Approche supervisée pour l'appariement d'entités dans le domaine Transport et Logistique

Yassine Guermazi, Sana Sellami, Omar Boucelma

Approche supervisée pour l'appariement d'entités dans le domaine Transport et Logistique

Yassine Guermazi
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
yassine.guermazi@etu.univ-amu.fr

Sana Sellami
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
sana.sellami@univ-amu.fr

Omar Boucelma
Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
omar.boucelma@univ-amu.fr

ABSTRACT

L'appariement d'entités (Entity Matching) est un problème crucial pour l'intégration de données. Dans cet article, nous nous intéressons à l'appariement d'entités dans le domaine du Transport et Logistique lesquelles peuvent être définies par une structure (raison sociale, adresse). Aux difficultés usuelles qui caractérisent la problématique (typos, données manquantes ou redondantes, similarités sémantiques, etc.), s'ajoutent des spécificités « domaine » comme les abréviations et les acronymes dans les noms de sociétés ou l'absence d'un format standard pour les adresses. La solution que nous proposons s'appuie sur un processus en deux phases : 1) Standardisation des entités en vue de leur prétraitement et du parsing d'adresses (données textuelles), et 2) Appariement par apprentissage supervisé sur des représentations vectorielles sémantiques des entités obtenues par des techniques de représentation distribuée des mots. Les expérimentations menées sur un jeu de données réel illustrent la performance de la solution.

KEYWORDS

Appariement d'entités, apprentissage supervisé, représentation distribuée des mots, domaine du transport et logistique

ACM Reference Format:
Yassine Guermazi, Sana Sellami, and Omar Boucelma. 2020. Approche supervisée pour l'appariement d'entités dans le domaine Transport et Logistique. In *Proceedings of BDA 2020: 36ème Conférence sur la Gestion de Données – Principes, Technologies et Applications. (BDA 2020).* , 10 pages.

1 INTRODUCTION

Le domaine du Transport et de la Logistique implique différents acteurs de l'industrie. L'identification de ces acteurs est indispensable pour le maintien des activités telles que des livraisons de colis ou encore la réalisation des transactions. Chacune de ces entités est définie par une structure comprenant la raison sociale et l'adresse. Cependant, l'utilisation des acronymes ou encore les abréviations dans les noms de sociétés, les problèmes typographiques, ou de données manquantes au niveau des adresses implique une vérification qui peut s'avérer coûteuse.

BDA 2020, 27–30 Octobre, 2020, Paris, France

Nous nous intéressons, dans ce travail, à la vérification de l'existence de ces entités par rapport à leurs adresses en France. Généralement, la vérification se fait auprès d'une base de références. Or, en France, il n'existe pas de base englobant toutes les entités du domaine, appelée souvent "ground truth database". Par conséquent, les organisations du Transport et Logistique maintiennent leurs propres référentiels. Faute de ressources et règles d'harmonisation les différents référentiels restent hétérogènes et de qualité insuffisante pour permettre une interopérabilité électronique complète des acteurs de la chaîne logistique. La table 1 décrit les différents problèmes de qualité que peuvent avoir ces données: 1) incomplétude, 2) absence d'un format standard de données, 3) incohérences sémantiques et 4) données non mises à jours.

L'objectif de notre travail est de vérifier l'existence de ces entités. Ceci implique d'une part la correction des anomalies liées aux données et d'autre part de valider les entités en réalisant l'appariement (Matching) des noms de sociétés et leurs adresses avec des données provenant d'annuaires tels que Kompass [1], societe [2] et infogreffe [3].

L'appariement d'entités est une tâche complexe étant donné les problèmes de qualité des données du domaine (voir section 2). Dans ce travail, nous proposons une solution d'appariement d'entités qui s'appuie sur un processus en deux phases: 1) Standardisation des entités en vue de leur prétraitement et du parsing d'adresses (données textuelles), et 2) Appariement par apprentissage supervisé sur des représentations vectorielles sémantiques des entités obtenues par des techniques de représentation distribuée des mots. Plus précisément, l'approche proposée permet:

- de prendre en considération l'aspect sémantique des adresses: des éléments d'adresse qui partagent le même contexte géographique sont similaires.
- de déterminer la similarité entre des données polluées: données injectées sous les mauvais attributs.
- la classification des adresses en plusieurs classes par rapport aux attributs d'adresse qui leur correspondent (Table 5).

Cet article est organisé comme suit. Dans la section 2, nous introduisons le principe de l'appariement des entités. La section 3 est consacrée à l'état de l'art sur le matching d'entités. Dans la section 4, nous décrivons notre approche d'appariement d'entités dans le domaine Transport et Logistique. Les expérimentations réalisées sont décrites dans la section 5. La section 6 conclut le travail et présente quelques perspectives.

[1] https://fr.kompass.com/
[2] https://www.societe.com/
[3] https://www.infogreffe.fr/

Table 1: Problèmes de qualités de données

Entités	Problèmes de qualités de données
FC logistique, 22 Ave des nationsPARIS NORD II, PARC SILIC Bt rotand villepint france	**Elément manquant:** 93420 (zipcode) **Abréviations:** Ave (Avenue) et Bt (bâtiment) **Erreur typographique:** nationsPARIS NORD II (nations PARIS NORD II) **Erreurs d'orthographes:** Villepint (Villepinte) et rotand (rostand)
GLM TRANSIT, 25 RUE JULES LEQUIER 22190 PLERIN	**Problème de fraîcheur**: Adresse correcte: 31 RUE LEQUIER 22190 PLERIN
NADINTER, ZONE DE FRET 2 93290 TREMBLAY-EN-FRANCE	**Problème d'incomplétude**: Adresse complète: 12 RUE DU CHAPITRE ROISSY CDG ZONE DE FRET 2 BAT.3702 93290 TREMBLAY-EN-FRANCE

Table 2: Exemple d'appariement d'adresses avec des données injectées sous le mauvais attribut

RoadName	City	Additional
LIEU DIT LE PORTEREAU	VERTOU	-

RoadName	City	Additional
-	VERTOU	LE PORTEREAU

Table 3: Exemple d'appariement sémantique d'adresses

Zone	RoadName	City
ZAC SATOLAS GREEN	-	PUSIGNAN

Zone	RoadName	City
-	CHEMIN DU BOIS DES AIES	PUSIGNAN

2 PROBLÈME D'APPARIEMENT D'ENTITÉS

Dans cette section, nous définissons le problème d'appariement d'entités dans le domaine du Transport et Logistique.

