Construction d'un graphe de dépendances fonctionnelles à partir de tableaux web

Tarek Benkhelif*, Clara Lebeau**

*Talend, 89 Boulevard de la Prairie au Duc, 44200 Nantes
tbenkhelif@talend.com
**clara.lebeau@hotmail.fr

Résumé. La découverte de dépendances est au cœur de nombreux efforts de profilage et de nettoyage des données. Parmi les dépendances les plus importantes pour les bases de données relationnelles, on trouve les dépendances fonctionnelles (DFs) qui représentent des contraintes entre les attributs d'un modèle de données relationnelles. Dans cet article, nous proposons une méthodologie permettant de construire à partir d'un corpus de tableaux web, une base de connaissance qui prend la forme d'un graphe dont les nœuds sont des types sémantiques et dont les arêtes représentent l'existence d'une dépendance fonctionnelle relaxée.

1 Introduction

L'exploitation des données par les entreprises privées et les acteurs du secteur public connaît une croissance exponentielle, les données sont un atout crucial et leur nettoyage est aujourd'hui devenu un enjeu majeur. Au cours des deux dernières décennies, des recherches intensives ont été menées pour développer des algorithmes et des outils de nettoyage des données. La découverte de dépendances dans un ensemble de données est au cœur de nombreux efforts de profilage et de nettoyage des données. Parmi les dépendances les plus importantes pour les bases de données relationnelles, on trouve les dépendances fonctionnelles (DFs) et les dépendances fonctionnelles relaxées (DFRs). Ces dernières représentent des DFs assouplies sur la méthode de comparaison et/ou sur la contrainte de satisfaction.

Deux types de colonnes peuvent être identifiées dans les tables de données : *atomiques* et *sémantiques*. Les types atomiques tels que les *chaînes de caractères*, les *booléens* ou les *entiers* donnent des informations techniques sur la structure des données stockées dans une table. Tandis que les types sémantiques tels que le *lieu*, la *date de naissance* ou le *nom* donnent des informations plus riches et précises sur les colonnes.

Il est très coûteux de calculer les dépendances fonctionnelles relaxées quand le nombre de colonnes dans les données devient important. Par exemple, Kruse et Naumann affirment dans (Kruse et Naumann, 2018) que leur méthode excéderait les 30 heures d'exécution pour un jeu de données contenant 35 colonnes et $\sim$30 000 lignes. Pour pallier ce problème, nous proposons dans cet article, une méthodologie permettant de construire une base de connaissance qui répertorie les DFRs fréquemment retrouvées dans les tableaux web.

Les contributions principales de notre travail sont (1) l'idée de construire une base de connaissance qui répertorie les DFRs qui occurrent fréquemment entre des colonnes de données caractérisées par leur type sémantique (2) l'idée d'exploiter cette base de connaissance, pour vérifier les DFRs connues sur de nouvelles données, moyennant une phase de détection de type sémantique (3) l'évaluation empirique de quelques stratégies pour la construction d'une telle base de connaissance. L'article est organisé comme suit. Dans la section 2, nous présentons un aperçu des travaux de recherche portant sur l'exploitation des tableaux web, les problèmes de détection de types sémantiques et celui de la découverte de dépendances fonctionnelles. Nous introduisons ensuite, dans la section 3, la méthodologie proposée. Nous poursuivons, dans la section 4, par les expérimentations et les évaluations menées sur le corpus VizNet. Enfin, nous concluons et présentons nos perspectives.

2 Travaux antérieurs

2.1 Tableaux web

Le World Wide Web est constitué d'une énorme quantité de données structurées sous forme de tableaux HTML (Nishida et al., 2017). Les premières études (Cafarella et al., 2008) ont examiné 14 milliards de tableaux et ont montré que 154 millions d'entre eux contenaient des données relationnelles. Des travaux récents, explorant la puissance des tableaux web, mettent en évidence des applications intéressantes : recherche de tableaux (Chapman et al., 2020), extension de tableaux (Lehmberg et al., 2015) et augmentation de la base de connaissance utilisée pour permettre l'interprétation sémantique du contenu des tableaux (Lehmberg, 2019). L'extraction de tables relationnelles depuis le web est une tâche longue et compliquée. Nous avons donc décidé d'utiliser le répertoire de données VizNet. VizNet est un grand corpus de tables web de bonne qualité (650 GB) extraites et rassemblées par Hu et al. (Hu et al., 2019) dans le but de faciliter les travaux de recherches en apprentissage automatique.

