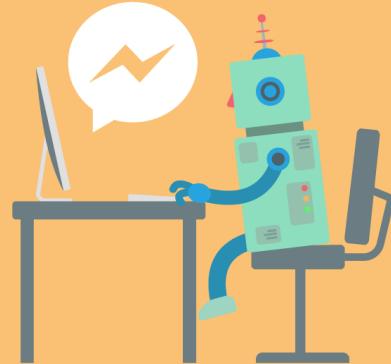


+



Big Data, IA et Chatbots

Gilbert NZEKA
pro@gilbertnzekwa.com
<https://www.linkedin.com/in/nzeka>

+

Qui suis-je ?

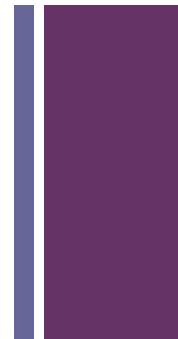
Gilbert NZEKA

- Formateur
 - En entreprise
 - Université
 - Ecoles supérieures
- Entrepreneur
 - Solution de paiement
 - Billetterie en ligne
 - Daily deals dans le sport
 - Legaltech
- Auteur et co-auteur



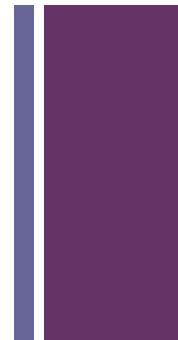
+

Et vous?



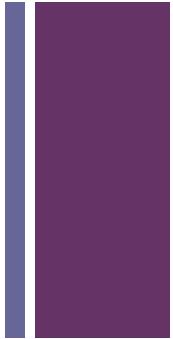
+

Vos attentes de la formation





Programme



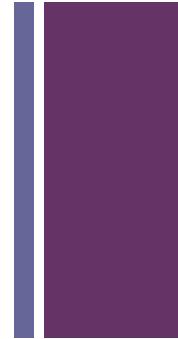
- Introduction
 - Big Data?
 - IA ?
 - Chatbot ?
 - Objectifs et potentiel de ces innovations
 - Retours d'expériences
- Crédit : création d'un premier chatbot
- Projet : créer le chatbot du site des services publics

+

Introduction



Big Data : définition et enjeux

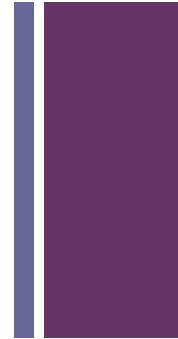


” Le big data, littéralement « grosses données », ou mégadonnées, parfois appelées données massives, désigne des ensembles de données devenus si volumineux qu'ils dépassent l'intuition et les capacités humaines d'analyse et même celles des outils informatiques classiques de gestion de base de données ou de l'information. ” Wikipedia





Big Data : définition et enjeux

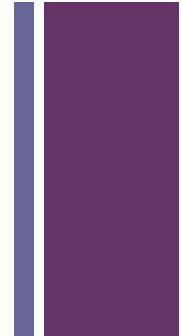


Cette explosion quantitative de données numériques demande de repenser la manière **capturer, stocker, rechercher, partager, analyser et visualiser les données**. Une nouvelle conception des traitements qui doit prendre en compte la règle des 3V : Volume, Vélocité et Variété des données





IA : définition et enjeux

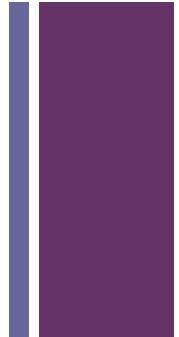


” L'intelligence artificielle (IA) est « l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence » ” Wikipedia





IA : définition et enjeux



” Le terme « intelligence artificielle », créé par John McCarthy, est souvent abrégé par le sigle « IA ». Il est défini par l'un de ses créateurs, Marvin Lee Minsky, comme « la construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique » ”





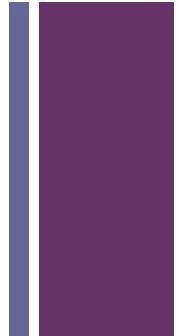
Chatbot : définition et enjeux

” Un chatbot, appelé parfois assistant virtuel ou dialogueur, est un programme informatique qui simule une conversation avec une personne, à l'écrit ou à l'oral. Quand un utilisateur lui pose une question ou formule une commande, le chatbot lui répondra ou exécutera l'action demandée. ”



+

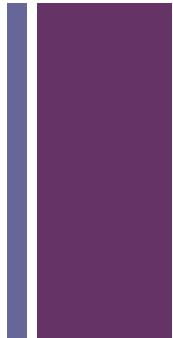
Le potentiel de ces innovations & technologies de rupture



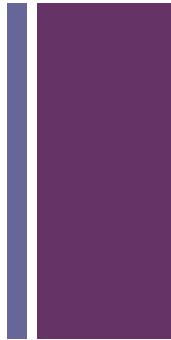
” Une technologie de rupture (dite aussi innovation de rupture ou rupture technologique) est une innovation technologique qui porte sur un produit ou un service et qui finit par remplacer une technologie dominante sur un marché. ”



Le potentiel de ces innovations & technologies de rupture



” Une technologie de rupture survient et domine un marché déjà existant soit en remplissant une fonction que la technologie traditionnelle ne pouvait pas remplir ou bien en augmentant progressivement les parts de marché au fur et à mesure que les performances augmentent, jusqu'à remplacer ceux qui étaient établis sur le marché.”



+

+

Quelques exemples d'application?

