Chaîne d'évaluation de compréhension du langage naturel

28 Janvier 2020

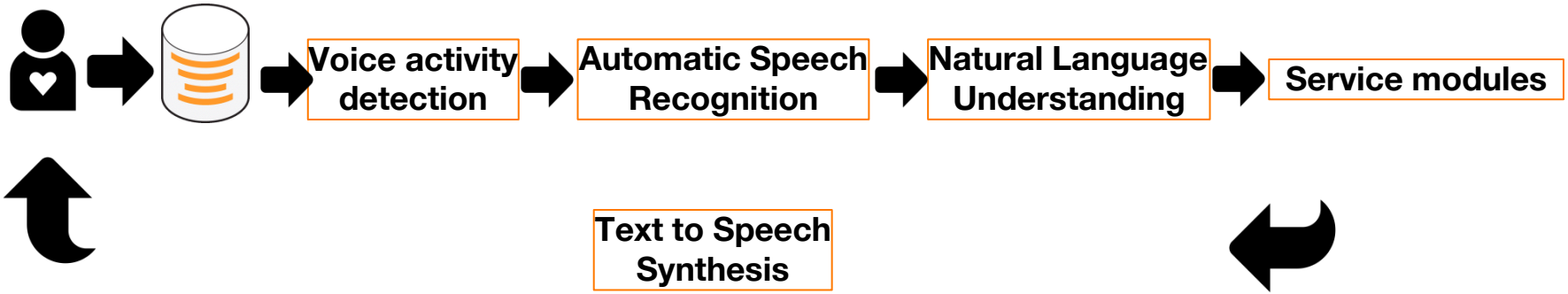
retour d'expérience sur l'assistant vocal Djingo d'Orange

Ghislain Putois

Orange Labs

Contexte

Architecture fonctionnelle d'un assistant vocal



Permet d'obtenir les news, la météo, de jouer de la musique, diffuser un programme télé, … cette présentation se concentre autour de l'extraction des informations des énoncés utilisateur par le composant NLU

NLU: Natural Language Understanding

Le NLU est l'étape d'interprétation de la requête d'un utilisateur

- **Interpréter, c'est au sens d'une ontologie** (*i.e.* une description de l'organisation des objets, concepts et relation du service)

- **Pour l'assistant vocal Djingo : en sortie de la reconnaissance vocale**
  - Travaille sur du texte, éventuellement bruité par les soucis de transcription
  - Fournit en sortie une « interprétation »

  Exemple :
  - énoncé = « mets la chanson joe le taxi »
  - sortie attendue= « **intent= music_play, title = joe le taxi** »

Comment marche le NLU ?

NLU: Natural Language Understanding

Deux tâches :

- **Trouver l'intention parmi les centaines définies dans l'ontologie : classification**
- **Trouver les valeurs dans le texte de l'énoncé, et associer les bons concepts : étiquetage**

Astuce : on peut regrouper ces deux tâches en une tâche globale d'étiquetage, en rajoutant un mot vide en début de phrase, qui recevra l'intention

Music_play	Mets O	la O	chanson O	Joe B-title	le I-title	taxi I-title	de O	Vanessa B-artist	Paradis I-artist

Solutions apprentissage automatique au problème d'étiquetage :

- **avant Conditional Random Fields**
- **maintenant, réseaux de neurones avec word embeddings (bi-lstm, BERT)**
  - Word embedding représentation vectorielle dense d'un mot (en contexte) apprise de manière non supervisée à partir de (très) grand volumes de textes bruts

Comment évaluer le NLU ?

Comment évaluer un système de classification ?

Sur quelles données ?

Comparer la sortie du NLU avec la sortie idéale, et compter les différences !

Quelle est la sortie idéale ?

Comment compter les différences ?

Exemple des défis internationaux : Kaggle

Fournies par les organisateurs

Sur quelles données ?

Comparer la sortie du NLU avec la sortie idéale, et compter les différences !

Fournie par les organisateurs

Quelle est la sortie idéale ?

Fourni par les organisateurs

Comment compter les différences ?

Conception d’un service

A définir

Sur quelles données ?