2.2 Détection de types sémantiques

La détection de types sémantiques est une étape cruciale vers l'automatisation du nettoyage de données. De nombreux systèmes utilisent des méthodes de comparaisons tels que les expressions régulières ou les dictionnaires pour détecter des types sémantiques au sein d'une table.

Le hachage sensible à la localité La découverte du type sémantique d'une colonne de données peut être vue comme un problème de recherche de similarité entre l'ensemble des valeurs qui apparaissent dans la colonne, et un dictionnaire qui contient les types sémantiques auxquels nous nous intéressons, attachés aux valeurs qui les définissent. L'idée de base du LSH (Indyk et Motwani, 1998) est d'utiliser des fonctions de hachage qui mettent en correspondance des objets similaires dans les mêmes segments de hachage avec une forte probabilité. L'exécution d'une recherche de similarité sur un index LSH se fait en deux étapes : (1) utiliser les fonctions LSH pour sélectionner des objets « candidats » pour une requête q donnée, et (2) classer les objets candidats en fonction de leur distance à q.

Sato Les méthodes basées sur des dictionnaires de valeurs sont peu robustes pour fonctionner avec des tables de mauvaise qualité et ne permettent pas de détecter un grand nombre de types sémantiques. Hulsebos et al. (Hulsebos et al., 2019) proposent l'utilisation d'un réseau de neurones, Sherlock, permettant d'améliorer grandement la qualité de la détection de types sémantiques en comparaison aux méthodes traditionnelles. Sherlock opère une détection uni-colonne basée sur l'extraction de 1 588 caractéristiques décrivant les propriétés statistiques, la distribution des caractères, l'imbrication des mots et des vecteurs rassemblant les valeurs d'une colonne. Zhang et al. (Zhang et al., 2019) vont plus loin en proposant un algorithme de détection multi-colonnes : Sato. Le but de cet algorithme est de détecter le type sémantique d'une colonne en se basant à la fois sur les valeurs de celle-ci, en utilisant un modèle de prédiction uni-colonne tel que Sherlock, mais aussi en se basant sur les valeurs des autres colonnes qui composent la table relationnelle. Ainsi, en utilisant une technique de modélisation de sujets, « Latent Dirichlet Allocation » (LDA) (Blei et al., 2003), cette méthode permet d'introduire un contexte à la détection.

2.3 Dépendances fonctionnelles relaxées

Étant donné un schéma de relation $R(A_1, A_2, \ldots, A_n)$, X et Y des sous-ensembles du groupe d'attributs $A_1, A_2, \ldots, A_n$, on dit que X détermine Y (ou Y dépend fonctionnellement de X) si et seulement s'il existe une fonction qui à partir de toute valeur de X détermine une valeur unique de Y. On note $X \rightarrow Y$. Les dépendances au sein d'un jeu de données réelles sont souvent approximatives. Découvrir des DFs dans des données réelles, avec leur lot d'incohérences, nécessite une relaxation de ces dernières. L'étude de Caruccio et al. (Caruccio et al., 2016) rassemble 35 types de DFRs et les classe selon leurs domaines d'applications et contraintes relaxées. Nous pouvons distinguer trois types de relaxations :

— la relaxation sur la contrainte de satisfiabilité, autorisant le fait qu'une contrainte ne soit pas satisfaite sur un sous-ensemble de tuples ;
— la relaxation sur le type de comparaison utilisée pour différencier les tuples et autorisant la similarité entre tuples plutôt que l'égalité stricte ;
— la relaxation hybride sur les contraintes de satisfiabilité et de comparaison.

La complexité de la découverte de DFRs augmente avec le nombre de relaxations autorisées. Il existe des algorithmes effectuant la détection d'un type spécifique de DFs ou DFRs. Carrucio et al. ont proposé l'algorithme Dime (Caruccio et al., 2020). Cet algorithme part de seuils spécifiés par l'utilisateur, et effectue une génération niveau par niveau des dépendances candidates à valider successivement. Pyro (Kruse et Naumann, 2018) implémente une stratégie de recherche « diviser pour mieux régner » guidée par échantillonnage qui détecte rapidement les dépendances candidates et les vérifie. Dans le cadre de ce travail, nous nous intéressons uniquement aux dépendances relaxées sur la contrainte de satisfiabilité.

