Gawaboumga

Geste de savoir

Résolution d'entité - Entity resolution

jeudi 04 juillet 2024

Table des matières

Introduction

Cet article est la première partie d'un diptyque concernant l'identification et la gestion des données redondantes dans des jeux de données. Dans un premier temps, on s'intéressera aux problèmes de résolution des entités ou, plus communément appelé, *entity resolution* au travers de ses enjeux, problèmes et solutions éventuelles. Dans la seconde partie du diptyque, on se concentrera sur les aspects liés à la gestion des données maîtres ou *master data management*.

Avec la multiplication des données sources et de leurs usages, il n'est pas rare de se retrouver avec l'information d'une même *entité* à plusieurs endroits au sein d'une même société. Par exemple, les vendeurs notent les informations concernant leurs clients dans des spreadsheets Excel. Ces clients sont alors intégrés dans les services informatiques que proposent l'entreprise et l'information se retrouve dans ses très nombreuses bases de données. Finalement, la comptabilité et le service de facturation emploient un ERP, complètement détaché du reste pour facturer au client. Maintenant, que se passe-t-il si l'on souhaite changer l'adresse de livraison ou ajouter une information en plus ? Il faut alors aller modifier aux différents endroits et espérer n'avoir rien oublié …

Afin d'éviter toute cette confusion, on aimerait bien pouvoir exploiter les données dans une source unique de vérité où il sera alors plus facile de faire des liens et de fournir de la valeur. Par exemple, on pourra ainsi mieux mesurer l'efficacité des campagnes marketing, de la politique de

prix sur les ventes et la logistique, … Simplifier les processus de mises-à-jour des données et ainsi économiser tant en temps humain que technique (économie de coûts opérationnels). La gestion des données maîtres ou *master data management* essaye de répondre à cette problématique en fournissant une uniformisation et une vue consistante des données et de leurs usages. On emploie souvent les termes de *golden record*, *master record* ou version consolidée de la donnée afin de désigner la version finale qui fait figure d'autorité.