+

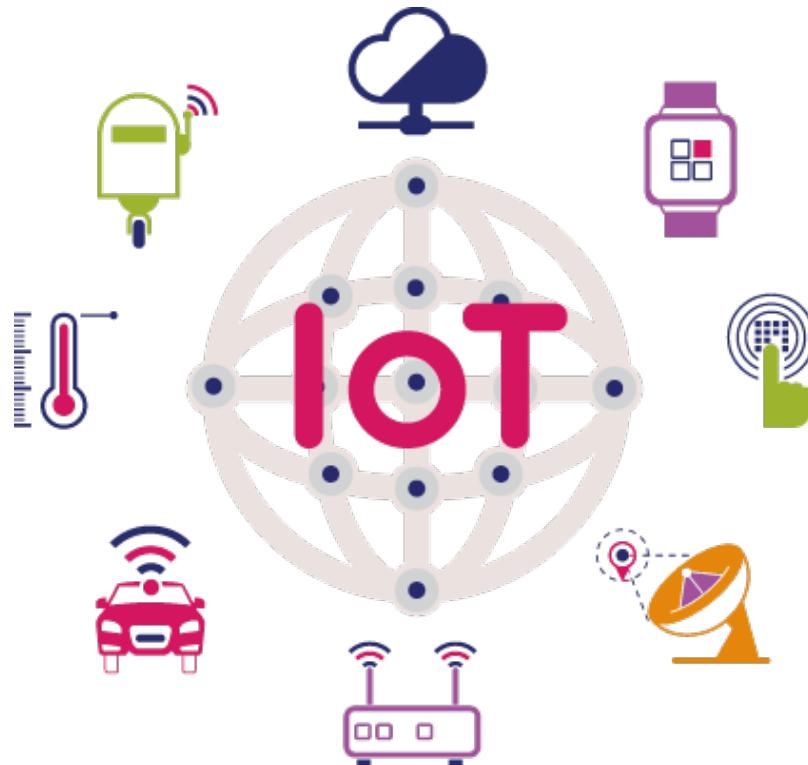
Quelques exemples d'application?



« Un véhicule autonome est un véhicule automobile apte à rouler, sur route ouverte, sans intervention d'un conducteur. »

+

Quelques exemples d'application?



« l'Internet des objets caractérise des objets physiques connectés ayant leur propre identité numérique et capables de communiquer les uns avec les autres. Ce réseau crée une passerelle entre le monde physique et le monde virtuel. »



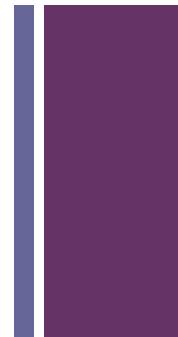
Quelques exemples d'application?



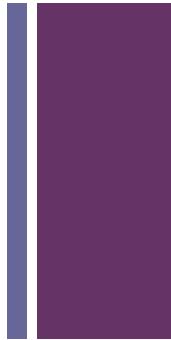
« La banque et l'assurance peuvent profiter énormément de ces technologies de ruptures et de nombreuses néo-banques le font déjà pour améliorer le service, les produits, l'amélioration des conseils, la réduction des tarifs, etc. »

+

Vos retours d'expérience !



Have an Idea?



+

+

Mon premier Chatbot



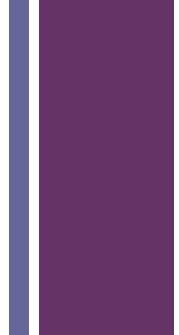
Chatbot : définition et enjeux

” Un chatbot, appelé parfois assistant virtuel ou dialogueur, est un programme informatique qui simule une conversation avec une personne, à l'écrit ou à l'oral. Quand un utilisateur lui pose une question ou formule une commande, le chatbot lui répondra ou exécutera l'action demandée. ”





Mon premier chatbot



” La construction des chatbots n'a jamais été aussi simplifiée. La majorité des éditeurs et réseaux sociaux proposent des APIs ouvertes pour la création de bots: *Facebook, Skype, Telegram, Slack, Microsoft, IBM...* Sur ces plateformes des millions d'utilisateurs effectuent déjà des achats, réservent des trains, des chambres d'hôtel et interagissent avec des chatbots en charge de la relation client. ”

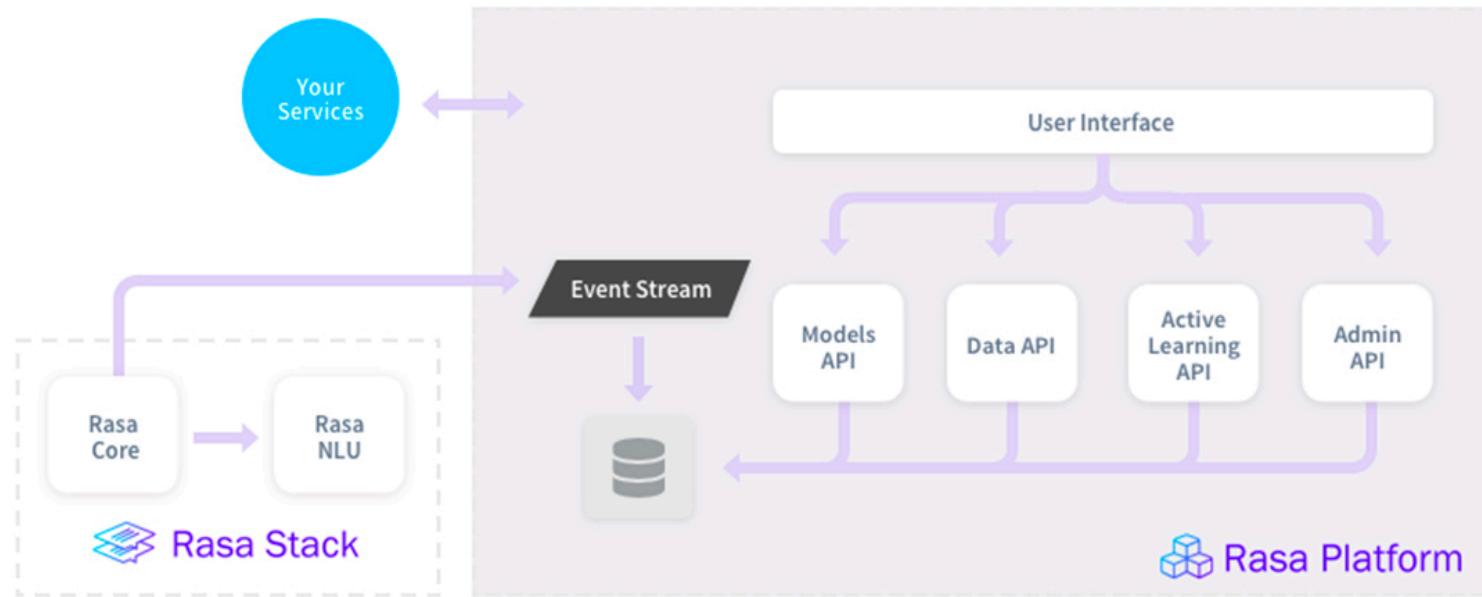


Mon premier chatbot avec RASA

- RASA est une plateforme open source créée par une entreprise allemande spécialisée dans les solutions de chatbot. Elle accompagne ses clients dans la mise en place d'une plateforme pour l'intégration de son agent conversationnel, notamment en fournissant des conteneurs Docker qui permettent de le déployer rapidement et de passer à l'échelle industrielle. Elle fournit également le support de sa solution pour résoudre les incidents, déployer sans délai avec une installation automatisée et notifier des prochaines mises à jour.



