



CogniTALK





Agence
DATASERVICES



L'IA au service de l'emploi



Intro

Sommaire du document



Agence
DATASERVICES



pôle emploi

- 1
- 2
- 3
- 4
- 5
- 6
- 7

Agence Data Service

Projet SCANLAB

Projet DAC

Word_n_Fun

Projet MAIL

Word2Job

Stark



AGENCE DATA SERVICE

Une agence
interne à Pôle Emploi
dédiée à l'IA
et la datascience

Comment tirer le meilleur de l'IA
et de la data pour servir le retour
à l'emploi et l'attractivité des
entreprises

1
Nous
sommes...

2
Historique

3
Les piliers
de l'IA
à la DSI



/ Nous sommes / **Agiles et orientés innovation**

25 produits
développés
par l'agence

3 sites
Géographiques
(Nantes, Aix en Provence &
Montreuil)

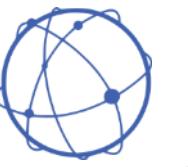
60 collaborateurs
(products managers, PO,
datascientists,
développeurs...)



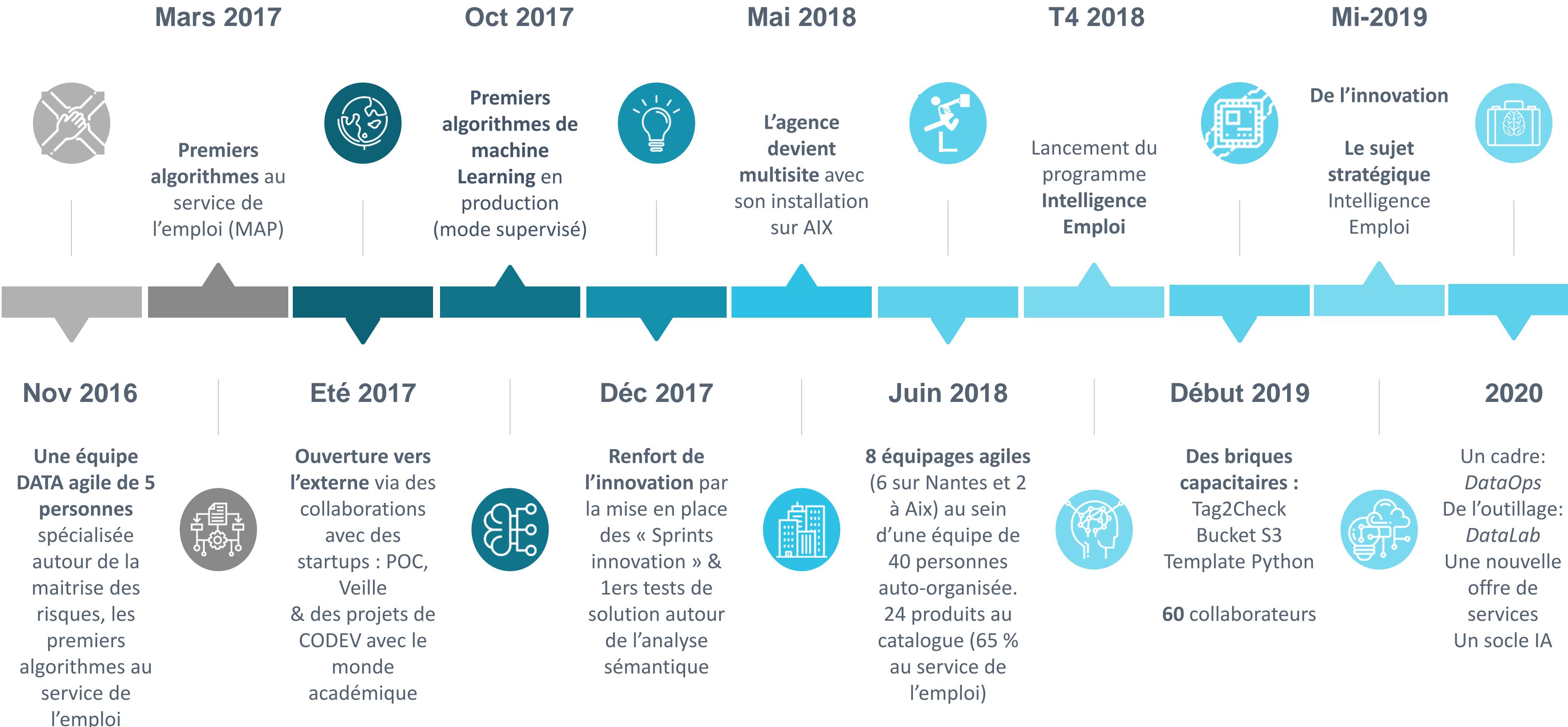
**Rotation, auto-
affectation des
équipes, autonomie**

**Choix de la démarche
Agile par les équipes**
(kanban, scrum,
scrumban, lean startup...)

**¼ du temps consacré
à l'innovation**



Une usine en forte croissance depuis 2016





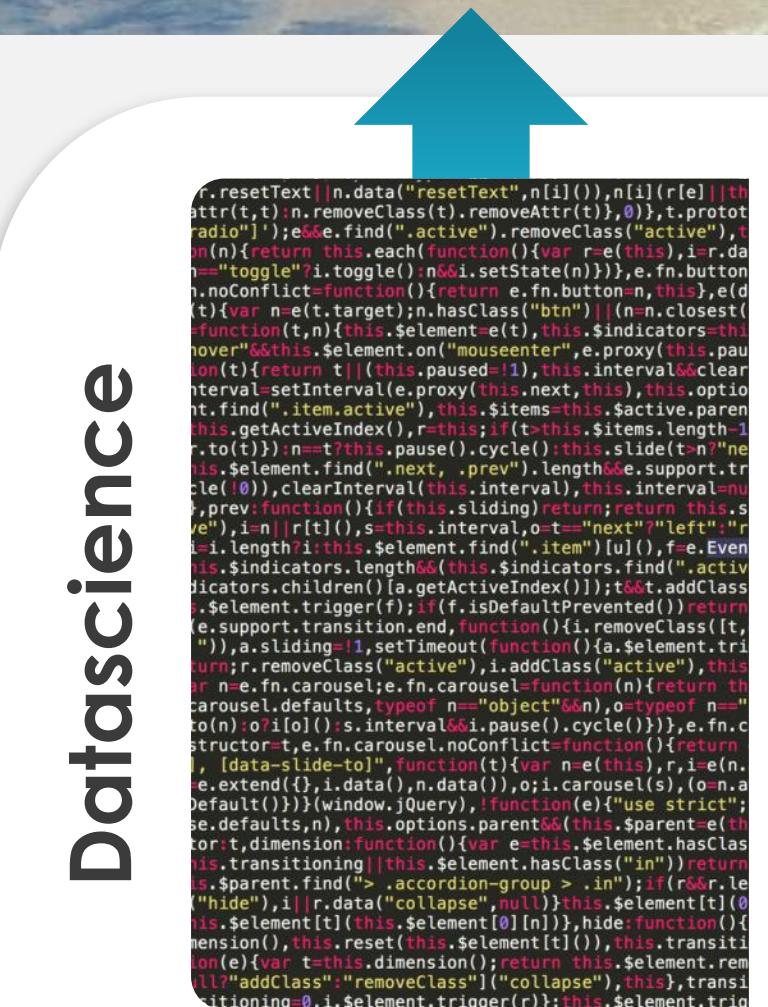
Les piliers de l'IA à la DSI de Pôle emploi et le positionnement de l'ADS

