

Disambiguation of inventor datasets

Projet de deuxième année du parcours ingénieur mathématiques-informatique

Responsable : François Maublanc

Encadrant : Pierre Andry

BEN MOSBAH Iyad

LIU Charles

OUESLATI Salim

TCHABO Bacarie



Sommaire

01

Présentation
du sujet

02

Base de
données

03

Algorithme

04

Interface
graphique



I°) Présentation du sujet

Contexte et problématique

Les bases de données de brevets, comme celle de l'EPO (European Patents Office), contiennent des millions d'informations sur les inventeurs. Cependant, **l'absence d'identifiant unique** rend difficile leur suivi : un même inventeur peut apparaître sous plusieurs variantes de nom, ou différents individus peuvent partager le même nom.

Problème posé : Comment identifier de manière fiable un inventeur à partir de données partielles et retrouver tous ses brevets sans générer de faux positifs ni de faux négatifs ?

Outils mis à disposition

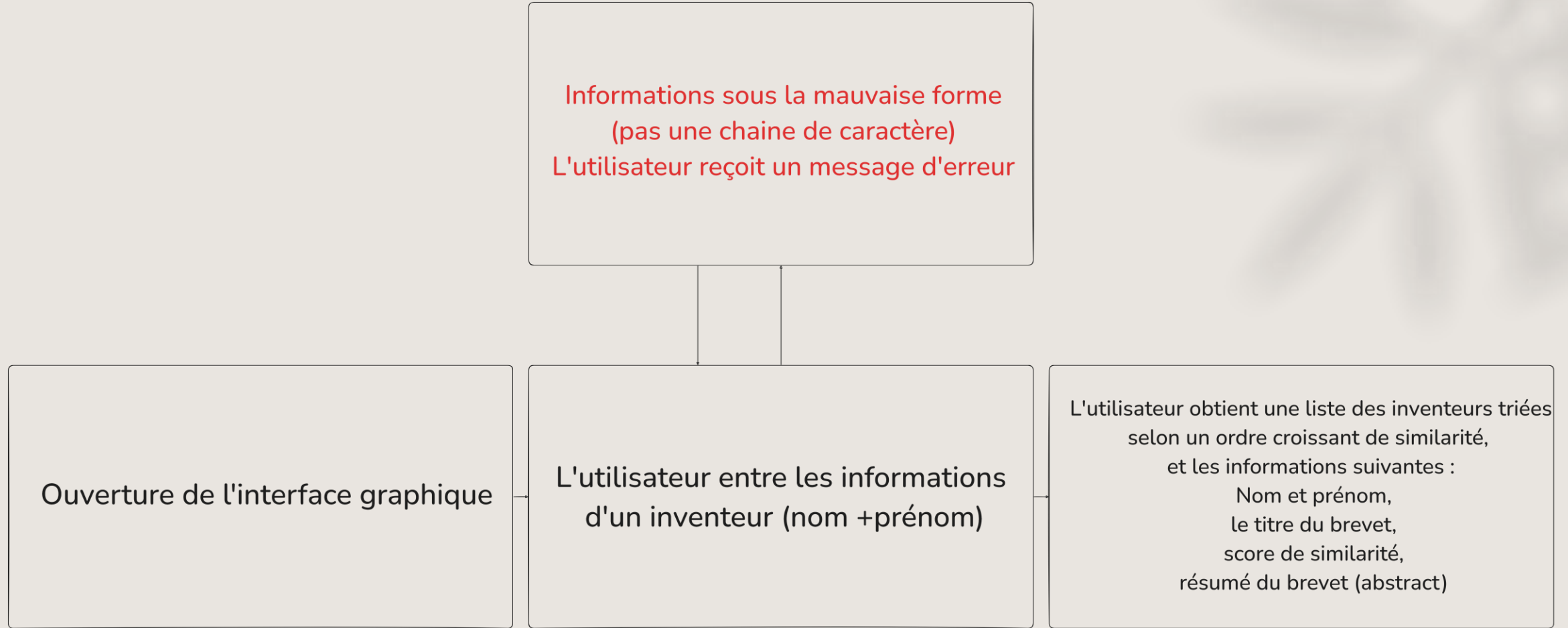
- USPTO (United States Patent and Trademark Office) a mis sa solution en libre accès en ligne. (codé en Python)
 - Projet entièrement en Python
- Base de données teseo_inventors

Objectifs


Le but est de créer une application ergonomique permettant de lever les potentiels ambiguïtés dans un dataset d'inventeurs.

Les utilisateurs pourront rentrer les informations d'un inventeur. L'application retourne l'identifiant de l'inventeur ainsi que ses brevets d'invention.