Définition d'une entité Une entité est une organisation de transport et logistique définie par un nom et une adresse. Une adresse comporte différents attributs qui sont ceux qu'on retrouve dans une adresse postale (numéro de rue, nom de la rue, la ville, le pays) mais également des attributs permettant de représenter les environnements ruraux (e.g zone industrielle, parc logistique). Pour définir le modèle d'adresse, nous avons étendu le modèle défini par UPU (Universal Postal Union) [4] représentant les adresses postales françaises, pour qu'il tienne compte des zones rurales ainsi que des points d'intérêts (POI) de transport et logistique comme par exemple: les aéroports, les terminaux et les cargo-ports. La figure 1 présente le modèle proposé. Le champ "Additional" représente un complément d'adresse.

Définition du matching d'entités (EM): Le matching d'entités vise à trouver toutes les paires de tuples (t, t') qui sont similaires, c'est-à-dire qui correspondent à la même entité du monde réel, entre deux tables T et T' tels que ($t \in T$, $t' \in T'$).

[4] http://www.upu.int/fileadmin/documentsFiles/activities/addressingUnit/fraFr.pdf

Table 4: Exemples d'appariement de noms de sociétés

Nom A	Nom B	Appariement
Transport Express Caretto	Monsieur STEPHANE CARETTO	"Match"
A.T.B SARL	AFFRETEMENTS ET TRANSPORTS BECKER	"Match"

Element 1	InBuilding (IB)
Element 2	ExtBuilding (EB)
Element 3	PoiLogistic (PL)
Element 4	Zone (Z)
Element 5	HouseNum (HN)
Element 6	RoadName (RN)
Element 7	PoBox (PB)
Element 8	Zipcode (ZC)
Element 9	City (C)
Element 10	Country (Co)
Element 11	Additional (A)

Figure 1: Modèle d'adresse proposé

Les deux tables suivent le même schéma de données $\{N, A0..A9\}$ avec N qui représente le nom de l'entité et $\{A0..A9\}$ qui représentent les différents attributs d'une adresse.

Le problème d'appariement entre la paire de tuples t et t' est représenté comme un problème de classification: nous attribuons une paire de classes L=(ln, ladd) aux paires de tuples (t,t') tels que:

- $ln \in$ ["*Match*", "*NoMatch*"] est la classe qui représente le résultat de Matching entre les deux noms de sociétés des deux tuples.
- $ladd \in$ ["*NoMatch*", "*CityLevel*", "*ZoneLevel*", "*PoiLevel*", "*RoadNamelevel*", "*HouseNumLevel*", "*ExtBuildingLevel*", "*InBuildingLevel*", "*AdditionalLevel*"] est la classe qui représente

le résultat de Matching entre les deux adresses des deux tuples (table 5).

Deux entités sont similaires si L= ("*Match*", "*laddM*") tels que:

"*laddM*" ∈ ["*CityLevel*", "*ZoneLevel*", "*PoiLevel*", "*RoadNamelevel*", "*HouseNumLevel*", "*ExtBuildingLevel*", "*InBuildingLevel*", "*AdditionalLevel*"]. La figure 2 présente un exemple d'appariement d'entités.

Problème de Matching d'entités On distingue plusieurs types de problème de matching d'entités liés aux types des données et leurs degrés de qualité. Les données peuvent être structurées ou bien textuelles. De plus, elles peuvent être polluées: données incomplètes ou redondantes, données incorrectes associées à un attribut, données contenant des erreurs typographiques et d'orthographe.

Dans notre travail, le matching est réalisé sur des adresses et des noms de sociétés. Les tables 2, 3 et 4 présentent des exemples de défis de matching. Comme l'illustre la table 4, le nom d'une société peut avoir plusieurs appelations et abbréviations. De plus, les attributs d'une adresse peuvent être mal placés (table 2) ou bien il peut exister une similarité sémantique entre deux différents types d'attributs (table 3).

3 TRAVAUX EXISTANTS

Nous décrivons dans cette section les travaux existants qui portent sur le matching d'entités de manière générale ainsi que les travaux sur le matching d'adresses et noms de sociétés.

3.1 Matching d'entités

Nous distinguons entre deux catégories d'approches de matching d'entités: 1) basées sur des règles [4, 21, 22] et 2) basées sur l'apprentissage automatique [2, 8, 16, 17]. La première catégorie d'approches est basée sur des règles déterminées par des experts qui fixent les conditions de matching des paires d'entités. Des mesures de similarité (e.g distance de Levenshtein [11], distance de Jaro-Winkler [24], distance de Jaccard [7]) sont généralement utilisées pour la comparaison des attributs. Dans cet état de l'art, nous mettons l'accent sur la deuxième catégorie d'approches basée sur l'apprentissage automatique qui considère le matching d'entités comme étant un problème de classification. L'apprentissage peut être soit supervisé soit non-supervisé.

Konda et al. [8] ont proposé un système de matching d'entités, appelé Magellan, qui couvre toute la chaîne de traitement comportant le *debugging*, *l'échantillonnage* (sampling), le *blocking* et le *matching*. Dans l'étape de matching, le système génére un vecteur, contenant des caractéritiques pour chaque paire d'entité (e.g des scores de similarités) en utilisant plusieurs techniques de similarités (e.g distance de Levenshtein, distance de Jaccard). Il effectue ensuite une validation croisée sur l'ensemble de ces vecteurs pour sélectionner l'algorithme d'apprentissage le plus performant parmi ceux fournis par Magellan (e.g arbres de décision, Machine à vecteurs de support). De 2016 à ce jour, Magellan est considéré comme l'outil le plus utilisé pour le matching d'entités dans diverses applications du monde réel [5].

A la différence de Magellan, Papadaki et al. [18] proposent un système de matching d'entités nommé Jedai, basé sur des techniques d'apprentissage non supervisé sur des données structurées et semi structurées. La phase de matching de données comporte deux sous étapes: 1) *Entity Matching* dans laquelle Jedai compare les paires d'entités, associe chaque paire à une valeur de similitude dans [0,1] en utilisant plusieurs techniques (e.g similarité cosinus, distance de Jaccard) et génére en sortie un graphe de similarités, c'est-à-dire un graphe pondéré non orienté où les nœuds correspondent à des entités et les bords relient des paires d'entités comparées et 2) *Entity Clustering* qui prend en entrée le graphe de similarités produit par l'étape précédente et le partitionne en un ensemble de clusters. Chaque cluster correspond à un objet du monde réel distinct. Parmi les techniques de clustering utilisées on trouve: *Merge-Center Clustering*, *Correlation Clustering* et *Best Assignment Clustering*. Le résultat d'évaluation de ce système sur 4 jeux de données réelles [9] montre que Jedai est plus performant que Magellan sur 3 jeux de données et compétitif avec lui sur le quatrième jeu. De plus, Jedai est plus rapide et moins dépendant des experts que Magellan vu qu'il ne nécessite pas une labellisation des données.