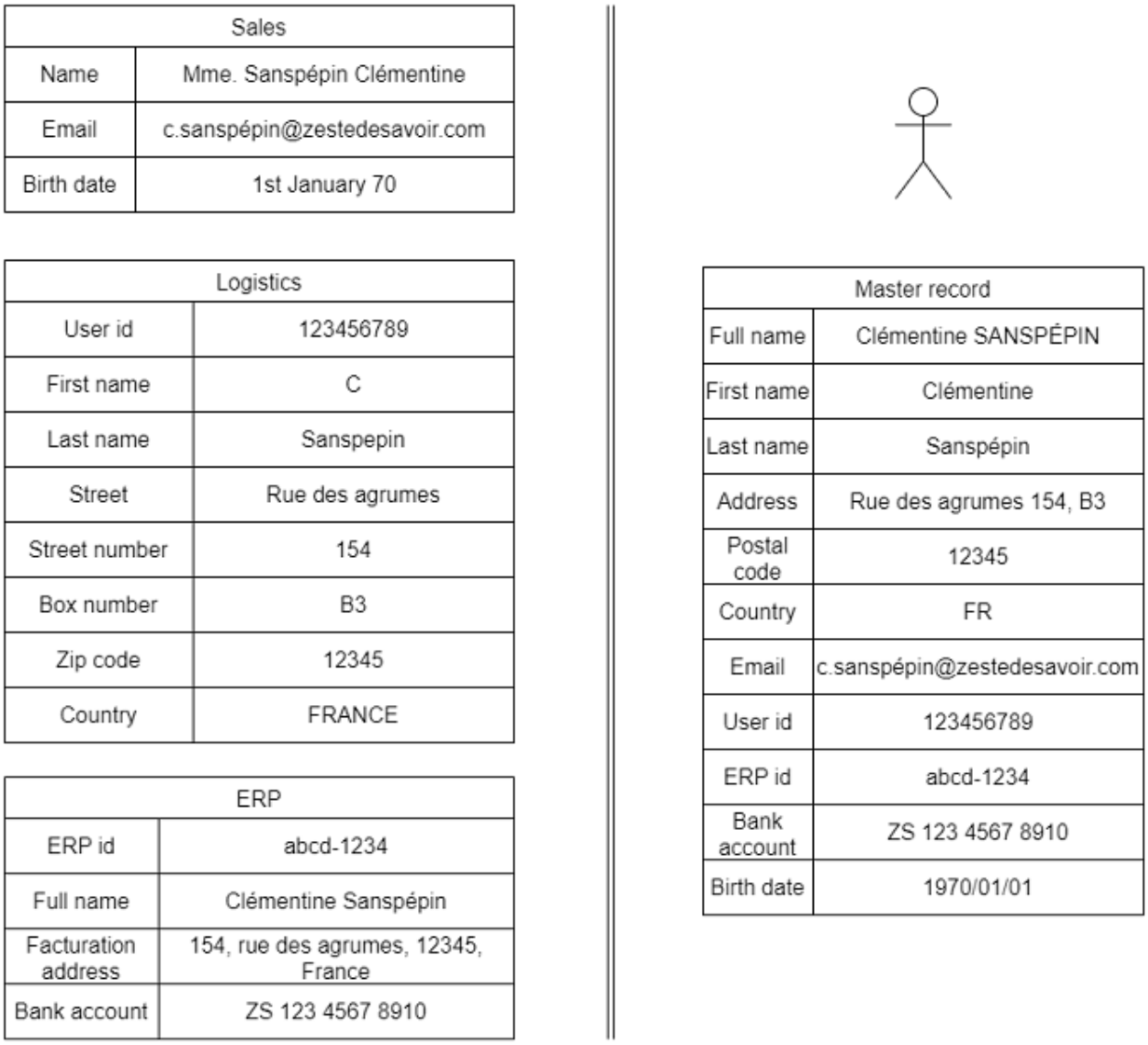

FIGURE 0.1. – On part d'informations parcellaires dans les différents systèmes et on forge une vision cohérente, un master record

Pour cela, il faut d'abord être capable d'identifier où apparaissent ces différentes *entités* au sein de l'entreprise. Ensuite, il faut chercher à faire correspondre les différentes versions d'une même entité en une vue unifiée. Ici, on s'attarde au premier aspect, comment faire en sorte d'identifier que IBM (USA) du département des ventes correspond bien à International Business Machines Corporation (États-Unis) du service comptabilité ? Et bien, ce concept s'appelle *entity resolution*, on cherchera à définir plus précisément ce concept et son importance dans la

gouvernance des données ainsi que les grandes étapes que l'on peut rencontrer et concepts qui lui sont liée.

Ensuite, on insistera tout particulièrement aux étapes de prétraitement de la donnée, en essayant de puiser un maximum d'information des sources qui peuvent exister. Il est illusoire de croire qu'un travail de qualité peut se passer de cette étape indispensable : *garbage in, garbage out*. On en profitera pour aborder un ensemble de problèmes que l'on peut rencontrer dans les processus d'*entity resolution* telles que l'ambiguïté, le **record linkage**, le **matching** ou la gestion des données plus ou moins cohérentes et de leur quantité.

Finalement, on présentera des pistes de solutions qui peuvent être mises en place afin de répondre à ce genre de thématiques, aux approches probabilistes et de machines learnings, ou celles davantage axées sur un ensemble de règles définies, ou quelque chose de plus hybride.

On espèrera qu'à la fin, vous aurez une meilleure compréhension de la thématique et des solutions disponibles afin d'assurer une gestion fiable et précises des données et de leur gestion. Dans la deuxième partie du diptyque, nous verrons comment nous pouvons combiner les informations des différentes entités résolues afin de présenter la vue unifiée.

