

# HackaTAL 2024

**Commande vocale de robots**

**Université Toulouse Jean-Jaurès, les 8 et 9 juillet 2024**

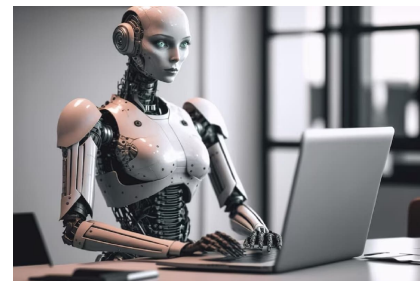
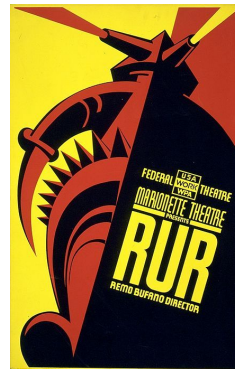


# Éditions du HackaTAL

- 2016 Paris (Google)
  - Détection d'évènements
  - Gestion de dialogues
- 2017 Orléans (LAB'O)
  - Résumé automatique de commentaires sur des produits en ligne
  - Identification des tendances stratégiques liées aux brevets
- 2018 Rennes (IRISA)
  - Repérer et catégoriser les fausses informations
  - Agrégation des fausses nouvelles pour vision synthétique
- 2019 Toulouse (Université Toulouse 1)
  - Analyses du grand débat national
  - Chatbots juridiques
- 2020-2023 nulle part (aka blackout raoultien)
- 2023 Paris (SCAI)
  - Wikipédia & Politique
  - Détection de Citations
- 2024 Toulouse (Université Toulouse Jean-Jaurès)
  - Commande vocale de robots

# Commande vocale de robot

- Robot
  - Mot tchèque dérivé (1920) de *robota* (« travail, besogne, corvée »)
  - Réalisation de tâches (industrie, cuisine, aspirateurs, tondeuses, drones, voitures)
  - Humanoïdes dans l'imaginaire, pas tellement en pratique (ou jeu)
  - Capacités cognitives (compréhension, planification) et physiques (senso-motrices)
- Commander un robot
  - Interfaces explicites (boutons) ou « commandes » plus ou moins naturelles (voix)
    - Capacités de compréhension, d'interprétation, de dialogue (bot génératif)
    - Converger vers un langage pour formuler des instructions
  - Objectif distant à atteindre par planification d'actions (étapes)
  - Dépendance forte aux contraintes physiques du monde
- Quelles capacités (machines ou humains)
  - Charge cognitive des tâches à réaliser
  - Dissocier le contrôle de la supervision (voitures) et de la conscience
  - Confiance et dystopie de la perte du contrôle



# Organisation

## Organisateurs

- Audran Bert (LINAGORA)
- Florian Boyer (Alrudit)
- Chloé Braud (IRIT)
- Angélique Burault (Alrudit)
- Isabelle Ferrané (IRIT)
- Kévin Gravouil (Alrudit)
- Philippe Lebas (Alrudit)
- Jean-Pierre Lorre (LINAGORA)
- Damien Nouvel (Inalco ERTIM)
- Julien Piquier (IRIT)
- Kate Thompson (LINAGORA)

## Étudiants en support (merci !)

- Othman Belgnaoui
- Cyprien Gay
- Samuel Gendre
- Mathilde Lalanne

# Programme

## **Lundi 8 juillet** dans la Bulle

- 10h-11h : introduction, présentation du hackathon
- 11h-12h : discussions par groupes
- 12h-14h : pause déjeuner (crous)
- 14h-17h : développements par équipes
- 18h-19h : présentations intermédiaires (concertation)
- 19h-21h : cocktail et buffet
- 21h-23h : développements par équipes (évaluations qualitatives)

## **Mardi 9 juillet** dans la Bulle et en hybride (lien Zoom à venir)

- 10h-11h : présentation des travaux réalisés par les équipes
- 11h-12h : remise des prix (vote) et conclusion

# Défis

1. Un membre de l'équipe interagit avec le robot pour piloter en temps réel
2. Un membre de l'équipe vocalise l'ensemble des instructions en une fois, le robot doit ensuite les exécuter
3. Un manuel réalisé par l'équipe est donné à une personne extérieure qui en prend connaissance et interagit avec le robot

# Défis

- Les équipes devront réaliser des parcours reproduisant une figure en un minimum d'étapes / instructions
  - les équipes peuvent proposer eux-mêmes une figure
  - ou en tirer une au sort
  - ou on impose 2 figures par équipes ?



