Déduplication sur des types d'attributs hétérogènes

Loujain Liekah*, Yacine Gaci*, George Papadakis**

* LIRIS - University of Claude Bernard Lyon 1, Villeurbanne, France
loujain.liekah5@gmail.com, ey_gaci@esi.dz
** National and Kapodistrian University of Athens, Athenes, Greece
gpapadis@di.uoa.gr

Résumé. La déduplication est une tâche qui consiste à reconnaître plusieurs représentations d'un même objet du monde réel. La majorité des solutions existantes se concentrent sur les données textuelles et souvent négligent les attributs booléens et numériques, tandis que le problème des valeurs manquantes n'est pas suffisamment couvert. Les solutions supervisées ne peuvent être appliquées sans un nombre adéquat d'exemples étiquetés, ce qui implique des processus d'étiquetage coûteux en temps. Nous proposons dans ce papier D-HAT, un pipeline non supervisé qui est intrinsèquement capable de traiter des types d'attributs de haute dimension, épars et hétérogènes. Au cœur de ce pipeline se trouvent : (i) une nouvelle fonction de matching qui résume efficacement les signaux de correspondance multiples, et (ii) *MutMax*, un algorithme de regroupement glouton qui désigne comme doublons les paires ayant un score de matching mutuellement maximal. Nous évaluons D-HAT sur cinq datasets réels, et démontrons que notre approche surpasse significativement l'état de l'art.

1 Introduction

Le présent article est un résumé de l'article publié dans la conférence *Advanced Data Mining and Applications* (Liekah and Papadakis, 2022). L'intégration d'ensembles de données qui se chevauchent et se complètent est un processus courant qui crée des connaissances nouvelles et précieuses (Chen et al., 2014). Une tâche importante de l'intégration consiste à identifier les données qui représentent la même entité du monde réel, comme les produits, les instituts ou les patients. Cette tâche est appelée *déduplication* (Dong and Srivastava, 2015), matching d'entités (Konda et al., 2016), résolution d'entités (Papadakis et al., 2020b) ou couplage d'enregistrements. La déduplication améliore la qualité des données en réparant et en conservant les sources de données (Fan et al., 2014), en réduisant la taille du stockage et en préparant les données pour les applications en aval (Dong and Srivastava, 2015).

La majorité des solutions pour la déduplication sont basées sur le calcul de scores de similarité par paire à partir d'un ou plusieurs attributs (Christophides et al., 2021). Les méthodes *non supervisées* créent un graphe de similarité, où les nœuds correspondent aux enregistrements et où les arêtes sont pondérés par les scores de matching des nœuds adjacents (Hassanzadeh et al., 2009). Le graphe est ensuite divisé en groupes de telle sorte que tous les nœuds de chaque

groupe correspondent à des doublons. Ces approches calculent généralement les scores de matching en traitant tous les attributs comme des données textuelles (Christophides et al., 2021). Cependant, les données du monde réel comportent des types d'attributs hétérogènes, numériques, catégoriques et booléens. Le fait de considérer ces types comme des chaînes de caractères peut conduire à des scores de similarité inexacts. Par exemple, les prix "14" et "14,00" sont identiques en tant que nombres, mais partiellement similaires lorsqu'ils sont comparés en tant que séquences de caractères et totalement différents lorsqu'ils sont traités comme des tokens. Par conséquent, les techniques non supervisées doivent modéliser et prendre en charge correctement des types d'attributs hétérogènes.

D'autre part, les méthodes *supervisées* modélisent généralement la déduplication comme une tâche de classification binaire (Konda et al., 2016) où les paires d'enregistrements se voient attribuer des étiquettes de similarité. Un classifieur est ensuite entrainé sur des vecteurs de fetures correspondant aux paires pour prédire le statut des paires non étiquetées. Cependant, ces approches sont confrontées à de nombreux défis : (i) La classification devient plus difficile avec une haute dimensionnalité. (ii) Les données étiquetées sont rares et leur obtention par le biais du crowd-sourcing est coûteuse et lente(Wang et al., 2012). De plus, la taille et la qualité des données affectent le résultat final dans une large mesure (Mudgal et al., 2018). (iii) Les méthodes supervisées nécessitent de longs temps d'apprentissage (Mudgal et al., 2018).