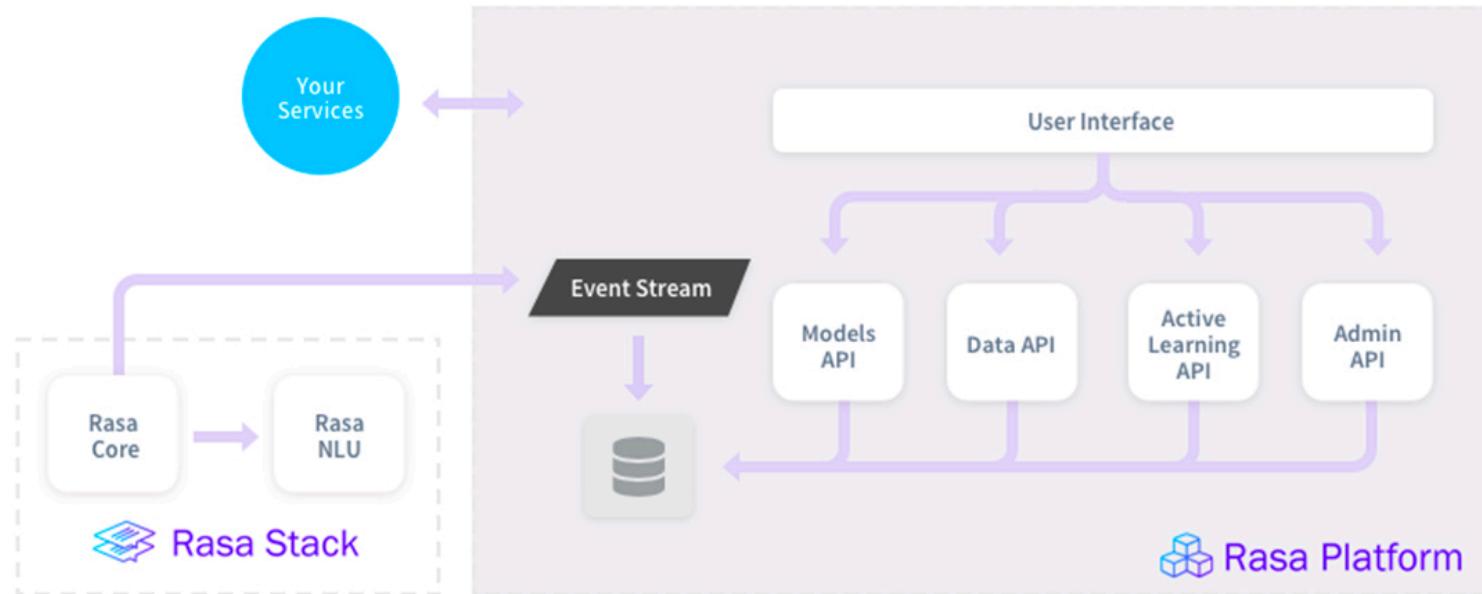
Plateforme Premium de RASA



- **Models API**: cet outil permet de tester un nouveau modèle et de comprendre quand il échoue. Il fournit aussi la possibilité d'update ou rollback du modèle via l'interface d'administration ou via une API.



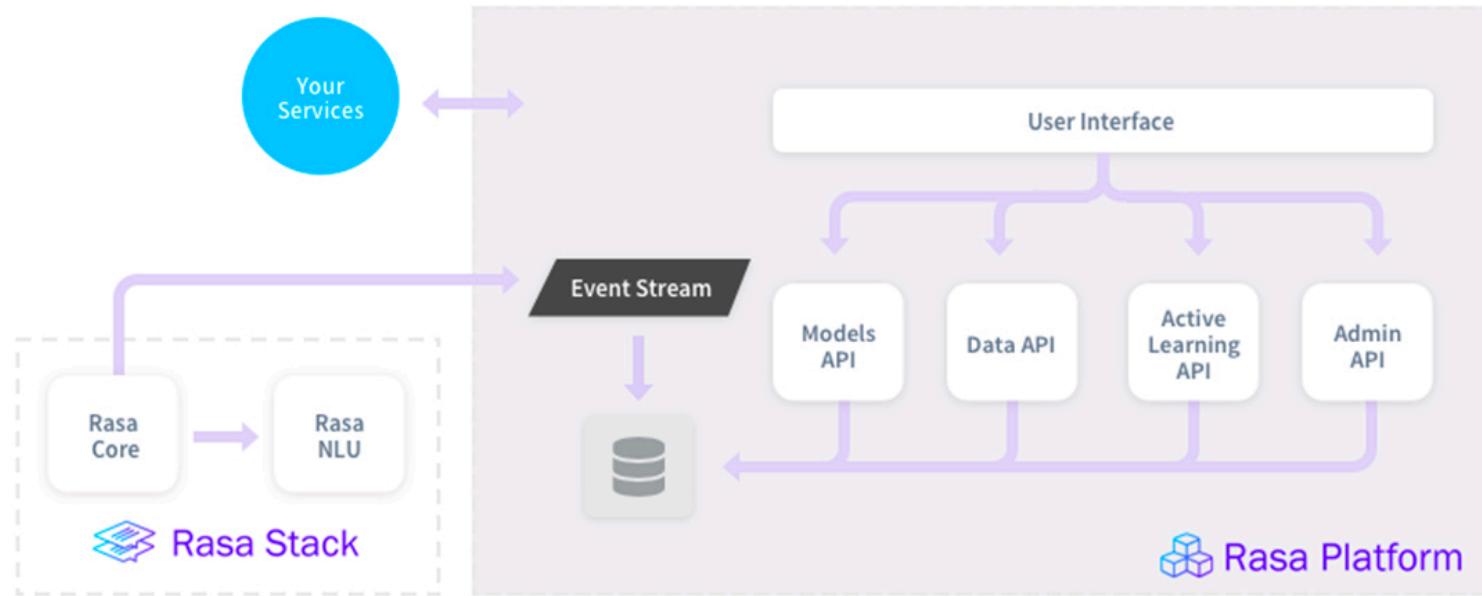
Plateforme Premium de RASA



- **Active learning API** : permet l'analyse les logs de tous les utilisateurs, l'aide au débogage et fournit des améliorations du modèle grâce à la suggestion de dialogue. Il est aussi possible de tester le Chatbot dans l'interface d'administration.