Magellan et Jedai sont deux exemples de systèmes de matching d'entités qui utilisent principalement des techniques de similarité entre les chaînes de caractères en combinaison avec des algorithmes d'apprentissage supervisé ou non. Cependant, ces sytèmes ne sont pas applicables sur des jeux de données non structurés et ne prennent pas en compte la sémantique lors de l'appariement des entités: ils échouent à réaliser la correspondance entre des entités qui partagent le méme sens ou bien le méme contexte et qui ne sont pas forcément similaires syntaxiquement.

D'autres travaux [2, 16] ont proposé l'utilisation de l'apprentissage profond. Dans [2], Ebraheem et al. utilisent Glove [19], une technique de représentation distribuée de mots pour attribuer des vecteurs pour chaque mot d'un attribut appartenant à un tuple. La représentation vectorielle d'un attribut est la moyenne des représentations de chaque mot. Ensuite, les vecteurs de similarité entre les attributs de la paire de tuples sont calculés en utilisant la similarité cosinus. Enfin, un réseau de neurones densément connecté mappe le vecteur de similarités à une sortie binaire, 1 s'il existe un matching entre la paire d'entités, et 0 sinon. L'évaluation de l'approche est réalisée sur 7 jeux de données réelles [5] / [6] / [7] couvrant divers domaines tels que les citations, le commerce électronique et la protéomique. Ces jeux de données sont classés en "Easy" et "Challenging" suivant la nature des données et leurs degrés de bruit. L'approche a été comparée par rapport à Magellan et les résultats ont montré qu'elle était performante sur les jeux données pollués et contenant des attributs non structurés. Par contre, l'approche est compétitive avec Magellan sur des données structurées et peu bruitées.

Mudgal et al. [16] proposent une approche similaire à celle de Ebraheem et al [2] mais en utilisant des techniques différentes. En effet, la représentation vectorielle des mots est réalisée avec "*fastText*", appartenant à la famille de *character embedding*. Ensuite, plusieurs mèthodes sont proposées pour la représentation vectorielle des attributs (e.g *Recurrent Neural Network (RNN)*, *Attention Mechanism*). Le calcul de similarités entre les attributs est effectué en utilisant deux différentes familles : 1) *Fixed distance* (cosine, Euclidean) et 2) *Learnable distance*. Enfin, la classification est réalisée

[5] https://dbs.uni-leipzig.de/en/research/ projects/object_matching/fever/benchmark_datasets_for_entity_resolution
[6] http: //www.cs.utexas.edu/users/ml/riddle/data.html
[7] https://sites.google.com/site/anhaidgroup/ projects/data.

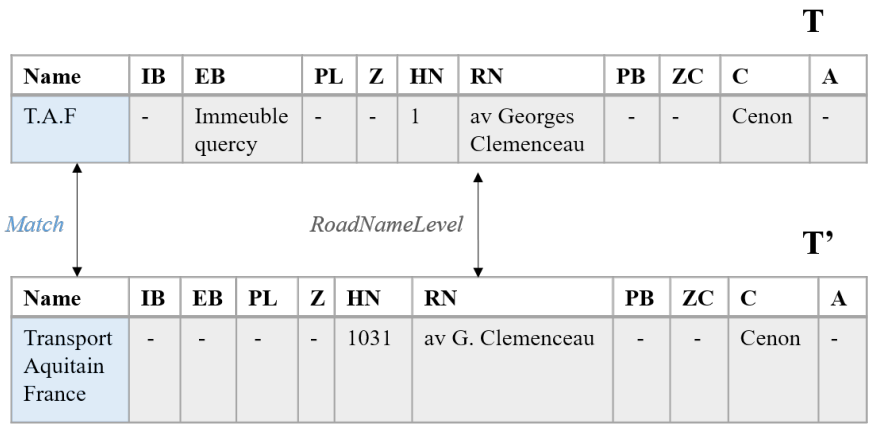

Figure 2: Exemple d'appariement d'entités

Table 5: Niveaux d'appariement d'adresses

Attributs appariés	Niveaux
-	NoMatch
City	CityLevel
City + Zone	ZoneLevel
City + (Zone) + PoiLogistic	PoiLevel
City + (Zone) + (PoiLogistic) + RoadName	RoadNameLevel
City + (Zone) + (PoiLogistic) + RoadName + HouseNum	HouseNumLevel
City + (Zone) + (PoiLogistic) + (RoadName) + ExtBuilding	ExtBuildingLevel
City + (Zone) + (PoiLogistic) + (RoadName) + ExtBuilding + InBuilding	InBuildingLevel
City + Additional	AdditionalLevel

par un algorithme d'apprentissage profond: *"ReLU HighwayNet"* [23]. L'approche a été évaluée par rapport à Magellan sur différents jeux de données réelles structurées, textuelles et bruitées. Les évaluations ont montré qu'elle était plus performante sur les données textuelles et bruitées.

Notre étude de l'existant nous permet d'avoir les observations suivantes: les méthodes basées sur l'apprentissage profond ne sont performantes que si le jeu d'apprentissage est volumineux. Les résultats d'application de ces méthodes sur des jeux de données de petite taille, qui représente notre cas de figure dans ce travail, n'apportaient par forcément des améliorations par rapport aux méthodes existantes. Cette observation est confirmée par les expérimentations réalisées dans [16]. De plus, ces méthodes supportent uniquement la classification binaire des entités en Match/NoMatch.

Dans l'industrie, des outils comme Informatica Data Quality [8], DataMatch [9] ou Tamr [10] ont été proposés pour réaliser l'Entity Matching. Une étude comparative a été réalisé par [3] sur 15 solutions commerciales d'Entity Matching. La plupart de ces outils réalisent le matching d'entités en utilisant des méthodes à base de règles.