Nouveaux services à base d'IA



Données

- Datalake Pôle Emploi



Data science

- Des compétences « data » pour exploiter les volumes de données



Open innovation

- Monde académique
 - Startups
 - Communauté Open source
 - Organismes publics

Offre de service l'agence





Computer Projet Scanlab Vision

1
Données

2
Approche

3
Modélisation

4
Industrialisation

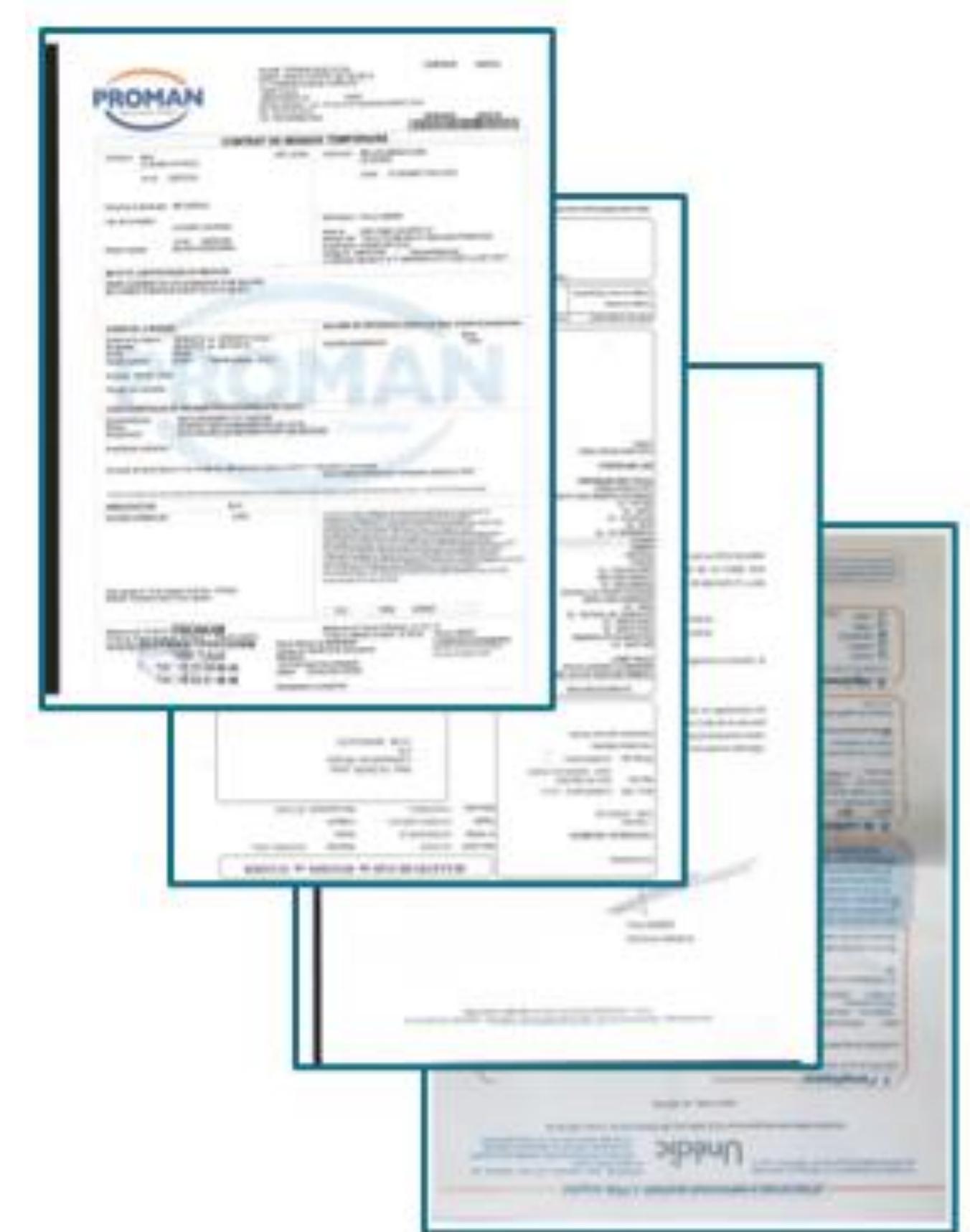
Problématique :

- De nombreuses pièces jointes sont uploadées sur le site de PE
- Nécessaire de découper les dossiers complets
- Objectif: faciliter la gestion de ses documents sans avoir à découper les documents manuellement



Données

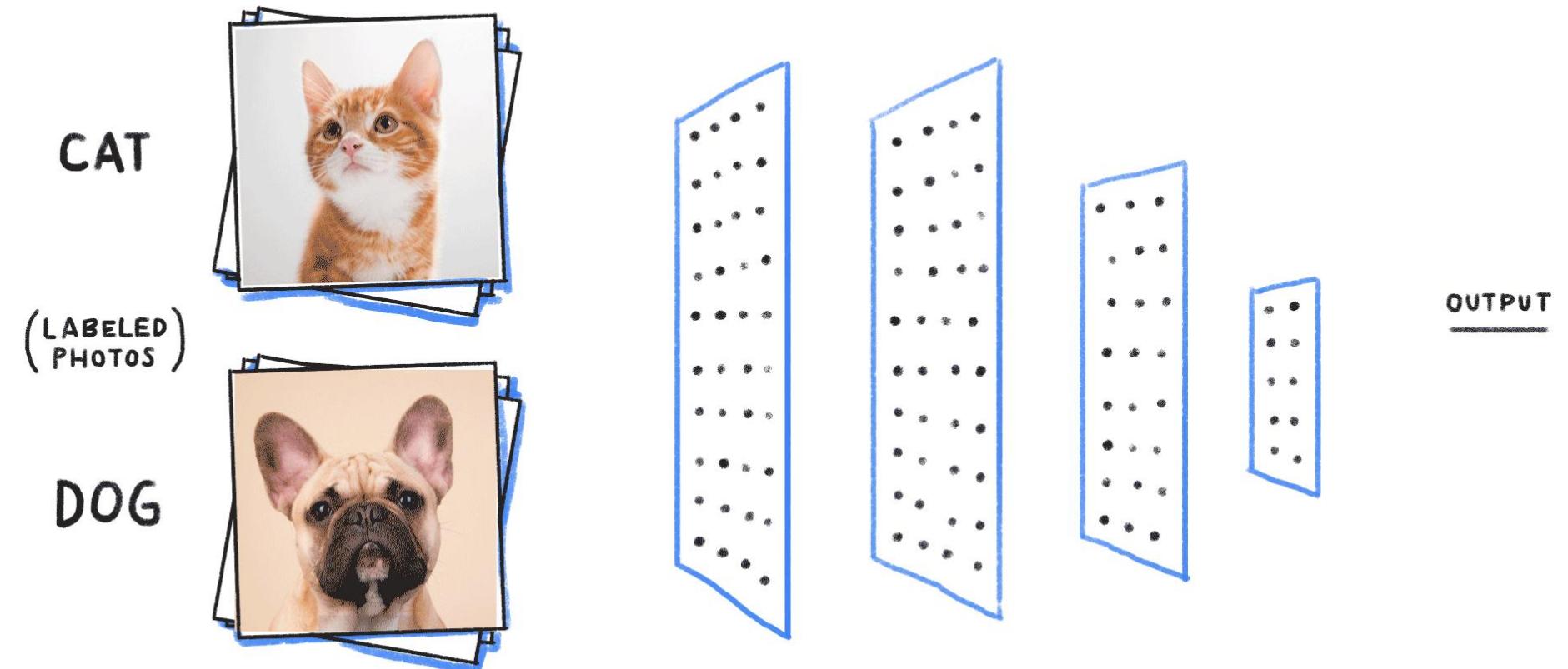
- 22 000 documents PDF
 - => 37 000 images
 - Des attestations employeur, des bulletins de salaire, des certificats de travail, des contrats de travail et autres
- Des photos, mal cadrées, pas normalisées, etc...





Approche

- Problème de classification **supervisée multi-classe**
 - 8 classes : Attestation employeur (p1 et p2+); Bulletin de salaire (p1 et p2+); Certificat de travail; Contrat de travail; Page blanche; Divers
- **Pré-processing**
 - Découpage des PDF en images
 - Suppression des marges
 - Resizing





Approche

- Outil maison pour le labélisation manuel

Attestation employeur (p.1)

Attestation employeur (p.2+)

Bulletin de salaire (p.1)

Bulletin de salaire (p.2+)

Certificat de travail

Contrat de travail

Page blanche

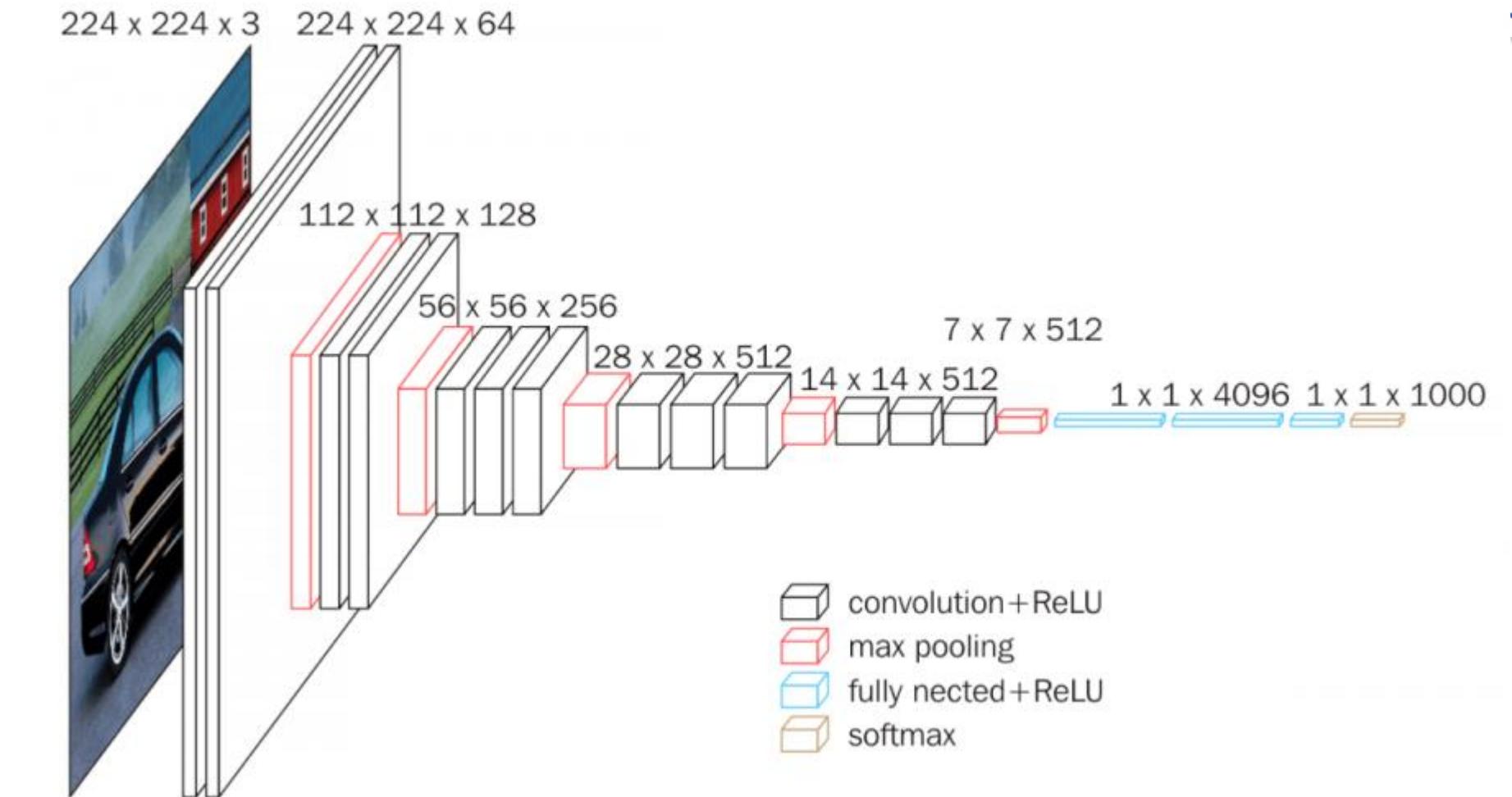
Divers

?