Évaluation et mesures d'erreurs Afin d'évaluer les DFRs candidates, plusieurs mesures d'erreurs ont été proposées. Considérons les trois mesures g_1, g_2 et g_3 à valeurs dans $[0, 1]$, dont les spécificités sont expliquées dans l'article de Kivinen et al. (Kivinen et Mannila, 1995). Pour une table relationnelle *r*, si l'erreur vaut 0, alors la DF $X \rightarrow Y$ est respectée sur r. Si l'erreur se rapproche de 1, alors pour tous les tuples $u, v \in r$ avec $u \neq v$, on a $u[X] = v[X]$ et

$u[Y] \neq v[Y]$ et la DF n'est pas du tout respectée. Ainsi, nous pouvons affirmer que les DFRs les plus pertinentes seront celles ayant des mesures g_i proches de 0.

$$G_1(X \to Y, r) = |\ \{\ (u, v)\ |\ u, v \in r, u[X] = v[X], u[Y] \neq v[Y]\ \}\ |, \tag{1}$$

$$g_1(X \to Y, r) = G_1(X \to Y, r)/|r|^2, \tag{2}$$

$$G_2(X \to Y, r) = |\ \{\ u\ |\ u \in r, \exists v \in R : u[X] = v[X], u[Y] \neq v[Y]\ \}\ |, \tag{3}$$

$$g_2(X \to Y, r) = G_2(X \to Y, r)/|r|, \tag{4}$$

$$G_3(X \to Y, r) = |r| - max\ \{\ |s|\ |s \subseteq r, s \models X \to Y\ \}, \tag{5}$$

$$g_3(X \to Y, r) = G_3(X \to Y, r)/|r|. \tag{6}$$

3 Description de la méthodologie

La solution proposée consiste à utiliser le corpus Viznet pour découvrir des dépendances fonctionnelles relaxées, fréquemment découvertes entre un ensemble de colonnes possédant des types sémantiques données. Les différentes phases de la méthodologie sont décrites ci-dessous :

1. **Filtrage des tables et détection de types sémantiques :** afin d'obtenir des tables de bonne qualité pour la recherche de DFRs, plusieurs filtres sont appliqués aux tables du corpus VizNet. Le processus d'extraction de ces tables se déroule en plusieurs étapes. Le corpus est parcouru en appliquant les étapes suivantes pour chaque table :
   (a) Filtrage sur les nombres minimaux de lignes et de colonnes de la table ;
   (b) Détection des types sémantiques présents dans cette table ;
   (c) Filtrage sur le nombre minimal de types sémantiques découverts dans la table.
2. **Fusion de tables :** cette étape consiste à fusionner les tables dont les colonnes correspondent à des types sémantiques identiques et a pour but d'obtenir des tables avec plus d'enregistrements afin d'augmenter la qualité des DFRs découvertes.
3. **Extraction de dépendances fonctionnelles relaxées :** plusieurs approches de découverte de dépendances fonctionnelles peuvent être considérées : recherche exacte, recherche approchée et les approches basées sur l'apprentissage automatique.
4. **Construction du graphe de dépendances :** dans cette dernière étape, une base de connaissance est construite, elle prend la forme d'un graphe dont les nœuds sont des types sémantiques et dont les arêtes représentent l'existence d'une dépendance fonctionnelle relaxée.

4 Expérimentation

Dans cette section, nous présentons d'abord le protocole expérimental ainsi que le score que nous proposons pour évaluer les dépendances. La première expérimentation vise à comparer trois approches de détection de types sémantiques différentes, la deuxième a pour but d'évaluer la pertinence de la phase de fusion.