Comparer la sortie du NLU avec la sortie idéale, et compter les différences !

A définir

Quelle est la sortie idéale ?

A définir

Comment compter les différences ?

Collection des énoncés, annotation, et constitution de corpus

- **Une démarche en trois étapes :**
  - Collecter des énoncés
    - un énoncé = une requête utilisateur prononcé à un instant précis
  - Annoter les énoncés = définir la « sortie idéale »
    - en accord avec le guide d'annotation et l'ontologie
  - Construire les corpus : échantillonnages représentatifs

Typologie des énoncés en entrée

Enoncés « In-domain »

- **Énoncé qui se projette dans l'ontologie courante**
  - Ex : « mets la chanson Joe le taxi »
  - Sortie attendue : interprétation correcte de l'énoncé
    - Intent=music_play & concept=title & concept value=« joe le taxi »

Enoncés « Out-of-domain »

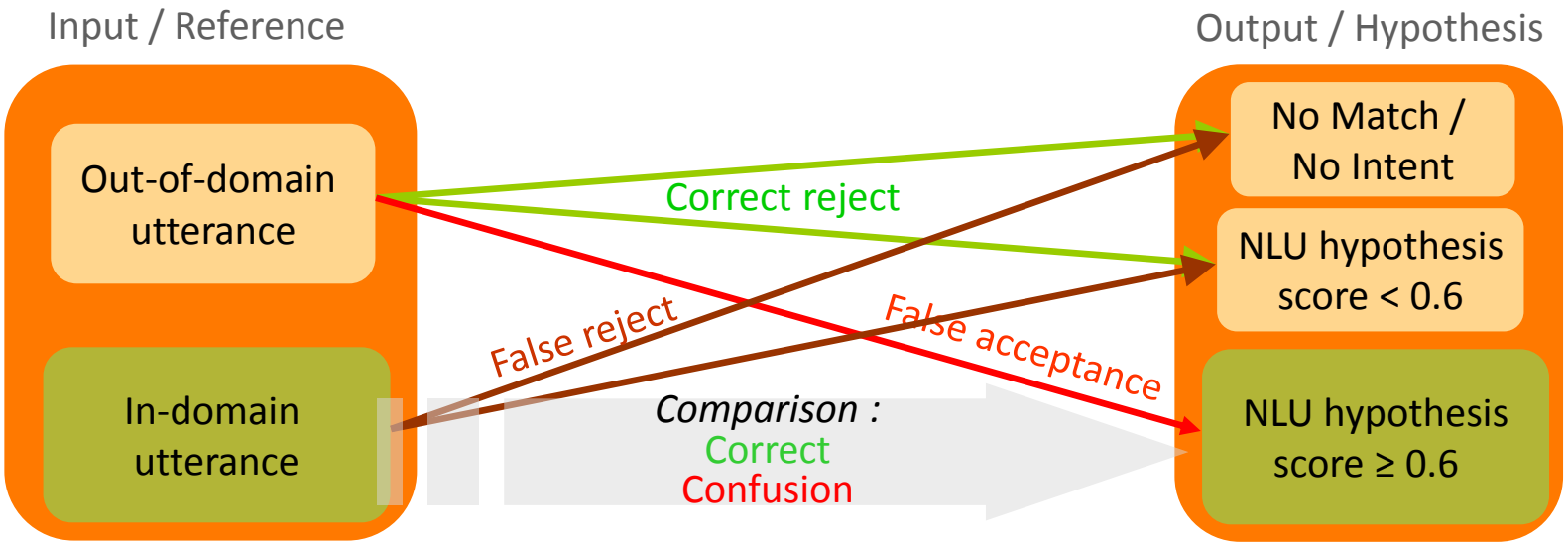
- **Un énoncé qui ne se projette pas dans l'ontologie courante (ou sur un concept « poubelle / autres »)**
  - Ex : « ma chemise est sale »
  - Sortie attendue : rejet de la requête

Sortie d’un classifieur NLU

- **Rejet explicite : « No Match » , « No Intent »**
  - Le système décide qu’il n’a pas d’interprétation pour l’énoncé dans l’ontologie courante