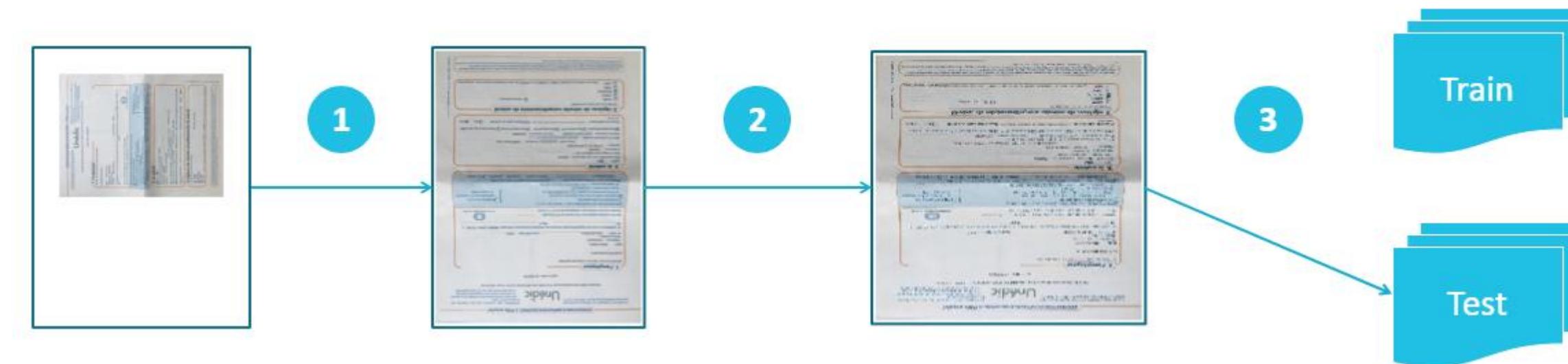


Modélisation

1. Un **CNN** classique maison



2. **Transfer learning** en utilisant ResNet & VGG16 + fine tuning



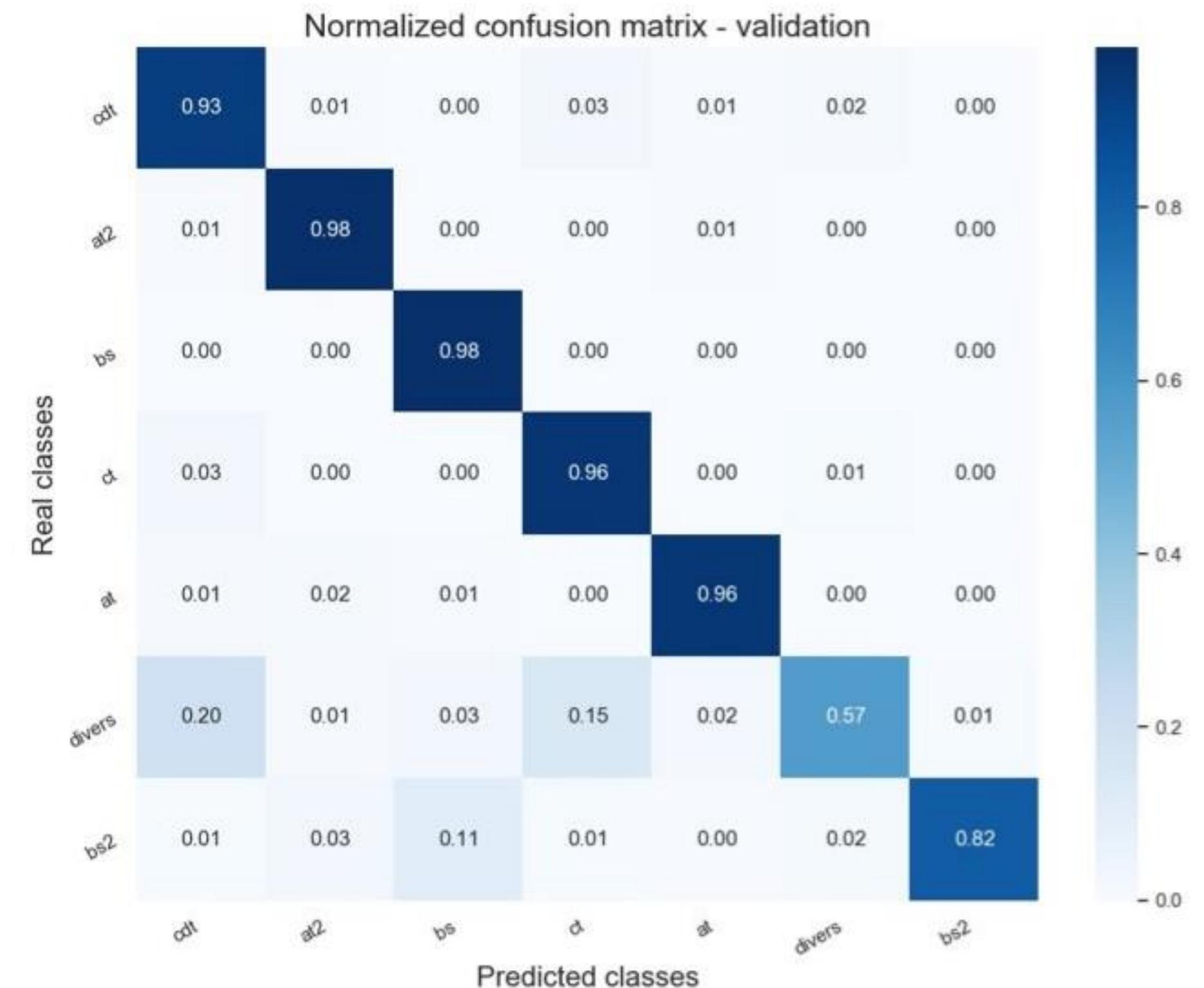


Modélisation

1. Un **CNN** classique maison

2. **Transfer learning** en utilisant ResNet & VGG16 + fine tuning

- Globalement > 90% accuracy
- Le modèle maison > transfert learning





Industrialisation



- V3.6+
- Le langage par défaut à l'ADS
- Idéal pour le prototypage rapide de nos modèles



- Une des références en matière de graphe computationnel
- Idéal pour l'implémentation de DL
- A des APIs pour de nombreux langages



Pour stocker les modèles finaux



Pour pouvoir faire de l'inférence à partir du modèle depuis nos applications java via tensorflow java
(ou alors en Python + Falcon)



Cible



Recommandation Projet DAC Engine



1 Objectifs & Données

2 Approche

3 Modélisation

4 Industrialisation

Problématique :

- Refonte du système d'actualisation mensuel de situation
 - Permet aux demandeurs d'emploi de compléter leur profil (nouvelles compétences, CV ...)
 - A la fin de l'exercice d'actualisation : 3 suggestions sont présentées au DE (participer à un atelier; suivre une formation; compléter son CV; etc...)



Objectifs & Données

Objectif :

- Offrir des **recommandations personnalisées** et pertinentes aux DEs
- Découvrir/faire **émerger** des « chemins » dans les suggestions & profils qui facilitent **le retour à l'emploi**
- Ne pas commettre « d'impairs » (ie en proposant une formation avec une composante physique à une personne handicapée moteur)

Les données :

- Toutes les données du SI PE relatives aux DEs (formations, métier recherché, situation socio-pro, localisation, CV, etc...)
- Données de navigation (accès ou non aux services)
- Données d'actualisation





Approche

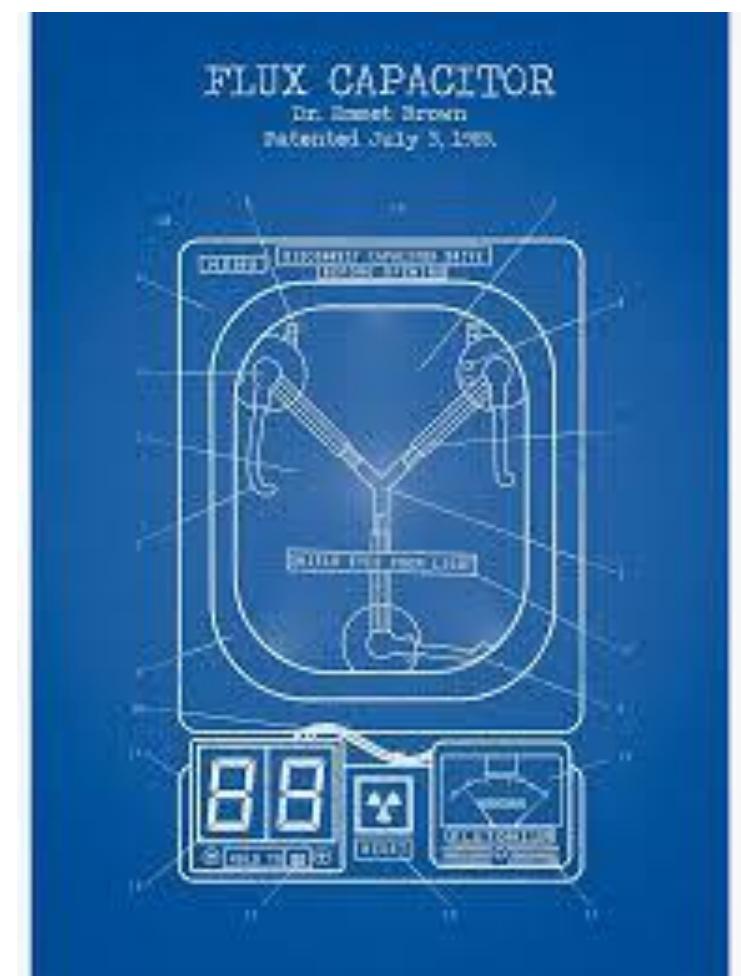
Difficultés

- **Coldstart** (item + individus) : données non existantes
- Problème de la **confiance** vis-à-vis du métier
- Besoin de rester **cohérent** dès le départ vis-à-vis des DEs