4.1 Protocole expérimental

Nous déclinons l'ensemble des expérimentations pour les dépendances fonctionnelles d'arité 2, mais l'étude peut être généralisée à des tailles plus importantes en utilisant notamment les algorithmes Pyro (Kruse et Naumann, 2018) ou Dime (Caruccio et al., 2020) pour la phase d'extraction de dépendances relaxées. Les expérimentations sont menées avec l'ensemble des algorithmes et paramètres qui suivent :

— *Filtrage des tables.* Nous évaluons uniquement les tables avec 20 lignes ou plus, et dont les colonnes correspondent à au moins 2 types sémantiques différents.

— *Ensemble de types sémantiques.* Nous construisons la liste ***type44*** pour laquelle nous sélectionnons 44 types sémantiques possédant un ensemble de valeurs fini, nous permettant ainsi, de construire un index LSH autour de cette dernière.

— *Détection de types sémantiques.* Trois approches sont considérées ici, comme première méthode de détection, nous utilisons une version pré-entrainée de ***Sato***. Sato a été entrainé sur 80 000 tables extraites depuis le corpus VizNet afin de détecter des types sémantiques parmi une liste de 78 types sémantiques qui inclut la liste ***type44***. Nous choisissons l'approche ***LSH*** comme deuxième méthode de détection. Cette dernière requiert l'utilisation d'un dictionnaire de valeurs pour chaque type sémantique que nous voulons détecter. Nous utilisons la librairie Datasketch pour construire un index de type Forêt LSH [1] avec 256 permutations. ***Comparaison d'entêtes.***, la troisième et dernière méthode de détection envisagée est la comparaison des noms des colonnes des tables avec une liste de noms de types sémantiques. Ainsi, en considérant que la première ligne d'une table relationnelle représente les noms de ses colonnes et en appliquant un prétraitement à ces derniers, nous pouvons vérifier si ces noms correspondent aux types sémantiques présents dans la liste utilisée. Pour réduire le temps de calcul l'ensemble de ces approches sont exécutées sur un sous-échantillon d'enregistrements si le nombre de ces derniers dépasse 1000.

— *Découverte de dépendances fonctionnelles.* Nous évaluons l'ensemble des couples de types sémantiques qui cooccurrent dans les tables extraites, pour chaque couple (X, Y) et après une étape de fusions facultative de toutes les tables contenant X et Y, nous calculons le DF_{score} (présenté ci-après) de la dépendance $X \rightarrow Y$ ainsi que celui de $Y \rightarrow X$. Si DF_{score} est supérieur ou égal à un certain seuil la DF est considérée comme valide selon l'approche.

— *Création d'un ensemble de DFRs annoté.* Afin d'évaluer les différentes méthodes, nous annotons toutes les DFs de taille 2 possibles pour l'ensemble ***type44***. Cette étape consiste à juger chacune des DFs candidates, obtenues en générant toutes les combinaisons de taille 2 possibles, comme étant vraisemblable ou non.

4.2 Score de dépendances

Nous proposons un score d'évaluation de dépendance combinant d'une part les mesures d'erreurs g_1, g_2 et g_3 (présentées dans la section 2.3), et d'une autre, des informations observées dans les tables. Nous menons un ensemble d'expérimentations pour déterminer les composantes du score et les poids de chaque composante, par souci de brièveté nous ne présentons ici, que les paramètres retenus finalement :

1. `http://ekzhu.com/datasketch/lshforest.html`

$$DF_{score}(X \to Y, r) = 0.7 - \left(\frac{1}{6} \cdot g_1(X \to Y, r) + \frac{3}{6} \cdot g_2(X \to Y, r) + \right. \tag{7}$$

$$\left. \frac{2}{6} \cdot g_3(X \to Y, r)\right) + 0.3 \cdot variety \tag{8}$$

$$variety(X \to Y, r) = min\left(\frac{n_{diff}(X)}{N} \cdot \frac{n_{diff}(Y)}{N} \cdot \frac{1}{0.25}\ ,\ 1\right) \tag{9}$$