- Les équipes devront reproduire des lettres en un minimum d'étapes / instructions pour former l'un des mots suivants :
  - JEP
  - TALN
  - HACKATAL

# Prix

Sur la base d'un vote des participants et des organisateurs :

- **prix Alrudit, créativité**  
Utilisation innovante et disruptive des briques logicielles par un assemblage original ou un détournement dans leur utilisation
- **prix LINAGORA, commande vocale**  
Meilleure performance dans les défis avec bonus pour l'utilisation de l'outil LinTO pour la reconnaissance de la parole.





## **Robot Explorateur**

Projet étudiant de 1A SRI

Adapté pour le HackaTal 2024

### **Contributeurs SRI 1A :**

- Othman Belgnaoui
- Cyprien Gay
- Samuel Gendre
- Mathilde Lalanne

Isabelle Ferrané – Julien Pinquier - IRIT



## Architecture générale



Serveur / Back-end



Interface utilisateur / Front-end

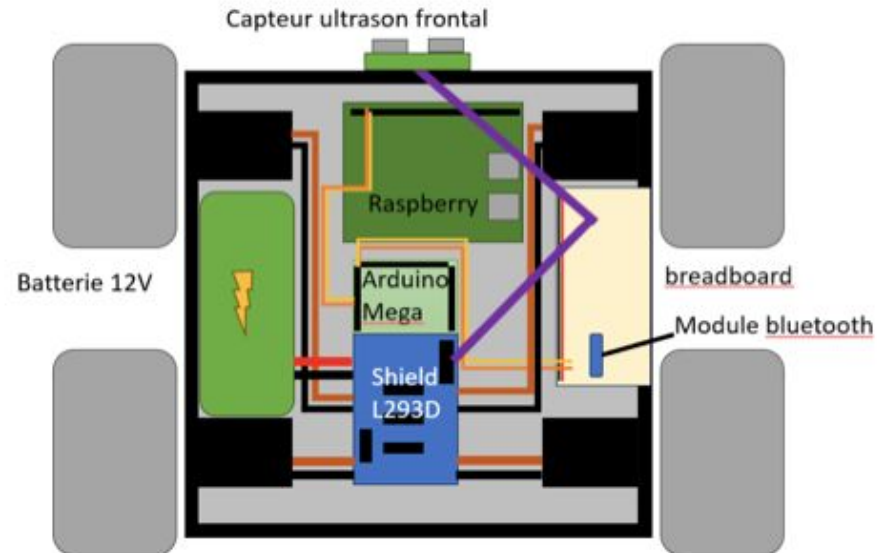
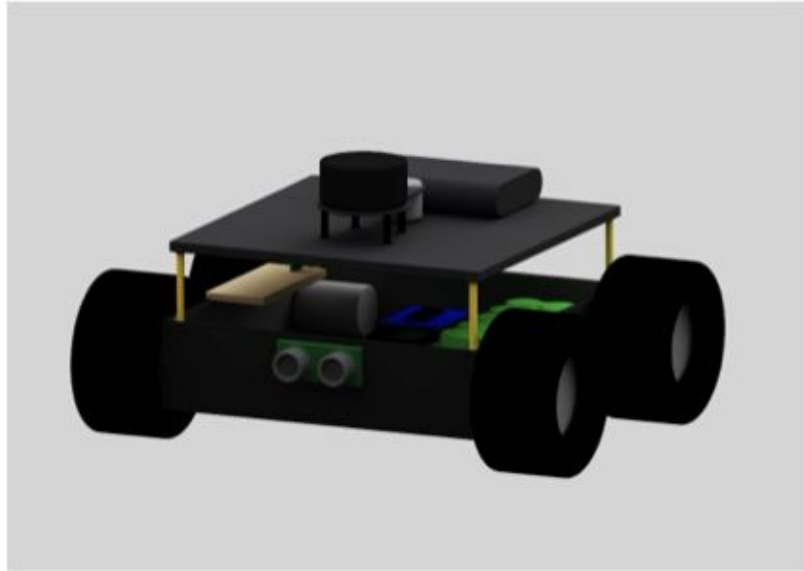