3.2 Matching d'adresses et de noms de sociétés

La plupart des travaux de matching d'adresses et de noms de sociétés dans la littérature [6, 10, 13] reposent sur le matching syntaxique en utilisant une combinaison de techniques de similarités de chaînes de caractéres avec des règles ou bien avec des algorithmes d'apprentissage classique.

Dans [6], les auteurs proposent un système de *Record Linkage* pour le matching des sociétés. Ils utilisent une combinaison des distances de Levenshtein [11] et Jaccard [7] pour calculer un score de similarité entre les noms de sociétés. Des poids différents sont attribués pour les *tokens* des noms de sociétés (poids important pour l'alias du nom de la société et un poids faible pour le lieu, le statut juridique). Cette approche a été évaluée par rapport au système d'indexation distribué *ApacheSolr*[11]. Les résultats d'évaluation ont montré que l'approche de [6] est plus performante que Apache Solr en termes de précision et rappel.

Dans [1], les auteurs construisent des vecteurs de mots pour chaque attribut d'adresse en utilisant Word2vec [15], une méthode qui permet la représentation distribuée des mots basée sur des réseaux de neurones à deux couches et cherche à apprendre les représentations vectorielles des mots composant un corpus de telle sorte que les mots qui partagent des contextes similaires soient

[8] https://kb.informatica.com/h2l/HowTo%20Library/1/0816-IdentifyingDuplicateRecordsinIDQ9.pdf
[9] https://dataladder.com/products/datamatch-enterprise/
[10] https://docs.tamr.com/new/docs/overall-workflow-mastering
[11] https://lucene.apache.org/solr/

représentés par des vecteurs numériques proches [12]. Ensuite, un vecteur de comparaison pour chaque paire d'adresses est construit contenant la valeur de similarité entre les vecteurs de mots déjà crées par Word2vec en utilisant la similarité cosinus. Enfin, trois méthodes de classification sont testées pour classifier les paires en Match/Nomatch: le modèle de regression logistique, les forêts d'arbres décisionnels (Random forest) et XGBoost. Le matching est réalisé entre 2 bases de données comportant des adresses appartenant au Royaume-Uni. Cependant, vu que le vecteur de comparaison utilisé entre les adresses n'est composé que des valeurs de similarité entre les attributs d'adresses de même nature, cette approche ne tient pas compte de matching des données injectées sous les mauvais attributs.

Dans [12], une approche de matching d'adresses combinant la technique de plongement de mots et l'apprentissage profond a été proposée. Les adresses sont, dans un premier temps, transformées en des vecteurs en utilisant Word2vec. Ensuite, un modèle d'inférence séquentielle amélioré (ESIM) est appliqué sur ces vecteurs. ESIM est un modèle de correspondance de texte approfondi permettant de calculer la similarité syntaxique et sémantique entre les adresses. L'approche est testée sur des adresses réelles de la ville de Shenzhen en Chine. Les résultats obtenus montrent que l'approche proposée est plus performante (avec un F1-score de 0.97) que d'autres approches, non basées sur l'apprentissage profond. Cependant, cette approche nécessite un jeu de données volumineux pour l'apprentissage. De plus, aucun détail n'est fourni sur l'extensiblité de ces méthodes pour tenir compte de la classification multi-classes des adresses.

3.3 Synthèse

Les méthodes basées sur les techniques de similarité en combinaisons avec des règles ou bien avec des techniques d'apprentissage sont performantes que dans le cas d'appariement syntaxique sur des données peu bruitées. Contrairement à ces méthodes, les techniques d'appariement combinant la représentation distribuée des entités avec des techniques d'apprentissage profond résolvent le problème d'appariement sémantique des entités et sont aussi performantes sur des données non-structurées polluées. Cependant, ces techniques nécessitent un jeu d'apprentissage volumineux et par suite un grand effort de labellisation de données. De plus, elles permettent de réaliser que la classification binaire des entités en Match/NoMatch. A la différence des approches existantes, notre solution proposée permet de résoudre tous les problèmes précédemment cités.

4 MÉTHODOLOGIE PROPOSÉE

Nous proposons une approche de Matching des entités définies par les noms des sociétés et leurs adresses dans le domaine Transport et Logistique (Figure 3). L'approche comporte deux phases: 1) la standardisation des entités et 2) l'appariement par apprentissage supervisé sur des représentations vectorielles sémantiques des entités obtenues par des techniques de représentation distribuée des mots.

[12] dataanalyticspost.com/Lexique/Word2vec/

Table 6: Keyword-Abbreviation list

Attributs	Mots clés	Abréviations
RoadName	BOULEVARD AVENUE	BLVD-BLD-BVD AV-AVE
InBuilding	ETAGE APPARTEMENT	ETG APT
ExtBuilding	BATIMENT IMMEUBLE	BAT IMM
PoiLogistic	AEROGARE AEROPORT	AERG AER
Zone	ZONE INDUSTRIELLE ZONE LOGISTIQUE	ZI ZL
PoBox	BOITE POSTALE CASE POSTAL	BP CP

4.1 Standardisation des entités

La standardisation désigne la transformation des données vers un format standard normalisé, en utilisant deux étapes: 1) le prétraitement et 2) le parsing.

Pré-traitement Le but de cette étape est la normalisation des entités et la correction des typos et des fautes d'orthographe. La normalisation consiste à étendre les abréviations, capitaliser les mots, identifier les acronymes (e.g A.T.B. vs AFFRETEMENTS ET TRANSPORTS BECKER) et supprimer les signes de ponctuation.

L'expansion consiste à reconnaître les abréviations et à les étendre à leurs mots correspondants. Nous avons utilisé une liste *Keyword-Abbreviation list* (Table 6) qui contient un ensemble de mots clés susceptibles d'être utilisés pour définir les composants du modèle d'adresse (par exemple InBuilding, ExtBuilding, RoadName, City) et leurs abréviations. Pour ce faire, nous avons extrait des données de sources officielles comme la Poste, le service INSEE [13] et des sources non officielles qui ont généralement des abréviations communes, comme la liste des abréviations reconnues par les outils d'interrogation de OpenStreetMap [14]. De plus, nous nous sommes intéressés à la correction des erreurs d'orthographe principalement dans les mots clés des adresses. Ces mots ont un rôle primordial dans le parsing d'adresses. Pour cela, nous avons utilisé un correcteur orthographique *pyspellchecker* [15]/ [16] qui, lorsqu'il détecte une faute d'orthographe dans un mot, propose une liste de candidats potentiels pour remplacer ce mot. Ensuite, nous avons effectué une correspondance exacte entre chaque mot de cette liste et les mots clés des éléments d'une adresse.