Notre choix

- **Système expert + pondérations** des suggestions

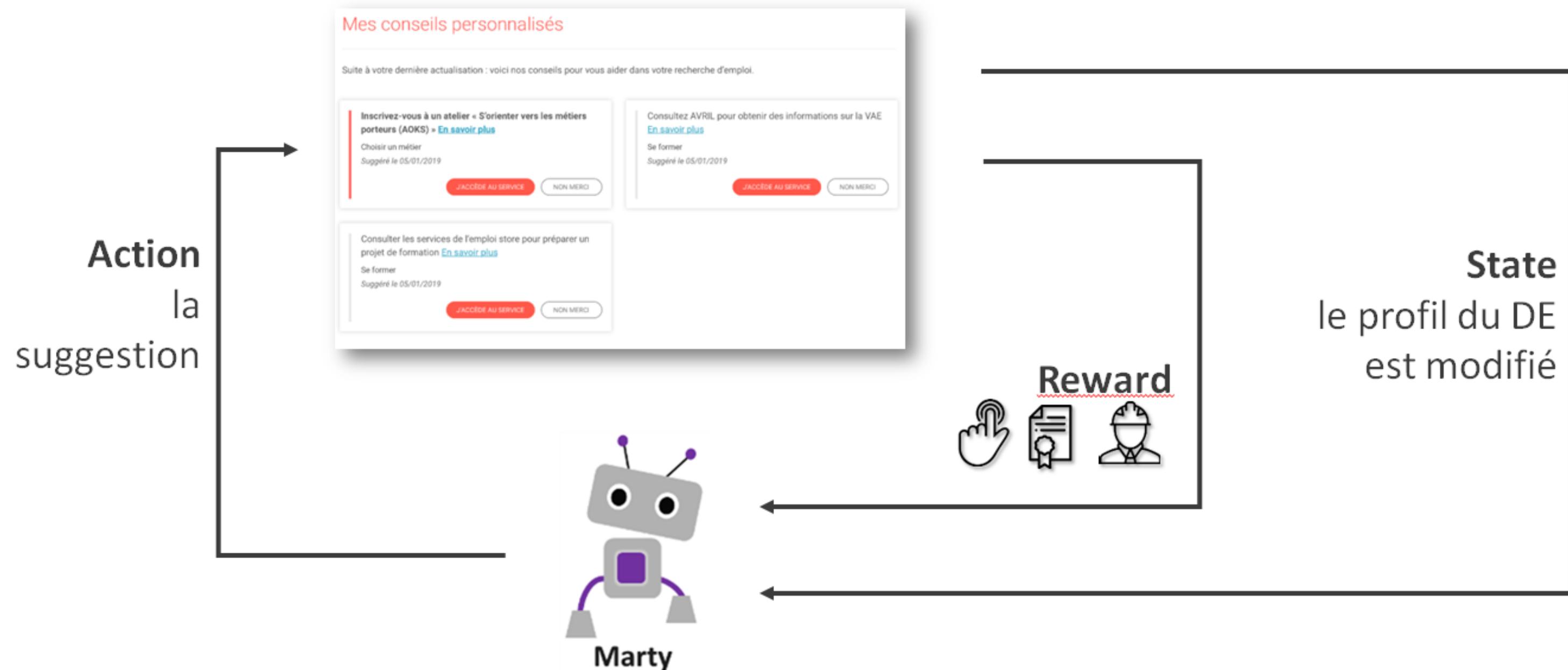
- Le système expert permet de **guider** les suggestions et de lancer le système
- Les pondérations permettent d'essayer d'optimiser les retours





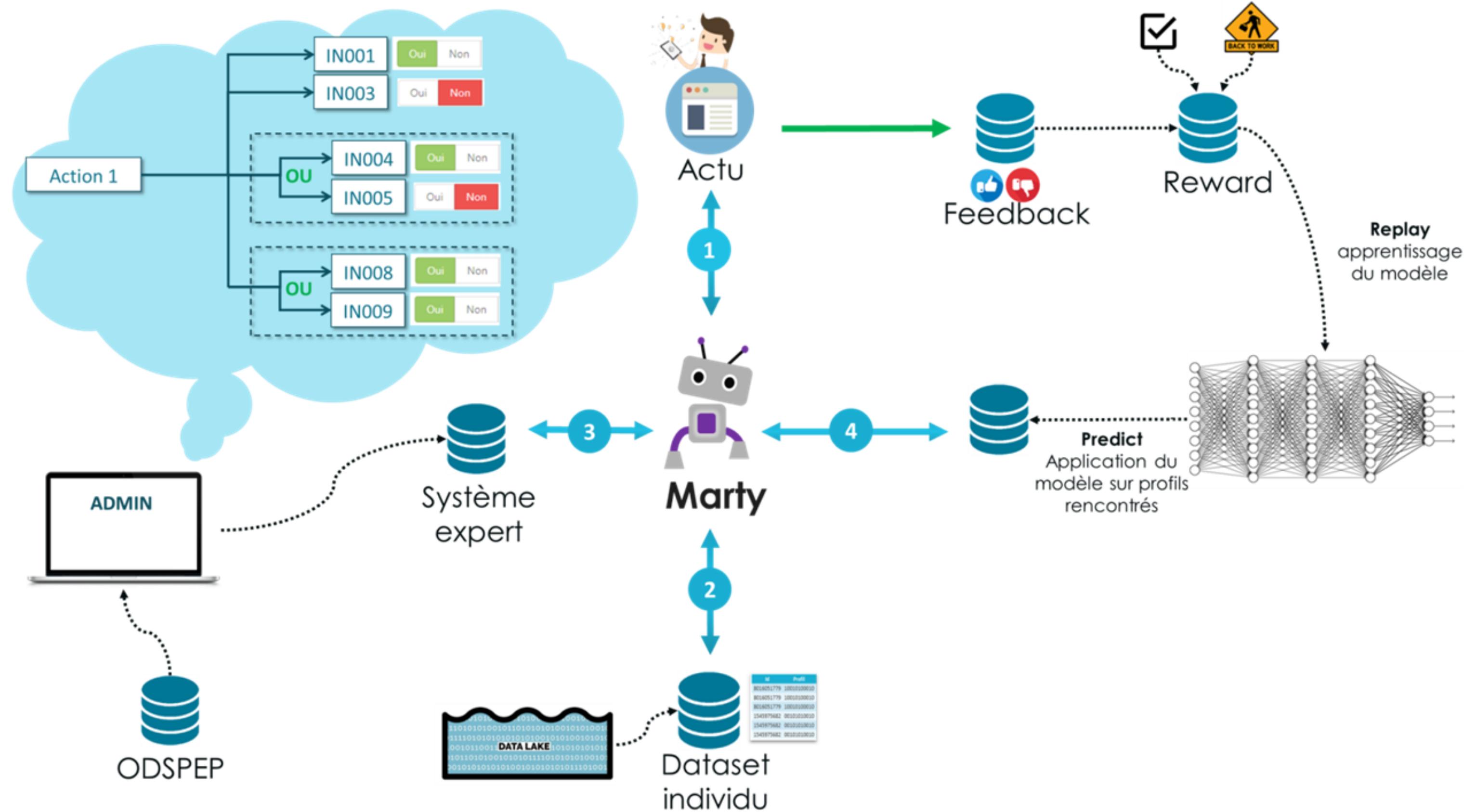
Modélisation

Par renforcement : Batch Deep Q Learning





Modélisation





Modélisation

ETAPE 1

Clic?



La navigation sur une suggestion permet de savoir si elle a un intérêt pour lui

Les clics utilisés sont :

Le clic sur « en savoir plus »

Le clic sur « j'accède au service »

Le clic sur « Non merci » (et du motif choisi)

Chaque clic a un poids permettant de quantifier son importance dans l'apprentissage (REWARD)

ETAPE 2

Action réalisée?



Une action est suggérée et le DE la réalise? On peut donc penser que la suggestion était bonne

Pour savoir si un DE réalise une action suggérée on :

Utilise les données XITI pour tracer la navigation sur les sites de pe.fr

Utilise les données du SI pour tracer les ateliers et prestations réalisés

Il n'est à l'heure actuelle pas possible de tracer la navigation sur l'emploi store

ETAPE 3

Retour à l'emploi?