Posture de l'utilisateur



Management du projet



Agile PROJET ING2

Active Timeline Mine All + Nouveau

Projets

Planning 1

Documentation & reproductibilité

100,00 %

+ Nouveau projet

In Progress 2

Algorithme de désambiguïsation

66,67 %

Interface Utilisateur

66,67 %

+ Nouveau projet

Groupes masqués

- Paused 0
- Backlog 0
- Done 1

Done 1

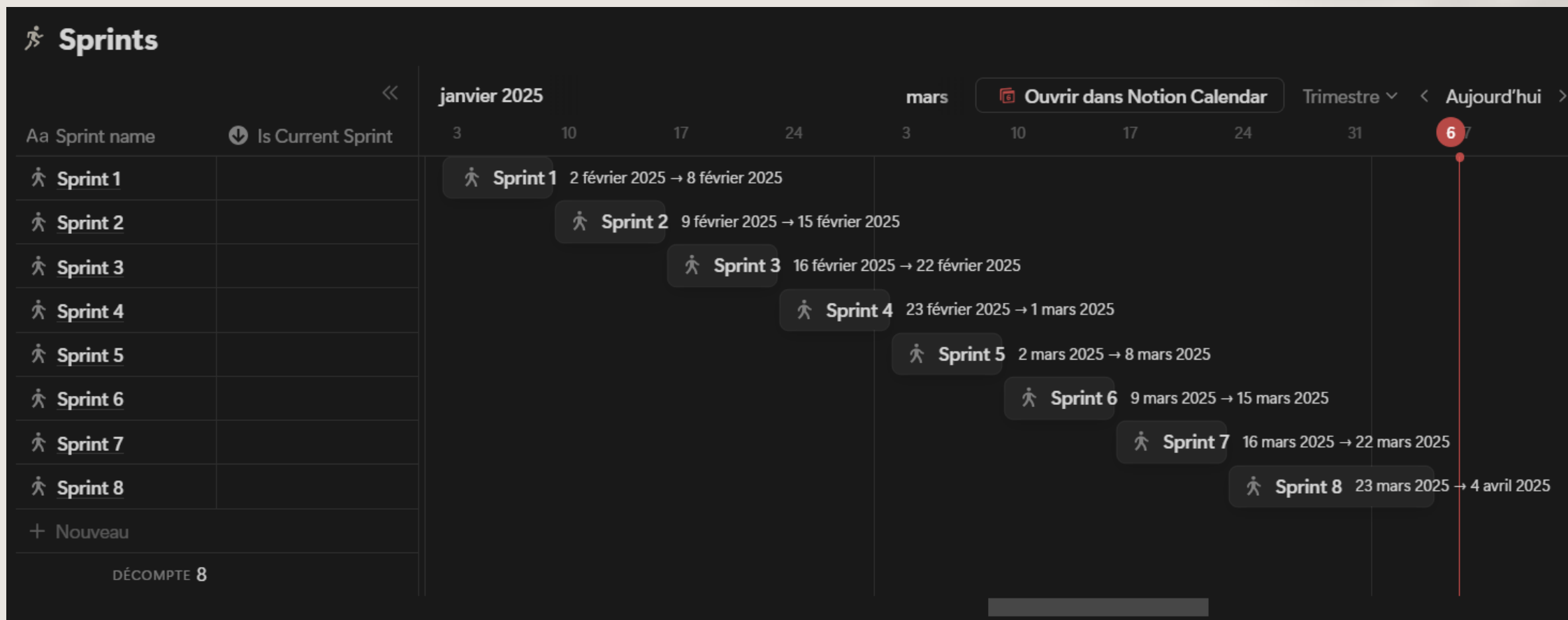
Rechercher une page...

Importation et traitement des bases de données

100,00 %

Par projet This sprint Backlog Mine People

Calendrier



Aa	Statut	Assign	Due	Sprint	Is Current Sprint	Projet
Extrait le fichier bak.	Ouvrir	Done	8 février 2025	Sprint 1	<input type="checkbox"/>	Importation et traitement des bases de données
+ Nouvelle tâche						
Décompte 1						
Aa	Statut	Assign	Due	Sprint	Is Current Sprint	Projet
Chargement et prétraitement des datasets de brevets, incluant le nettoyage et la suppression des données dupliquées.	Done		15 février 2025	Sprint 2	<input type="checkbox"/>	Importation et traitement des bases de données
Lire la base de donnée pour en comprendre la structure et les champs associés à chaque table	Done		15 février 2025	Sprint 2	<input type="checkbox"/>	Importation et traitement des bases de données
+ Nouvelle tâche						

Backlog – partie 2

▼ 👤 Sprint 3 2						
Aa	⚙ Statut	👤 Assign	📅 Due	↗ Sprint	🔍 Is Current Spri...	↗ Projet
📄 Compréhension du code.	● Done		1 mars 2025	👤 Sprint 4 👤 Sprint 3	<input type="checkbox"/> <input type="checkbox"/>	? Algorithme de désambiguïsation
📄 Lecture du document de recherche afin d'assurer une compréhension de la méthodologie appliqué.	● Done		22 février 2025	👤 Sprint 3	<input type="checkbox"/>	? Algorithme de désambiguïsation
+ Nouvelle tâche						
DÉCOMPTE 2						
▼ 👤 Sprint 4 2 ... +						
Aa	⚙ Statut	👤 Assign	📅 Due	↗ Sprint	🔍 Is Current Spri...	↗ Projet
📄 Adaptation du code à notre cas.	● In Progress		8 mars 2025	👤 Sprint 4 👤 Sprint 5 👤 Sprint 6	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	? Algorithme de désambiguïsation
📄 Compréhension du code.	● Done		1 mars 2025	👤 Sprint 4 👤 Sprint 3	<input type="checkbox"/> <input type="checkbox"/>	? Algorithme de désambiguïsation

