

# Le chatbot

Présentation rapide de sa mise en place



# Introduction

Ce powerpoint a pour but d'expliquer brièvement le chatbot réalisé et de donner quelques détails sur l'apprentissage du langage et du chatbot

NB : Le chatbot n'a pas été entraîné sur la reconnaissance de tous les noms des arrêts, il est donc normal qu'il se trompe parfois. Pour éviter cela, mieux vaut choisir des noms d'arrêts en un mot, ou alors reliés par des tirets.

# Les fonctionnalités du chatbot

Ce chatbot a pour vocation de répondre aux questions de l'utilisateur sur les transports en commun dans la région parisienne.

Ces questions peuvent porter sur plusieurs aspects :

- le retard moyen sur une ligne ou sur un arrêt
- le nombre de validations moyen sur une ligne ou sur un arrêt
- l'ensemble des lignes passant par un arrêt
- les arrêts d'une ligne permettant de changer de ligne
- l'arrêt d'une ligne le plus proche
- l'horaire du prochain passage à un arrêt

Seuls la reconnaissance du langage et le fonctionnement du chatbot ont été réalisés ici : il n'y a pas d'appel à la base de données Dataviz-neo4J, seulement des données fictives codées en dur.



Sur slack




 **Camille** 14 h 58  
bnjour


 **transportbot** APPLI 14 h 58  
Bonjour ! Que puis-je faire pour vous ?


 **Camille** 14 h 58  
Je cherche des informations sur le rer B


 **transportbot** APPLI 14 h 58  
Que voulez vous savoir sur la b ?

 **Camille** 14 h 58  
mmh.. est-ce qu'il y a du monde, en général ?

 **transportbot** APPLI 14 h 58  
Il y a en moyenne 585250 validations par jour sur la ligne b

 **Camille** 14 h 59  
D'accord. Et est-ce qu'il y a du retard ?

 **transportbot** APPLI 14 h 59  
Je vous dis ça dans un instant.  
Le retard moyen sur la ligne b est de 2 minutes et 35.0 secondes  
Que puis-je faire d'autre pour vous aider ?

 **Camille** 14 h 59  
Comment est-ce que je fais pour joindre la ligne B ?


 **transportbot** APPLI 14 h 59  
Pour cela, j'aurais besoin de vos coordonnées GPS. Est-ce que vous les connaissez ?

Veuillez entrer votre latitude

 **Camille** 15 h  
48.7747526

 **transportbot** APPLI 15 h  
Veuillez entrer votre longitude

 **Camille** 15 h  
2.2393253999999843

 **transportbot** APPLI 15 h  
L'arrêt de la ligne b le plus proche de vous est Antony, de latitude 48.80469043670592 et de longitude 2.070612019531268.  
Puis-je faire autre chose pour vous aider ?



Oui, j'aimerais des informations sur Antony, cette fois-ci



nouveaux messages

# La structure des fichiers



# La structure des fichiers

Pour cadrer l'apprentissage du robot, 6 fichiers sont indispensables :

- nlu\_model\_config.yml
- domain.yml
- data/data.json
- data/stories.md
- actions.py
- endpoints.yml

Les fichiers .py autres que action.py permettent soit de lancer l'apprentissage du robot, soit de lancer le robot lui-même

Les autres fichiers sont en général générés lors de la création du modèle de langage ou du modèle du robot

# Nlu\_model\_config.yml

- Permet de choisir la pipeline utilisée (les principales sont détaillées dans la doc de rasa nlu)
- Permet de choisir le langage utilisé





# Domain.yml

C'est ici qu'est vraiment défini le cadre dans lequel évolue le robot

- On y trouve ce que l'utilisateur lui dit : intents et entities
- On y trouve ce que le robot répond à l'utilisateur : utter et actions



# Data/data.json

C'est ce fichier qui permet l'apprentissage du langage grâce aux exemples qu'on y trouve. Grâce à lui, le robot apprend à reconnaître les intentions de l'utilisateur et les entités qu'il doit extraire pour comprendre ce qu'on lui demande (par exemple, le nom de l'arrêt, le but de la question posée...).

Ce fichier peut être fastidieux à écrire à la main, d'où l'avantage de l'outil développé par rasa, `rasa-nlu-trainer`



# Data/stories.md

C'est grâce à ce fichier que le robot apprend les réactions qu'il doit avoir en fonction de ce que lui dit l'utilisateur.