1. Définitions et concepts

1.1. Définition & enjeux

La *résolution d'entité* ou *entity resolution* est le processus qui consiste à identifier et lier différentes données référençant une même *entité* au sein de multiples jeux de données. Cela se traduit par l'analyse et la comparaison d'attributs et de caractéristiques qui permettent de déterminer si une entité est unique ; et si plusieurs sources mentionnent bel et bien la même entité. Les objectifs sont multiples :

— Supprimer les doublons et informations redondantes. Ce qui peut éventuellement permettre de réduire les coûts en simplifiant les opérations de toute part ; l'information se trouve à un seul endroit unique et donc réduire les opérations manuelles ; ce qui permet, *in fine*, de mieux gérer les changements au sein des données[1].
— Améliorer la qualité globale des données afin de soutenir l'analyse des données et la prise de décisions.
— Créer une vue unifiée des données, afin de faciliter l'intégration et la consolidation de ces dernières et éventuellement satisfaire des exigences de régulation[3].

On emploiera le terme « entité » pour désigner le groupe résolu de « *records* » (les enregistrements bruts) et « *cluster* » pour désigner sa cohorte (l'ensemble des *records* constituant cette *entité*). Généralement, les entités représentent un concept primordial, qui a besoin d'être identifié de manière unique, comme un individu, une société, une adresse, un produit, un compte en banque, un événement médiatique, …

1. Les anglophones parles de *data drift*.
2. Vous entendrez souvent parler de *compliance*.

1.1.1. Enjeux

Comme le monde est complexe, avant de se lancer dans un un processus d'*entity resolution*, nous devons définir notre objectif et notre cadre :

— Vaut-il mieux minimiser les faux positifs ou les faux négatifs ?
— Vaut-il mieux assimiler trop de records ensemble quitte à faire des erreurs ou être précautionneux mais manquer de nombreuses relations ?
— Quels sont les risques en cas d'erreurs ?
— Sera-t-on capable de les résoudre manuellement ?

1.2. Étapes

L'*entity resolution* se compose généralement de plusieurs grandes étapes classiques mais qui peuvent être optionnelles ou répétées aux besoins, avec des paramètres ou critères qui diffèrent. On retrouve :