Une action est suggérée , le DE la réalise, et retrouve un emploi? On peut peut-être penser que la suggestion était juste

Sujet en cours d'étude



Industrialisation



- V3.6+
- Le langage par défaut à l'ADS
- Idéal pour le prototypage rapide de nos modèles



- Permet d'encapsuler les applications python



- Pour générer les datasets, calculer les rewards
- Pour stocker les recommandations (hbase)



Pour servir les suggestions



Analyse Sémantique

Un sujet récurrent

- Beaucoup de données textuelles
 - Libres
 - Descriptifs d'offres
 - CVs
 - Mails
 - Contenus de formations, etc...
 - Structurées
 - Référentiels internes (ROME....)
 - Référentiels externes
- Globalement sous-utilisées dans les applications historiques



Analyse Sémantique

Brique sémantique

Words_n_fun

Repo Model

Use cases

Indus

Objectif

- Mutualiser au maximum le fruit des projets sémantiques
- Faciliter la phase exploratoire
- Permettre d'accélérer la production de nouveaux cas d'usages
- Rationnaliser la mise en production



Words_n_fun

Projet GIT

- Officiellement `pe_semantic`
- 100% **Python**
- Vocation à être **open-sourcé**
- Facilite et **unifie** les opérations de pré-processing (**primordial** pour de nombreuses analyses sémantiques)
 - Compatible **SKLearn** pipeline, **data agnostic**
- Permet de **mutualiser** plus facilement les différents travaux sémantiques de l'ADS

```
pipeline = ['remove_non_string', 'to_lower', 'remove_punct', 'remove_stopwords']
preprocessor = wnf.get_preprocessor(pipeline)
text = preprocessor.transform(text)
```

"@FollowSavvy I never found her :(. everytime I click on her twitter thing through your myspace..... it goes to some dude's page"

Texte brut

"I never found her . everytime I click on her twitter thing through your myspace..... it goes to some dude's page"

Texte nettoyé

[I', 'never', 'found', 'her', :, 'everytime', 'I', 'click', 'on', 'her', 'twitter', 'thing', 'through', 'your', 'myspace', '...', 'it', 'goes', 'to', 'some', 'dude', "s", 'page']

Tokenisation

[I', 'never', 'found', 'everytime', 'I', 'click', 'twitter', 'thing', 'myspace', 'goes', 'dude', 'page']

Suppression de la ponctuation, des symboles, chiffres, stopwords,... et passage en lowercase

[I', 'never', 'found', 'everytim', 'I', 'click', 'twitter', 'thing', 'myspac', 'go', 'dude', 'page']

Lemmatisation et Stemming

[I', 'never', 'found', 'everytim', 'I', 'click', 'twitter', 'thing', 'myspac', 'go', 'dude', 'page']

Texte Final



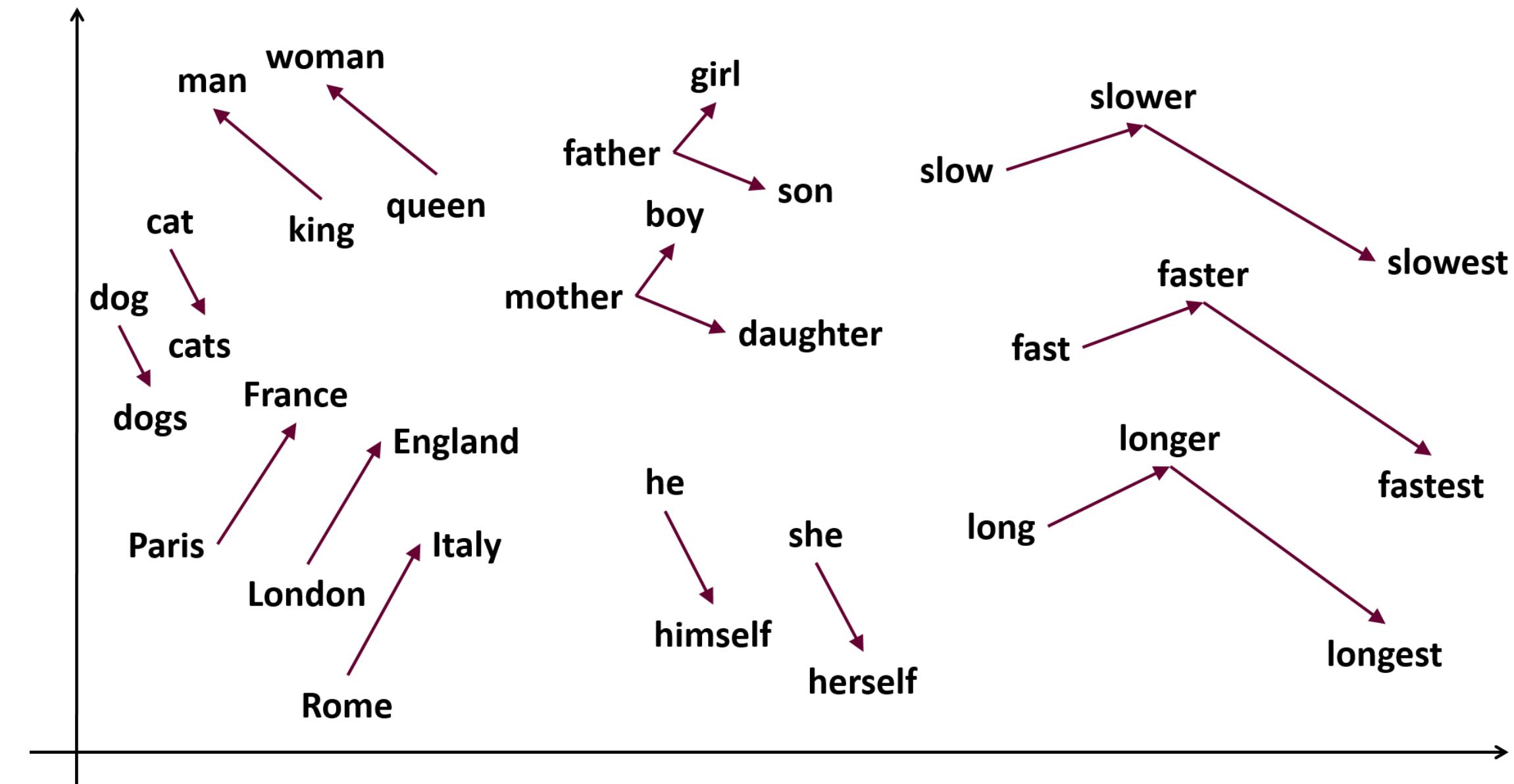
Model repository

Bucket S3

- Permet de partager les **embeddings**
 - Mapping word => vecteur
- Les classiques **pré-entraînés** (word2vec, fasttext)
- Des embeddings « **maison** » : stand-alone, agrégé avec les modèles pré-entraînés ou fine-tuned sur nos données



fastText





Cas d'usages

- Text2job / Text2app
 - **Rapprocher** du texte libre d'un code ROME ou d'une appellation ROME
 - **Classification multi-classe**
 - Différentes stratégies de **pré-processing** (stemmatization / lemmatization / avec ou sans stopwords / etc...)
 - Text2job : TF/IDF + SVM, **OneVsRest**
 - 90k documents, 500 classes, 70/30 split, **~90%** de précision
 - Text2app: **LSTM / CNN / TFIDF + Dense**
 - 1.5m documents, 8k classes, 75/25 split, **~85%** de précision

Saisir un texte à analyser :

pizzas classiques et élaborées, les entrées et les desserts, et les recettes de pâtes fraîches et de bruschetta.
il fabrique les différentes pâtes à pizza utilisées (pâtes levées, pâtes levées composées, pâtes « traiteurs »), ou les reçoit de son fournisseur. Il prépare les garnitures simples (tomates, les différents fromages, les herbes aromatiques) et les garnitures élaborées (les sauces d'accompagnement, les fruits de mer, charcuterie, légumes), assure le pétrissage et la panification de la pâte, effectue la mise en place de la garniture puis la cuisson de la pizza.

Analyser !

Codes ROME suggérés par l'algorithme :

- | | | |
|------|---|------------|
| 0.72 | Fabrication de crêpes ou pizzas (G1604) | Pourquoi ? |
| 0.06 | Personnel de cuisine (G1602) | Pourquoi ? |
| 0.04 | Manutention manuelle de charges (N1105) | Pourquoi ? |
| 0.03 | Conduite d'équipement de production alimentaire (H2102) | Pourquoi ? |
| 0.01 | Accueil et renseignements (M1601) | Pourquoi ? |



Industrialisation



- V3.6+
- Acquisition des données + pré-processing + entraînement + prédition
- Web Service avec Flask ou Falcon / Gunicorn



- Permet d'encapsuler les applications python



Pour l'IHM



Projet MAIL Natural Language Processing

1 Objectifs & Données

2 Approche

3 Modélisation

4 Industrialisation

Problématique :

- Un constat : le nombre de mails de demandeurs d'emploi explose.
 - ❖ 17,6 millions en 2016
 - ❖ 33,7 millions en 2018

Irritants détectés :

- Identifier l'émetteur et l'objet du mail est parfois compliqué.
- Orienter le mail vers la personne la plus à même de répondre est parfois difficile.

Preuve de valeur :

- réduction du temps de traitement des mails



Objectifs & Données

Objectif

- Identifier le demandeurs d'emploi expéditeur du mail
- Classifier les mails parmi 21 catégories définies par le métier.

Les données

- 20 millions de mails bruts sur l'année 2019, envoyé depuis des messageries personnelle pour la plus part.