Les stories sont en quelques sorte des dialogues types entre l'utilisateur et le robot, où figurent non pas des phrases, mais des intents, des entities, des utter et des actions.

Il faut les générer à la main, même s'il me semble qu'il y a des outils pour faciliter leur génération.

Pour simplifier leur écriture et leur structuration, on peut aussi utiliser les checkpoints (repérés par > checkpoint), qui relient des blocs de stories entre eux.

Si l'utilisateur ne peut communiquer qu'en utilisant des boutons, il se peut que ces stories écrites à la main et reliées entre elles par des checkpoints suffisent à couvrir tout le domaine d'utilisation du robot.

Sinon, il faut aussi générer des stories en apprenant au robot (grâce à `train_online.py`). Cet apprentissage permet de recouvrir tous les cas qui n'ont pas été pensés en amont.

# Actions.py

C'est ici que sont écrites les actions plus compliquées qu'un simple affichage.



# Endpoints.yml

Détermine l'url où sont lancées les actions.



# L'environnement du transportBot



# Le contenu de domain.yml

- Intents
  - greet : salue le robot
  - thankyou : le remercie
  - goodbye : lui dit au revoir
  - inform : lui donne des informations sur ce qu'il veut savoir
- Entities
  - info\_type : route ou stop, c'est ce sur quoi va porter la question de l'utilisateur
  - stop\_name : le nom de l'arrêt sur lequel on demande l'information
  - route\_name : le nom de la ligne sur laquelle on demande l'information
  - aim : delay, traffic, change, nearest\_stop, ce que l'utilisateur veut savoir
  - transport\_type : train, rer\_train, metro\_train, bus, tramway : le type de transport que l'utilisateur veut prendre. Cette entité n'a pas pour l'instant pas d'utilité car elle n'aurait servi qu'au moment de lancer la requête pour trouver la ligne demandée par l'utilisateur (qui n'a pas été codée ici)
  - latitude : pour nearest\_stop
  - longitude : pour nearest\_stop
  - gps\_coord\_known : pour nearest\_stop

# Les questions possibles

Aim	Pour les arrêts	Pour les lignes
Delay	Retard moyen à l'arrêt	Retard moyen sur la ligne
Traffic	Nombre de validations à l'arrêt	Nombre de validations sur la ligne
Change	Les lignes passant par cet arrêt	Les arrêts de la ligne permettant un changement vers une autre ligne
Next_train	L'horaire du prochain passage	/
Nearest_stop	/	L'arrêt de la ligne le plus proche de l'utilisateur





# L'apprentissage du langage

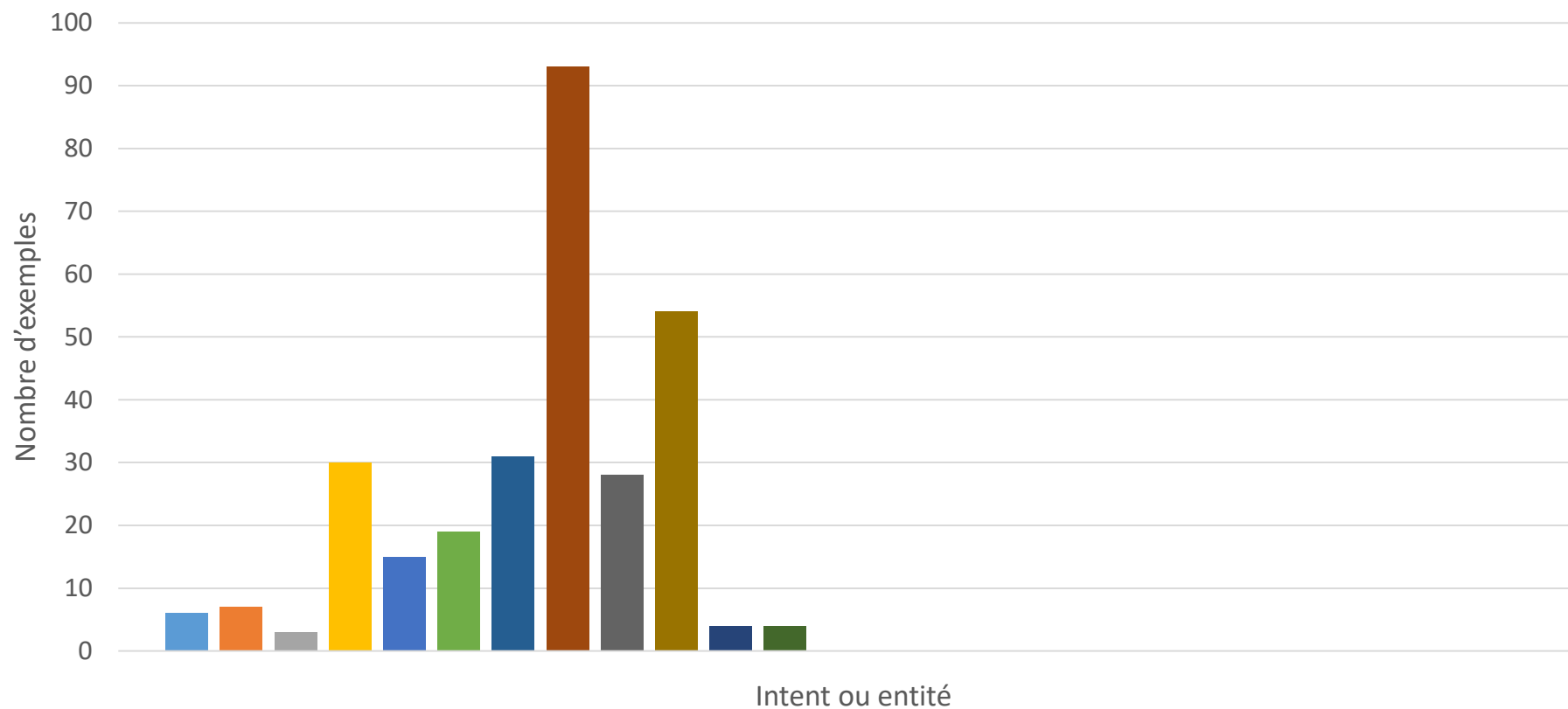
Se fait par des exemples sous format json