Pour remédier à ces problèmes, nous présentons D-HAT (Deduplication over Heterogeneous Attribute Types), un nouveau pipeline basé sur le clustering pour la déduplication de bout en bout. D-HAT se distingue de la litérature de trois façons : (i) Il prend en charge des données comportant des types d'attributs hétérogènes et une grande partie de valeurs manquantes. (ii) Il supporte et exploite les données à haute dimensionnalité. (iii) Il obtient des résultats de pointe sans nécessiter de données étiquetées. Nous menons des expériences sur des ensembles de données réels, montrant que : D-HAT surpasse les méthodes supervisées et non supervisées de l'état de l'art, en performance et en temps d'exécution. Nous rendons publiques toutes les données et le code utilisés dans nos expériences par le biais de `https://github.com/Loujainl/D-HAT`.

2 Etat de l'Art

La recherche croissante sur la déduplication reflète son importance grandissante (Christen, 2012a; Christophides et al., 2021; Dong and Srivastava, 2015). L'un des principaux défis de la déduplication est sa complexité quadratique : dans le pire des cas, elle examine toutes les paires d'enregistrements possibles. Le *Blocking* est généralement utilisé pour atténuer cette complexité, surtout avec des données volumineuses (Christen, 2012b; Papadakis et al., 2020b). Le blocking rassemble les enregistrements similaires en groupes appelés blocs en appliquant des schémas ou des fonctions de blocage. Une fonction de blocage extrait les signatures de chaque enregistrement, divisant l'ensemble des données d'entrée en un ensemble de blocs qui se chevauchent – Les comparaisons sont réduites à des *candidats*, c'est-à-dire à des paires d'enregistrements partageant au moins un bloc, ce qui réduit considérablement le coût de calcul. Cependant, cette efficacité en termes de temps d'exécution s'accompagne du risque de diminution dans la précision(Papadakis et al., 2016).

Après le blocking, le *matching* est effectuée pour déterminer le degré de similitude entre les paires. Essentiellement, le matching applique des fonctions de similarité aux valeurs des

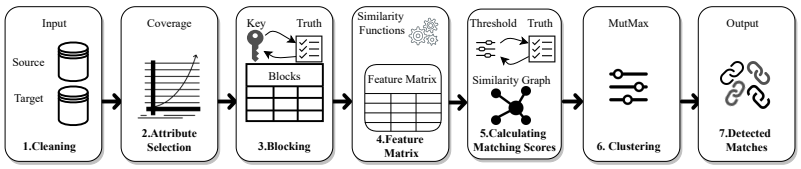

FIG. 1 – *Le pipeline de bout en bout de D-HAT.*

attributs sélectionnés. Ensuite, il détermine si la similarité est suffisante pour désigner deux enregistrements comme doublons. Nous distinguons deux types d'algorithmes de matching : algorithmes supervisés et non supervisés.

La première catégorie comprend un ensemble de méthodes fournies par JedAI (Papadakis et al., 2020a, 2018) et Stringer (Hassanzadeh et al., 2009), tandis que *ZeroER* (Wu et al., 2020) constitue l'approche non supervisée de pointe qui représente chaque paire de candidats comme un vecteur de caractéristiques. Contrairement aux méthodes supervisées, elle ne nécessite pas d'ensemble d'apprentissage, mais repose sur l'observation que la distribution des caractéristiques pour les enregistrements doublons diffère de celle des enregistrements non doublons.

Parmi les méthodes supervisées, la plus populaire est *Magellan* (Konda et al., 2016), qui est un système combinant une variété de caractéristiques en utilisant les principaux classifieurs d'apprentissage automatique, tels que les arbres de décision, la régression logistique et les machines à vecteurs de support. Partant d'un échantillon annoté de paires T, le matching est effectué en entraînant un classificateur sur T. Magellan propose également un ensemble de méthodes de blocking.

3 Approche

Le pipeline de notre approche est illustré dans la Figure 1.

