Des données tabulaires aux graphes de connaissances : état de l'art des méthodes d'interprétation sémantique de tables

Jixiong Liu, Viet-Phi Huynh, Yoan Chabot, Raphaël Troncy

Des données tabulaires aux graphes de connaissances : état de l'art des méthodes d'interprétation sémantique de tables

Jixiong Liu[1,2], Viet-Phi Huynh[1], Yoan Chabot[1], Raphaël Troncy[2]

[1] Orange, France
[2] EURECOM, Sophia Antipolis, France

yoan.chabot@orange.com

Résumé

Les données tabulaires sont omniprésentes sur le Web et dans les entrepôts de données des entreprises. Ces tableaux contiennent des informations pouvant potentiellement devenir des connaissances après une étape d'interprétation sémantique de tables se basant sur un graphe de connaissances. Ce papier propose un état de l'art des différentes tâches et méthodes existantes pour mener à bien cette interprétation. Dans un premier temps, nous proposons une nouvelle classification des tableaux reflétant la diversité et la complexité de ces structures. Nous décomposons ensuite le problème de l'interprétation sémantique en cinq sous-tâches et passons en revue trois familles d'approches au travers du prisme des corpus d'évaluation proposés par la communauté.

Mots-clés

Interprétation sémantique de tables, Annotation, Données tabulaires, Graphes de connaissances

Abstract

Tabular data are widely spread on the Web and in corporate data repositories. They contain information that can potentially become knowledge after a step called semantic interpretation based on a knowledge graph. This paper provides a state of the art of the different tasks and methods to carry out this interpretation. First, we propose a new classification of tabular data reflecting the diversity and complexity of these structures. We then decompose the problem of semantic interpretation of tables into five sub-tasks and review three families of approaches through the prism of evaluation corpora proposed within the community.

Keywords

Semantic Table Interpretation, Table annotation, Tabular data, Knowledge graph

1 Introduction

Les formats de données tels que CSV et XLS sont couramment utilisés en entrée d'algorithmes d'apprentissage automatique. Ainsi, l'interprétation des données tabulaires est une tâche ayant attiré beaucoup d'attention ces dernières années, avec notamment la cristallisation des efforts de recherche autour de challenges scientifiques comme SemTab [23]. Pour rendre les données tabulaires intelligibles, l'idée principale est de trouver des correspondances entre les éléments composant le tableau et les entités/concepts/relations décrits dans les graphes de connaissances (KG) encyclopédiques comme DBpedia [7] et Wikidata [53], ou spécifiques à l'entreprise. Ce problème est connu sous le nom d'annotation de données tabulaires (ou STI en anglais pour Semantic Table Interpretation). Les KG peuvent être utilisés pour guider l'interprétation sémantique tout en étant eux-mêmes les artefacts pouvant être enrichis ou corrigés par le résultat de l'interprétation. L'annotation des données tabulaires avec des entités sémantiques ouvre la voie à une utilisation plus intelligente des données. Elle permet d'envisager notamment des services basés sur ces nouvelles informations sémantique (e.g., indexation, recherche et recommandation d'informations à un niveau conceptuel) et de contribuer à l'amélioration des systèmes de questions/réponses.

L'interprétation automatique des données tabulaires est un problème complexe en raison du contexte limité disponible pour résoudre les ambiguïtés, du format de présentation des tableaux, et du caractère incomplet des KG vis à vis des connaissances présentes dans les tableaux. Ce papier vise à définir les différentes sous-tâches d'annotation de données tabulaires et à passer en revue les méthodes qui ont été proposées à ce jour, ainsi que leurs performances sur des ensembles de données d'évaluation bien établis dans la communauté.

La structure de ce papier est la suivante. La section 2 définit les notions essentielles inhérentes aux données tabulaires et propose une nouvelle taxonomie des types de tableaux. La section 3 définit ensuite les tâches liées à l'annotation de données tabulaires. La section 4, passe en revue les représentants de trois familles d'approches (non mutuellement exclusives) respectivement basées sur des heuristiques, sur de l'ingénierie de caractéristiques (ou feature engineering) et de l'apprentissage profond. La section 5 évalue les performances des systèmes STI puis la section 6 liste les défis scientifiques subsistants. Le lecteur peut se référer à [28] pour plus d'informations sur les jeux de données, les benchmarks ou encore les méthodes utilisées dans chacune des approches de l'état de l'art.