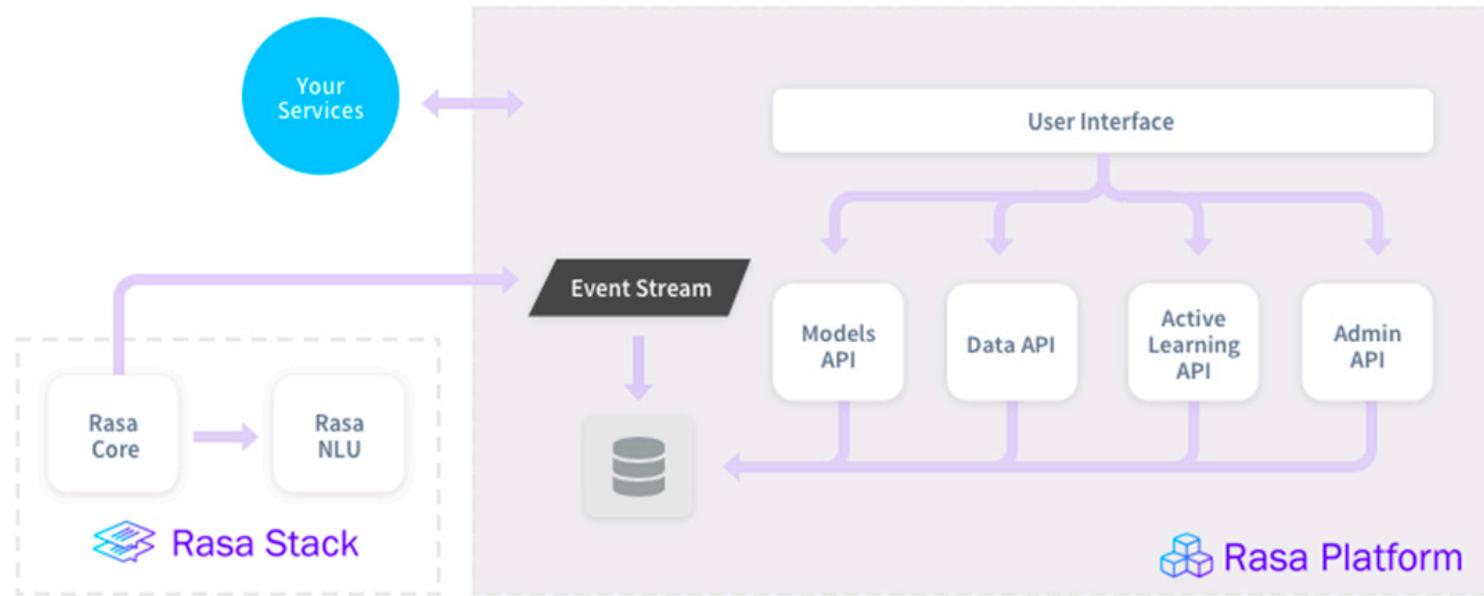
Plateforme Premium de RASA



- **Data API** : cet outil de data versioning permet de contrôler la qualité et l'accès aux données. Il assure par ailleurs l'intégrité du backend.

+

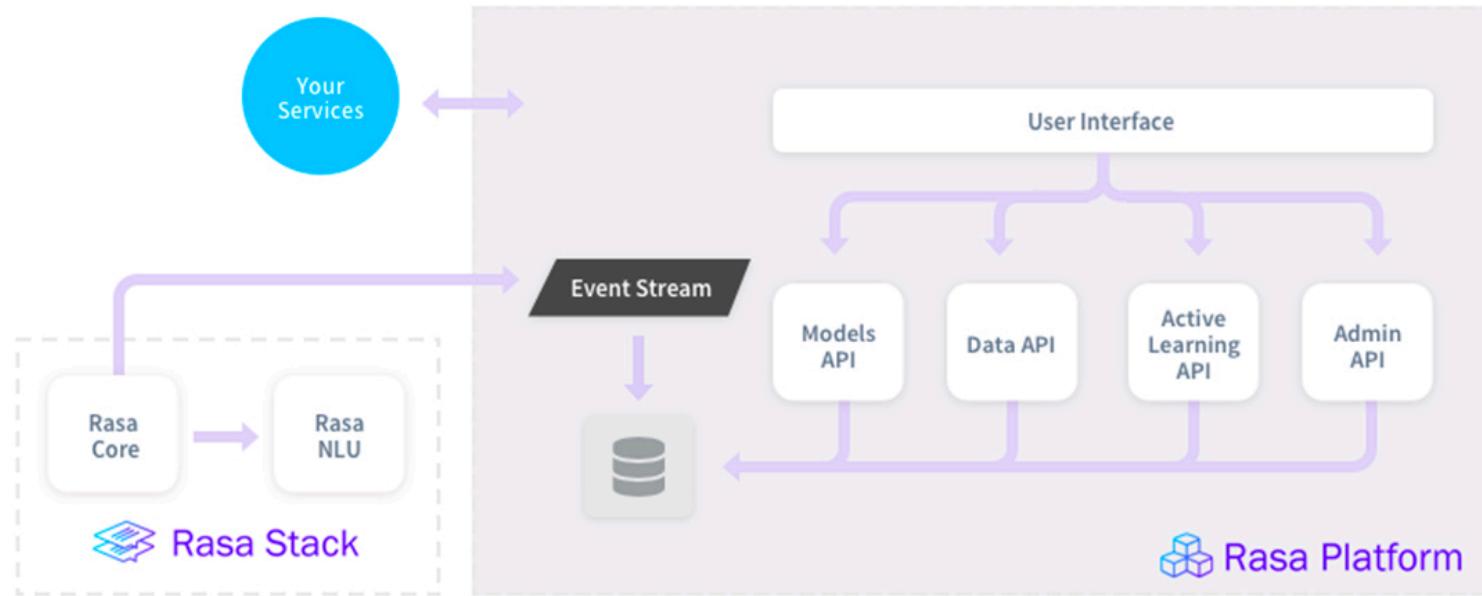
Plateforme Open Source de RASA



- **RASA-NLU** qui gère la partie compréhension du langage de l'agent



Plateforme Open Source de RASA



- **RASA-CORE** qui s'occupe de la partie stockage d'information, du choix de la réponse et de la connexion avec l'interface de discussion.