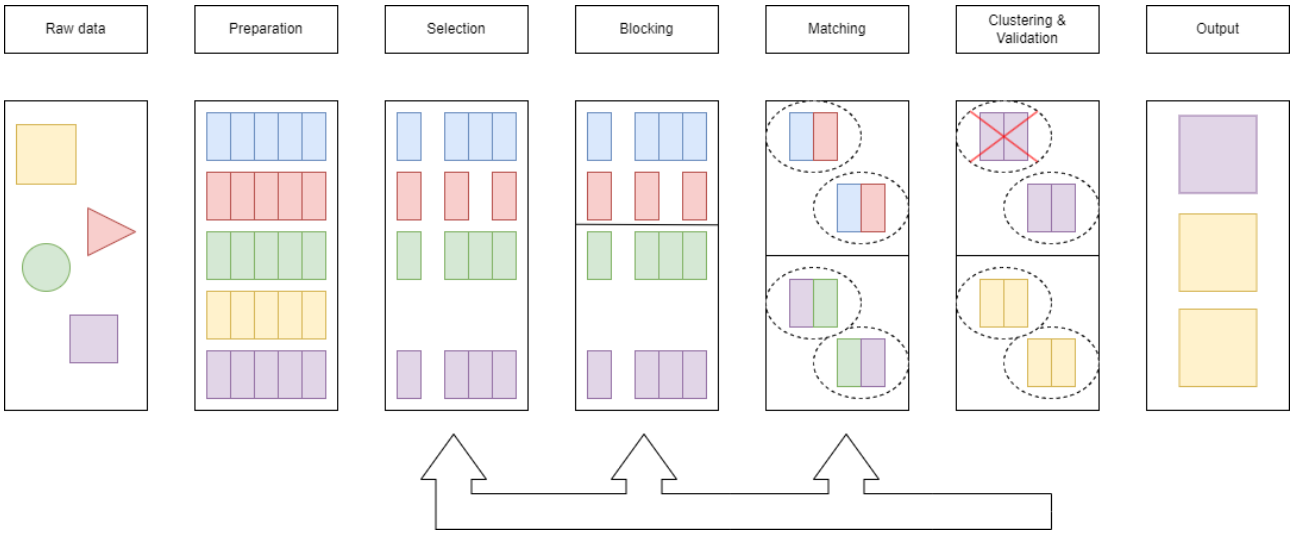

FIGURE 1.2. – Processus de résolution des entités

1.2.1. Prétraitement des données

Étape plus que primordiale, elle consiste au nettoyage et à la standardisation des données en entrée afin d'assurer une certaine consistance et compatibilité entre des données qui peuvent provenir de sources souvent très hétérogènes en qualité. Le but étant évidemment de fournir des données ayant un maximum d'information exploitable afin d'aider les différentes étapes qui suivront ; cela se traduit par la suppression de données invalides, la correction des erreurs, la gestion des valeurs manquantes ou mal classifiées. On fait généralement la distinction entre :

— Le *cleansing* : le nettoyage des données consiste à la détection et à la rectification des erreurs à proprement parler (comme des doublons ou des erreurs typographiques), ou la gestion des données qui n'ont pas de valeurs sémantiques : n.a. pour 'not available' ou encore la gestion des problèmes de consistances (comme un mauvais code postal associé à une ville).

— La *normalization* : généralement liée à des aspects plus techniques comme l'encodage en UTF-8, la suppression des 'espaces' en trop, … Cela s'accompagne le plus souvent par la décomposition en unités atomiques d'information telle que la scission d'une adresse en rue, ville, code postal, …
— La *standardization* : le but est d'obtenir une représentation commune d'une donnée, peu importe sa forme originelle, par exemple : numéro de téléphone en +XXX, code postal luxembourgeois de type L-XXXX ou n'avoir que des « avenue » et non « av. », … Ou encore, traiter des problèmes de translittération, traduction ou de champs multi-langues. L'important est de résoudre toutes les variations que l'on peut retrouver dans la représentation de la donnée afin de ne proposer qu'une vue 'unique'.

	Brut	**Cleansing**	**Normalization**	**Stan**
Nom complet	ClÃ©mentine, SanspÃ©pin	ClÃ©mentine, SanspÃ©pin	Clémentine Sanspépin	Clém
Nom	/			Clém
Prénom	Unknown			Sansp

1.2.2. Sélection

Pour démarrer le processus d'*entity resolution*, il faut bien commencer par prendre un record et sélectionner tous ceux qui lui sont affiliés. Toute l'astuce réside dans la définition du concept de similarité/dissimilarité afin de retrouver tous les records qui pourraient être des candidats à cette résolution, à ce cluster. Il y a tout un spectre de puissance d'identification, des éléments qui sont fortement discriminants : un code de sécurité social, un numéro de série ou un numéro de passeport, d'autres qui ne prennent leur force que par leur combinaison : même date de naissance, prénom et nom de famille et enfin ceux qui n'ont que très peu de valeur : numéro de téléphone de la société pour laquelle la personne travaille. Une grande partie de la subtilité réside également dans le contexte dans lequel ces informations apparaissent : un nom de famille peut être très commun (M. Leclercq) en France, mais très peu présent au Japon.

1.2.3. Blocking.

C'est une étape complémentaire au processus de sélection, elle consiste à diviser le jeu de données (éventuellement sélectionnés) en plus petits ensembles sur base de critères ou attributs. Cette étape fait surtout sens lorsque l'on a des indications additionnelles sur ce que l'on cherche (code d'activité d'une entreprise ou société pour laquelle une personne travaille). Elle vise principalement à réduire la complexité en réduisant l'espace de recherche et déjà trouver des premiers ensembles. Si un nombre insuffisant de résultats ressort de cette opération, on peut revenir à l'étape de sélection en élargissant le spectre et ainsi faire un ping-pong.

1.2.4. Similarité ou matching

Une fois que des records potentiels sont identifiés, on peut chercher à identifier à quel point des éléments sont similaires ou dissimilaires entre eux. De nouveau, toute une couche d'arbitraire

s'opère à cette étape, tant sur la comparaison des champs en eux-même que sur la combinaison de ces différentes métriques en un score « unique ».