Parsing Le but du parsing est d'identifier les différents éléments d'une adresse définis dans le modèle (figure 1):

- identification des éléments d'adresse tels que RoadName, InBuilding, ExtBuilding, PoiLogistic, Zone et PoBox en se basant sur la liste de mots clés (Keyword-Abbreviation list). Par exemple, l'adresse suivante: "Rue de Quebec Zone industrielle Chef de Baie BP 2088" est divisée en éléments d'adresses suivants: "**Rue** de Quebec", "**Zone industrielle** Chef de Baie","**BP** 2088".

[13] https://www.sirene.fr/sirene/public/variable/typvoie
[14] https://wiki.openstreetmap.org/wiki/Name_finder:Abbreviations
[15] https://readthedocs.org/projects/pyspellchecker/downloads/pdf/latest/
[16] https://norvig.com/spell-correct.html

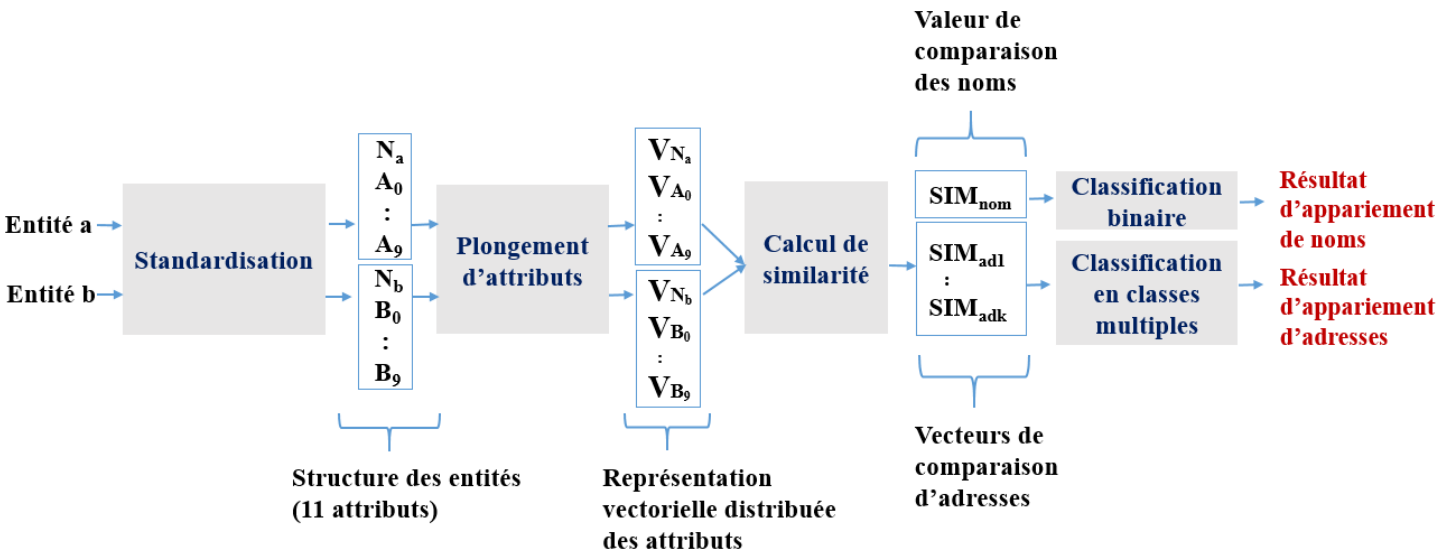

Figure 3: Approche d'appariement d'entités

- identification des chiffres dans l'adresse comme HouseNum et ZipCode dans l'adresse, en utilisant les règles définies par le Service National de l'adresse (SNA) [17], un service développé par La Poste pour favoriser l'amélioration des fichiers courrier utilisés par les différents utilisateurs des services postaux. Les régles sont décrites comme suit:
  - un numéro de rue (HouseNum) doit être placé avant le nom de la rue (RoadName) et doit avoir un maximum de 4 chiffres.
  - Un code postal (ZipCode) doit contenir 5 chiffres et doit précéder une ville.
- identification de la ville (City) représentée par un jeton ou une séquence de jetons. Nous identifions deux cas:
  - un nom de ville connu appartenant au UN-LOCODE gazetteer [18], qui contient les villes et communes françaises avec des informations sur leurs subdivisions et coordonnées géographiques.
  - une ville inconnue mais précédée directement d'un code postal ou suivie d'un nom de pays (France). Dans ce cas, la validation du nom de la ville sera réalisée à l'étape de matching d'adresses.

4.2 Appariement d'entités

L'objectif de cette étape est de réaliser la mise en correspondance entre les différentes entités (noms de sociétés et adresses). Dans un premier temps, nous attribuons manuellement des labels aux paires d'adresses et des noms de sociétés. Les labels d'adresses sont les niveaux de matching d'adresse décrits dans la table 5. Les labels de noms de sociétés sont soit "Match" soit "NoMatch". Puis, nous appliquons les techniques de plongement d'attributs (attribute embedding) nous permettant d'avoir une représentation vectorielle des mots. Ensuite, nous calculons les similarités entre les différents vecteurs. Enfin, nous appliquons un algorithme d'apprentissage supervisé sur les vecteurs de comparaison et les labels afin de classifier les paires d'entités. L'alogrithme 1 décrit, particulièrement, le processus de matching d'adresses.

Plongement d'attributs (Attribute Embedding) La création des vecteurs de représentation des noms des sociétés et des attributs d'adresses est réalisée avec fastText qui est une méthode de représentation distribuée des mots de la famille *Character Embedding*. fastText est une variante de Word2vec incluant des séquences de caractères dans l'apprentissage de plongement (*Embedding*). A la différence de Word2vec, fastText représente les mots par l'ensemble de ses n-grammes. Le vecteur de représentation d'un mot est alors la somme des vecteurs de représentation de tous ses n-grammes.

Calcul de similarité entre les vecteurs de représentation d'entités A l'issue de l'étape de plongement d'attributs, chaque entité sera représentée par plusieurs vecteurs (un vecteur par nom de société et par attribut d'adresse). Le calcul du vecteur de comparaison entre deux entités est réalisé en utilisant la similarité cosinus (équation (1)). En effet, cette mesure calcule la similarité entre des vecteurs (de nombres) contrairement à plusieurs techniques de similarité qui comparent des champs textuels. De plus, elle permet de capturer la proximité sémantique entre les représentations vectorielles des mots [2, 12, 14].