# Entraînement du NLU

Nombre d'exemples comportant l'intent ou l'entité

Nombre total d'exemples : 225



greet goodbye thankyou aim{next\_train} aim{delay} aim{change}  
aim{traffic} stop\_name route\_name info\_type latitude longitude

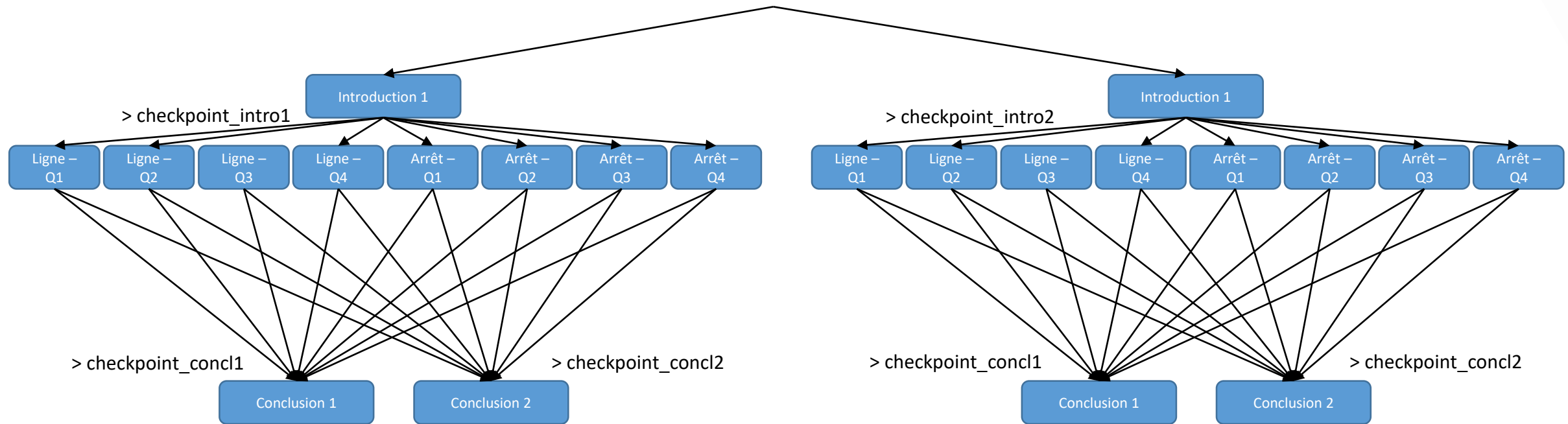
# L'apprentissage du robot

Se fait par la création de stories



# Mise en place des stories – théorie

Soit l'arbre suivant qui décrit les cours de discussion possibles :