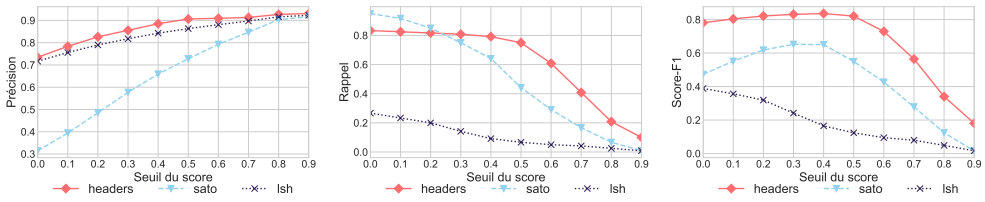
$$n_{diff}(Z) = \textit{nombre de lignes différentes sur la colonne Z} \tag{10}$$

$$N = \textit{nombre total de lignes dans la table} \tag{11}$$

DF_{score} varie dans $[0, 1]$. Un poids global de 70% est attribué aux erreurs g_1, g_2 et g_3. Un poids de 30% est attribué à la proportion du nombre de lignes différentes pour chacun des attributs concernés par la DF. Si le ratio entre le nombre de lignes différentes et le nombre de lignes total pour un attribut est proche de 1 alors cet attribut est proche d'une clé primaire, si ce ratio se rapproche de 0, alors toutes les lignes tendent à être identiques. Dans les deux cas la DF considérée n'est pas pertinente. Ainsi, un ratio autour de 0.5 nous assure la pertinence de la DF, et justifie le facteur $\frac{1}{0.5 \cdot 0.5}$.

4.3 Comparaison des approches de détection de types sémantiques

Le but de cette expérimentation est de déterminer l'approche de détection de types sémantique qui permet la meilleure découverte de dépendances. Nous utilisons l'ensemble ***type44*** sur 100 000 tables extraites et comparons la qualité des DFRs retournées par chaque approche en utilisant l'ensemble annoté.



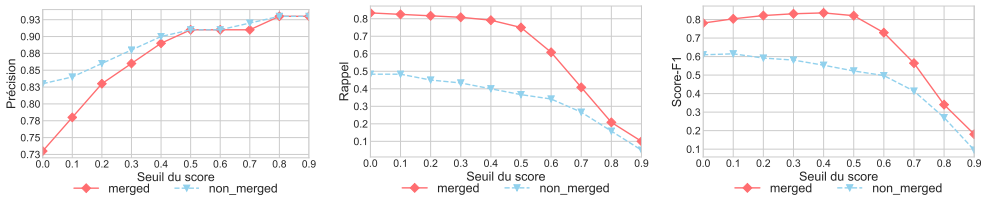
(a) Précision (b) Rappel (c) F1-score

FIG. 1: Comparaison entre les méthodes de détection de types sémantiques

La précision, le rappel et le f1-score des trois méthodes en fonction du seuil de score sont représentés respectivement, Figures 1a, 1b et 1c. Les trois graphiques montrent que la méthode par *comparaison d'entêtes* (courbes avec le marqueur losange) est celle qui offre les meilleurs résultats, et ce, indépendamment du seuil de DF_{score} considéré. Ceci prouve la qualité du corpus VizNet, qui comporte un grand nombre de tables relationnelles avec des entêtes. La précision de *LSH* (courbes avec le marqueur x) semble meilleure que celle de *Sato* (courbes avec le marqueur triangle), mais son rappel médiocre nous encourage à favoriser une approche telle que *Sato*, dans le cadre de l'exploitation d'un corpus de moindre qualité. Le seuil optimal pour obtenir une base de connaissance de bonne qualité est autour de 0.5 pour *comparaison d'entête*.

4.4 Pertinence de la fusion

Le but de cette expérimentation est de juger de la pertinence de l'étape de fusion des tables avec des types sémantiques identiques. Nous utilisons l'ensemble ***type44***, la *comparaison d'entêtes* et développons les deux approches, sur 100 000 tables extraites. Enfin, nous comparons la qualité des DFRs retournées par chaque approche en utilisant l'ensemble annoté.



(a) Précision (b) Rappel (c) F1-score

FIG. 2: Comparaison entre les DFRs obtenues avec et sans fusion des tables

La précision, le rappel et le f1-score des trois méthodes en fonction du seuil de score sont représentés respectivement, Figures 2a, 2b et 2c. Les précisions mesurées avec fusion (courbes avec le marqueur losange) et sans fusion (courbes avec le marqueur triangle) sont assez proches, cependant, le rappel des DFRs obtenues avec fusion surpasse largement celui mesuré avec l'alternative sans fusion. Cela s'explique par la nature du DF_{score} qui défavorise les attributs proches d'une clé primaire, dont les valeurs ne se répètent que très rarement dans les tables non-fusionnées. La fusion augmente ainsi leur chance d'être répliquées dans plusieurs enregistrements.