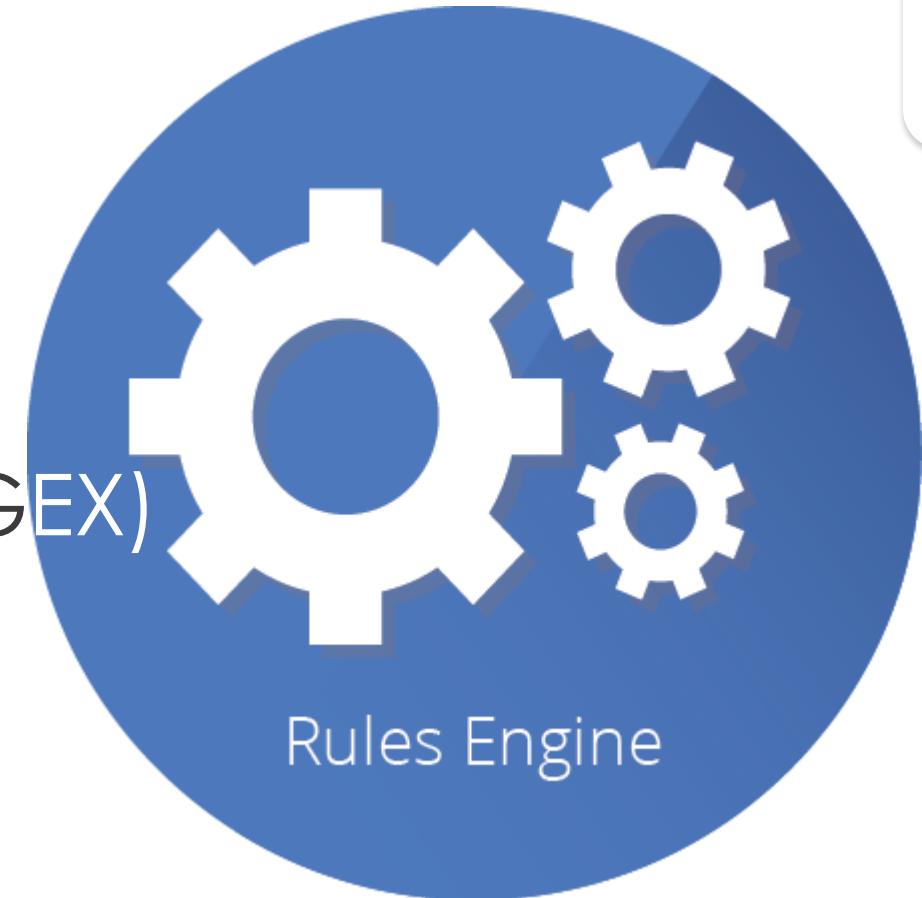




Approche

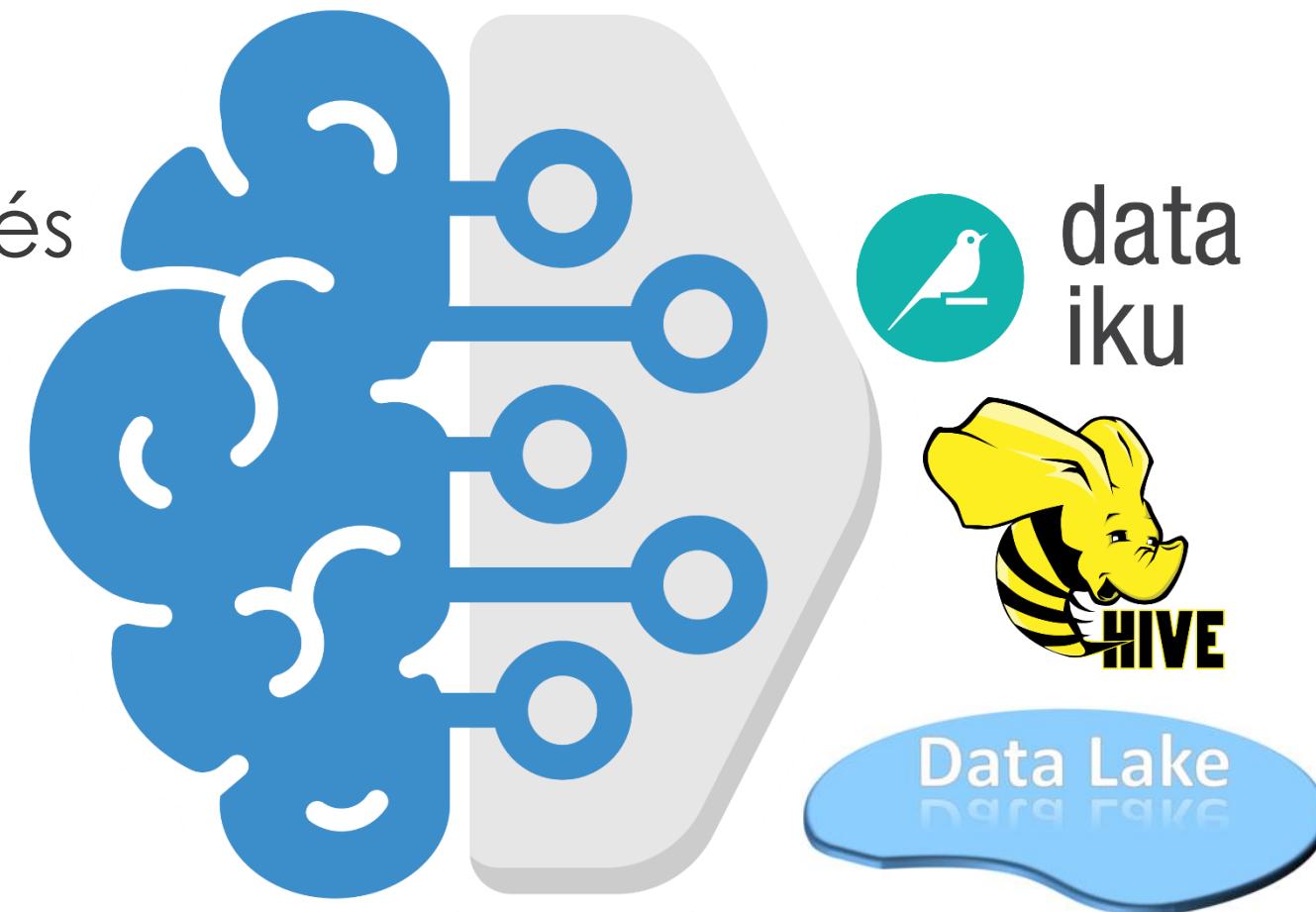
Identifier le demandeurs d'emploi : Moteur de règles (REGEX)

- Déetecter identifiant
- Déetecter nom/prénom



Classifier les mails parmi 21 catégories

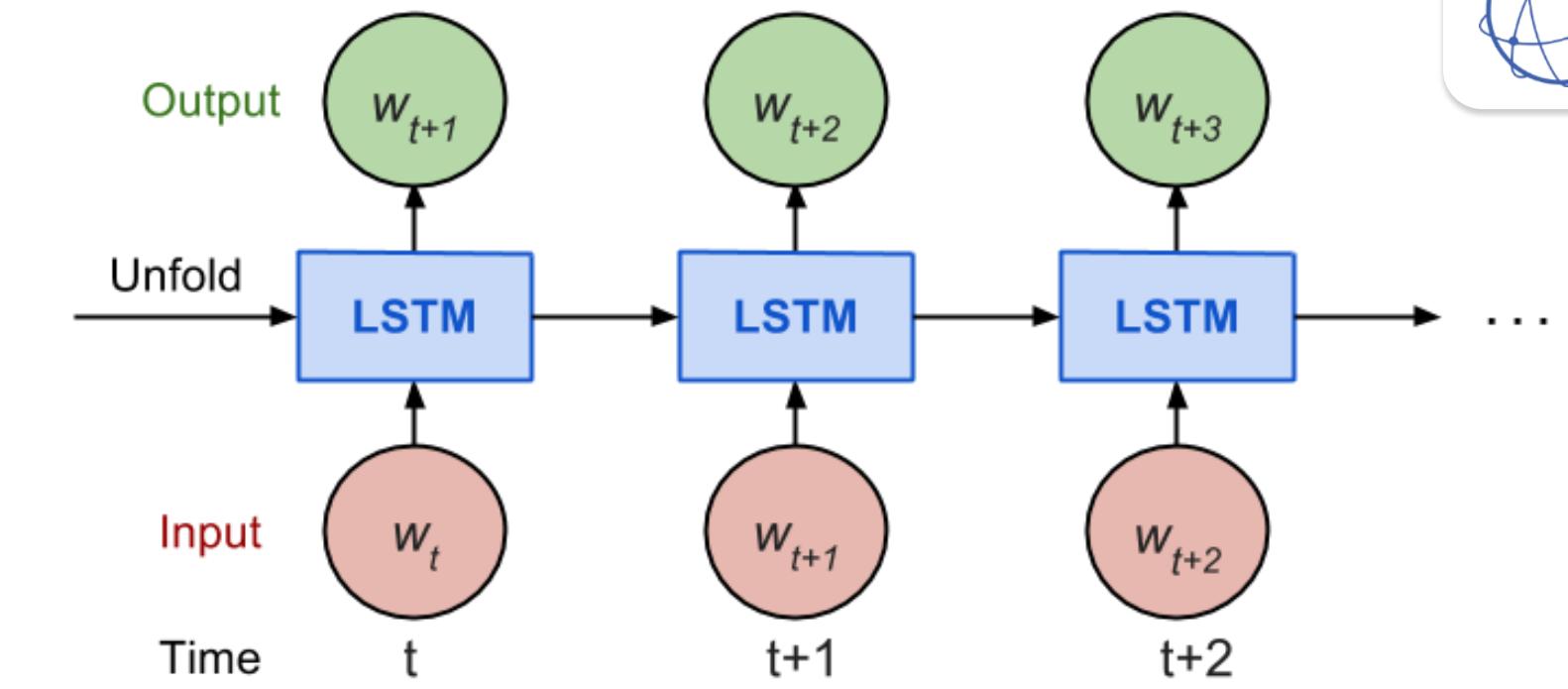
- Approche supervisé : 13 000 mails labélisés
- Dataset de référence
- Word_n_Fun
- Embedding
- Algorithmes deep learning
- Industrialisation





Modélisation

- Réseau **LSTM** 3 couches
- 66% de rappel en moyenne sur les 21 catégories
- Atelier de labélisation en cours



Pistes d'amélioration envisagé :

- Activ'Learning pour la sélection des mails à labéliser
- Transfer Learning (CamenBert?)
- Amélioration des pré-traitement
- Amélioration de l'embedding