2 Données tabulaires

La première source d'information d'un système STI est la table elle-même (dans la suite du papier, les termes "table" et "tableau" seront utilisés de manière équivalente). Un tableau est un arrangement bidimensionnel de données comportant n lignes et m colonnes. Une cellule est l'élément de base d'un tableau où $\mathcal{T}_{ij}$ $(0 \leq i \leq n-1, 0 \leq j \leq m-1)$ indique la cellule de la ligne i et de la colonne j du tableau $\mathcal{T}$. Outre les données qu'elles contiennent, les métadonnées et le contexte dans lequel les tables apparaissent constituent des informations précieuses pour l'inteprétation. Par exemple, si un tableau a été publié sur une page web décrivant la Bundesliga, cette table est probablement plus en rapport avec le football qu'avec n'importe quel autre sport. Il est ainsi utile de collecter à la fois le tableau lui-même et ses métadonnées lors de l'extraction des données.

Avant d'interpréter un tableau, il est important d'identifier son type afin de prendre en compte ses spécificités dans le processus de STI. Étant donné l'importante hétérogénéité des tables en termes de format, de provenance et d'utilisation, nous introduisons dans cette section une classification des types de tableaux (figure 1) basée sur les classifications existantes avec une analyse plus approfondie des tables relationnelles. Cette classification des tables vise à faciliter la définition du champ d'application des approches et à aider à mieux décrire les défis liés aux tâches de STI.

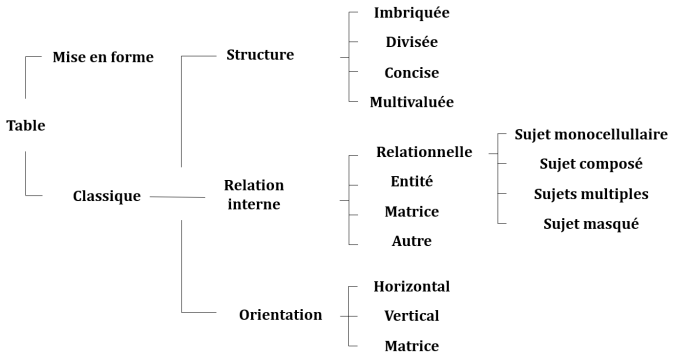

FIGURE 1 – Classification des types de tables.

Les tableaux peuvent être divisés en deux grandes catégories. **Les tables de mise en forme** sont utilisées pour structurer et formater les pages Web. Ces tables ne contiennent pas de relations sémantiques et sont utilisées pour organiser visuellement le contenu d'une page afin de maximiser le confort de l'utilisateur et l'ergonomie d'un site.

Les tables dites classiques contiennent des connaissances. Ces tableaux présentent un niveau élevé de cohérence (syntaxique et sémantique) entre les lignes et les colonnes. Les tableaux de cette classe contiennent des connaissances qui peuvent être interprétées et constituent donc des données d'entrée pour le processus de STI.

Nous proposons ensuite de classer les tableaux classiques en fonction de trois dimensions non mutuellement exclusives : la structure, les relations internes et l'orientation. Les types de tableaux sont ensuite formés par la composition de ces dimensions. Par exemple, le tableau représenté dans la figure 2(b) sur les lignes de chemin de fer est un tableau concis (dimension structurelle), un tableau horizontal (orientation) et un tableau relationnel à sujet composé (relation interne). Dans la suite, nous définissons plus en détail chacune de ces trois dimensions.

La sous-classe **structure** de notre classification est divisée en quatre types de tableaux. Les **tableaux imbriqués** contiennent un ou plusieurs tableaux dans une ou plusieurs de leurs cellules. Les **tableaux divisés** sont des tableaux pouvant être divisés en sous-tableaux. Les **tableaux concis** contiennent des cellules fusionnées afin d'éviter les répétitions de cellules faisant référence au même contenu dans les lignes et/ou les colonnes. Les **tableaux multivalués** contiennent plusieurs valeurs (sous la forme d'une énumération non structurée par exemple) dans une seule cellule.