NLP et théories de base

- Le cycle de vie d'un bot consiste à attendre des messages des utilisateurs, traiter les informations dans ces messages et poster une réponse à chacun d'entre eux.
- Comme vous devez vous en douter, le plus gros défis dans la construction d'un chatbot reste la compréhension des messages en langage naturel envoyés par les utilisateurs.
- Pour y arriver, on va devoir utiliser des algorithmes de **classification automatique** et de **reconnaissance d'entités nommés**.



NLP et théories de base

- Dans le premier cas, on identifie l'**intention** de chaque message envoyé par l'utilisateur. C'est à dire, on range chaque message dans un ensemble de types de message prédéfinis qui sert à orienter le bot vers le type de traitement à appliquer à chaque requête.
- Les algorithmes de reconnaissance d'entités nommés sont utilisés pour identifier dans un message textuel l'ensemble de **noms / entités** dans un document. Ces entités peuvent être utilisées comme des paramètres dans le traitement d'une requête.



NLP et théories de base : exemples!

- “Je veux réserver un hôtel à Zurich pour le prochain Noël”
- “Peux-tu me divertir en me racontant une blague ?”
- “Où en est ma demande de remboursement de l'ordinateur de ma fille ?”

Quels sont les intentions et entités de chaque demande?

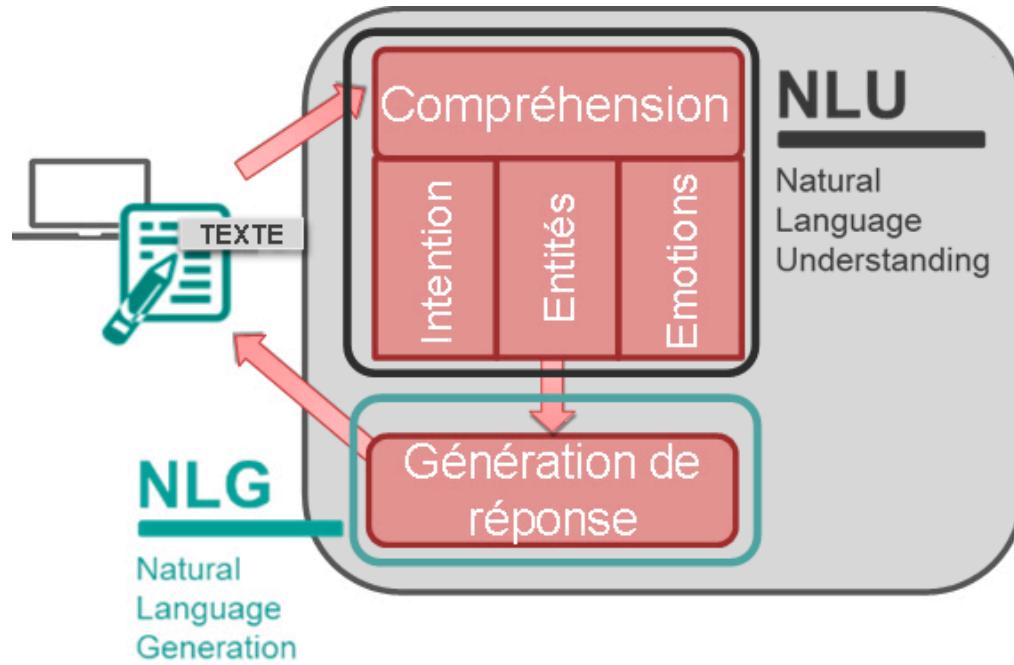


NLP et théories de base : exemples!

- “Je veux réserver un hôtel à Zurich pour le prochain Noël”
 - **Intention** : *Réservation d'une chambre d'hôtel*
 - **Entité(s)** : *Zurich, prochain Noël*
- “Peux-tu me divertir en me racontant une blague ?”
 - **Intention** : *Divertir*
 - **Entité(s)** : *Blague*
- “Où en est ma demande de remboursement de l'ordinateur de ma fille ?”
 - **Intention** : *Suivi de dossier*
 - **Entité(s)** : *remboursement, ordinateur, fille*

+

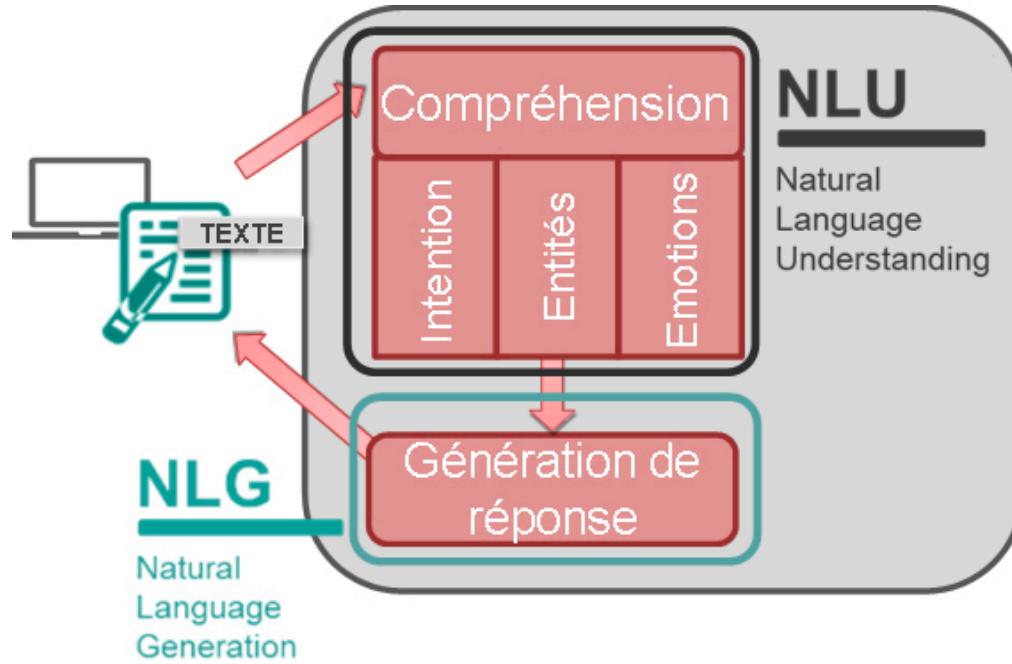
Mon premier chatbot avec RASA



- Sous RASA, le NLU est le système qui effectue un « mapping » de langage naturel en une représentation intention / entités, de la même façon qu'une classification.