		Confusion matrix, without normalization																						
		I02	I03	I04	I05	I06	I07	I08	I09	I10	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	TZ1	TZ0	
		102	103	104	105	106	107	108	109	110	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	TZ1	TZ0	
I02		32	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	6	
I03		0	24	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
I04		6	1	32	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	2	
I05		2	0	0	16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
I06		1	0	0	0	4	0	0	0	0	3	0	0	0	0	0	0	0	0	0	1	0	2	
I07		4	1	1	0	2	17	0	0	1	1	0	0	0	1	0	0	4	0	0	0	0	0	
I08		3	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	0	0	0	0	0	4	
I09		1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	
I10		1	0	0	0	0	0	0	0	21	2	0	0	0	1	0	2	0	0	0	3	0	3	
P01		2	0	0	0	0	0	0	0	2	92	0	0	0	2	0	3	0	0	0	0	0	1	
P02		0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1	0	0	0	1	0	0	
P03		0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	0	0	0	0	0	1	
P04		2	0	0	0	0	0	0	0	0	1	0	0	7	0	0	7	0	0	0	1	0	1	
P05		0	0	0	0	0	0	0	1	5	1	0	0	23	0	7	2	0	0	0	0	0	2	
P06		0	0	0	0	1	0	0	0	0	0	0	0	1	2	2	0	0	0	0	0	0	0	
P07		1	0	1	0	0	0	0	0	6	0	0	0	1	5	1	102	3	0	0	0	0	3	
P08		0	0	0	0	0	0	0	0	0	0	0	0	5	0	1	2	0	0	0	0	0	1	
P09		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	
P10		1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	4	0	0	1	
P11		0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	6	0	3	0	
TZ1		0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
TZ0		13	0	1	0	0	0	0	0	3	2	0	0	0	6	0	6	0	0	0	0	0	23	



Industrialisation



- V3.6+
- Le langage par défaut à l'ADS
- Idéal pour le prototypage rapide de nos modèles



- Une des références en matière de graphe computationnel
- Idéal pour l'implémentation de DL
- A des APIs pour de nombreux langages



Pour stocker les modèles finaux



Pour pouvoir faire de l'inférence à partir du modèle depuis nos applications java
(ou alors en Python + Falcon)

Cible



Auto-complétion Suggestion de contenu **Word2job**



Modélisation

Industrialisation

Difficultés

Objectif

Faciliter la recherche sur pe.fr en suggérant des codes ROMEs et codes Compétences

- Concepts centraux de tout le SI PE (et de pe.fr)
- Issu d'un sprint innovation



Word2job

Modélisation

- Algorithme maison en deux étapes
 - **TF/IDF** sur les référentiels ROMES / Compétences pour identifier les mots les plus **discriminants**
 - **Word2vec** entraîné sur 1M de descriptifs d'offres pour identifier les **synonymes** de ces mots discriminants
- Export d'un **index mot -> ROME** (Compétence) correspondant soit à son score TF/IDF soit au score TF/IDF du synonyme le plus proche du code pondéré par la distance cosinus word2vec

$$TFIDF_{t,d,D} = \underbrace{TF_{t,d}}_{\text{Importance d'un terme } t \text{ dans un document } d} \times \underbrace{IDF_{t,D}}_{\substack{\text{Fréquence d'un terme } t \text{ dans un document } d \\ \text{Importance du terme } t \text{ dans l'ensemble des documents } D}}$$



Industrialisation



- V3.6+
- Le langage par défaut à l'ADS
- Idéal pour le prototypage rapide de nos modèles



Pour certaines manipulations sur les matrices TF/IDF



- Permet d'encapsuler les applications python



- Basé sur Apache lucene
- Idéal pour récupérer rapidement les valeurs correspondantes à un index str



Word2job

Difficultés:

- Algorithme **non-supervisé**
- Difficile de **tester automatiquement** les résultats
 - Dataset de « référence » des suggestions retournées pour les 1000 requêtes les plus fréquentes
 - Validation **manuelle**
 - Comparaison manuelle en cas de divergence suite à une mise à jour du modèle
- Besoin d'un **feedback loop** [en cours]

A screenshot of a search interface, likely a browser extension or a dedicated tool. The search bar at the top contains the text "langu". Below the search bar is a list of ten job titles, each preceded by a small icon of a briefcase and a person. The job titles are:

- Lecteur / Lectrice de langue étrangère dans l'enseignement secondaire
- Lecteur / Lectrice de langues dans l'enseignement supérieur
- Professeur / Professeure en langue des signes
- Formateur / Formatrice d'espagnol
- Professeur / Professeure de langues vivantes
- Professeur / Professeure d'espagnol
- Formateur / Formatrice d'allemand
- Formateur / Formatrice de Français Langue Etrangère - FLE -
- Formateur / Formatrice d'italien
- Professeur / Professeure d'allemand



Extraction d'informations

STARK INDUSTRIES

Approche

Labellisation

Modélisation

Objectif

- Les descriptifs d'offres : regorgent d'informations pertinentes
- Ces informations ne sont pas captés par le SI
- Il faut donc les extraire automatiquement
 - Pour caractériser les offres plus finement
 - Pour capter des informations ne correspondant à aucun référentiel
 - Sprint innovation



Stark

Modélisation

- Les informations **pertinentes** ont été **présélectionnées** par le métier
- Problème de **classification supervisée** avec 9 labels + environ 120 sous-labels (distribution hétérogène parmi les labels)
 - **Labélisation** (pô facile !)

Back in 2000 , People Magazine PUBLISHER highlighted Prince Williams' PERSON style who at the time was a little more fashion-conscious , even making fashion statements at times .

Now-a-days the prince mainly wears navy COLOR suits ITEM (sometimes double-breasted DESIGN) , light blue COLOR button-ups ITEM with classic LOOK pointed DESIGN collars PART , and burgundy COLOR ties ITEM .

But who knows what the future holds ...

Duchess Kate PERSON did wear an Alexander McQueen BRAND dress ITEM to the wedding OCCASION in the fall of 2017 SEASON .



Stark

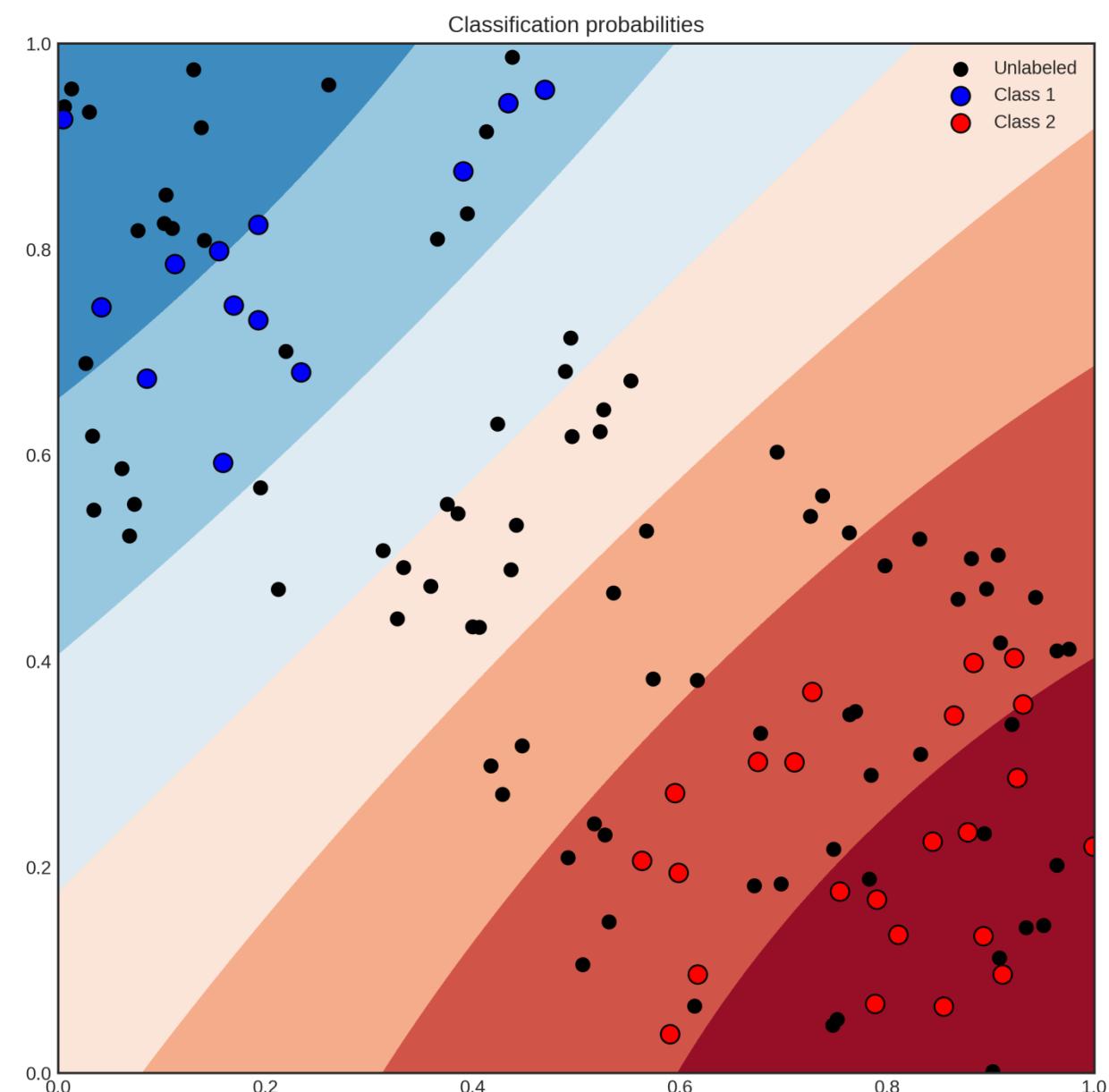
Labellisation



Stark

Labellisation

- Difficile de mobiliser; exercice **pénible**
 - ~700 phrases par jour
 - 133m de lignes dans la table offres
 - Beaucoup d'éléments **ambigus**
- Active learning**
 - Plutôt que de labéliser des échantillons aléatoires : on **cible** les zones d'incertitudes de l'espace
 - Module maison basé sur le package modAL