La sous-classe **relation interne** prend en compte les relations sémantiques entre les cellules. Dans notre classification, nous proposons les types suivants. Les **tableaux relationnels** sont des structures dans lesquelles chaque ligne (ou colonne) fournit des informations sur une entité spécifique, et les colonnes (ou lignes) correspondantes représentent des attributs qui décrivent l'entité. Les **tables d'entité**, également appelées tables attribut-valeur, sont utilisées pour décrire une entité unique. Une table d'entité énumère les attributs de l'entité (e.g. infobox Wikipédia). Les **matrices** présentent un arrangement bidimensionnel de données qui doivent être lues simultanément horizontalement et verticalement. Une matrice associe des paires $(\text{row}, \text{column})$ aux valeurs des cellules par le biais d'une propriété unique. Les **autres** tables contiennent des informations mais ne correspondent pas aux types susmentionnés (e.g. les énumérations et les calendriers).

La littérature considère les tables relationnelles comme une feuille dans les taxonomies de types de tables [26]. Cependant, les tables relationnelles présentent une diversité importante, notamment dans la représentation des entités. Nous proposons de les classer plus finement en fonction des caractéristiques de leurs sujets. Le **sujet** d'une ligne d'un tableau relationnel horizontal (resp. d'une colonne d'un tableau relationnel vertical) est une entité qui est décrite par les ensembles de cellules dans cette ligne (resp. colonne). Nous introduisons quatre sous-types de tableaux relationnels (figure 2).

Les **tableaux à sujet monocellulaire** associent chaque ligne d'un tableau horizontal (ou chaque colonne d'un tableau vertical) à un seul sujet. Les mentions (i.e. label représentant une entité) des sujets sont indiquées dans une seule colonne (resp. ligne). Par exemple, dans la figure 2(a), la colonne "Department" contient les sujets. Les autres colonnes décrivent les sujets. Les **tableaux à sujet composé** nécessitent la combinaison de plusieurs cellules pour former le sujet de chaque ligne (resp. colonne). Par exemple, dans le tableau de la figure 2(b), on peut identifier les sujets (classes de train particulières) en fusionnant les colonnes "Lines", "Manufacturer" et "Class". Les **tableaux à sujets multiples** contiennent des cellules qui se réfèrent à des sujets différents tout en étant dans la même ligne.

(a)

Department	Area (km²)	Population (2011)[37]	Municipalities
Paris (75)	105.4	2 249 975	1 (Paris)
Hauts-de-Seine (92)	176	1 581 628	36 (list)
Seine-Saint-Denis (93)	236	1 529 928	40 (list)
Val-de-Marne (94)	245	1 333 702	47 (list)
Petite Couronne	657	4 445 258	123
Paris + Petite Couronne	762.4	6 695 233	124

(b)

Lines	Manufacturer	Class	Image	Number	Car numbers	Built
BART main lines	Rohr	A		59	1164–1276	1968–1975
	Rohr	B		380	1501–1913	1971–1975
	Alstom	C1		150	301–450	1987–1989
	Morrison-Knudsen	C2		80	2501–2580	1994–1996[67]
	Bombardier	D		*310*	3001–3310	2012–
	Bombardier	E		*465*	4001–4465	2012–
Oakland Airport Connector	DCC Doppelmayr	Cable Liner		4	1.3–4.3	2013
eBART	Stadler	GTW		8	101–108	2014–2018

(c)

Release year	Album	Artist/s	Nationality	Worldwide sales (in millions)	Ref(s)
2002	*Come Away With Me*	Norah Jones	United States	**23.9**	[3]
2000	*The Marshall Mathers LP*	Eminem	United States	**23.29**	[4]
2002	*The Eminem Show*	Eminem	United States	**22.95**	[5]
2000	*Hybrid Theory*	Linkin Park	United States	**20.8**	[6]
2015	*25*	Adele	United Kingdom	**20.41**	[7]

(d)

3	11 juin 1998	Italie	**2 - 2**	Chili
4	11 juin 1998	Cameroun	**1 - 1**	Autriche
19	17 juin 1998	Chili	**1 - 1**	Autriche
20	17 juin 1998	Italie	**3 - 0**	Cameroun
33	23 juin 1998	Italie	**2 - 1**	Autriche
34	23 juin 1998	Chili	**1 - 1**	Cameroun