Backlog – partie 3

▼ 👤 Sprint 5 1						
Aa	⚙ Statut	👤 Assign	📅 Due	↗ Sprint	🔍 Is Current Spri...	↗ Projet
📄 Adaptation du code à notre cas.	● In Progress		8 mars 2025	👤 Sprint 4 👤 Sprint 5 👤 Sprint 6	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	? Algorithme de désambiguïsation
+ Nouvelle tâche						
DÉCOMPTE 1						
▼ 👤 Sprint 6 2 ... +						
Aa	⚙ Statut	👤 Assign	📅 Due	↗ Sprint	🔍 Is Current Spri...	↗ Projet
📄 Adaptation du code à notre cas.	● In Progress		8 mars 2025	👤 Sprint 4 👤 Sprint 5 👤 Sprint 6	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	? Algorithme de désambiguïsation
📄 Conception et développement d'une interface utilisateur ergonomique	● Done		15 avril 2025	👤 Sprint 6	<input type="checkbox"/>	🖥 Interface Utilisateur

Backlog – partie 4

▼ Sprint 7 2						
Aa	⚙ Statut	👤 Assign	📅 Due	🚩 Sprint	🔍 Is Current Spri...	🚩 Projet
📄 Test d'ergonomie et d'accessibilité	● Done		22 avril 2025	🚩 Sprint 7	<input type="checkbox"/>	📄 Interface Utilisateur
📄 Ajout du module Elasticsearch pour assurer une fonctionnalité accrue.	● In Progress		22 mars 2025	🚩 Sprint 7	<input type="checkbox"/>	📄 Interface Utilisateur
+ Nouvelle tâche						
DÉCOMPTE 2						
▼ Sprint 8 4 ... +						
Aa	⚙ Statut	👤 Assign	📅 Due	🚩 Sprint	🔍 Is Current Spri...	🚩 Projet
📄 Création du PPT	● Done		4 avril 2025	🚩 Sprint 8	<input type="checkbox"/>	📄 Documentation & reproductibilité
📄 Rédaction du rapport 📄 OUVRI	● Done		4 avril 2025 💬 📄	🚩 Sprint 8	<input type="checkbox"/>	📄 Documentation & reproductibilité
📄 Organisation du code pour faciliter sa réutilisation	● Done		4 avril 2025	🚩 Sprint 8	<input type="checkbox"/>	📄 Documentation & reproductibilité
📄 Rédaction d'une documentation technique complète pour assurer la compréhension et la maintenance du code.	● Done		4 avril 2025	🚩 Sprint 8	<input type="checkbox"/>	📄 Documentation & reproductibilité

II°) Base de données

User Story 1 : Importation et traitement de la base de données

Numéro de story : US-01

Durée estimée : Sprint 1 et sprint 2 (~14 jours)

Description :

En tant que utilisateur,

Je veux extraire la base de données à partir de la sauvegarde et nettoyer les données brutes des inventeurs et brevets,

afin de garantir une qualité optimale pour les étapes ultérieures de traitement.

Prérequis : Fichier de sauvegarde

Critères d'acceptation :

Les champs manquants critiques sont complétés ou signalés.

Exploitation de la base de données

Pourquoi SQL Server Management Studio (SSMS) ?

- Permet d'ouvrir le fichier reçu teseo_inventor_bakup que d'autre applications ne peuvent pas ouvrir
- Base unique pour données brutes, nettoyage et résultats
- Peut traiter un gros volume de données
- Peut exécuter Python/R

patents

3 929 636 entrées

patents_abstract

3 750 048 entrées

patents_applicants

4 050 148 entrées

patents_inventors

10 396 265 entrées

patents_tech

134 954 263 entrées

patents_titles

3 754 118 entrées

Structure des tables de données

Table

patents

patents_inventors

patents_applicants

patents_title

patents_abstracts

patents_tech

Contenu principal

Infos générales sur les brevets (dates, type)

Liste des inventeurs liés aux brevets

Liste des demandeurs (entreprises, universités, etc.)

Titres des brevets (en plusieurs langues)

Résumés des brevets (textes longs)

Classification technologique (domaines, IPC...)