Robot mobile





## Architecture matérielle





## Interface utilisateur

Code fourni – basé Node.js Angular html



Activation enregistrement

Activation de l'interprétation  
du fichier audio





## Déplacement du robot

### COMMANDES de BASE

- **F : Forward**, avancer : met en marche avant les moteurs à la vitesse 200
- **B : Backward**, reculer : met en marche arrière les moteurs à la vitesse 200
- **L : Left**, gauche : fait tourner le robot sur lui-même vers la gauche
- **R : Right**, droite : fait tourner le robot sur lui-même vers la droite
- **S : stop** : arrête le robot. Il faut noter que l'arrêt n'est pas instantané et donc anticiper la distance de freinage, parfois jusqu'à 20cm selon la vitesse
- **+** : accélère : augmente le paramètre speed des moteurs et augmente la vitesse du robot
- **-** : ralenti : réduit le paramètre speed des moteurs et réduit la vitesse du robot
- **V** : met le paramètre speed à puissance maximum, soit à 100%

### DÉPLACEMENT

Distance de base  
~1 mètre fonction du délai

Angle 90°



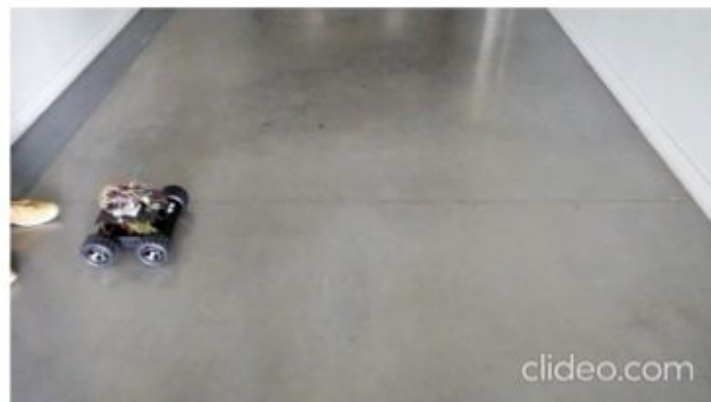
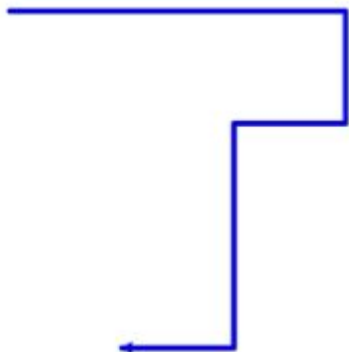


## Déplacement du robot

**SIMULATION**

**DÉPLACEMENT RÉEL**

['F', 'F', 'F', 'R', 'F', 'L', 'B', 'L', 'F', 'F', 'R', 'F']





## Environnement fourni

### CODE DISPONIBLE

#### Archive HACKATAL\_2024

```
#% RECOUPÉRIER DE L'ARCHIVE HACKATAL_2024.zip
## Lien Télécharger : https://filesender.renater.fr/?download&token=7f154962-aa26-4884-91a2-4776e352ee08

#####
## A AN INSTALLATION = A FAIRE UNE SEULE FOIS SI VOUS AVEZ LE CHALLENGE #####
#####

## 1. AN GESTION D'UN ENVIRONNEMENT VIRTUEL

## Installation de micromamba pour gérer l'environnement (si vous ne l'avez pas déjà)
https://docs.mamba.sh/getting-started/install/#install
```