+

Mon premier chatbot avec RASA



- Le NLG (Natural Language Generation) permet l'écriture automatique de texte de qualité similaire à celui écrit par un humain à partir de données structurées. Nous n'irons pas plus loin sur ce point, car ce composant ne fait pas partie de la solution RASA.



Mon premier chatbot avec RASA : d'autres concepts !

- Les **slots** agissent comme un dépôt de valeurs importantes, pouvant être utilisés pour stocker les informations fournies par l'utilisateur (par exemple, leur ville de résidence) ainsi que les informations recueillies sur le monde extérieur (par exemple, le résultat d'une requête de base de données). La plupart du temps, les slots ont une influence sur la progression de la discussion.

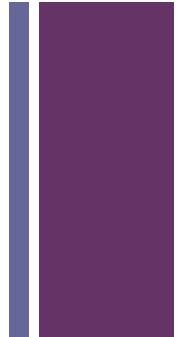


Mon premier chatbot avec RASA : d'autres concepts !

- Les **actions** permettent de remplir les slots et de choisir la réponse qui sera envoyée à l'utilisateur. Il y en a 3 :
 - **Default** : listen, restart et fallback, pour respectivement se mettre à l'écoute, redémarrer une conversation et ignorer le dernier message entrant.
 - **Basic** : sert à renvoyer un message à l'utilisateur à partir de l'intention
 - **Custom** : faire des opérations pluscomplexes à partir de l'intention et des slots avec des APIs & co.



Mon premier chatbot avec RASA



- Les **stories** qui sont des exemples de discussions créées à partir de l'enchaînement d'intentions de l'utilisateur et d'actions prises par l'agent selon les intentions reçues. Ces stories constituent la base d'apprentissage pour le modèle de discussion.



Mon premier chatbot : un peu de pratique ?

- Le principal travail va être de créer des fichiers de données d'entraînement. Ce sont des phrases types que l'utilisateur pourrait dire et dans lesquelles l'intention et les entités sont précisées. Il y aura également des fichiers de configuration pour RASA.
 - **nlu_config.yml** : de base, RASA est brut de décoffrage car il ne comprend rien ! Il faudra ajouter des librairies de prétraitements comme spaCy ou tensorflow. C'est dans ce fichier qu'on définit cela.
 - **nlu.md** : C'est dans ce fichier qu'on définit les intentions attendues et comment les reconnaître.
 - ...



Mon premier chatbot : un peu de pratique ?

- Le principal travail va être de créer des fichiers de données d'entraînement. Ce sont des phrases types que l'utilisateur pourrait dire et dans lesquelles l'intention et les entités sont précisées. Il y aura également des fichiers de configuration pour RASA.
 - ...
 - **stories.md** : Il faut maintenant indiquer au chatbot comment manier les entités et répondre à l'utilisateur grâce aux Stories.
 - **domain.yml** : Ce fichier permet de lier toutes les briques !



Mon premier chatbot : Installation

- Etape 1 : Installer Python 3.6 & mettre à jour PIP (“***pip3.6 install --upgrade pip***”)
- Etape 2 : Installer la Rasa Stack sur votre ordinateur avec “***pip3.6 install -U rasa_core***”
- Etape 3 : Installer la brique NLU sur votre ordinateur avec SpaCy (“***pip3.6 install rasa_nlu[spacy]***”) ou Tensorflow (“***pip3.6 install rasa_nlu[tensorflow]***”)
- C'est bon !



+

Mon premier chatbot : Création de nlu.md et nlu_config.yml

- **nlu_config.yml** : de base, RASA est brut de décoffrage car il ne comprend rien ! Il faudra ajouter des librairies de prétraitements comme spaCy ou tensorflow. C'est dans ce fichier qu'on définit cela.
- **nlu.md** : C'est dans ce fichier qu'on définit les intentions attendues et comment les reconnaître.





Mon premier chatbot : Création de nlu.md et nlu_config.yml

```
## intent:greet
- hey
- hello
- hi
- good morning
- good evening
- hey there
```

```
## intent:goodbye
- bye
- goodbye
- see you around
- see you later
```

```
## intent:good_affirm
```

nlu.md

```
language: "en"
pipeline: spacy_sklearn
```

nlu_config.yml



Mon premier chatbot : Entrainement du modèle NLU

- Installer les fichiers de langues de SpaCy avec : “***python3.6 -m spacy download en***”
- Appeler RASA avec la commande : “***python3.6 -m rasa_nlu.train -c nlu_config.yml --data nlu.md -o models --fixed_model_name nlu --project current --verbose***”

```
[MacWork:FormationChatbots khaalel$ python3.6 -m rasa_nlu.train -c nlu_config.yml --data nlu.md -o models --fixed_model_name nlu --project current --verbose
2019-04-15 19:17:23 INFO     rasa_nlu.utils.spacy_utils - Trying to load spacy model with name 'en'
2019-04-15 19:17:24 INFO     rasa_nlu.components - Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
2019-04-15 19:17:24 INFO     rasa_nlu.training_data.loading - Training data format of nlu.md is md
2019-04-15 19:17:24 INFO     rasa_nlu.training_data.training_data - Training data stats:
    - intent examples: 39 (6 distinct intents)
    - Found intents: 'goodbye', 'mood_unhappy', 'greet', 'mood_denry', 'mood_affirm', 'mood_great'
    - entity examples: 0 (0 distinct entities)
    - found entities:

2019-04-15 19:17:24 INFO     rasa_nlu.model - Starting to train component nlp_spacy
2019-04-15 19:17:24 INFO     rasa_nlu.model - Finished training component.
2019-04-15 19:17:24 INFO     rasa_nlu.model - Starting to train component tokenizer_spacy
2019-04-15 19:17:24 INFO     rasa_nlu.model - Finished training component.
2019-04-15 19:17:24 INFO     rasa_nlu.model - Starting to train component intent_featurizer_spacy
2019-04-15 19:17:24 INFO     rasa_nlu.model - Finished training component.
2019-04-15 19:17:24 INFO     rasa_nlu.model - Starting to train component intent_entity_featurizer_regex
2019-04-15 19:17:24 INFO     rasa_nlu.model - Finished training component.
2019-04-15 19:17:24 INFO     rasa_nlu.model - Starting to train component ner_crfs
2019-04-15 19:17:24 INFO     rasa_nlu.model - Finished training component.
```