5 Conclusion

Dans cet article, nous avons présenté notre approche pour la construction d'un graphe de connaissance à partir du corpus VizNet. Les résultats obtenus sur l'ensemble de données annoté ont montré la pertinence du score de dépendance proposé et ont permis de guider le choix des paramètres. Le f1-score a dépassé 80% pour la méthode de détection de types sémantiques basée sur la *comparaison d'entêtes*. Par ailleurs, notre étude est axée sur les dépendances fonctionnelles d'arité 2, nous comptons la généraliser à des tailles plus importantes. Nous prévoyons également de généraliser notre méthodologie pour la prise en compte de contraintes supplémentaires telles que les *combinaisons uniques relaxées* ou les *contraintes de déni* (« denial constraints »).

Références

Blei, D. M., A. Y. Ng, et M. I. Jordan (2003). Latent dirichlet allocation. *Journal of machine Learning research 3*(Jan), 993–1022.

Cafarella, M. J., A. Halevy, D. Z. Wang, E. Wu, et Y. Zhang (2008). Webtables : exploring the power of tables on the web. *Proceedings of the VLDB Endowment 1*(1), 538–549.

Caruccio, L., V. Deufemia, et G. Polese (2016). Relaxed functional dependencies—a survey of approaches. *IEEE Transactions on Knowledge and Data Engineering 28*(1), 147–165.

Caruccio, L., V. Deufemia, et G. Polese (2020). Mining relaxed functional dependencies from data. *Data Mining and Knowledge Discovery 34*(2), 443–477.

Chapman, A., E. Simperl, L. Koesten, G. Konstantinidis, L.-D. Ibáñez, E. Kacprzak, et P. Groth (2020). Dataset search : a survey. *The VLDB Journal 29*(1), 251–272.

Hu, K., S. Gaikwad, M. Hulsebos, M. A. Bakker, E. Zgraggen, C. Hidalgo, T. Kraska, G. Li, A. Satyanarayan, et Ç. Demiralp (2019). Viznet : Towards a large-scale visualization learning and benchmarking repository. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–12.

Hulsebos, M., K. Hu, M. Bakker, E. Zgraggen, A. Satyanarayan, T. Kraska, Ç. Demiralp, et C. Hidalgo (2019). Sherlock : A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1500–1508.

Indyk, P. et R. Motwani (1998). Approximate nearest neighbors : Towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, New York, NY, USA, pp. 604–613. Association for Computing Machinery.

Kivinen, J. et H. Mannila (1995). Approximate inference of functional dependencies from relations. *Theoretical Computer Science 149*(1), 129–149.

Kruse, S. et F. Naumann (2018). Efficient discovery of approximate dependencies. *Proceedings of the VLDB Endowment 11*(7), 759–772.

Lehmberg, O. (2019). *Web table integration and profiling for knowledge base augmentation*. Ph. D. thesis.

Lehmberg, O., D. Ritze, P. Ristoski, R. Meusel, H. Paulheim, et C. Bizer (2015). The mannheim search join engine. *Journal of Web Semantics 35*, 159–166.

Nishida, K., K. Sadamitsu, R. Higashinaka, et Y. Matsuo (2017). Understanding the semantic structures of tables with a hybrid deep neural network architecture. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Zhang, D., Y. Suhara, J. Li, M. Hulsebos, Ç. Demiralp, et W.-C. Tan (2019). Sato : Contextual semantic type detection in tables. *arXiv preprint arXiv :1911.06311*.

Summary

Discovering dependencies in a dataset is at the heart of many data profiling and cleansing efforts. Among the most important dependencies for relational databases are functional dependencies (FDs), which represent constraints between the attributes of a relational data model. FDs can be relaxed on the comparison method and/or on the satisfaction constraint, this relaxation makes their discovery more complex than that of exact dependencies, which is already difficult. In this paper, we propose a methodology to build from a corpus of web tables, a knowledge base that takes the form of a graph whose nodes are semantic types and whose edges represent the existence of a relaxed functional dependency.