Étape 1 : Nettoyage des données. La première étape prépare les données en déterminant les caractéristiques essentielles de leurs attributs [1], c'est-à-dire qu'elle calcule le nombre de valeurs uniques et le type de données par attribut. Les attributs qui ont deux valeurs uniques sont convertis en booléens pour obtenir un degré de similarité plus précis. Les attributs ayant très peu de valeurs uniques ($<$10) sont traités comme des variables catégorielles. Les attributs numériques sont identifiés par des expressions régulières qui détectent des quantités, éventuellement accompagnées d'une unité de mesure. Par exemple, une valeur d'attribut `largeur = "42.8 in"` est transformée en `largeur = 42.8` et est marquée comme un type de données numériques. Une normalisation min-max est ensuite effectuée sur les valeurs des attributs numériques.

Étape 2 : Sélection des attributs. La *couverture d'un attribut* a exprime la proportion de valeurs non vides dans a sur l'ensemble des enregistrements d'entrée ; moins il y a de valeurs manquantes, plus la couverture est élevée. Nous définissons formellement la couverture c de chaque attribut comme suit : $c(a) = 1 - \frac{|r_i.a=N/A:r_i \in T|}{|T|}$. Cette étape élimine les attributs dont

1. Dans le cas de Record Linkage, nous supposons que les schémas sont alignés.

la couverture est inférieure à un seuil spécifique. Des expériences préliminaires ont démontré que 0.1 constitue une valeur efficace.

Étape 3 : Blocking. Cette étape est cruciale car elle détermine deux choses : (i) Efficience temporelle, car le temps de traitement des étapes suivantes est déterminé par le nombre de candidats dans les blocs résultants. (ii) Efficacité, parce que les paires d'enregistrements faussement négatifs, qui n'ont aucun bloc en commun, ne peuvent pas être détectées par les étapes suivantes, et sont exclues du résultat final.

Il est donc crucial que le blocage trouve un équilibre entre ces deux objectifs concurrents : la réduction de l'espace de recherche et une plus grande efficacité. D-HAT est suffisamment générique pour s'adapter à toute méthode de blocage répondant à cette exigence. Des expériences préliminaires ont indiqué que le *overlap blocker* de Magellan (Konda et al., 2016) est une approche robuste pour créer des blocs de haute performance (voir la Section 4 pour plus de détails). Elle définit comme paires candidates celles qui partagent au moins un jeton dans les valeurs d'un attribut spécifique. D-HAT applique l'overlap blocker à tous les attributs textuels dans les ensemble de données(s) donnés et opte pour celui qui minimise le nombre de candidats, tout en maximisant la couverture – une couverture élevée indique implicitement un rappel élevé.

Étape 4 : Matrice de caractéristique. Comme pour les approches supervisées, D-HAT représente chaque paire d'enregistrements sous la forme d'un vecteur de caractéristique en appliquant des fonctions de similarité normalisées spécifiques au type des attributs sélectionnés. Contrairement aux approches supervisées, ces vecteurs ne sont pas étiquetés. Plus précisément, D-HAT crée un vecteur de caractéristique $V_{i,j}$ pour chaque paire d'enregistrements candidates $(r_i, r_j) \in B$, où B est l'ensemble des blocs produits par l'étape précédente et la k^{ime} caractéristique/dimension dans $V_{i,j}$, $V^k_{i,j}$, provient d'une fonction de similarité compatible avec le type de l'attribut k^{ime}, a_k. Si la valeur d'un enregistrement pour a_k est vide ou incorrecte (c'est-à-dire incompatible avec le type de a_k), $V^k_{i,j}$ ='N/A', ce qui signifie qu'une caractéristique est manquante. Cette étape ne requiert aucune connaissance du domaine de la part de l'utilisateur. D-HAT détecte automatiquement le type d'attribut et applique les fonctions de similarité appropriées afin de créer les caractéristiques.

Nous référons les lecteurs intéressés à l'article original (Liekah and Papadakis, 2022) pour plus de détails sur les fonctions de similarité.

Étape 5 : Scores de matching. L'objectif de cette étape est d'estimer la probabilité de matching pour chaque paire de candidats sur la base de la matrice des caractéristiques de l'étape précédente. Cette opération s'effectue en deux étapes : (i) *Binarisation des vecteurs de caractéristiques.* D-HAT traite chaque caractéristique comme un vote pour une décision de "match" (1) ou de "no-match" (0). En dehors des attributs booléens et catégoriels qui sont déjà binaires, D-HAT binarise les attributs numériques et textuels en considérant comme "match" les attributs ou la valeur de la caractéristique dépasse un seuil de similarité θ. Toutes les dimensions ayant une valeur "N/A" sont ignorées. (ii) *Score d'estimation.* Le score de matching de deux enregistrements est la moyenne de leurs caractéristiques binarisées à l'issue de l'étape précedente.