Les valeurs du vecteur de comparaison des adresses ont 3 origines:

- valeurs représentant les similarités entre les mêmes types d'attributs.
- valeurs représentant les similarités entre l'attribut "Additional" et les attributs ExtBuilding, PoiLogistic, Zone et RoadName. Nous avons calculé ces valeurs de similarité pour tenir

[17] https://www.definitions-marketing.com/definition/service-national-de-l-adresse/
[18] https://service.unece.org/trade/locode/fr.htm

compte des éléments d'adresse qui sont injectés dans les mauvais attributs à l'issue de la phase de parsing d'adresses.
- valeurs représentant les similarités entre toutes les combinaisons possibles de paires incluant les attributs ExtBuilding, PoiLogistic, Zone, RoadName (sauf la combinaison de paires de vecteurs appartenant au même type d'attribut dont on a déjà tenu compte). Nous avons calculé ces valeurs pour traiter les cas où on a une similarité sémantique, qui peut avoir lieu entre des attributs de différentes nature (e.g. calculer la similarité entre les attributs RoadName et Zone puisque la zone peut contenir la rue donc forcément on aura une similarité entre les deux attributs).

$$\cos(\mathbf{V}, \mathbf{V'}) = \frac{\mathbf{V}\mathbf{V'}}{\|\mathbf{V}\|\|\mathbf{V'}\|} = \frac{\sum_{i=1}^{n} \mathbf{V}_i \mathbf{V'}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{V}_i)^2}\sqrt{\sum_{i=1}^{n} (\mathbf{V'}_i)^2}} \quad (1)$$

Algorithm 1 Appariement d'adresses

Entrée: Tables T, T' (chaque tuple est composé d'une adresse avec K + 1 attributs ; $K \in \{0..9\}$), Base d'adresses DB
Sortie: Résultat d'appariement (classe correspondante) pour chaque tuple d'adresses
// Pour chaque tuple, calculer la représentation distribuée de tous ses attributs
Réaliser l'apprentissage d'un modèle de représentation distribuée de mots WE en utilisant la base d'adresses DB.
pour chaque paire de tuples dans T x T' **faire**
    Calculer la représentation vectorielle V_k pour chaque valeur d'attribut de t (t[A_k]) en utilisant le modèle WE.
    Calculer la représentation vectorielle V'_k pour chaque valeur d'attribut de t' (t'[A_k]) en utilisant le modèle WE.
fin pour
// Créer le vecteur de comparaison W pour chaque paire de tuples en utilisant la similarité cosinus (cos)
$W := \emptyset$; $W1 := \emptyset$; $W2 := \emptyset$; $W3 := \emptyset$
pour chaque paire de tuples dans T x T' **faire**
    $W1 := cos(V_k, V'_k)$
    $W2 := cos(V_9, V'_i) \cup cos(V_i, V'_9)$; $i \in \{1, 2, 3, 5\}$
    $W3 := cos(V_x, V'_y)$; $x \in \{1, 2, 3, 5\}$, $y \in \{1, 2, 3, 5\}$ et x ≠ y
    $W := W1 \cup W2 \cup W3$
fin pour
// Classification
Diviser T x T' en jeu d'apprentissage S et jeu de test TS
Réaliser l'apprentissage sur les vecteurs de comparaisons de S ainsi que les labels des paires d'adresses en se basant sur un classificateur C.
pour chaque paire de tuples (t,t') dans TS **faire**
    Prédire la classe d'appariement de (t, t') en utilisant C
fin pour

Apprentissage supervisé pour la classification des paires d'entités Une fois que nous obtenons le vecteur de comparaison pour chaque paire d'entités, nous divisons le jeu de données en un jeu d'apprentissage et un jeu de test. Ensuite, nous réalisons l'apprentissage, en utilisant le jeu d'apprentissage, sur les vecteurs de comparaisons ainsi que les labels des paires d'adresses et ceux des pairs de noms de sociétés en se basant sur plusieurs méthodes d'apprentissage supervisé (section 5.2). Le jeu de tests est ensuite utilisé pour évaluer la classification des différents modèles d'apprentissage utilisés.

5 EXPÉRIMENTATION ET RÉSULTATS

Nous décrivons, dans cette section, les expérimentations réalisées afin d'évaluer notre approche de matching d'adresses et de noms de sociétés.

5.1 Jeux de données

Nous disposons d'un jeu de données réelles comprenant 3712 entités françaises à vérifier qui représentent des noms de sociétés (issus du domaine du transport et logistique) et leurs adresses.

Afin de vérifier ces entités, nous avons comparé notre jeu de données avec des entités présentes dans des annuaires tels que Kompass [19]. Nous avons extrait les données de l'annuaire en utilisant la bibliothèque python BeautifulSoup [20]. Nous obtenons alors un jeu de données composé de 3123 paires d'entités: l'entité initiale à vérifier et le(s) résultat(s) récupéré(s) de kompass. Comme notre approche de matching d'entités est basée sur l'apprentissage supervisé, nous avons alors procédé à la labellisation manuelle de ce jeu de données.

5.2 Etude expérimentale et évaluation

Nous avons réalisé plusieurs expérimentations afin d'évaluer la qualité de notre approche de matching.

Métrique d'évaluation Durant nos expérimentations, nous avons utilisé F1-score comme métrique d'évaluation:

$$F1 - score = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

tels que:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$$
$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

TP, FP and FN représente respectivement le nombre de *True Positives*, *False Positives* et *False Negatives*.