Table patents_inventors

Champs importants :

- person_name : nom saisi tel quel → souvent source d'ambiguïtés
- appln_id : identifiant du brevet associé
- person_ctype_code, person_address : informations contextuelles utiles
- psn_id, psn_name : identifiants harmonisés → peu fiables, doivent être revérifiés

Pourquoi cette table est cruciale :

- C'est elle qui contient les noms des inventeurs, au cœur de la problématique du projet.
- Elle reflète les variations d'écriture, erreurs de saisie, affiliations, etc.
- Toutes les tentatives de désambiguïsation (nettoyage, regroupement, scoring...) partent de ces données.
- C'est la table qui relie directement un inventeur à un brevet via appln_id.

	appln_id	appln_auth	person_id	inv_seq_nr	person_name	person_name_orig_lg	psn_name	psn_id
1	1	EP	2	1	Lipponen, Markku	Lipponen, Markku	LIPPONEN, MARKKU	19669542
2	1	EP	3	2	Laitinen, Timo	Laitinen, Timo	LAITINEN, TIMO	18561041
3	1	EP	4	3	Aho, Ari	Aho, Ari	AHO, ARI	420702
4	1	EP	5	4	Knuutila, Jarno	Knuutila, Jarno	KNUUTILA, JARNO	17435717
5	2	EP	9	1	Griffiths, Andrew David	Griffiths, Andrew David	GRIFFITHS, ANDREW DAVID	10937127
6	2	EP	10	2	Hoogenboom, Hendricus Renerus Jacobus Mattheus	Hoogenboom, Hendricus Renerus Jacobus Mattheus	HOOGENBOOM, HENDRICUS RENERUS JACOBUS MATTHEUS	13132864
7	2	EP	11	3	Marks, James David	Marks, James David	MARKS, JAMES DAVID	20936434
8	2	EP	12	4	McCafferty, John	McCafferty, John	MCCAFFERTY, JOHN	21412537
9	2	EP	13	5	Winter, Gregory Paul	Winter, Gregory Paul	WINTER, GREGORY PAUL	35376353
10	2	EP	14	6	Grigg, Geoffrey Walter	Grigg, Geoffrey Walter	GRIGG, GEOFFREY WALTER	10939289
11	3	EP	22	1	Wieczorek, Herfried, Philips Corporate	Wieczorek, Herfried, Philips Corporate	WIECZOREK, HERFRIED, PHILIPS CORPORATE	35092425
12	3	EP	23	2	Schneider, Stefan, Philips Corporate	Schneider, Stefan, Philips Corporate	SCHNEIDER, STEFAN, PHILIPS CORPORATE	28752234
13	3	EP	24	3	Lauter, Josef, Philips Corporate	Lauter, Josef, Philips Corporate	LAUTER, JOSEF, PHILIPS CORPORATE	18788104
14	4	EP	27	1	Chittipeddi, Sailesh	Chittipeddi, Sailesh	CHITTIPEDDI, SAILESH	4950267

psn_sector	han_name	han_harmonized	han_id	person_address	person_ctype_code
	Lipponen, Markku	0	100000002	Simo Kaarion katu 1 A 2,33720 Tampere	FI
	Laitinen, Timo	0	100000003	Peiponkatu 6.37830 Viala	FI
	Aho, Ari	0	100000004	Elementinpolku 13 A 6.33720 Tampere	FI
	Knuutila, Jarno	0	100000005	Matti Tapijon katu 1 F 17.33720 Tampere	FI
	Griffiths, Andrew David	0	100000009	28 Lilac Court, Cherry Hinton Road,Cambridge CB1 4...	GB
	Hoogenboom, Hendricus Renerus Jacobus Mattheus	0	100000010	1 Hauxton Road, Little Shelford,Cambridge CB2 5JH	GB
	Marks, James David	0	100000011	107 Ardmore,Kesington, CA 94707	US
	McCafferty, John	0	100000012	32 Wakelin Avenue,Sawston, Cambridgeshire CB2 4DA	GB
	Winter, Gregory Paul	0	100000013	c/o Trinity College,Cambridge CB2 1TQ	GB
	Grigg, Geoffrey Walter	0	100000014	352 Burns Bay Road, Lane Cove,Linley Point, NSW 2...	AU
	Wieczorek, Herfried, Philips Corporate	0	100000022	Intellectual Property GmbH, Habsburgerallee 11,5206...	DE
	Schneider, Stefan, Philips Corporate	0	100000023	Intellectual Property GmbH, Habsburgerallee 11,5206...	DE
	Lauter, Josef, Philips Corporate	0	100000024	Intellectual Property GmbH, Habsburgerallee 11,5206...	DE
	Chittipeddi, Sailesh	0	100000027	308 Lenape Trail,Allentown, PA 18104	US

Prétraitements nécessaires

- **Suppression des lignes non exploitables**
→ Lignes sans nom, avec NULL, ou -NOT AVAILABLE-
- **Création d'index SQL**
→ Accélère les requêtes sur person_name, appln_id, etc.
- **Utilisation critique de psn_name**
→ Champ harmonisé utile mais insuffisant : vérification manuelle ou scriptée nécessaire
- **Objectif global**
→ Réduire le bruit et faciliter le travail de l'algorithme de désambiguïsation