- **Interprétation avec un score de confiance**
  - Le système propose une interprétation de l’énoncé, accompagné en général d’un score de confiance
  - La décision d’action dépend alors d’un seuil de confiance :
    - Si le score de confiance est supérieur au seuil, le système accepte l’interprétation proposée
    - Sinon, il rejette l’interprétation, car il n’a pas d’interprétation suffisamment fiable

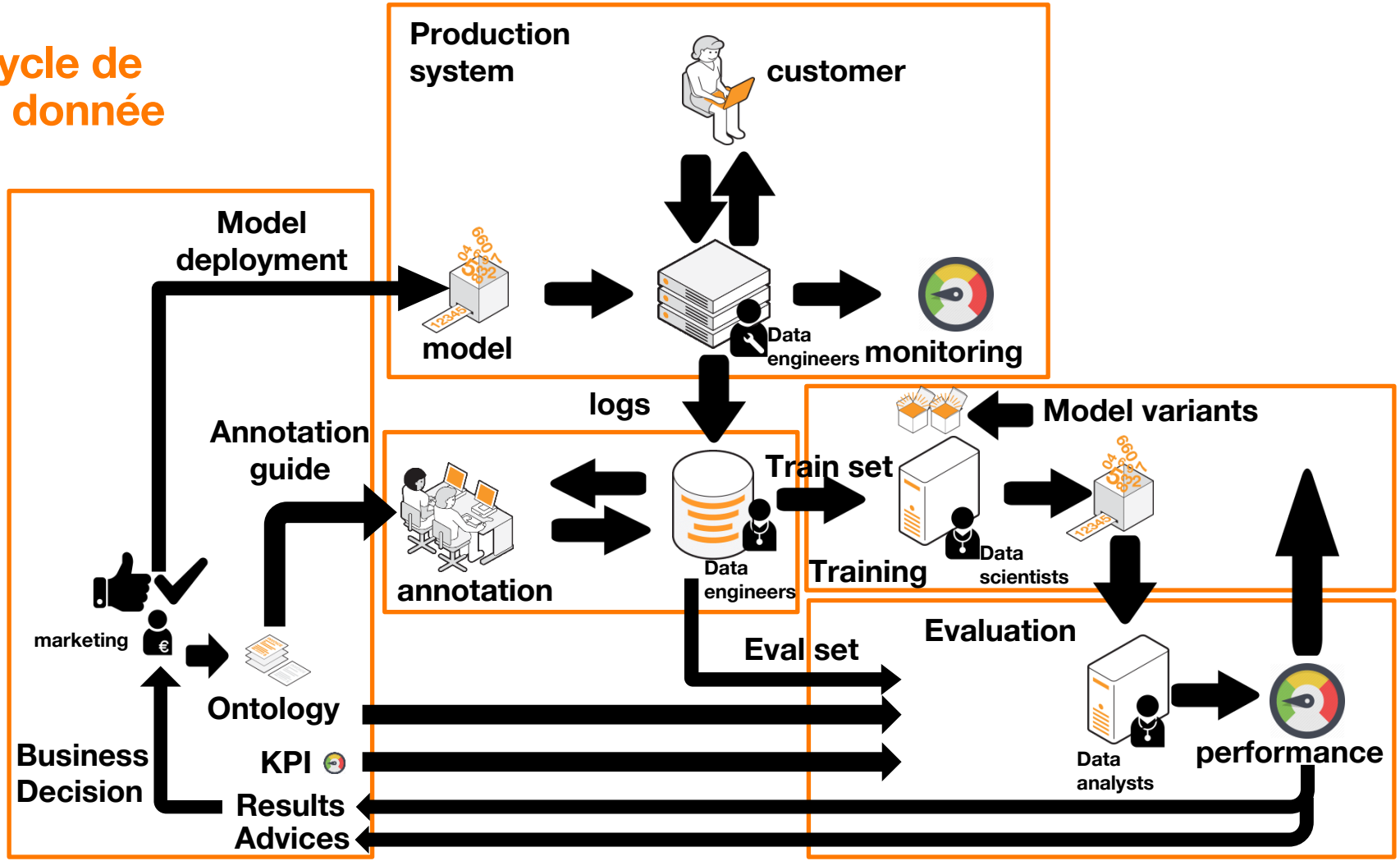
Evaluation du NLU : les différents cas



- Correct reject: énoncé out-of-domain rejeté par le système (directement ou à cause d'un score de confiance trop bas)
- False acceptance: énoncé out-of-domain accepté à tort par le système
- False reject: énoncé in-domain rejeté par le système
- Correct: énoncé in-domain reconnu correctement
- Confusion: énoncé in-domain reconnu, mais mal interprété

