleurs et appareils IoT.

- Intégration avec API cloud et edge computing
 - Déploiement flexible sur cloud (AWS, Google Cloud, Azure) pour un traitement haute performance ou edge computing (Raspberry Pi, Jetson Nano) pour une exécution locale et une latence réduite.
 - Interconnexion via API REST / MQTT / WebSockets pour faciliter la communication entre modules ML et dispositifs IoT.

Synthèse des Contraintes Techniques

Critère	Exigence	Solution proposée
Précision	Classification dialectale > 85%	Fine-tuning sur corpus annoté, embeddings spécialisés
Latence	Temps de réponse < 2s	Accélération GPU/TPU, compression des modèles
Sécurité	Protection des échanges ML-IoT	Chiffrement TLS/SSL, authentification des commandes
Fiabilité	Gestion des erreurs et confirmations	Post-traitement NLP, validation des actions critiques
Compatibilité	Support multi-appareils	Déploiement sur cloud et edge computing
Interopérabilité	Communication IoT	Intégration API REST, MQTT, WebSockets

Scénarios d'Utilisation

Pour mieux illustrer les fonctionnalités du système, voici quatre scénarios d'interaction principaux:

Scénario 1: Interaction Personne-à-Personne

- Description: Un utilisateur envoie un message vocal à une autre personne via un appareil IoT.
- Exemple: "Envoie un message à Ali pour lui dire que je suis en retard."
- Étapes:
 1. L'utilisateur parle à l'appareil.
 2. L'appareil transmet l'audio au serveur cloud.
 3. Le serveur convertit l'audio en texte et analyse l'intention.

4. Le serveur prépare et envoie le message à l'appareil de l'autre personne.
5. L'appareil destinataire lit le message à haute voix.

Scénario 2: Contrôle d'Appareils IoT

- Description: Un utilisateur contrôle un appareil IoT à distance via des commandes vocales.
- Exemple: "Allume la lumière du salon."
- Étapes:
 1. L'utilisateur donne une commande vocale.
 2. L'appareil transmet la commande au serveur cloud.
 3. Le serveur analyse la commande et envoie une instruction à l'appareil IoT concerné.
 4. L'appareil IoT exécute la commande.
 5. Une confirmation est renvoyée à l'utilisateur.

Scénario 3: Requête d'Information

- Description: Un utilisateur demande des informations via un appareil IoT.
- Exemple: "Quelle est la météo pour demain?"
- Étapes:
 1. L'utilisateur pose une question vocale.
 2. L'appareil transmet la question au serveur cloud.
 3. Le serveur recherche l'information demandée.
 4. Le serveur envoie la réponse à l'appareil de l'utilisateur.
 5. L'appareil lit la réponse à haute voix.

Scénario 4: Automatisation Basée sur des Routines

- Description: Un système exécute des routines prédéfinies basées sur des commandes vocales.
- Exemple: "Active le mode nuit."
- Étapes:
 1. L'utilisateur active une routine vocale.
 2. La commande est transmise au serveur cloud.
 3. Le serveur déclenche une série d'actions prédéfinies.
 4. Les appareils exécutent les actions de la routine.
 5. Une confirmation est envoyée à l'utilisateur.

Cahier des Charges Technique

Architecture du Système

Pipeline de Traitement

1. **Acquisition Audio** : Enregistrement vocal via micro.
2. **Speech-to-Text (STT)** : Transcription des phrases en darija en texte.
3. **Classification des Dialectes** : Identification du type de darija (Casawia, Marakchia, Tetouania...).
4. **Traduction** : Conversion du texte en darija vers l'anglais.
5. **Intégration IoT** : Envoi des résultats vers un système connecté (ex: affichage sur un écran, synthèse vocale, commandes vocales pour domotique...).

Communication entre le Système ML et l'IoT

- **Protocole de Communication** : MQTT, Bluetooth, ou Wi-Fi.
- **Serveur de traitement** : Un serveur local (Raspberry Pi, PC avec GPU) ou un cloud pour exécuter les modèles.
- **Interactions** : L'IoT reçoit des commandes/textes et les affiche ou les utilise pour une action spécifique (ex: commande vocale pour un appareil).

Besoins Matériels et Logiciels

Matériel

- **GPU** : Pour l'entraînement des modèles deep learning (NVIDIA RTX, Google Colab, serveur cloud...).
- **Micro et Haut-parleurs** : Pour capturer et restituer l'audio.
- **Arduino/Raspberry Pi** : Pour l'intégration IoT et la communication avec le système ML.
- **Capteurs et Afficheurs** : Écran LCD/OLED, LED pour feedback, modules Bluetooth/Wi-Fi pour communication.