À la fin de ces deux étapes, les scores de matching de toutes les paires sont stockés dans une matrice M. Les enregistrements et la matrice définissent un graphe pondéré $G(V, M)$, où l'ensemble des nœuds V représente les enregistrements d'entrée, et M est la matrice d'adjacence des poids. On appelle $G(V, M)$ le *graphe de similarité*.

TAB. 1 – *Caractéristiques techniques des ensembles de données de test.* $|S|$, $|T|$ *et* $|D|$ *représentent respectivement le nombre d'enregistrements sources, d'enregistrements cibles et de paires de doublons.*

Dataset	$\|S\|$	$\|T\|$	$\|D\|$	#Attributs	#Numérique	#Bool. & Cat.	#Textuel	#Sélectionnés
Amazon-Google	1 363	3 226	1 298	4	1	0	2	3
Abt-Buy	1 081	1 092	1 095	3	1	0	2	3
DBLP-ACM	2 614	2 294	2 223	4	1	1	2	3
Fodors-Zagats	533	331	112	5	0	0	5	5
Immucare	305	310	305	213	32	6	37	75

Étape 6 : MutMax Clustering. L'étape finale reçoit en entrée le graphe de similarité $G(V, M)$ et le partitionne en un ensemble de clusters disjoints, tels que chaque cluster correspond à une entité unique, contenant tous les enregistrements dupliqués la décrivant. Le partitionnement est effectué par **MutMax**, une approche gloutonne qui définit comme doublons les paires d'enregistrements ayant des scores mutuellement maximaux. Plus précisément, MutMax fonctionne comme suit : Pour chaque enregistrement r_i, tous les candidats sont triés par scores de matching décroissants et le plus élevé $r^i_{max} = r_j$ est sélectionné comme un match potentiel, si r_i a été défini comme la correspondance potentielle pour r_j, les enregistrements r_i et r_j sont désignés comme des correspondances. Le reste des paires candidates est ignoré.

En termes de complexité temporelle, le coût des étapes 1, 2 et 3 est linéaire avec le nombre d'attributs dans l'ensemble de données donné T, $O(|T.A|)$. Pour les étapes 4 et 5, le coût est de $O(|B|)$. Pour l'étape 6, aucun tri n'est nécessaire. D-HAT itère une fois sur toutes les cellules du tableau bidimensionnel M. En pratique, une table de hachage peut être utilisée pour stocker les similarités estimées. Par conséquent, la complexité temporelle et spatiale de l'étape 6 (et de l'ensemble de l'algorithme) est linéaire par rapport au nombre de paires candidates après blocage $O(|B|)$.

4 Evaluation

Benchmarks. Nous utilisons cinq ensembles de données connus, de plusieurs domaines : produits, bibliographie, restaurants et soins de santé. Immucare est un ensemble de données sur les soins de santé correspondant à deux visites différéntes à l'hôpital d'un même patient. Tous les détails techniques de ces ensembles de données (Konda et al., 2016; Wu et al., 2020) sont résumés dans le tableau 1.

Bases de référence. Nous comparons les performances de D-HAT avec Magellan (Konda et al., 2016) et ZeroER (Wu et al., 2020). Pour le premier, nous utilisons l'arbre de décision comme algorithme de classification, tandis qu'aucune configuration n'est nécessaire pour le second.

Mesures d'évaluation. Nous utilisons les mesures standard de rappel, de précision et de score F1. Nous indiquons également le temps d'exécution global.