FIGURE 2 – (a) Table à sujet monocellulaire [a], (b) Table à sujet composé [b], (c) Table à sujets multiples [c], (d) Table à sujet masqué [d].

a. https://en.wikipedia.org/wiki/France#Major%20cities
b. https://en.wikipedia.org/wiki/Bay_Area_Rapid_Transit
c. https://en.wikipedia.org/wiki/List_of_best-selling_albums_of_the_21st_century
d. https://fr.wikipedia.org/wiki/Coupe_du_monde_de_football_1998

Dans la figure 2(c), une ligne est composée de deux sujets : “Artist(s)” est le sujet de la colonne “Nationality”, tandis que “Album” est le sujet des colonnes “Release year”, “Artist(s)”, “Worldwide sales” et “Ref(s)”. Les **tableaux à sujet masqué** ne mentionnent pas explicitement le sujet de chaque ligne (resp. colonne). Par exemple, dans la figure 2(d), chaque ligne décrit le résultat d’un match de football, mais la mention du match lui-même n’est pas explicite dans le tableau.

La sous-classe **orientation** spécifie la direction des relations. Connaître le sens des relations structurant un tableau simplifie son interprétation en permettant d’associer les bons attributs à un sujet donné par exemple. Dans les **tableaux horizontaux**, les sujets sont décrits horizontalement, ce qui signifie que chaque ligne décrit un sujet différent. Dans les **tableaux verticaux**, les sujets sont décrits verticalement, ce qui signifie que chaque colonne décrit un sujet différent. Les **matrices**, quant à elles, doivent être interprétées cellule par cellule (et non ligne par ligne ou colonne par colonne) tout en tenant compte des en-têtes horizontaux et verticaux.

3 Interprétation automatique de tables

Après avoir donné les définitions nécessaires à l’étude du domaine, cette section présente le processus de STI en détaillant les cinq tâches qui le composent.

Une tâche d’annotation peut être définie par les éléments du tableau qui doivent être annotés et par le type de candidats considérés (individus, concepts ou propriétés du KG).

Nous proposons de décomposer le domaine de l’interprétation de tables en cinq tâches principales illustrées dans la figure 3 : “cell-entity annotation”, “column-type annotation”, “columns-property annotation” [23], la thématisation (“topic annotation”), et la tâche de correspondance ligne-instance [41].

L’annotation de cellules avec des entités (CEA) a pour but d’annoter une cellule avec une entité d’un KG. Par exemple, dans la figure 3, la tâche de CEA permet de faire correspondre la mention “Suisse” avec l’entité Wikidata Q165141. **L’annotation de colonnes avec des types** (CTA) a pour objectif de faire correspondre une colonne avec un type sémantique (classe) du KG. La difficulté de la tâche CTA réside dans la sélection du type le plus pertinent. Une entité peut être associé à plusieurs types représentés dans des arbres hiérarchiques complexes (par exemple, la topologie des types de Wikidata). Le type sélectionné pour une colonne donnée doit être représentatif des individus qu’elle contient et véhiculer un maximum d’informations. Si le type choisi est trop général (par exemple, la deuxième colonne du tableau de la figure 3 est annotée comme une “geographic entity” (Q27096213) plutôt que comme une “city of Switzerland” (Q1545591)), l’annotation portera peu d’informations. Inversement, un type trop spécifique peut être moins représentatif des valeurs d’une colonne. **L’annotation de colonnes avec des propriétés** (CPA) vise à annoter une paire de colonnes avec une propriété du KG. Par exemple, la relation entre la dernière colonne et la colonne entourée en orange dans la figure 3 doit correspondre au prédicat “number of points/goals/set scored” (P1351). **La thématisation** vise à annoter l’en-