On peut par exemple sep oser les questions suivantes :

— deux dates de naissance sont similaires si on ne possède que l'année ? Quid des formats américains MM/DD/YYYY vs civilisés DD/MM/YYY ?
— des chaînes de caractères, quelles métriques employer ? Levenshtein, les indels, Jaro–Winkler ?
— deux adresses peuvent être fort différents mais référencer la 'même' chose (p.o. box vs adresse 'physique').

Lors de l'étape de *matching* il faudra enfin décider s'il vaut-mieux avoir quatre champs identiques, mais qui ne diffèrent que par un seul autre, ou ne posséder que l'information sur un champ « fort » même si tous les autres diffèrent ?

1.2.5. Évaluation et validation

Une fois que les records qui forment un cluster ont été identifiés, est-ce que certains éléments ne peuvent pas être regroupés pour une raison extérieure ? Approche prudente des identificateurs forts (numéro de passeport) afin d'éviter de se retrouver avec deux numéros de passeport pour un même individu ou dissimilarité trop forte au niveau des âges ou adresses. Si le cluster est dégénéré et contient un nombre anormalement élevé de records, est-il pertinent ? Ou alors selon des critères externes, si on suppose que l'entité est ainsi résolue, est-ce que l'information est cohérente ?

Exemple : Jeff Bezos est un nom assez rare mais si je vois qu'un record est lié à un professeur du secondaire et l'autre au patron d'Amazon, peu de chances qu'il s'agit du même individu en pratique. C'est une étape qui peut être plus ou moins manuelle et nécessite une expertise très forte de la part d'un individu afin d'affirmer ou non la pertinence de la résolution.

1.2.6. Génération des résultats

Finalement, on veut récupérer l'ensemble des entités qui ont été résolues, afin de pouvoir être exploités à d'autres fins et ainsi proposer une vue holistique des entités au travers des différentes sources de données. Il peut également être important d'enregistrer les raisons qui ont mené à la création de cluster en particulier à des fins d'audit.

2. Prétraitement des données

On ne le répètera pas suffisamment assez mais le traitement des données en amont de l'entity resolution joue un rôle plus que primordial dans le résultat de ce dernier. Malheureusement, cette étape est particulièrement fastidieuse et ne reçoit pas l'attention qu'elle devrait, on se contente souvent de données plus que moyennes en pratique … Amusant à l'heure à laquelle des grosses sociétés d'Intelligence Artificielle (qui servent souvent d'inspiration au senior management) prennent justement le temps d'améliorer leurs données …

2.1. Données

Voici une liste non-exhaustive de toutes les absurdités que l'on peut retrouver classiquement au niveau des données en elles-mêmes :

- Problèmes techniques : principalement issu d'un encodage erroné, de trop nombreux systèmes ne parviennent toujours pas à gérer la présence d'accents (ou de caractères Unicode) en 2023 … Un autre grand classique est la présence de caractères non désirés tels que des retours à la ligne, des tabulations, ou de délimitations …
- Données incomplètes : il a bien fallu qu', un jour, quelqu'un décide d'inscrire les données. Il n'est donc pas rare d'avoir des informations manquantes ou qui n'ont pas été remplies à l'époque parce que l'information était encore inconnue.
- Données inconsistantes ou erronées : plusieurs phénomènes sont possibles, des données peuvent contenir des erreurs (potentiellement volontaires), fautes (typographiques), mal formatées (au combien de personnes indiquent leur nom à la place de leur prénom et réciproquement) ou qui ne font pas de sens parce que contradictoire (ville et code postal qui ne correspondent pas).
- Données outre-datées : on se retrouve avec de l'information qui n'est pas à jour et plus forcément pertinente.
- Données non-pertinentes ou inaccessibles : certaines personnes cherchent volontairement à réduire la qualité de certaines données en introduisant des données non-pertinentes et tentant de noyer le poisson. D'autre part, des cadres légaux peuvent simplement forcer l'absence de certains champs (GDPR) ou le caviardage de ceux-ci (droit à l'oubli).