Applications

```
CREATE INDEX idx_person_name ON dbo.patents_inventors (person_name);  
CREATE INDEX idx_appln_id ON dbo.patents (appln_id);
```

```
DELETE FROM dbo.patents_inventors  
WHERE person_name IS NULL  
OR person_name = '-NOT AVAILABLE-'  
OR psn_id IS NULL;
```

	appln_id	appln_auth	person_id	inv_seq_nr	person_name	person_name_orig_lg	psn_name	psn_id	psn_sector	han_name	han_harmonized	han_id	person_address	person_ctr_code
1	399127	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
2	399108	EP	263	2			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
3	399108	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
4	397978	EP	263	3			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
5	397882	EP	263	3			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
6	397882	EP	263	2			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
7	397882	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
8	397791	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
9	397373	EP	263	2			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
10	394329	EP	263	3			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
11	394329	EP	263	2			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
12	394329	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		
13	394281	EP	263	1			-NOT AVAILABLE-	23866221	UNKNOWN		0	100000263		

Ambiguïtés des données

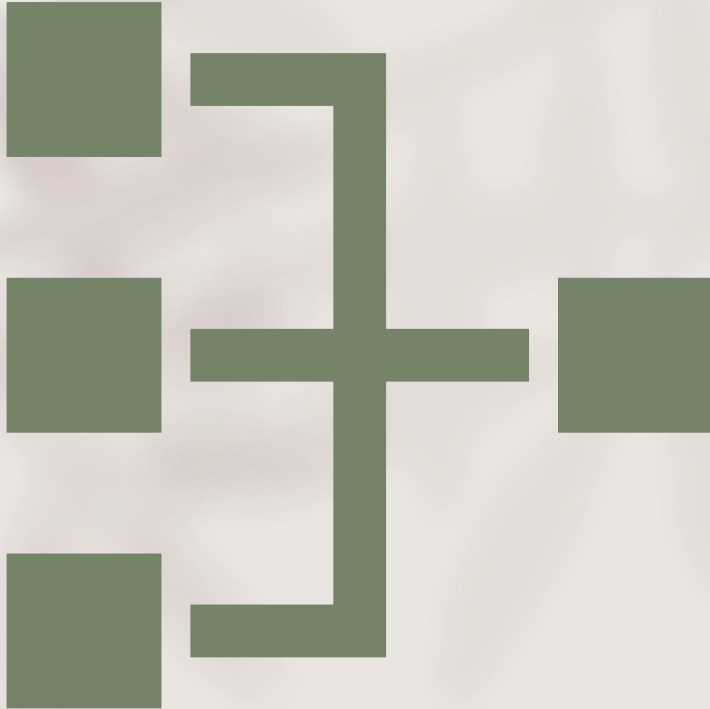
Avec des requêtes SQL simples, nous pouvons identifier des ambiguïtés au sein des données.

```
USE teseo_inventors
GO
SELECT * from patents_inventors WHERE person_name LIKE 'Dupont, Jean-Fabien' OR person_name LIKE 'Dupont, Jean-Fabien%'
```

	appln_id	appln_auth	person_id	inv_seq_nr	person_name	person_name_orig_lg	psn_name	psn_id
1	143806	EP	169135	2	DUPONT, Jean-Fabien Kodak Industrie	DUPONT, Jean-Fabien Kodak Industrie	DUPONT, JEAN-FABIEN KODAK INDUSTRIE	7483080
2	15936177	EP	1271062	1	Dupont, Jean-Fabien	Dupont, Jean-Fabien	DUPONT, JEAN-FABIEN	7483079
3	15936199	EP	1271092	1	Dupont, Jean-Fabien, c/o Kodak Industrie, Dep-Brev	Dupont, Jean-Fabien, c/o Kodak Industrie, Dep-Brev	DUPONT, JEAN-FABIEN, C/O KODAK INDUSTRIE, DEP-BREV	7483081

psn_sector	han_name	han_harmonized	han_id	person_address	person_etry_code
	DUPONT, Jean-Fabien Kodak Industrie	0	100169135	Département Brevets CRT - Zone Industrielle,F-711...	FR
	Dupont, Jean-Fabien	0	101271062	c/o Kodak Industrie, Dep. Brevets, CRT-Zone Ind.,7...	FR
	Dupont, Jean-Fabien, c/o Kodak Industrie, Dep-Brev	0	101271092	CRT 60/2, Zone Industrielle,71102 Chalon sur Sao...	FR

III°) Algorithme



User Story 2 : Adaptation de l'algorithme de désambiguïsation.

Numéro de story : US-02

Durée estimée : 3 Sprints (~21 jours)

Description :

En tant que utilisateur,
je veux utiliser un algorithme basé sur des règles pour identifier les inventeurs similaires,
afin de regrouper correctement les brevets par inventeur unique.

Prérequis : US-01

Critères d'acceptation :

L'algorithme peut détecter des correspondances basées sur le nom, l'adresse et l'affiliation et les éventuels co-inventeurs.

Compréhension de l'algorithme

Objectif : Regrouper tous les brevets se référant à un même inventeur.

Enjeux :

- Homonymie : plusieurs inventeurs avec le même nom
- Hétérogénéité des données : fautes de frappe, abréviations, traductions, etc.