# Mise en place des stories – théorie

Ce qui donne la structure des stories suivante :

## Introduction 1  
Blabla  
> checkpoint\_intro1

## Introduction 2  
Blabla  
> checkpoint\_intro2

## Ligne – Q1 11  
> checkpoint\_intro1  
Blabla  
> checkpoint\_concl1

## Ligne – Q1 12  
> checkpoint\_intro1  
Blabla  
> checkpoint\_concl2

## Ligne – Q1 21  
> checkpoint\_intro1  
Blabla  
> checkpoint\_concl2

## Ligne – Q1 22  
> checkpoint\_intro2  
Blabla  
> checkpoint\_concl2

## Conclusion 1  
> checkpoint\_concl1  
Blabla

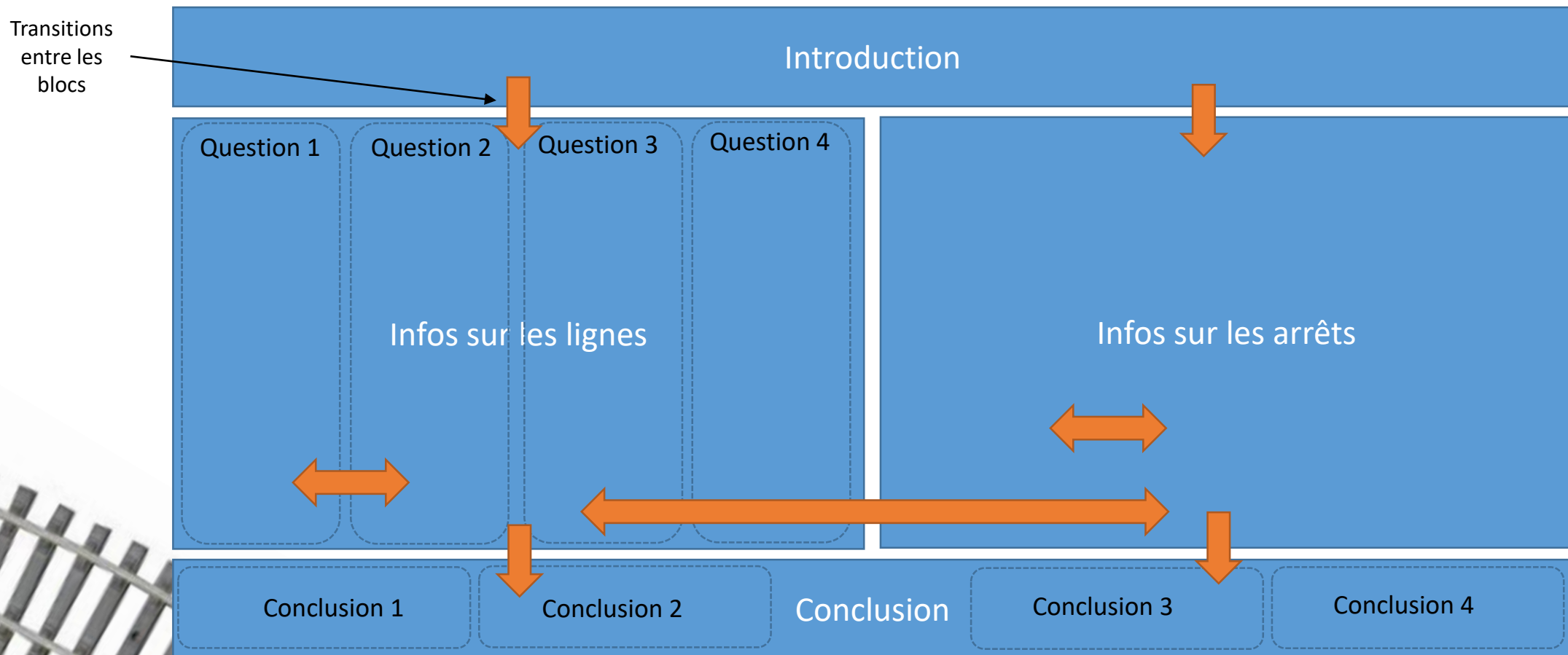
## Conclusion 2  
> checkpoint\_concl2  
Blabla

Idem pour toutes les  
autres questions

NB : Si on veut passer d'une question à une autre, il suffit d'ajouter des stories de transition, associées au checkpoint ask\_any\_other\_question