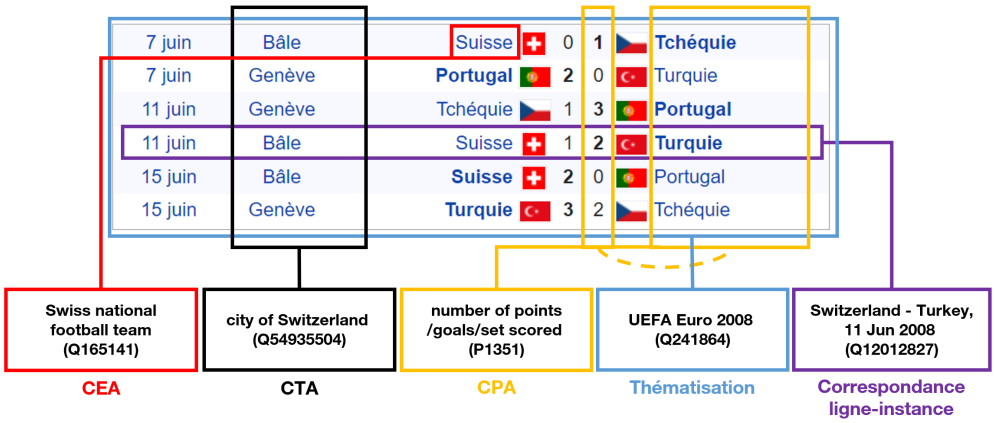

FIGURE 3 – Illustration des cinq tâches de STI sur une table décrivant les résultats du groupe A de l'Euro UEFA 2008 [a].

a. https://fr.wikipedia.org/wiki/Championnat_d%27Europe_de_football_2008#1er_tour_-_phase_de_groupes

semble du tableau avec un concept ou une entité du KG cible. La figure 3 révèle que le tableau entier est lié à l'entité "UEFA Euro 2008" (Q241864) dans Wikidata. **La correspondance ligne-instance** annote une ligne entière d'une table relationnelle avec une entité du KG. Dans cette tâche, chaque ligne est traitée comme une entité, qui est considérée comme le sujet de la ligne. La tâche de correspondance ligne-instance diffère de la tâche CEA car elle peut permettre de découvrir davantage d'entités en s'appuyant sur le contexte de la ligne, en particulier dans le cas où le sujet de la ligne est caché. Par exemple, dans la figure 3, la quatrième ligne est représentée par ("Switzerland - Turkey, 11 Jun 2008" (Q12012827)) qui ne peut pas être identifiée par le CEA. L'ensemble de ces tâches permettent d'établir des correspondances entre les tableaux et le graphe de connaissances. Cette étape constitue une première étape pour la génération de triplets à partir du contenu du tableau.

Enfin, les tâches de STI utilisent les KG comme sources d'information et comme références pour la production d'annotations. La colonne KG de la Table 1 fournit les KG qui ont été utilisés dans chaque système STI examiné dans ce papier. Les KG sont des éléments essentiels pour soutenir le processus de STI. En effet, comprendre le contenu d'un tableau revient à identifier les entités mentionnées dans les cellules du tableau et les relations entre elles. En fonction de leur contenu, les KG peuvent être classés en KG spécifiques à un domaine, en KG encyclopédiques ou en KG de sens commun. Les KG les plus couramment utilisés pour l'interprétation des tableaux sont **DBpedia** [7], **Wikidata** [53], **Freebase** [8] et **YAGO** (Yet Another Great Ontology) [50]. Les KG spécifiques à un domaine sont très peu utilisés à ce jour pour les tâches de STI comme en témoigne la table 1.

4 Approches

Dans cette section, nous passons en revue les approches de STI. Plusieurs jeux de données ont été proposés pour évaluer les approches de STI. Certains d'entre eux sont des références pour l'évaluation dans lesquels les composants des tableaux (cellules, lignes, colonnes ou paires de cellules) sont associés à des éléments de KG (entité, classe ou propriété), tandis que d'autres jeux de données proposent des tableaux de qualité pour permettre l'entrainement de système. Plusieurs jeux de données sont utilisés pour comparer les approches de STI dans cet état de l'art : Limaye [27], T2D [43], WDC [26], TabEL [5], Zhang et al. [59] et SemTab 2019, 2020, 2021 [1]. Parmi les cinq tâches présentées dans la section précédente, la littérature se concentre principalement sur le CTA, le CEA et le CPA. Nous proposons de classer les systèmes STI selon trois familles représentatives de leur méthodologie intrinsèque : les méthodes heuristiques (section 4.1), les méthodes basées sur l'ingénierie de caractéristiques (section 4.2) et les méthodes basées sur l'apprentissage profond (section 4.3). La table 1 présente plus de détails sur cette classification, notamment les algorithmes représentatifs, les tâches ciblées, les éléments de tableau utilisés, les KG cibles et l'année de publication.