Logiciel & Outils

- **Modèle Speech-to-Text** : Whisper (OpenAI), DeepSpeech, Google Speech API.
- **Modèle de Classification** : CNN, RNN ou Transformers (ex: BERT, Wav2Vec).
- **Traduction** : NLLB (No Language Left Behind), MarianMT, Google Translate API.
- **Frameworks** : TensorFlow, PyTorch.
- **Communication IoT** : MQTT (Mosquitto), Bluetooth (BlueZ), Wi-Fi (ESP8266, ESP32).

Collecte et Préparation des Données

- **Création du Dataset :**
 - Enregistrements audio de phrases en différents dialectes de darija.
 - Annotation manuelle : transcription en darija, classification du dialecte, traduction en anglais.
- **Prétraitement :**
 - Filtrage du bruit et normalisation audio.
 - Alignement texte-audio.

Développement et Intégration

Développement des Modèles

- Entraînement du modèle Speech-to-Text sur des données de darija.
- Fine-tuning du modèle de classification dialectale.
- Développement du modèle de traduction en anglais.

Déploiement et Optimisation

- Déploiement sur un serveur local ou cloud.
- Intégration avec Raspberry Pi ou Arduino pour interaction IoT.
- Tests de latence et optimisation pour usage en temps réel.

Sécurité et Fiabilité

- Protection des données utilisateurs (anonymisation).
- Sécurisation des communications MQTT/Wi-Fi (chiffrement SSL/TLS).
- Gestion des erreurs et des cas extrêmes (ex: bruit élevé, accents).

Cas d'Usage IoT

- **Affichage dynamique :** Traduction instantanée sur un écran.
- **Commande vocale :** Contrôle d'appareils via reconnaissance des dialectes.
- **Application éducative :** Apprentissage des dialectes et traduction en temps réel.

Planification et Gestion du Projet

Mois 1

- **Semaine 1 à 2**
On commence par l'**étude de faisabilité**. On va analyser si le projet est réalisable, et définir les technologies qu'on va utiliser, le temps nécessaire, et le budget. On va aussi chercher des solutions similaires qui existent déjà.
- **Semaine 3 à 4**
Ensuite, on passe à la **rédaction du cahier des charges**. On va y définir en détail ce que le projet doit faire, les technologies à utiliser, les coûts, et les délais. Ce document va nous servir de guide pendant tout le projet.

Mois 2

- **Semaine 1 à 2**
Pendant cette période, on va créer un **prototype**. On va tester un modèle sur un dataset existant pour voir si les modèles choisis fonctionnent bien et aussi tester comment intégrer Arduino et Raspberry Pi.
- **Semaine 3 à 4**
Ensuite, on passe à la **conception et l'architecture du système**. Ici, on va définir comment la plateforme va être structurée. On choisira les serveurs à utiliser, comment les données vont circuler, et comment intégrer IoT.

Mois 3

- **Semaine 1 à 2**
Cette période sera consacrée à la **collecte et préparation des données**. On va chercher des enregistrements audio de personnes parlant Darija avec différents accents, et on les annotera pour entraîner les modèles de transcription et de traduction.
- **Semaine 3 à 4**
Ensuite, on commence le **développement des fonctionnalités principales**. On va développer les principales fonctionnalités de la plateforme comme l'interface utilisateur, et les modèles pour la classification des accents et la transcription.

Mois 4

- **Semaine 1**
Avant de mettre le projet en ligne, on va faire des **tests et validations**. On va tester la précision de la classification des accents, la transcription vocale et la traduction. Des tests utilisateurs seront aussi faits pour s'assurer que tout fonctionne bien.
- **Semaine 2 à 3**
Quand tout sera validé, on procédera au **déploiement et à la mise en production**. On déploiera la plateforme en ligne sur des serveurs qui pourront gérer la charge. On vérifiera aussi que l'intégration avec Alexa fonctionne bien.

- **Semaine 4**

Une fois le projet lancé, on va préparer des **guides et des tutoriels** pour aider les utilisateurs à comprendre comment utiliser la plateforme. On mettra aussi en place un support technique pour répondre aux questions.

Mois 4 et 15 jours

- **Semaine 1 à 2 (deux premières semaines)**

Après le lancement, on va faire du **suivi et de l'amélioration continue**. On recueillera les retours des utilisateurs et si nécessaire, on ajustera les modèles, on réentraînera l'IA, ou on améliorera des fonctionnalités.

- **Semaine 3 à 4**

Enfin, on lancera une **campagne de communication et de marketing**. Et intégration de IoT.

Bibliographie et Ressources

- Articles sur le speech-to-text en arabe
- Recherches sur la classification des dialectes
- Documentation sur l'intégration IoT et NLP