Organisation et conseils

Cycle de la donnée



Comment compter les différences ?

- **« Ontologie » hiérarchique Djingo :**
  - Divisée en domaines métier
    - Chaque domaine métier a un ensemble fini d'intents
    - Chaque intent admet une liste finie de concepts (qui prennent leur valeur dans l'énoncé)

- **Les évaluations suivent cette hiérarchie :**
  - Différents niveaux d'erreur d'importance décroissante :
    - domaine métier
    - intent
    - concept
    - valeur du concept

Sortie idéale

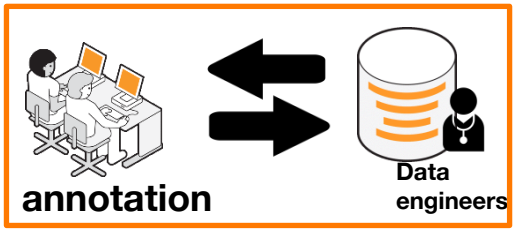
Une sortie complètement annotée, non ambiguë, indépendante du contexte, et alignée avec l'ontologie courante

Indépendante du contexte : le NLU ne résout pas les ambiguïtés contextuelles. Cette résolution se fait soit dans les modules de service, soit à travers les consignes définies dans le guide d'annotation

Non-ambiguë : requiert un accord inter-annotateurs formalisé dans un guide d'annotation, qui doit être validé par toutes les parties prenantes (métier/marketing, développeurs et annotateurs)

Alignée avec l'ontologie : l'ontologie évolue régulièrement avec les améliorations de la compréhension métier des besoins des utilisateurs

- **il faut versionner les ontologies et les annotations**
- **il y a un travail important lors des évolutions d'ontologies**



Aperçu d’un outil d’annotation

Apprentissage et évaluation

1) Les jeux de données doivent être distincts :

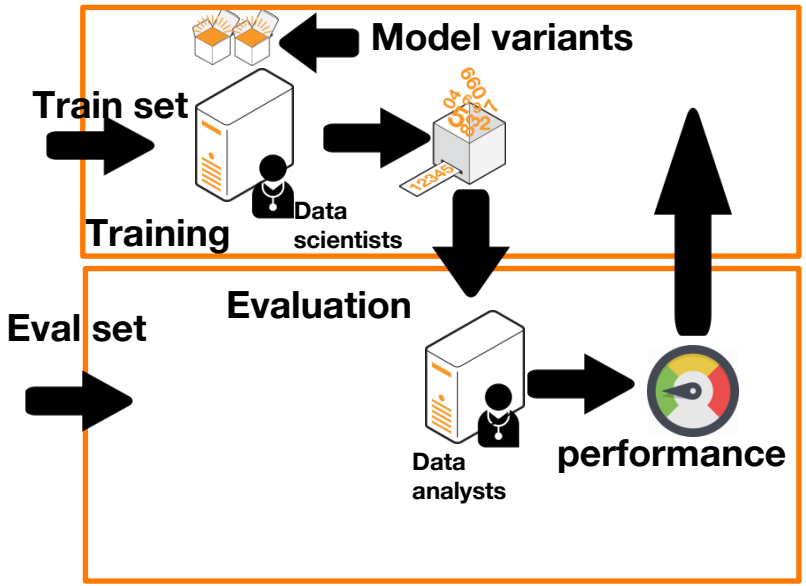
- en termes d'énoncés, pas de transcription
- les données d'évaluation ne doivent pas être utilisées lors de l'apprentissage
- certaines transcriptions peuvent être communes!