4.1 Approches heuristiques

La famille des approches heuristiques regroupe diverses approches de STI utilisant des algorithmes simples et ne nécessitant pas d'efforts significatifs d'ingénierie de caractéristiques ou d'apprentissage. En effet, les tâches de STI sont ici effectuées à l'aide de techniques heuristiques telles que les mesures de similarité de chaînes de caractères [31, 39, 55], le vote majoritaire [61], TF-IDF [39, 51] ou les méthodes probabilistes [35]. Le contexte du tableau, y compris l'en-tête, le titre et les cellules voisines [22, 55] peuvent être pris en compte par ce type d'approches mais pas systématiquement. Nous identifions deux sous classes d'approches heuristiques. Tout d'abord, les approches basées sur les opérations de lookup travaillent avec un ensemble initial d'entités candidates déterminé par un service de recherche. Après avoir généré des candidats, ces méthodes les classent à l'aide de différentes métriques reposant sur les éléments du tableau (par exemple, les cellules,

1. http://www.cs.ox.ac.uk/isg/challenges/sem-tab/

TABLE 1 – Les approches de STI sont classifiées en trois familles. "R2I" correspond à la tâche de correspondance ligne-instance ; "TA" correspond à la thématisation ; "$\mathcal{T}_{i*}$" indique que des informations de la ligne sont utilisées pour annoter une cellule donnée ; "$\mathcal{T}_{*j}$" indique que l'approche tire parti des informations portées par la colonne étudiée (CEA, CTA) ou les colonnes voisines (CPA) ; "$\mathcal{T}_{0*}$" signifie que l'approche utilise les informations contenues dans l'en-tête ; "$\mathcal{T}_{**}$" indique que l'approche entraine un modèle sur l'ensemble des éléments contenus dans la table incluant les influences entre colonnes ; "$\mathcal{T}_{out}$" indique que l'approche utilise, en complément de la table, des métadonnées et le contexte associés à la table.