D'autre part, si on possède une donnée, celle-ci peut se présenter sous différentes formes. Il n'est pas rare d'avoir plusieurs langues au sein d'un même pays, ou plusieurs systèmes d'écriture, voire de numérotation. Comment traiter les problèmes de traduction, de translittération ou de graphie / représentation ?

2.2. Concepts

Finalement, au niveau des champs en eux-mêmes, on se retrouve souvent confronté aux diversités du monde. Et des choses qu'on pense pourtant bien défini, commencent à prendre des sens beaucoup plus flous …

- Les noms : l'exemple classique est la séparation en "prénom" et "nom", certaines cultures ne font tout simplement pas cette distinction. Mais cela s'étend bien au delà. On peut citer la question des honorifiques (Mme., Dr., …), des affixes (mac, -son, …), des suffixes (Jnr., III, …), les noms de jeunes filles, les surnoms ou les diminutifs (hypocorisme), … ou même s'ils ont simplement un nom.
- Les adresses : de nombreuses régions du monde possède une autre forme d'adresse. En effet, pourquoi ne pas simplement employer des coordonnées géographiques ? Au Japon, il n'y a pas vraiment de distinction entre la ville, le code postal, la rue, … le tout forme l'adresse. Tous les pays ne possèdent pas de systèmes de code postal, de boîte postale (p.o. box), de 'c/o' pour 'care of', …
- Les numéros de téléphone : il n'est pas rare qu'une même personne possède plusieurs numéros de téléphone et qu'un même numéro de téléphone correspond à plusieurs personnes.

Il est d'usage de taper : 'falsehood' accompagné du type de la donnée afin d'avoir une vague idée de ce qu'il serait possible de retrouver dans ce champ.

2.3. Abréviations et synonymes

Quand on cherche à standardiser une donnée, il faut également prêter attention aux abréviations et aux synonymes. L'astuce étant que celles-ci dépendent souvent à minima de la langue employée et, le plus souvent, également du contexte socio-culturel / du pays. Une même abréviation peut exister dans de très nombreux pays (S.A. pour sodiedad anonima) mais avoir des significations fort différentes (Son Altesse). De même, les synonymes présentent des difficultés analogues, mais avec une variation souvent encore plus grande. Il faut souvent aller plus loin d'une simple liste de correspondances pour résoudre ce genre de problèmes, un arbre de dépendance est déjà un meilleur choix.

2.4. Guesstimates et complétion

Enfin, on peut chercher à inférer des informations sur base de celles fournies. Que ce soit pour gérer des problèmes d'abréviations ou améliorer la standardisation. Il est plus simple d'extraire un code postal quand on sait que l'adresse est belge que si l'adresse est émirati (et il n'y a de fait, pas de code postal). On peut également tenter d'inférer des informations comme, sur base de l'adresse, chercher à remplir le code postal ou la région. Toute information peut éventuellement servir à améliorer la qualité des données. Mais il est d'usage de bien noter quelles informations ont été fournies, celles construites et celles inférées dans les processus.

3. Entity resolution

Une fois que les données sont à peu près propres, on peut enfin s'attaquer à la pièce de résistance, la partie *entity resolution*. Sur base d'un ensembles de records, on cherche à créer un ensemble de clusters tels que chacun correspond à une entité cohérente. On parle généralement de **linkage** pour cette étape puisqu'on va relier le concept d'une entité avec tous les records qui la composent.

Pour cela, de nombreuses techniques existent mais se regroupent en deux grandes catégories obéissant à des concepts assez similaires.

3.1. Principe général

En soit, le processus d'entity resolution n'est pas particulièrement complexe à comprendre. Mais le diable se cache dans les **très** nombreux détails qui le composent, ce qui le rend particulièrement difficile à implémenter en pratique. L'optimisation est également particulièrement complexe, naïvement, la complexité est au minimum quadratique, traite essentiellement des chaînes de caractères, et les dépendances contraignent certaines possibilités de parallélisme. Bref, ce n'est pas une mince affaire mais dans les grandes lignes :

— On prend une entité, et on identifie ce qui la rend unique (combinaison de différents attributs).
— On cherche celles qui partagent un ou plusieurs de ces attributs.
— On mesure alors le niveau de similarité entre ces différentes entités afin de définir des clusters.
— On répète ces étapes avec les entités restantes jusqu'à avoir tout traité.