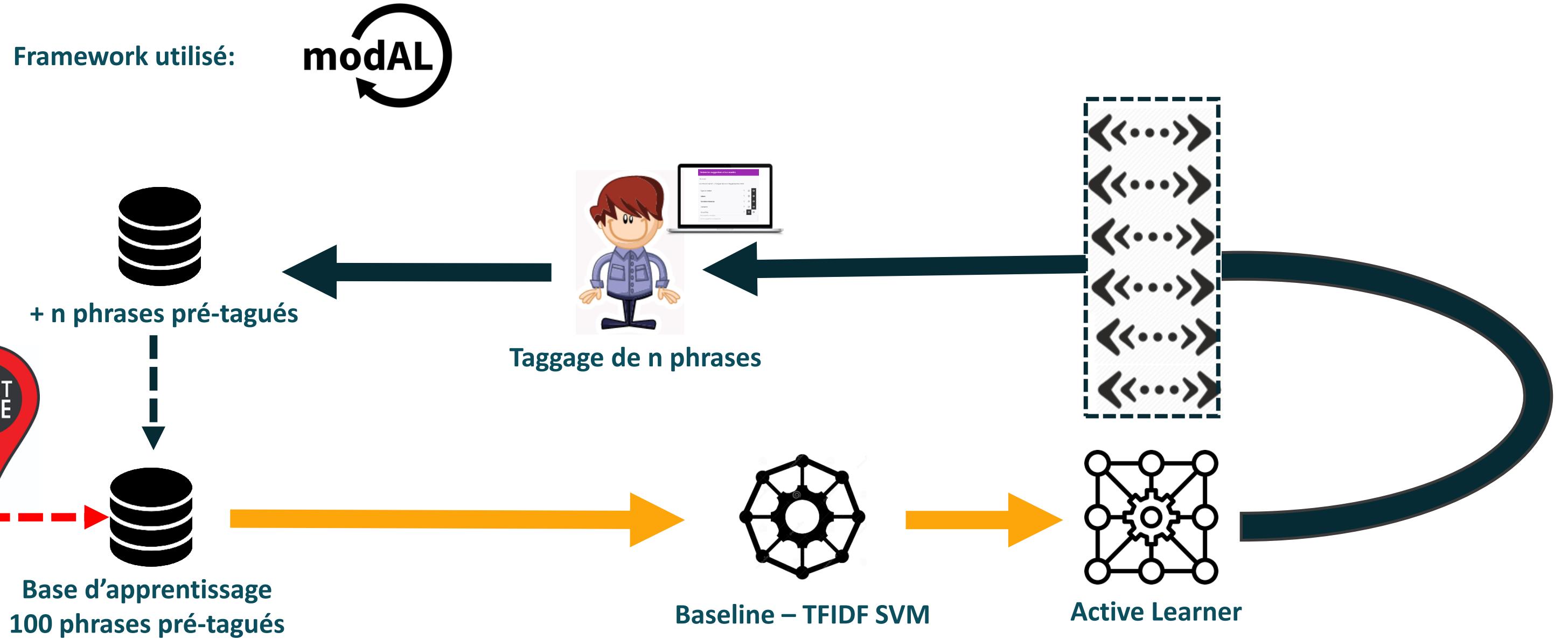


Stark

Objectif – Utiliser le machine learning pour soulager l'effort d'annotation / taggage

Framework utilisé:

modAL

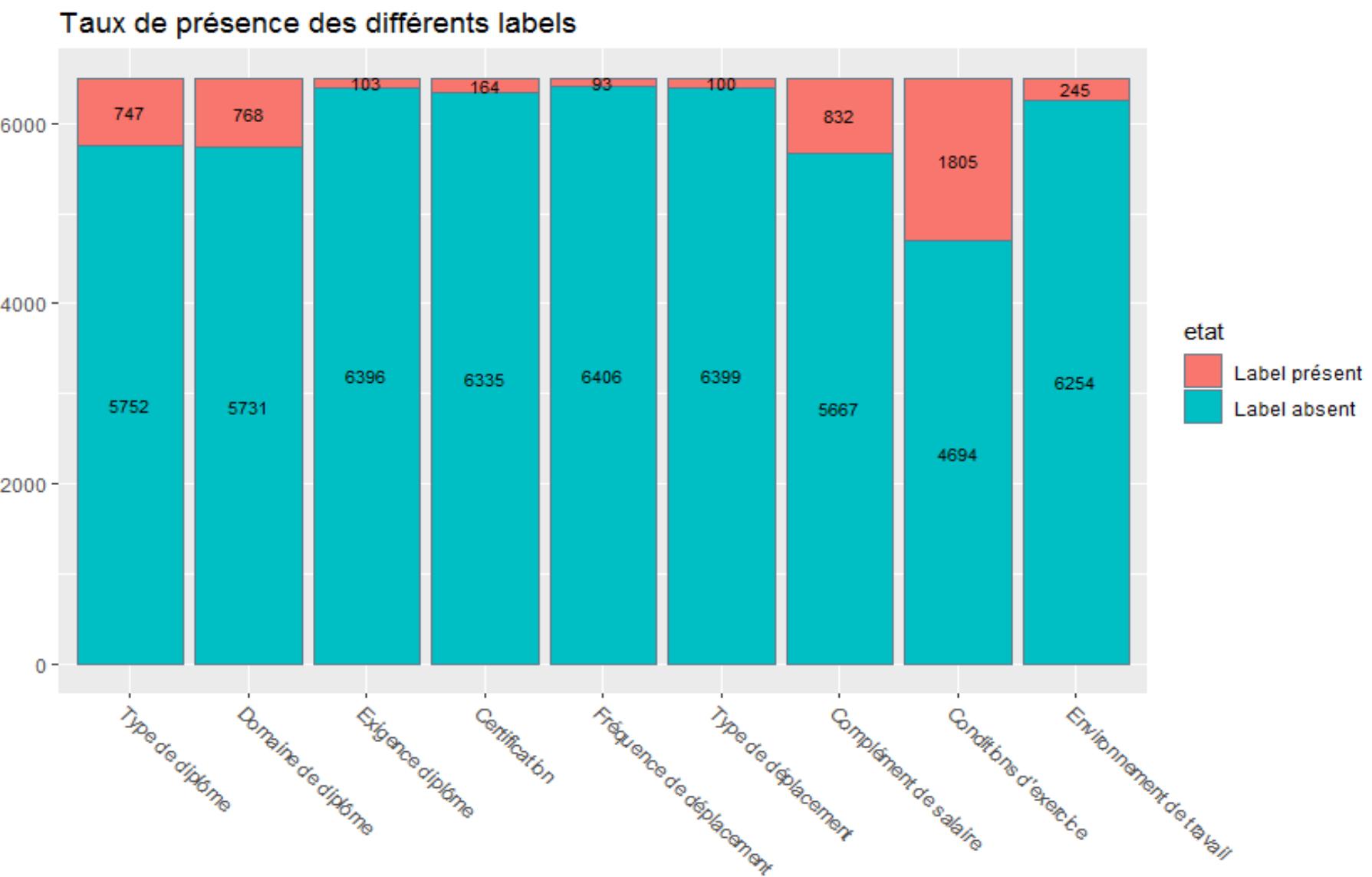




Stark

Labellisation

- Difficile de mobiliser; exercice **pénible**
 - ~700 phrases par jour
 - 133m de lignes dans la table offres
 - Beaucoup d'éléments **ambigus**
 - **Active learning**
 - Plutôt que de labéliser des échantillons aléatoires : on **cible** les zones d'incertitude de l'espace
 - Module maison basé sur le package modAL





Stark

Modélisation

- Difficile d'utiliser les modèles de **Named Entity Recognition**
 - Vocabulaire hautement **spécifique** (acronymes, beaucoup de chiffres) qui n'apparaît pas forcément dans les corpus d'entraînement
- Deux approches envisagées :
 - 1 **TF/IDF + SVM** au niveau des 9 labels & 9 TF/IDF + SVM pour les sous labels
 - 1 archi **deep learning** (TF/IDF + Dense ou **Embedding Maison + Attention** / CNN / LSTM) avec deux outputs : les labels et sous labels
 - En labo, ajouter une sortie correspondant aux labels « guide » la back propagation et améliore les performances de classification au niveau sous-label

Label	Taux de reconnaissance
Type de diplôme	95.4%
Exigence diplôme	Pas suffisamment de données
Certification	Pas suffisamment de données
Fréquence de déplacement	Pas suffisamment de données
Type de déplacement	Pas suffisamment de données
Complément de salaire	95%
Conditions d'exercice	88.6%
Environnement de travail	Pas suffisamment de données

Conclusion

- Beaucoup de projets **passionnants** (vision, nlp, reco, etc...)
- Difficultés d'**industrialisation** : passage lab / prod encore hésitant (choix technologiques, intégration dans le SI)
- Difficultés d'**acculturation** : faire comprendre aux **métiers** l'étendu des possibles, se mettre d'accord sur les **métriques**
- Grand projet transverse à venir : mesure de la **valeur**; feedback loop



Merci !
Pour votre
attention



N'hésitez pas à **venir échanger**
avec nous !



CogniTALK