#### Fichiers

HACKATAL\_2024.zip (84 Mo) : 4 Téléchargements  
install\_env\_python\_HACKATAL\_light.md (12.5 ko) : 6 Téléchargements  
install\_env\_python\_HACKATAL.md (16.9 ko) : 5 Téléchargements  
environment\_robot\_backend\_V2.yml (328 o) : 4 Téléchargements

< > HACKATAL\_2024

Nom

Interface\_Commande\_Vocale  
Interface\_Initiale\_Mode\_Manuel  
interface\_robot\_explorateur  
Traitement\_ROBOT

Front-end  
Back-end





**HACKATAL ...**

**... A VOUS DE JOUER !!!**



# API TAL

Hackatal 2024



**Phillipe Lebas**  
*Fondateur et CEO*



**Kévin Gravouil**  
*Responsable R&D*



# Comment y accéder

1. Récupérer un token d'authentification auprès d'Airudit
2. Aller à : `http://<ADRESSE_IP>:8008/docs`
3. Via Python :

```
import requests
BASE_URL = "http://<ADRESSE_IP>:8008/"
headers = {
    "X-Airudit-Token": "airu-XXX",
    "Content-Type": "application/json",
}
your_data = { ... }
resp = requests.post(f"{BASE_URL}/multitask/predict", headers=headers, json=your_data)
print( resp.status_code ) # 200=OK, 401=unauthorized, 500=venez me voir
print( resp.headers ) # peut contenir des infos utiles
print( resp.json() ) # contient la résultat de l'analyse demandée
```

# Fonctionnalités

/check_auth_token	Vérifie si le token d'authentification est correct
/tokenize	Tokenize une phrase
/matcher_fast	Match many keywords on a sentence
/matcher_complex ⚠	Match many “NLP patterns” on a sentence
/number_handler ⚠	Détection des expressions et valeurs numériques
/temporal_handler ⚠	Détection des informations temporelles
/multitask/predict	Prédit les entités nommées, l’arbre syntaxique et les POS tags
/generate	Génère la suite des tokens probables du texte en entrée à l'aide d'un LLM
/nli	Estimation de l'adéquation entre un <i>statement</i> et une liste de labels candidats

**http://<ADRESSE\_IP>:8008/docs**

**ou venez me voir**

# Prix

Votre **créativité** dans la manière  
où vous exploitez et assemblez  
les différents outils existants



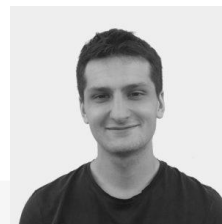
**LinTO-STT : API de transcription**



**Jean-Pierre Lorré**  
R&D Director



**Kate Thompson**  
NLP



**Audran Bert**  
Speech recognition



# LinTO-STT, qu'est ce que c'est ?

## C'est quoi ? Une API

- De transcription
- Open source
- Écrite en Python
- Capable de fonctionner en mode temps réel ou non

## Comment ça marche ?

- Une image Docker à déployer
- Simple requête HTTP
- En mode temps réel il faut se connecter en WebSocket

## Avantages ?

- Modèle haute performance : Whisper
- Ou modèle moins performant mais plus léger : kaldil
- Fonctionne sur CPU et GPU
- Capable de transcrire dans plusieurs langues
- Ou de traduire vers l'anglais



# LinTO-STT, Comment l'utiliser ?

## Docker

- Installer Docker
- Choisir son image
- Définir les différentes options
- Lancer l'image

[Tutoriel pour déployer l'API](#)

## Paramètres

- Choix du modèle
- Choix de la langue
- Choix du device (GPU/CPU)
- Mode streaming (temps réel)

## Requêtes

- Mode « offline » :  
`curl -X POST "http://ADDRESS"...`
- Mode streaming :
  - Exemple dans le [Repo GitHub](#).  
Connexion via le package websockets en Python

Gitlab:

[https://gitlab.irit.fr/melodi/hackatal\\_robot24\\_public](https://gitlab.irit.fr/melodi/hackatal_robot24_public)