Résultats.
Pour des fins d'équité, nous appliquons la même méthode de blocking au même attribut clé pour les deux systèmes de base. Notez que pour Amazon-Google, ZeroER n'a pas pu créer sa matrice de caractéristiques dans un délai de 6 heures. Pour compléter l'évaluation, nous l'avons combiné avec les vecteurs de caractéristiques créés par Magellan. Par conséquent, les

TAB. 2 – *Efficacité du matching dans D-HAT, Magellan et ZeroER sur tous les ensembles de données. Le meilleur F1 est en gras.*

Dataset	D-HAT									Magellan			ZeroER		
	Features Syntaxiques			Features Sémantiques			Features Hybrides								
	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1	Pr	Re	F1
A-G	0,904	0,479	0,626	0,828	0,349	0,534	0,925	0,532	**0,675**	0,513	0,573	0,542	0,663	0,385	0,487
A-B	0,818	0,402	**0,539**	0,635	0,174	0,274	0,824	0,346	0,487	0,440	0,443	0,442	0,220	0,601	0,322
D-A	0,992	0,956	0,974	0,995	0,980	**0,987**	0,997	0,974	0,985	0,980	0,983	0,981	0,936	0,945	0,940
F-Z	0,981	0,929	**0,954**	0,971	0,911	0,940	0,981	0,929	**0,954**	0,939	0,969	**0,954**	1,000	0,312	0,476
CA	0,993	0,987	**0,990**	0,990	0,987	0,988	0,993	0,987	**0,990**	0,968	1,000	0,984	1,000	0,487	0,655

performances de ZeroER pourraient être légèrement différentes de celles rapportées dans (Wu et al., 2020).

Les performances de tous les algorithmes en matière de précision (Pr), de rappel (Re) et de f-mesure (F1) figurent dans le Tableau 2, tandis que les temps d'exécution correspondants sont indiqués dans la Figure 2. Notez qu'après des expériences préliminaires, nous avons fixé $c_{min} = 0,1$ et $\theta = 0,7$ pour D-HAT dans tous les cas.

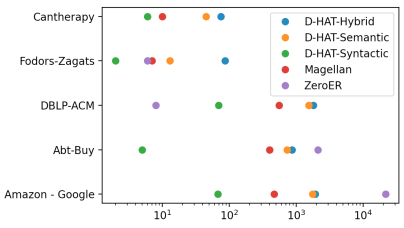

FIG. 2 – *Temps d'exécution (sec.)*

En comparant les différents groupes de caractéristiques entre eux, nous observons que les caractéristiques syntaxiques sont systématiquement plus performantes que les caractéristiques sémantiques. La raison est que les ensembles de données contiennent une terminologie spécifique au domaine. Par conséquent, word2vec et GloVe souffrent des termes hors vocabulaire.

En termes d'efficacité temporelle, l'avantage des fonctions de similarité syntaxique est clair, comme le montre la Figure 2. Le temps d'exécution de D-HAT augmente d'un ordre de grandeur dans presque tous les cas lorsqu'on remplace les fonctions de similarité syntaxique par les fonctions sémantiques. Cela est dû au grand nombre de recherches et de calculs nécessaires pour convertir chaque valeur d'attribut en un vecteur à haute dimension et aussi pour calculer les scores de similarité.

Il est intéressant d'examiner si la combinaison des similarités syntaxiques et sémantiques (lourdes en calcul) est justifiée par une augmentation de l'efficacité. Cela n'est vrai que dans le cas d'Amazon-Google, où le F1 des caractéristiques hybrides est supérieur à celui des syntaxiques de $\sim$10%. Dans tous les autres cas, les caractéristiques hybrides se situent entre les deux autres groupes de caractéristiques, généralement plus proches du groupe le plus performant. Par conséquent, *D-HAT devrait être exclusivement combiné avec le groupe de caractéristiques syntaxiques*.

Comparé à ZeroER, le Tableau 2 montre que D-HAT utilisé avec des caractéristiques syntaxiques est nettement supérieur. Sa F1 est supérieure de 50%, en moyenne, sur les cinq tâches. En même temps, la Figure 2 montre que D-HAT est systématiquement plus rapide que ZeroER par des ordres de grandeur entiers (par exemple, 1 minute contre 6 heures sur Amazon-Google).

Par rapport à Magellan, dans les deux premiers ensembles de données, D-HAT obtient une f-mesure supérieure de plus de 13%, tandis que dans les trois ensembles de données suivants, les deux méthodes affichent des performances pratiquement identiques. La performance

compétitive de Magellan provient de sa fonctionnalité supervisée, tandis que D-HAT est non supervisé. En termes d'efficacité temporelle, nous observons sur la Figure 2 que D-HAT prend une nette avance, autour d'un ordre de grandeur entier (par exemple, 35 contre 400 secondes pour Abt-Buy).