Configuration expérimentale Notre approche est basée sur la combinaison d'une technique de représentation distribuée de mots avec des algorithmes de classification supervisé. Nous avons appliqué deux méthodes de représentation vectorielles des mots qui sont: Word2vec et fastText. Nous avons utilisé la bibliothèque Gensim [20] de python afin de les implémenter. Le jeu d'apprentissage de Word2vec et fastText est le jeu de données composé des paires d'entités initiales augmenté par une base d'adresses françaises [21], réelle et publique, qui comporte 1 million d'adresses. Cette base contient des adresses des établissements français transmises à l'Institut national de la propriété industrielle (INPI) dans le cadre des ses missions. Dans le cas de matching de noms de sociétés, le

[19] https://fr.kompass.com/
[20] https://www.crummy.com/software/BeautifulSoup/bs4/doc/
[21] https://public.opendatasoft.com/explore/dataset/inpi-liste-des-etablissements/table/?disjunctive.nom_greffe

Table 7: Résultats d'appariement d'entités

Appariement de noms	Appariement d'adresses	Fréquence
NoMatch	-	1111
Match	NoMatch	320
Match	CityLevel	527
Match	ZoneLevel	177
Match	PoiLevel	12
Match	RoadNameLevel	302
Match	HouseNumLevel	587
Match	ExtBuildingLevel	14
Match	InBuildingLevel	0
Match	AdditionalLevel	73

jeu d'apprentissage de fastText est le jeu de données composé des paires des noms de sociétés.

Les algorithmes de classification utilisés dans notre approche sont: *Support Vector Machine (SVM)*, *Decision Tree*, *Random Forest*, *XGBoost* et *Multi-layer Perceptron* (avec une seule couche cachée). De plus, nous avons utilisé *Stratified K-fold Cross-Validation*, tel que k=10, comme méthode d'évaluation des modèles des deux approches. Cette méthode permet la division des données en 10 sous-échantillons de taille égale. Les données sont distribuées de telle sorte que chaque sous-échantillon contient les différents types de labels avec des pourcentages égaux avec le jeu de données complet. En total, 10 itérations sont effectuées: dans chaque itération, l'apprentissage est réalisé sur 9 sous-échantillons et le test est réalisé sur un sous-échantillon. La moyenne des F1-score dans les 10 itérations est utilisée dans l'évaluation des résultats.

5.3 Résultats expérimentaux

Nous avons réalisé plusieurs expérimentations afin d'évaluer notre approche d'appariement d'entités du Transport et Logistique.

Résultats d'appariement d'entités La table 7 présente le résultat d'appariement d'entités. Les paires d'entités qui possédent le label "NoMAtch", soit dans l'appariement de noms soit dans l'appariement d'adresses, contiennent des entités de notre jeu de données non validées par rapport aux données de Kompass, soit un total de 1431 entités. D'après les résultats trouvés, nous pouvons conclure que la vérification d'entités avec un seul annuaire n'est pas suffisante (54% d'entités sont non validées). La vérification auprès d'autre annuaires comme societe.com et infogreffe est donc nécessaire.

Comparaison avec des méthodes existantes Nous avons comparé notre approche d'appariement d'adresses avec l'outil Magellan. La table 8 illustre les résultats obtenus en termes de F1-scores. Nous remarquons que notre approche est plus performante (L'écart de F1-score entre notre approche et Magellan est 0.04) . En effet, notre approche tient compte du matching sémantique entre les adresses, ce qui n'est pas la cas avec Magellan qui se base principalement sur la similitude lexicale.

Nous avons également évalué l'appariement des noms des sociétés. La table 9 illustre les résultats obtenus en termes de F1-scores en comparaison avec Magellan. Nous remarquons que notre approche est compétitive avec Magellan. En effet, les problèmes existants (e.g erreurs d'orthographe, abbréviations et acronymes, permutation de mots) dans les noms de sociétés peuvent être résolus avec des approches de matching syntaxique.

Table 8: Comparaison des F1-scores pour les différentes méthodes de matching d'adresses

Méthodes	F1-scores
Magellan	0.907
fastText + SVM	0.942
fastText + Decision Tree	0.938
fastText + Random Forest	0.932
fastText + XGBoost	0.938
fastText + MLP	0.94

Table 9: Comparaison des F1-scores pour les différentes méthodes de matching de noms de sociétés

Méthodes	F1-scores
Magellan	0.917
fastText + SVM	0.925
fastText + Decision Tree	0.917
fastText + Random Forest	0.894
fastText + XGBoost	0.915
fastText + MLP	0.917

Table 10: Comparaison de différentes méthodes de représentation distribuée des mots

	Word2vec	fastText
SVM	0.914	0.942
Decision Tree	0.906	0.938
Random Forest	0.891	0.932
XGBoost	0.895	0.938
MLP	0.912	0.940

Comparaison de différentes méthodes de représentation distribuée des mots Nous avons comparé deux différentes méthodes : Word2vec et fastText. La table 10 montre que l'utilisation de fastText permet d'obtenir des résultats plus précis que ceux obtenus avec Word2vec. Ceci est principalement dû au fait que fastText tient compte des erreurs d'orthographe dans la représentation vectorielle des mots (e.g fastText attribue des représentations vectorielles similaires pour les deux mots MARCHAL et MARECHAL suite à l'appariement de *AVENUE MARCHAL FOCH* vs *AVENUE MARECHAL FOCH*).

Evaluation de la variation de la taille du jeu d'apprentissage du word embedding Nous avons étudié l'impact de la variation de la taille du jeu d'apprentissage utilisé pour fastText. Pour cela, nous avons pris deux jeux d'apprentissage de tailles différentes : *small Training set* qui représente le jeu données constitué d'adresses à vérifier et celles extraites de Kompass et *large Training set* qui comporte 1 million d'adresses Françaises collectées par INPI. La table 11 montre les résultats d'évaluation de notre approche avec

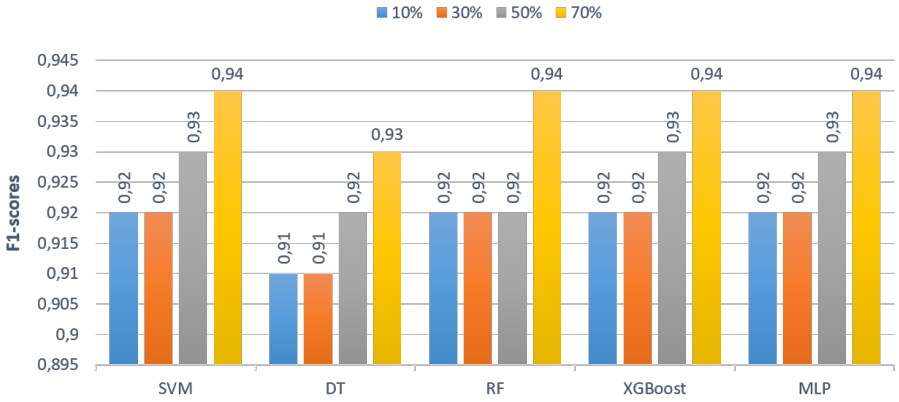

Figure 4: Impact de la variation de la taille du jeu d'apprentissage supervisé sur la performance de notre approche

Table 11: Impact de la variation de la taille de jeu d'apprentissage de fastText

	fastText + small Training set	**fastText + Large Training set**
SVM	0.916	0.942
Decision Tree	0.920	0.938
Random Forest	0.912	0.932
XGBoost	0.914	0.938
MLP	0.920	0.940

les deux différents jeux d'apprentissage de fastText. Nous remarquons que la performance augmente avec l'utilisation d'un jeu d'apprentissage volumineux pour fastText. En effet, les méthodes de word embedding nécessitent une grande quantité de données d'apprentissage afin de mieux détecter les relations sémantiques entre les mots.