Approches			Tâches					Elements utilisés					KG	Source	Année de publication
Famille		Algorithme	CEA	CTA	CPA	R2I	TA	$\mathcal{T}_{i*}$	$\mathcal{T}_{*j}$	$\mathcal{T}_{0*}$	$\mathcal{T}_{**}$	$\mathcal{T}_{out}$			
Heuristique	Lookup	Venetis et al. [52]		√	√				√				Ad-hoc	Web Tables	2011
		Wang et al. [55]		√				√	√	√			Probase	Wikipedia Tables	2012
		Deng et al. [15]		√					√				FreeBase, YAGO	Wikipedia Tables	2013
		Sekhavat et al. [44]			√				√				DBpedia	Web Tables	2014
		TabEL [5]	√					√	√				YAGO	Limaye	2015
		ADOG [39]	√	√	√			√	√				DBpedia	SemTab 2019	2019
		Tabularisi [51]	√	√	√				√				DBpedia	T2D, VizNet	2019
		C^2 [25]		√					√	√	√		DBpedia, Wikidata	Limaye, ISWC2017, SemTab 2019, T2D	2020
		Magic [47]	√	√	√				√	√			DBpedia, Wikidata	SemTab 2021	2021
		Alobaid et al. [2]		√						√			DBpedia	SemTab 2021, T2D	2022
	Itérative	Zwicklbauer et al. [61]		√					√				DBpedia	Wikipedia Tables	2013
		T2K [42]			√	√	√	√	√				DBpedia	T2D	2015
		TableMiner+ [59]			√	√	√	√	√	√		√	Freebase	Limaye, IMDB, MusicBrainz	2017
		LOD4ALL [31]	√	√	√			√	√				DBpedia	SemTab 2019	2019
		CSV2KG [48]	√	√	√			√	√	√			DBpedia	SemTab 2019	2019
		MTab [35, 37, 38]	√	√	√			√	√	√			DBpedia, Wikidata	SemTab 2019-2021	2019
		LinkingPark [12]	√	√	√			√	√				Wikidata	SemTab 2020	2019
		DAGOBAH SL [20, 21, 22]	√	√	√			√	√				DBpedia,Wikidata	SemTab 2019-2022	2019
		MantisTable [14, 13]	√	√	√	√		√	√				DBpedia, Wikidata	SemTab 2019-2021	2019
		JenTab [1]	√	√	√			√	√				DBpedia, Wikidata	SemTab 2020-2021	2020
Ingénierie de caractéristiques		Limaye et al. [27]	√	√	√			√	√	√			YAGO	Limaye	2010
		Mulwad et al. [33, 32]	√	√	√			√	√	√			Wikitology	Limaye	2010
		SemanticTyper [40]		√					√				DBpedia	Museum	2015
		DSL [30]		√					√				DBpedia	City, Museum, Weather, Custom Soccer	2016
		Neumaier et al. [34]		√					√				DBpedia	Portail de données gouvernementales	2016
		NUMER [24]		√					√		√		DBpedia	NumDB	2018
Apprentissage profond	Modélisation du KG	Vasilis et al. [17]	√					√	√				Wikidata	Limaye, T2D, Wikipedia	2017
		Biswas et al. [6]					√					√	DBpedia	Wikipedia infobox	2018
		DAGOBAH Embeddings [10]	√	√				√	√	√			DBpedia, Wikidata	SemTab 2019	2019
		Radar Station [29]	√						√				Wikidata	Limaye, T2Dv2, SemTab 2020	2022
	Modélisation de la table	Sherlock [19]		√					√				DBpedia	T2D, VizNet	2019
		Sato [57]		√					√		√		DBpedia	VizNet	2019
		ColNet [11]		√				√	√				DBpedia	Limaye, T2Dv2	2019
		Guo et al. [18]		√					√			√	DBpedia	T2Dv2	2020
		Zhang et al. [58]	√		√			√	√				DBpedia	T2Dv2	2020
		TURL [16]	√	√	√			√	√	√	√	√	DBpedia	WikiGS, WikiTable, T2D	2020
		TCN [54]		√	√			√	√		√	√	-	Web Tables, WikiTable [16]	2021
		DUDUO [49]		√	√			√	√		√		-	WikiTable, VizNet	2021
		Singh et al. [46]			√				√	√		√	DBpedia	T2Dv2	2021
		Zhou et al [60]		√					√		√		DBpedia	Wikipedia Tables	2021

le type de colonnes, etc.). **Venetis et al.** [52] et **TabEL** [5] sont deux approches notables de cette sous-classe. Deuxièmement, les approches itératives sont construites à partir d'un système à base de lookup, avec une étape supplémentaire de désambiguïsation pour reclasser les entités candidates. Les techniques de désambiguïsation itératives jouent un rôle important dans l'amélioration des performances et de nombreuses approches performantes de l'état de l'art appartiennent à cette sous-classe, notamment **T2K** [42], **MTab** [35], **LinkingPark** [12], **JenTab** [1] et **DAGOBAH SL** [20, 21, 22].

4.2 Approches basées sur l'ingénierie de caractéristiques

Cette famille de méthodes extrait des caractéristiques statistiques et lexicales (telles que la distribution des valeurs numériques, l'occurrence des mentions de cellules, la similarité textuelle, etc.) des lignes et des colonnes du tableau et les utilise dans des modèles d'apprentissage automatique. Les algorithmes utilisés par cette famille sont, par exemple, SVM [33], Random Forest [30] et K-Nearest Neighbor [34]. La quantité et la qualité des données d'apprentissage, et par conséquent la qualité des caractéristiques d'entrée, ont un impact significatif sur la performance des modèles, comme indiqué dans [30]. En outre, nous observons que les méthodes à base d'apprentissage ciblent la tâche CTA en particulier, car les colonnes peuvent fournir plus de caractéristiques statistiques que d'autres cibles d'annotation. **Limaye et al.** [27] et **Mulwad et al.** [33] sont deux approches importantes de cette famille.