5 Conclusions

Nous avons présenté D-HAT, un système de déduplication de bout en bout efficace et entièrement automatisé basé sur le clustering. D-HAT traite des ensembles de données de grande dimension avec des types d'attributs hétérogènes et des valeurs manquantes sans nécessiter l'intervention de l'utilisateur ni de données étiquetées. L'étude expérimentale sur des ensembles de données connus démontre que notre système surpasse les méthodes de l'état de l'art. Le principal avantage de D-HAT par rapport aux méthodes non supervisées est sa grande précision sur toutes les tâches standards, tandis que par rapport aux méthodes supervisées, D-HAT élimine le temps et les efforts supplémentaires nécessaires aux experts du domaine pour l'annotation des données [2].

Références

Chen, M., Mao, S., and Liu, Y. (2014). Big data : A survey. *MONET*, 19(2) :171–209.

Christen, P. (2012a). *Data Matching*. Springer.

Christen, P. (2012b). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Trans. Knowl. Data Eng.*, 24(9) :1537–1555.

Christophides, V., et al. and Stefanidis, . K. (2021). An overview of end-to-end entity resolution for big data. *ACM Comput. Surv.*, 53(6) :127 :1–127 :42.

Dong, X. L. and Srivastava, D. (2015). Big data integration. *Synthesis Lectures on Data Management*, 7(1) :1–198.

Fan, W., Ma, S., Tang, N., and Yu, W. (2014). Interaction between record matching and data repairing. *Journal of Data and Information Quality (JDIQ)*, 4(4) :1–38.

Hassanzadeh, O., et al. Framework for evaluating clustering algorithms in duplicate detection. *Proc. VLDB Endow.*, 2(1) :1282–1293.

Konda, P., Das, S., et al. (2016). Magellan : Toward building entity matching management systems. *Proc. VLDB Endow.*, 9(12) :1197–1208.

Liekah, L. and Papadakis, G. (2022). Deduplication over heterogeneous attribute types (d-hat). In *International Conference on Advanced Data Mining and Applications*, pages 379–391. Springer.

Mudgal, S., Li, H., Rekatsinas, T., Doan, A., Park, Y., et al. (2018). Deep learning for entity matching : A design space exploration. In *SIGMOD*, pages 19–34.

Papadakis, G., et al. Thanos, E., et al. (2020a). Three-dimensional entity resolution with jedai. *Information Systems*, 93 :101565.

2. Ce projet a été financé par le programme de recherche et d'innovation Horizon 2020 de l'Union européenne dans le cadre de la convention de subvention n° 875171.

Papadakis, G., et al. Blocking and filtering techniques for entity resolution : A survey. *ACM Computing Surveys (CSUR)*, 53(2) :1–42.

Papadakis, G., et al. Comparative analysis of approximate blocking techniques for entity resolution. *Proc. VLDB Endow.*, 9(9) :684–695.

Papadakis, G., et al. The return of jedai : End-to-end entity resolution for structured and semi-structured data. *Proc. VLDB Endow.*, 11(12) :1950–1953.

Wang, J., Kraska, T., Franklin, M. J., and Feng, J. (2012). Crowder : Crowdsourcing entity resolution. *arXiv preprint arXiv :1208.1927*.

Wu, R., Chaba, S., Sawlani, S., Chu, X., and Thirumuruganathan, S. (2020). Zeroer : Entity resolution using zero labeled examples. In *SIGMOD*, pages 1149–1164.

Summary

Deduplication is the task of recognizing multiple representations of the same real-world object. The majority of existing solutions focuses on textual data, this means that data sets containing boolean and numerical attribute types are rarely considered in the literature, while the problem of missing values is inadequately covered. Supervised solutions cannot be applied without an adequate number of labelled examples, but training data for deduplication can only be obtained through time-costly processes. To address these challenges, we go beyond existing works through D-HAT, a clustering-based pipeline that is inherently capable of handling high dimensional, sparse and heterogeneous attribute types. At its core lies: (i) a novel matching function that effectively summarizes multiple matching signals, and (ii) *MutMax*, a greedy clustering algorithm that designates as duplicates the pairs with a mutually maximum matching score. We evaluate D-HAT on five established, real-world benchmark data sets, demonstrating that our approach outperforms the state-of-the-art supervised and unsupervised deduplication algorithms to a significant extent.