Mon premier chatbot :

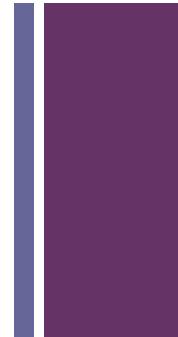
Tester le nouveau modèle

```
1  from rasa_nlu.model import Metadata, Interpreter
2  import json
3
4  def pprint(o):
5      # small helper to make dict dumps a bit prettier
6      print(json.dumps(o, indent=2))
7
8  interpreter = Interpreter.load('./models/current/nlu')
9  pprint(interpreter.parse(u"Hello"))
```

```
MacWork:FormationChatbots khaale1$ python3.6 step1-test-model.py
{
    "intent": {
        "name": "greet",
        "confidence": 0.40326015630626455
    },
    "entities": [],
    "intent_ranking": [
        {
            "name": "greet",
            "confidence": 0.40326015630626455
        },
        {
            "name": "mood_affirm",
            "confidence": 0.2565363722851674
        },
        {
            "name": "mood_greet",
            "confidence": 0.1468636277148325
        }
    ]
}
```



Mon premier chatbot : Apprendre au chatbot à répondre



- **stories.md** : Il faut maintenant indiquer au chatbot comment manier les entités et répondre à l'utilisateur grâce aux Stories.
- **domain.yml** : Ce fichier permet de lier toutes les briques (entités, actions, stories, etc.) !





Mon premier chatbot :

Apprendre au chatbot à répondre

```
1  ## happy path
2  * greet
3  | - utter_greet
4  * mood_great
5  | - utter_happy
6
7  ## sad path 1
8  * greet
9  | - utter_greet
10 * mood_unhappy
11 | - utter_cheer_up
12 | - utter_did_that_help
13 * mood_affirm
14 | - utter_happy
```

stories.md

```
1  intents:
2  | - greet
3  | - goodbye
4  | - mood_affirm
5  | - mood_DENY
6  | - mood_great
7  | - mood_unhappy
8
9  actions:
10 | - utter_greet
11 | - utter_cheer_up
12 | - utter_did_that_help
13 | - utter_happy
14 | - utter_goodbye
15
16 templates:
17 | utter_greet:
```

domain.yml



Mon premier chatbot : Entraîner le chatbot à répondre

- Appeler RASA avec la commande : “**python3.6 -m rasa_core.train -d domain.yml -s stories.md -o models/current/dialogue**”

```
MacWork:FormationChatbots khaalel$ python3.6 -m rasa_core.train -d domain.yml -s stories.md -o models/current/dialogue
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 3910.77it/s, # trackers=1]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 1753.65it/s, # trackers=4]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 533.41it/s, # trackers=12]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 504.35it/s, # trackers=12]
2019-04-15 19:45:07.973648: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that t
his TensorFlow binary was not compiled to use: AVX2 FMA
-----  

Layer (type)          Output Shape         Param #
-----  

masking (Masking)    (None, 5, 18)        0  

-----  

lstm (LSTM)          (None, 32)           6528  

-----  

dense (Dense)        (None, 12)            396  

-----  

activation (Activation) (None, 12)        0  

-----  

Total params: 6,924
Trainable params: 6,924
Non-trainable params: 0
```



Mon premier chatbot :

Parlons à notre chatbot ;)

- Lançons le chat en ligne de commande avec la commande :

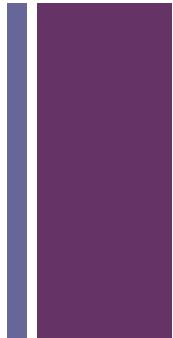
"python3.6 -m rasa_core.run -d models/current/dialogue -u models/current/nlu"

```
MacWork:FormationChatbots khaalel$ python3.6 -m rasa_core.run -d models/current/dialogue -u models/current/nlu
2019-04-15 19:49:04 INFO    root  - Rasa process starting
2019-04-15 19:49:05 INFO    rasa_nlu.components  - Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
2019-04-15 19:49:10.577312: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that t
his TensorFlow binary was not compiled to use: AVX2 FMA
2019-04-15 19:49:11 INFO    root  - Rasa Core server is up and running on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
Your input -> Hello
Hey! How are you?
127.0.0.1 - - [2019-04-15 19:49:26] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 185 0.132385
Your input -> Fine
127.0.0.1 - - [2019-04-15 19:49:32] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 122 0.016339
Your input -> perfect
Great carry on!
127.0.0.1 - - [2019-04-15 19:50:07] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 183 0.015132
Your input -> □
```

+

Mon premier chatbot :

Parlons à notre chatbot ;)



Ca marche !!!!



Mon premier chatbot : Créons des actions plus complexes !

- Ajoutons d'autres librairies avec : « **pip3.6 install -r requirements.txt** »
- Créons un fichier **actions.py** qui va appeler une API générant des phrases de Chuck Norris ;)

```
15  class ActionJoke(Action):  
16      def name(self):  
17          # define the name of the action which can then be included in training stories  
18          return "action_joke"  
19  
20      def run(self, dispatcher, tracker, domain):  
21          # what your action should do  
22          request = json.loads(  
23              requests.get("https://api.chucknorris.io/jokes/random").text  
24          ) # make an api call  
25          joke = request["value"] # extract a joke from returned json response  
26          dispatcher.utter_message(joke) # send the message back to the user  
27          return []
```



Mon premier chatbot : Créons des actions plus complexes !