Evaluation de la variation de la taille du jeu d'apprentissage supervisé La figure 4 montre l'effet de la variation de la taille de jeu d'apprentissage supervisé sur la performance de notre approche. Nous avons comparé la performance de notre approche avec 4 différentes tailles du jeu d'apprentissage: 10%, 30%, 50% et 70% du jeu de données initial (3123 pairs d'entités). Les résultats obtenus montrent que même avec un jeu d'apprentissage de petite taille, c'est à dire qui représente 10% (313 paires d'entité) du jeu de données initial, notre approche reste performante. Pour tous les algorithmes d'apprentissage utilisés, le maximum d'écart de F1-score entre un jeu d'apprentissage de taille 70% avec celui de 10% est égale à 0.02. La robustesse de notre approche est dûe principalement à la phase d'apprentissage non supervisé réalisée sur un large volume de données (1 million d'adresses) pour obtenir les représentations vectorielles des adresses.

6 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons décrit notre approche et solution pour le problème du matching d'entités dans le domaine *Transport et Logistique*, pour lequel la phase de matching est une étape cruciale pour la vérification des entités (entreprises avec leurs adresses). Nous avons illustré, à travers des exemples réels, quelle était la complexité du problème.

La solution que nous proposons s'appuie sur des techniques de word embedding et d'apprentissage supervisé. Une implémentation avec des expérimentations sur des données réelles "françaises" ont été réalisées. Les premiers résultats obtenus, avec des mesures F1 supérieurs à 0.9, sont encourageants mais nous devons les relativiser compte tenu de la taille des échantillons.

Il nous reste maintenant à étendre ce travail selon au moins deux directions : la gestion de données volumineuses et la prise en compte de données internationales.

REFERENCES

[1] Sam Comber and Daniel Arribas-Bel. 2019. Machine learning innovations in address matching: A practical comparison of word2vec and CRFs. *Transactions in GIS* 23, 2 (2019), 334–348.
[2] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq Joty, Mourad Ouzzani, and Nan Tang. 2018. Distributed representations of tuples for entity resolution. *Proceedings of the VLDB Endowment* 11, 11 (2018), 1454–1467.
[3] P. Konda et al. Magellan. 2016. Toward building entity matching management systems. http://www.cs.wisc.edu/~anhai/papers/magellan-tr.pdf. [Technical Report].
[4] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. 2009. Reasoning about record matching rules. *Proceedings of the VLDB Endowment* 2, 1 (2009), 407–418.
[5] Yash Govind, Pradap Konda, Paul Suganthan GC, Philip Martinkus, Palaniappan Nagarajan, Han Li, Aravind Soundararajan, Sidharth Mudgal, Jeff R Ballard, Haojun Zhang, et al. 2019. Entity Matching Meets Data Science: A Progress Report from the Magellan Project. In *Proceedings of the 2019 International Conference on Management of Data*. 389–403.
[6] Thomas Gschwind, Christoph Miksovic, Julian Minder, Katsiaryna Mirylenka, and Paolo Scotton. 2019. Fast record linkage for company entities. In *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 623–630.
[7] Paul Jaccard. 1908. Nouvelles recherches sur la distribution florale. *Bull. Soc. Vaud. Sci. Nat.* 44 (1908), 223–270.
[8] Pradap Konda, Sanjib Das, Paul Suganthan GC, AnHai Doan, Adel Ardalan, Jeffrey R Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, et al. 2016. Magellan: Toward building entity matching management systems. *Proceedings of the VLDB Endowment* 9, 12 (2016), 1197–1208.
[9] Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of entity resolution approaches on real-world match problems. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 484–493.
[10] Ioannis Koumarelas, Axel Kroschk, Clifford Mosley, and Felix Naumann. 2018. Experience: Enhancing address matching with geocoding and similarity measure

selection. *Journal of Data and Information Quality (JDIQ)* 10, 2 (2018), 1–16.
[11] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.
[12] Yue Lin, Mengjun Kang, Yuyang Wu, Qingyun Du, and Tao Liu. 2020. A deep learning architecture for semantic address matching. *International Journal of Geographical Information Science* 34, 3 (2020), 559–576.
[13] Dilek Küçük Matci and Uğur Avdan. 2018. Address standardization using the natural language process for improving geocoding results. *Computers, Environment and Urban Systems* 70 (2018), 1–8.
[14] Bridget T McInnes and Ted Pedersen. 2013. Evaluating measures of semantic similarity and relatedness to disambiguate terms in biomedical text. *Journal of biomedical informatics* 46, 6 (2013), 1116–1124.
[15] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).
[16] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. 2018. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 International Conference on Management of Data*. 19–34.
[17] George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, and Manolis Koubarakis. 2018. The return of jedai: End-to-end entity resolution for structured and semi-structured data. *Proceedings of the VLDB Endowment* 11, 12 (2018), 1950–1953.
[18] George Papadakis, Leonidas Tsekouras, Emmanouil Thanos, George Giannakopoulos, Themis Palpanas, and Manolis Koubarakis. 2020. Domain-and Structure-Agnostic End-to-End Entity Resolution with JedAI. *ACM SIGMOD Record* 48, 4 (2020), 30–36.
[19] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1532–1543.
[20] Radim Rehurek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer.
[21] Rohit Singh, Vamsi Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Generating concise entity matching rules. In *Proceedings of the 2017 ACM International Conference on Management of Data*. 1635–1638.
[22] Rohit Singh, Venkata Vamsikrishna Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. 2017. Synthesizing entity matching rules by examples. *Proceedings of the VLDB Endowment* 11, 2 (2017), 189–202.
[23] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387* (2015).
[24] William E Winkler. 1990. String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage. (1990).