3.2. Approches basées sur un ensemble de règles

On peut assez facilement imaginer des algorithmes qui prennent différentes règles (écrites manuellement) et qui commencent par les combinaisons d'attributs les plus discriminants et relâchant les règles jusqu'à avoir obtenu un résultat suffisant. Le peaufinage de la précision et du rappel (precision-recall / accuracy - l'arbitrage entre être précautionneux ou agressif dans la constitution de *clusters*) est alors assez simple puisqu'on peut facilement mesurer quelles règles influencent la formation des différentes entités.

Les avantages sont assez forts :

— Transparence : chaque étape peut être comprise et expliquée ; ce qui facilite très fortement les problèmes d'audit.
— Flexibilité : on peut facilement ajouter ou modifier des règles en fonction des données ou des besoins spécifiques et ainsi proposer plusieurs niveaux d'*entity resolution*.
— Contrôle : l'utilisateur a le contrôle complet sur la méthode, il peut décider de gérer des cas particuliers à son bon vouloir ou peut affiner les règles afin d'obtenir les résultats souhaités.

Mais les désavantages ne sont pas sans équivoques :

— Efficacité et scalabilité : avec l'augmentation de la taille des données et des cas qui peuvent être rencontrés, il faut essayer de garder une complexité computationnelle assez restreinte. L'implémentation étant déjà assez complexe, chercher à optimiser les opérations sur des ensembles devient vite difficilement gérable.
— Maintenance et compréhension : créer et gérer les règles pour chaque scénario possible peut se relever particulièrement difficile puisque le moindre changement à un endroit peut avoir des conséquences un peu partout. D'autant que les données ou les besoins peuvent évoluer avec le temps, ce qui prend du temps tant pour la partie business que celle technique.
— Difficulté de gestions des ambiguïtés : il existe tout un panel de données particulièrement floues qui obéissent éventuellement à plusieurs motifs en même temps. Multiplier toutes les possibilités est une course sans fin mais parfois nécessaire pour régler les 10% restants. Ce qui est d'autant plus gênant que des variations inattendues peuvent toujours survenir.

3.3. Approches basées sur le Machine Learning

Puisqu'une grande partie de la difficulté réside dans la définition et la gestion des règles, pourquoi ne pas déléguer cette tâche au Machine Learning ? Cela peut s'appliquer à plusieurs niveaux : au niveau du matching, de l'évaluation, de la recherche d'éléments similaires ou de l'exploitation des relations comme au travers des *graph convolutional networks* . Il y a clairement des questions

probabilistes qui interviennent dans ce domaine puisqu', intrinsèquement, des noms de famille sont plus rares que d'autres, des combinaisons d'attributs peuvent également l'être en fonction du contexte, …

Voici quelques avantages :

— Précision : en théorie, les algorithmes de machine learning sont capables de reconnaître des motifs plus complexes au sein des données et de mieux exploiter les relations.
— Scalabilité : Les infrastructures employées pour le machine learning sont généralement plus lourdes et permettent une meilleure mise à l'échelle. Même si certains modèles de langage ne sont pas connus comme étant particulièrement rapides …
— Automatisation et flexibilité : une fois un objectif atteint, il est plus simple de sauvegarder le modèle entraîné à un instant T. Introduire de nouvelles données ou motifs n'est plus un soucis qui doit être reconsidéré dans son ensemble, il suffit de réentraîner le modèle. On gagne surtout en temps humain du côté business.