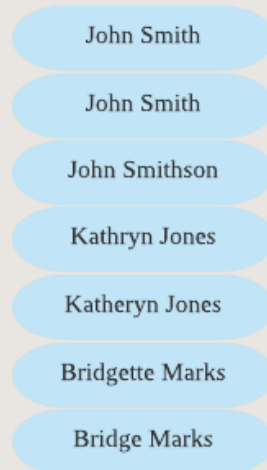
Méthodologie en 3 grandes étapes :

- **Canopy Clustering** : pré-filtrage pour limiter les comparaisons inutiles
- **Matching supervisé**
- **Clustering** (GRINCH) pour regrouper les id et brevet d'une même personne

Compréhension de l'algorithme

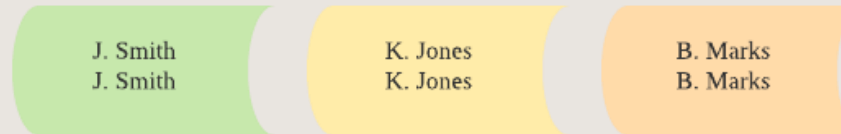
- Partie 1 : Création de "canopies"

Inventor Mentions



Grouping of Inventor Mentions into Canopies

Shared first initial, shared last name



Compréhension de l'algorithme

Partie 2 : Calcul de Similarité

- **Variables (features) :**
 - Nom complet
 - Co-inventeurs en commun
 - Localisation (ville, pays)
 - Organisation
 - Domaines techniques
- **Score final** : Probabilité que deux inventions soient la même personne

Compréhension de l'algorithme

Partie 3 : Clustering hiérarchique (GRINCH)

- Fonctionne en fusionnant les paires les plus similaires jusqu'à un seuil
- **Critère de fusion** : score de similarité $>$ seuil défini
- **Sortie** : Groupes d'inventions correspondant à un inventeur unique

Structure du code :

Build Features

```
python -m pv.disambiguation.inventor.build_assignee_features_sql  
python -m pv.disambiguation.inventor.build_coinventor_features_sql  
python -m pv.disambiguation.inventor.build_title_map_sql
```



Build Canopies

```
python -m pv.disambiguation.inventor.build_canopies_sql
```



Run clustering

```
wandb sweep bin/inventor/run_all.yaml  
wandb agent $sweep_id  
# or using slurm  
sh bin/launch_sweep.sh $sweep_id
```



Finalize

```
python -m pv.disambiguation.inventor.finalize
```



Compréhension du code

Deux éléments majeurs sont à prendre en compte :

- Les codes sont écrits pour émettre des requêtes SQL dans des base de données d'inventeur MySQL.

- La table SQL sur laquelle la requête est appliqué doit comporter les colonnes suivantes :

|uuid | patent_id | assignee_id | rawlocation_id | type |
name_first | name_last |

Build Features

```
python -m pv.disambiguation.inventor.build_assignee_features_sql  
python -m pv.disambiguation.inventor.build_coinventor_features_sql  
python -m pv.disambiguation.inventor.build_title_map_sql
```

Build Canopies

```
python -m pv.disambiguation.inventor.build_canopies_sql
```

Adaptation des requêtes SQL

Version originale du code	Adaptation
uuid	person_id
Patent_id	appln_id
name_first, name_last	Person_name

Les obstacles rencontrés

- Installation des module (grinch : `pip install git+https://github.com/iesl/grinch.git`)
- Utilisation de Weights and Biases

```
File "pv/disambiguation/inventor/run_clustering.py", line , in  
<module>  
    from pv.disambiguation.inventor.load_mysql import Loader  
ModuleNotFoundError: No module named 'pv'
```

IV°) Interface graphique



User Story 3 : Création d'une interface utilisateur simplifiée

Numéro de story : US-03

Durée estimée : 4 jours

Difficulté : 3/5

Importance pour le client : 4/5

Description :

En tant que chercheur en économie,

je veux disposer d'une interface graphique intuitive pour charger des données et exécuter l'algorithme,

afin de simplifier l'utilisation de l'outil sans avoir besoin de compétences en programmation.

Prérequis : US-02

Critères d'acceptation :

L'utilisateur peut charger des fichiers CSV via l'interface.

Les résultats sont exportables sous forme de tableau avec des IDs d'inventeurs désambiguïsés.

Algorithme de score de similarité

- Via le module FuzzyWuzzy.
- 1ere méthode : **fuzz.ratio()** , basé sur la distance de Levenshtein
 - Exemple : Plante plate -> 1 ; désambiguïsée, désambiguïser -> 1

Algorithme de score de similarité

- Via le module FuzzyWuzzy.
- 2eme méthode : `token_sort_ratio` :
 - **Découpe** les chaînes en **tokens** (mots séparés).
 - Trie les mots **par ordre alphabétique**.
 - **Recolle** les mots. Applique un **fuzz.ratio()** classique sur les chaînes triées.

Algorithme de score de similarité

- 3eme méthode : **fuzz.partial_ratio()** : recherche la **sous-chaîne la plus similaire** dans la chaîne la plus longue par rapport à la chaîne la plus courte.
 - Exemple : "apple" , "the big apple pie" -> 1

Améliorations possibles

- Ajouter des filtres pour une recherche plus précis (Adresse, domaine de recherche, etc...)