- Ajoutons la nouvelle action à stories.md et domain.yml !
- Appelons RASA avec la commande : “**python3.6 -m rasa_core.train -d domain.yml -s stories.md -o models/current/dialogue**”

```
MacWork:FormationChatbots khaalel$ python3.6 -m rasa_core.train -d domain.yml -s stories.md -o models/current/dialogue
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 3910.77it/s, # trackers=1]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 1753.65it/s, # trackers=4]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 533.41it/s, # trackers=12]
Processed Story Blocks: 100%|██████████| 4/4 [00:00<00:00, 504.35it/s, # trackers=12]
2019-04-15 19:45:07.973648: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that t
his TensorFlow binary was not compiled to use: AVX2 FMA
-----
Layer (type)          Output Shape         Param #
=====
masking (Masking)    (None, 5, 18)        0
-----
lstm (LSTM)          (None, 32)           6528
-----
dense (Dense)        (None, 12)            396
-----
activation (Activation) (None, 12)        0
=====
Total params: 6,924
Trainable params: 6,924
Non-trainable params: 0
```



Mon premier chatbot : Créons des actions plus complexes !

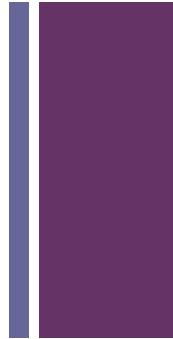
- Lançons le chat en ligne de commande avec la commande :
“python3.6 -m rasa_core.run -d models/current/dialogue -u models/current/nlu --endpoints endpoints.yml”

```
[MacWork:FormationChatbots khaalel$ python3.6 -m rasa_core.run -d models/current/dialogue -u models/current/nlu --endpoints endpoints.yml
/Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/h5py/_init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
    from ._conv import register_converters as _register_converters
2019-04-15 20:37:45 INFO      root  - Rasa process starting
2019-04-15 20:37:46 INFO      rasa_nlu.components  - Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
2019-04-15 20:37:51.451961: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2019-04-15 20:37:52 INFO      root  - Rasa Core server is up and running on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
[Your input -> Hi
Hey! How are you?
127.0.0.1 - - [2019-04-15 20:38:33] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 185 0.131695
[Your input -> perfect
Great carry on!
Bye
127.0.0.1 - - [2019-04-15 20:38:35] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 232 0.018108
[Your input -> Bye
Chuck Norris' childhood macaroni art is on display at the Smithsonian.
Bye
127.0.0.1 - - [2019-04-15 20:38:40] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 287 1.357904
[Your input -> ]
```



Mon premier chatbot :

Prendre en compte les entités



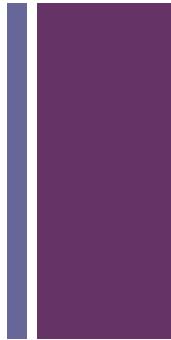
- Modifier domain.yml, stories.md et nlu.md pour prendre en compte les entités
- Relancer toutes les RASA de création des modèles :
“python3.6 -m rasa_nlu.train -c nlu_config.yml --data nlu.md -o models --fixed_model_name nlu --project current --verbose” puis “python3.6 -m rasa_core.train -d domain.yml -s stories.md -o models/current/dialogue”
- Relancer le custom action puis la cmd avec :
“python3.6 -m rasa_core_sdk.endpoint --actions actions” et “python3.6 -m rasa_core.run -d models/current/dialogue -u models/current/nlu --endpoints endpoints.yml”.



Mon premier chatbot :

Prendre en compte les entités

```
FormationChatbots — Python - Python -m rasa_core.run -d models/current/dialogue -u models/current/nlu --endpoints endpoints.yml -t actions actions
~ — -bash ...se/Titanium — -bash ...lrelaydev:~ — -bash ...oints endpoints.yml ...t --actions actions +
/ Library/Frameworks/Python.framework/Versions/3.6/lib/python3.6/site-packages/h5py/_init__.py:36: FutureWarning: Conversion of the second argument of issubdtype from `float` to `np.floating` is deprecated. In future, it will be treated as `np.float64 == np.dtype(float).type`.
from ._conv import register_converters as _register_converters
2019-04-15 21:13:44 INFO      root - Rasa process starting
2019-04-15 21:13:45 INFO      rasa_nlu.components - Added 'nlp_spacy' to component cache. Key 'nlp_spacy-en'.
2019-04-15 21:13:49.325649: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2019-04-15 21:13:50 INFO      root - Rasa Core server is up and running on http://localhost:5005
Bot loaded. Type a message and press enter (use '/stop' to exit):
[Your input -> I am Gigi
Nice to you meet you gigi. How can I help?
127.0.0.1 -- [2019-04-15 21:16:06] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 210 0.131534
[Your input -> Hi
Hey! How are you?
127.0.0.1 -- [2019-04-15 21:16:10] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 185 0.015588
[Your input -> perfect
Great carry on!
Bye
127.0.0.1 -- [2019-04-15 21:16:13] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 232 0.018592
[Your input -> My name is Gigi
Nice to you meet you gigi. How can I help?
127.0.0.1 -- [2019-04-15 21:16:22] "POST /webhooks/rest/webhook?stream=true&token= HTTP/1.1" 200 210 0.017997
Your input -> 
```



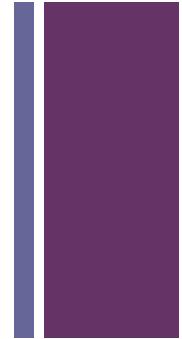
+

+

Projet : Créer un chatbot



Enoncé du projet

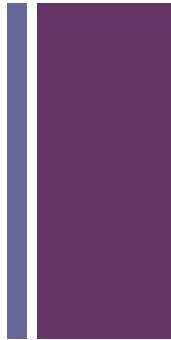


Créer un chatbot qui permet de trouver un festival, un événement ou une soirée (voir des informations sur un artiste ou un lieu de concert si vous avez le temps) en exploitant des API REST et RASA.

Vous pouvez exploiter les APIs suivantes ainsi que toutes APIs que vous pourrez trouver sur Internet ou Mashape (<https://market.mashape.com/>) :

- Eventbrite (<https://www.eventbrite.com/platform/api#/introduction>),
- Flickr (<https://www.flickr.com/services/api/>),
- LastFM (<https://www.last.fm/api>),
- Google Places,
- Wikipedia API,
- etc.

Le projet sera à livrer en ZIP à l'adresse pro@gilbertnzeka.com lors du dernier cours du 24/05/2019.



+



Des questions? Je reste joignable sur nzeka@nachosmedia.com !