Néanmoins, nous imaginons bien quelques désavantages que cette technique propose :

— Manque de transparence : le machine learning est généralement une boîte noire, dont la prise de décision est souvent difficile, ce qui restreint fortement les possibilités d'audits. Et les conséquences au niveau business ne sont pas des moindres puisque les résultats peuvent être complètement inattendus.
— Surapprentissage : les modèles de machine learning peuvent mener à du surapprentissage (*overfitting*), ce qui risque de mener à des résultats moindres lors de l'ajout de nouvelles données et il faudra alors relancer tout le processus d'apprentissage.
— Expertise : il faut une certaine expertise afin de peaufiner les modèles et leurs paramètres tout en ayant aucune garantie sur les résultats qui seront obtenus.

3.4. Remarques additionnelles

Les approches basées sur des ensembles de règles et le machine learning ne sont pas en opposition. On peut éventuellement penser à employer le machine learning afin d'aider à la découverte de ces règles ou restreindre le modèle sur base de règles qui font du sens pour le business.

En outre, il y a deux problèmes assez difficiles à régler pour les deux méthodes. La gestion est très empirique : tant qu'on n'a jamais vu de cygne noir, tous les cygnes sont blancs. Il est difficile d'investiguer quels sont les clusters qui sont erronés s'ils sont noyés dans une masse gigantesque. Et remonter à la compréhension sur ce qui a mené à cette situation peut être particulièrement consommateur en temps.

D'autant plus que les questions de l'évaluation sont souvent complètement éludées. En effet, quels sont les critères qui font que cette partition-là des records est la bonne, beaucoup de cas sont souvent très limites et il n'y a que peu de raisons de préférer l'un à l'autre. En particulier, on navigue souvent à vue puisqu'on ne possède pas d'horizon clairement défini juste des règles assez floues (éviter le regroupement d'entités sans rapport ou essayer de regrouper un maximum les entités entre-elles). Monitorer tant les *data drifts* que l'évolution des modèles et des clusters formés est très loin d'être une opération triviale.

La problématique des contrôles de qualité et d'assurance qualité sont loin d'être triviaux dans ce domaine. On cherchera à définir une mesure de partitionnement, puisqu'on connaît tous les éléments (records) mais pas les classes (entité), quelque chose qui peut se rapprocher des notions

de type « Jaccard Index » ou « Mutual information score », sachant pertinemment qu'on ne possède pas « LA » solution. On peut se limiter à une approche qualitative des groupes pris au hasard avec des tailles représentatives, l'analyse plus approfondie des entités importantes pour le business ou la présence d'entités pathologiques regroupant des milliers de records en un.

Rien n'empêche également de travailler avec « deux » systèmes d'*entity resolution*, un précautionneux fournissant les données officielles pour la production et un autre, reprenant les règles du premier, mais en étant plus laxiste. Cela permet de mieux constater quelles règles pourraient être pertinentes à rajouter en offrant une autre vision sur la donnée, tenter d'éclaircir les zones floues qui peuvent exister. Mieux trancher des cas transmis par des clients vis-à-vis de résolutions manquées et voir s'il faut appliquer un traitement *ad hoc* ou global à la donnée.

Conclusion

On espère qu'avec cet article vous aurez une meilleure vue du processus d'*entity resolution*, de ces problèmes, enjeux et comment aborder le problème. On a conscience d'avoir omis d'aborder de nombreuses thématiques, qui nécessitent quand même un regard : on pensera aux problèmes de respect de la vie privée et considérations éthiques ; comment gérer la modification des clusters générés (qui peuvent être créés, défaits, puis recréés) ; les problèmes de GDPR et de droit à l'oubli, et plus globalement les problèmes liés à la gouvernance des données ; l'importance de l'engagement des différentes parties-prenantes (*stakeholders*).

On ne saura insister sur l'importance d'avoir des données de qualité afin de conduire au meilleur résultat possible. Si cette étape est bâclée, ce qui sera construit autour de l'*entity resolution* verra son impact amoindri, ce qui ne pourra que nuire à la prise de décisions et la valeur ajoutée des produits. Mesurer la précision et la robustesse du système est aussi critique afin d'éviter les entités pathologiques ou *dégénérées* composées d'un trop grand nom de records ou maintenir un œil sur les entités qui incombent le plus au business.

Liste des abréviations

ERP Enterprise Resource Planning. 1