Avec une BDD désambiguïsé

- Pouvoir trouver des inventeurs avec un nom mal/autrement orthographié

Documentation et reproductibilité



- User Story 4 : Génération de rapports analytiques
- Numéro de story : US-04
- Durée estimée : 3 jours
- Difficulté : 2/5
- Importance pour le client : 4/5
- Description :
En tant que utilisateur,
je veux générer un rapport détaillé des résultats de désambiguïsation,
afin de comprendre les performances de l'outil et identifier d'éventuelles anomalies.
- Prérequis : US-03, US-02, US-01
- Critères d'acceptation :
 - Le rapport inclut des métriques (précision, rappel, etc.).
 - Une visualisation des groupes d'inventeurs désambiguïsés est fournie.

Discussion critique

Objectif:

- **Exploiter** la base de données des brevets européens
- **Désambiguïser** la base de données avec le code de **PatentsView**
- Créer une **interface graphique (GUI)**
- L'interface graphique accède la Bdd
Désambiguïsée pour suggérer un inventeur à un ou plusieurs brevets.

Discussion critique

Objectifs:

- **Exploiter** la base de données des brevets européens : **atteint**
- **Désambiguïser** avec le code de **l'USPTO** : **partiellement atteint**
- Créer une **interface graphique (GUI)** : **atteint**
- Le **GUI** accède à la Bdd **Désambiguïsée** pour suggérer un inventeur à un ou plusieurs brevets. **échec/atteint**

Discussion critique

- Le projet a montré les limites des approches classiques face à la complexité des données brevets.
- L'adaptation de l'algorithme PatentsView a bien débuté, mais son exécution reste à finaliser.
- **Perspectives** : finaliser le clustering dans un environnement Linux, adapter l'algorithme pour des données européennes, et intégrer des modèles d'IA pour améliorer la précision

Merci de votre attention !

Annexe 1.1 – Users stories 1 et 2

Importation et traitement des bases de données

Statut Done

Participants Vide

Taux d'avancement 100,00 %

Dates Vide

1 autre propriété

Commentaires

B Ajouter un commentaire...

User Story

- En tant que membre du projet, je veux pouvoir lire les données du fichier de sauvegarde pour ensuite nettoyer les données brutes des inventeurs de brevet.

Pour atteindre cet objectif, nous définissons les tâches suivantes :

- Extraire le fichier .bak.
- Lire la base de donnée pour en comprendre la structure et les champs associées à chaque table
- Chargement et prétraitement des datasets de brevets, incluant le nettoyage et la suppression des données dupliquées.

Algorithme de désambiguïsation

Statut In Progress

Participants Vide

Taux d'avancement 66,67 %

Dates Vide

1 autre propriété

Commentaires

B Ajouter un commentaire...

User Story

- En tant que chercheur, je veux appliquer l'algorithme de PatentsView pour désambiguïser les inventeur de la base de donnée de brevet européen.*

Pour atteindre cet objectif, nous définissons les tâches suivantes :

- Lecture du document de recherche afin d'assurer une compréhension de la méthodologie appliqué.
- Compréhension du code.
- Adaptation du code à notre cas.

Annexe 1.2 – Users stories 3 et 4

Interface Utilisateur

Statut

In Progress

Participants

Vide

Taux d'avancement

66,67 %

Dates

Vide

1 autre propriété

Commentaires

B Ajouter un commentaire...

User Story

- User Story : En tant qu'utilisateur, je veux une interface simple pour importer des fichiers et visualiser les résultats de désambiguïsation.*

Pour atteindre cet objectif, nous définissons les tâches suivantes :

- Conception et développement d'une interface utilisateur intuitive et ergonomique permettant d'interagir avec l'outil facilement.
- Ajout du module ElasticSearch pour assurer une fonctionnalité accrue.
- Tests d'ergonomie et d'accessibilité pour assurer une bonne expérience utilisateur.

Documentation & reproductibilité

Statut

Planning

Participants

Vide

Taux d'avancement

100,00 %

Dates

Vide

1 autre propriété

Commentaires

B Ajouter un commentaire...

User Story

- En tant que chercheur, je veux une documentation claire et détaillée afin de comprendre et utiliser efficacement l'outil.*

Pour atteindre cet objectif, nous définissons les tâches suivantes :

- Rédaction d'une documentation technique complète pour assurer la compréhension et la maintenance du code.
- Organisation du code pour faciliter sa réutilisation
- Création du PPT

Annexe 3 - Evaluation algorithme

- **Définitions :**
 - **Précision** = $TP / (TP + FP)$ → éviter les faux regroupements
 - **Rappel** = $TP / (TP + FN)$ → ne pas rater de vrais liens
 - **F1-score** = $2 * (Précision * Rappel) / (Précision + Rappel)$
- **Résultats observés normalement par USPTO** : F1-score > 0.95