2) Les données d'évaluation doivent refléter la distribution réelle d'usage :

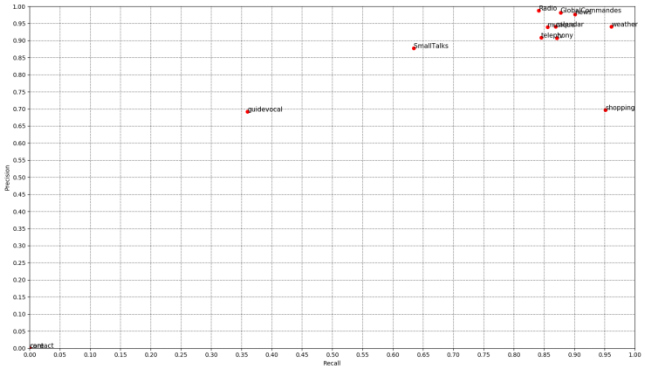
- respecter décomptes et fréquences
- collecter des volumes importants (« long tail »)
- pas de raison de cas spéciaux (« top requests », « dirty samples »)

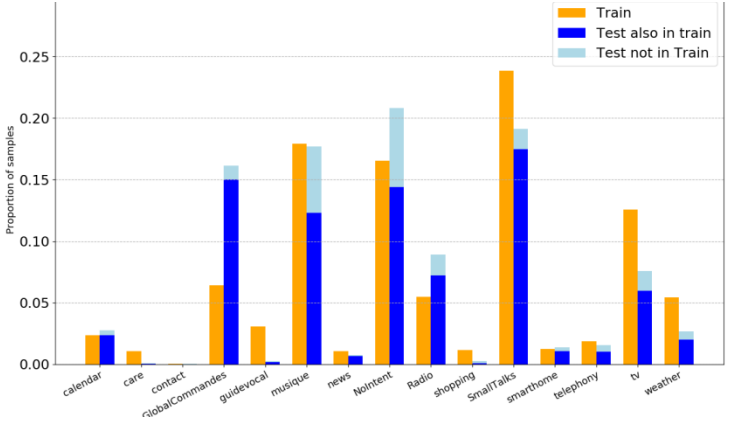
3) Les données d'apprentissage devraient refléter la distribution réelle d'usage :

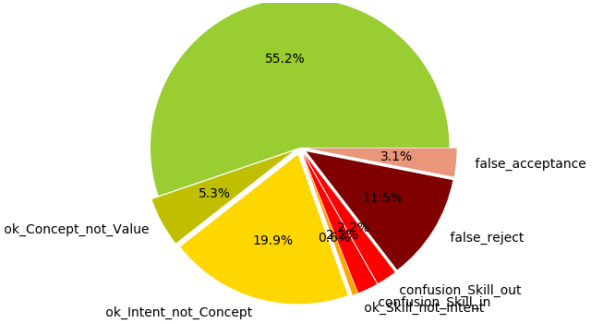
- meilleures performances

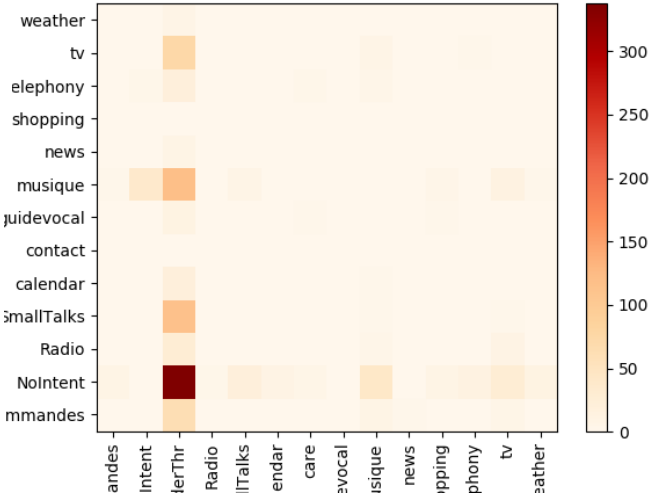


Data visualisation (illustration)









Merci

des questions ?