# Mise en place des stories – pour le transportbot

Les stories initiales sont réparties en plusieurs blocs :



Par la suite, des stories ont été ajoutées en entraînant le robot

# Mise en place des stories – pour le transportbot

En tout, on a les stories suivantes :

Nombre de questions

Type de story	Nombre de stories
Introduction	6
Ask about delay	3
Ask about traffic	3
Ask about next train	2
Ask about nearest stop	3
Ask about possible routes	3
Generated stories de correction	26
Ask any other question	11
Say something useless	3
Infinite hello	1
Tankyou	2
Conclusion	3

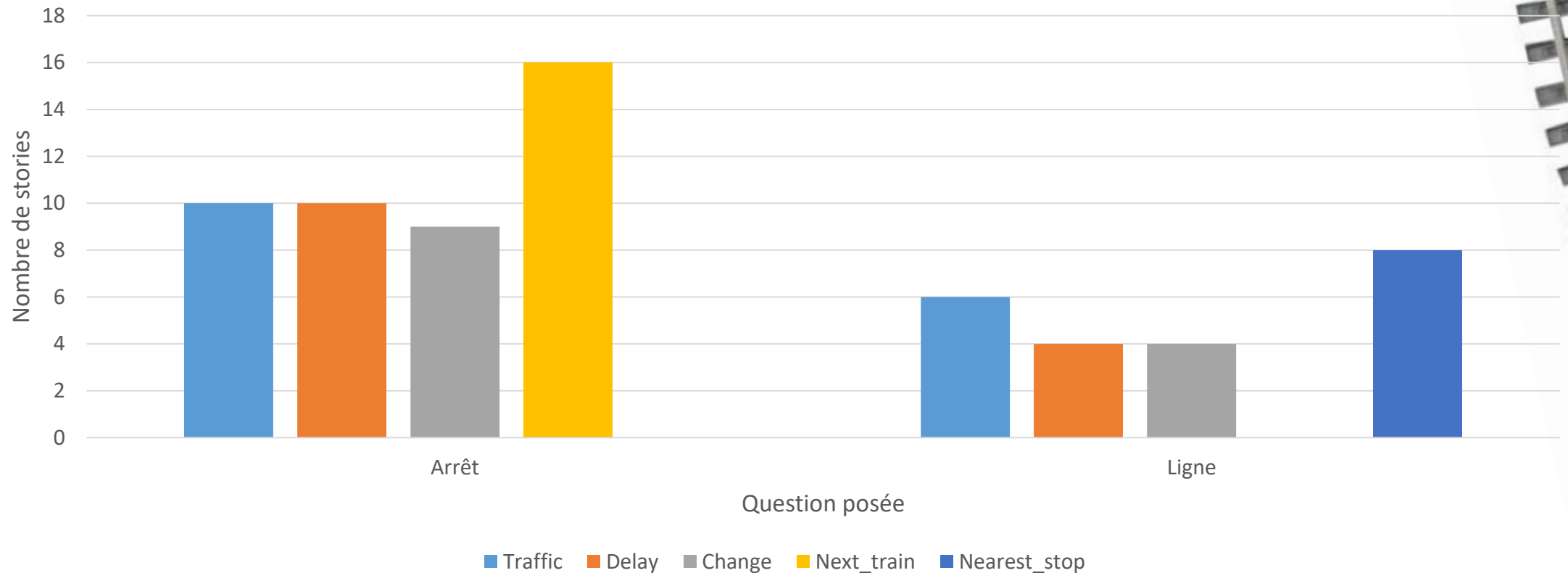
A peu près fixe, ce sont les différentes manières de poser la question pour l'utilisateur

Permet la transition entre les différentes parties et gère l'aléatoire de l'utilisateur  
Plus on prend en compte sa capacité à dire n'importe quoi, plus il y en a !

Dépend du nombre d'entraînements qu'on donne au robot  
Pas forcément utile dans le cas de boutons, mais sinon, permet de gérer les cas non prévus initialement

# Nombre de stories écrites

Nombre de stories écrites pour chaque question posée par l'utilisateur



Attention : il s'agit du nombre blocs de stories écrits, en réalité, grâce aux checkpoints, bien plus de stories sont générées. Il y a aussi des stories de transition, qui permettent de passer d'un bloc à l'autre (justement, grâce à ces checkpoints).