Annexe 4 - Modification du code

```
1 def build_granted(config):
2     # | uuid | patent_id | assignee_id | rawlocation_id | type | name_first |
3     name_last | organization | sequence |
4     feature_map = collections.defaultdict(list)
5     cnx = pvdb.granted_table(config)
6     # if there was no table specified
7     if cnx is None:
8         return feature_map
9     cursor = cnx.cursor()
10    query = "SELECT uuid, patent_id, assignee_id, rawlocation_id, type,
11    name_first, name_last, organization, sequence FROM rawassignee"
12    cursor.execute(query)
13    idx = 0
14    for rec in cursor:
15        am = AssigneeMention.from_granted_sql_record(rec)
16        feature_map[am.record_id].append(am.assignee_name())
17        idx += 1
18        logging.log_every_n(logging.INFO, 'Processed %s granted records - %s
19        features', 10000, idx, len(feature_map))
20    logging.log(logging.INFO, 'Processed %s granted records - %s features', idx,
21    len(feature_map))
22    return feature_map
```

```
1 def build_granted():
2     # | uuid | patent_id | assignee_id | rawlocation_id | type | name_first |
3     name_last | organization | sequence |
4     cnx = pyodbc.connect(f'DRIVER={{SQL Server}};SERVER={server};DATABASE={
5     database};Trusted_Connection=yes;')
6     cursor = cnx.cursor()
7
8     query = "SELECT person_id, appln_id, person_name from patents_applicants"
9     cursor.execute(query)
10    feature_map = collections.defaultdict(list)
11    idx = 0
12    for person_id, appln_id, person_name in cursor:
13        parts = list(map(str.strip, person_name.split(",")))
14        nom = parts[0]
15        prenom = parts[1] if len(parts) > 1 else ""
16
17        rec = [str(person_id), str(appln_id), str(person_id), None, prenom, nom,
18        None, None, None]
19
20        am = AssigneeMention.from_granted_sql_record(rec)
21        feature_map[am.record_id].append(am.assignee_name())
22        idx += 1
23        logging.log_every_n(logging.INFO, 'Processed %s granted records - %s
24        features', 10000, idx, len(feature_map))
25    return feature_map
```

Annexe 5 – Classe InventorMention

```
1  def __init__(self, uuid, patent_id, rawlocation_id, name_first, name_last
2  , sequence, rule_47, deceased,
3      document_number=None, city=None, state=None, country=None):
4      self.uuid = str(uuid)
5      self.patent_id = str(patent_id).replace('\n', '') if patent_id else None
6      self.rawlocation_id = str(rawlocation_id).replace('\n', '') if
7      rawlocation_id else ''
8      self.raw_last = str(name_last).replace('\n', '') if name_last else ''
9      self.raw_first = str(name_first).replace('\n', '') if name_first else ''
10     if type(sequence) is int:
11         self.sequence = str(sequence)
12     else:
13         self.sequence = sequence.replace('\n', '') if sequence else ''
14         self.rule_47 = str(rule_47).replace('\n', '') if rule_47 else ''
15         self.deceased = str(deceased).replace('\n', '') if deceased else ''
16         self.name = '%s %s' % (self.raw_first, self.raw_last)
17         self.document_number = str(document_number)
18
19     self.mention_id = '%s-%s' % (self.patent_id, self.sequence) if self.
20     patent_id is not None else 'pg-%s-%s' % (
21         self.document_number, self.sequence)
22     self.assignees = []
23     self.title = None
24     self.coinventors = []
25
26     self._first_name = None
27     self._first_initial = None
28     self._first_letter = None
29     self._first_two_initials = None
30     self._first_two_letters = None
31     self._middle_name = None
32     self._middle_initial = None
33     self._suffixes = None
34     self._last_name = None
35
36     self.city = str(city)
37     self.state = str(state)
38     self.country = str(country)
39
40     self.record_id = self.patent_id if self.patent_id else 'pg-%s' % self.
41     document_number
```

Annexe 6 – Erreur Weight and Biases

```
1 wandb: Starting wandb agent
2 2025-04-05 21:38:15,062 - wandb.wandb_agent - INFO - Running runs: []
3 2025-04-05 21:38:15,413 - wandb.wandb_agent - INFO - Agent received command: run
4 2025-04-05 21:38:15,413 - wandb.wandb_agent - INFO - Agent starting run with
  config:
5     chunk_id: 0
6     chunk_size: 10000
7     min_batch_size: 1
8     run_id: run_24
9 2025-04-05 21:38:15,416 - wandb.wandb_agent - INFO - About to run command: /usr/
  bin/env python pv/disambiguation/inventor/run_clustering.py --chunk_id=0 --
  chunk_size=10000 --min_batch_size=1 --run_id=run_24
10 Traceback (most recent call last):
11   File "pv/disambiguation/inventor/run_clustering.py", line 12, in <module>
12     from pv.disambiguation.inventor.load_mysql import Loader
13 ModuleNotFoundError: No module named 'pv'
```

Annexe 7 – Distance de Levenshtein

$$\text{lev}(a, b) = \begin{cases} \max(|a|, |b|) & \text{si } \min(|a|, |b|) = 0, \\ \text{lev}(a - 1, b - 1) & \text{si } a[0] = b[0], \\ 1 + \min \begin{cases} \text{lev}(a - 1, b) \\ \text{lev}(a, b - 1) \\ \text{lev}(a - 1, b - 1) \end{cases} & \text{sinon.} \end{cases}$$