4.3 Approches basées sur l'apprentissage profond

L'apprentissage profond a connu un grand succès dans plusieurs domaines grâce à la disponibilité d'énormes quantités de données et de puissantes ressources informatiques. Il a attiré de plus en plus l'attention de la communauté du STI au cours des dernières années. Nous identifions deux courants principaux dans cette famille d'approches. Premièrement, la modélisation de KG se concentre sur l'apprentissage de plongement des entités représentant les cellules des tables (et non les cellules elles-mêmes). Plus précisément, les techniques de plongements de KG (par exemple, TransE [9] et TransH [56]) sont utilisées pour plonger les entités et leurs relations dans un espace vectoriel. Les modèles de STI reposent sur l'intuition que les entités d'une même colonne doivent présenter des similitudes sémantiques. Elles doivent donc être proches les unes

des autres dans l'espace de plongement au regard de la distance de similarité cosinus [17] ou de la distance euclidienne [10]. **DAGOBAH Embeddings** [10] et le module **Radar Station** [29] sont deux approches utilisant la modélisation de KG. Deuxièmement, la modélisation des tableaux considère directement le contenu textuel du tableau ainsi que les interactions intra-table et inter-table. La représentation des éléments de base du tableau comme les cellules ou les colonnes est apprise à l'aide de réseaux de neurones profonds [11, 19] ou de modèles de langage comme BERT [16, 49, 60].

5 Evaluation

Dans cette section, nous analysons plus en détail les performances, les forces et les faiblesses de chaque famille d'approches de STI. La Table 2 résume les performances des trois meilleurs systèmes avec les scores F1, AP, ou AF1 les plus élevés pour les tâches CEA, CTA et CPA sur les ensembles de données couramment utilisés par la communauté. Les scores AP et AF1 sont utilisés pour la tâche de CTA afin de prendre en compte la multiplicité des annotations possibles, avec des types plus ou moins génériques/plus ou moins porteurs d'informations. Les ensembles de données, la méthode de collecte des résultats et les métriques d'évaluation sont présentés plus en détail dans [28].

Sur la base de notre classification des approches de STI, nous observons que les systèmes heuristiques apparaissent dans les trois premiers systèmes pour tous les ensembles de données et toutes les tâches. En particulier, aucun des systèmes à base d'ingénierie de caractéristiques ou des systèmes basés sur l'apprentissage profond n'a atteint le podium pour les tâches d'appariement d'entités (CEA et correspondance ligne-instance). L'une des principales raisons est que, contrairement aux tâches de CTA ou CPA, qui valorisent des caractéristiques sur les colonnes ou des paires de colonnes, les caractéristiques qui peuvent être utilisées pour annoter des cellules ou des lignes sont plus rares. Cela limite donc les performances de ces systèmes.

Méthodes d'appariement VS méthodes d'apprentissage. Nous avons observé que les approches de STI reposent soit sur l'appariement (association d'une entité du KG et d'une cellule de la table) soit sur de l'apprentissage. L'appariement est la base des approches heuristiques, tandis que l'ingénierie de caractéristiques et les méthodes basées sur l'apprentissage profond reposent sur l'apprentissage de la représentation du tableau d'entrée. Ces approches peuvent également être combinées : les annotations apprises par les réseaux de neurones sont affinées à l'aide de techniques d'appariement utilisées lors d'un post-traitement (DAGOBAH Embeddings [10] et ColNet [11] en sont deux exemples). D'après nos observations, l'efficacité des opérations d'appariement dépendent fortement de la compatibilité entre la table et le KG cible. Par conséquent, ce type de technique souffre de l'incomplétude du tableau et des problématiques de knowledge shifting du KG. Les méthodes d'appariement sont moins résistantes au bruit que les méthodes d'apprentissage. De leur côté, les méthodes d'apprentissage nécessitent de grands ensembles de données d'entraînement qui ne sont pas toujours simples à collecter ou à générer. Certaines approches d'apprentissage limitent toutefois le nombre de candidats cibles pour pallier au manque de données d'apprentissage. La taille des tableaux constitue un autre défi pour les méthodes d'apprentissage. Certaines méthodes s'appuient sur les caractéristiques statistiques calculées à partir du tableau (par exemple, la longueur des mentions). Ces caractéristiques ne sont pas statistiquement stables si le nombre de cellules du tableau est faible.

L'essor de l'apprentissage profond. A partir de 2017, l'apprentissage profond a fait son entrée dans le domaine du STI. Par rapport aux approches d'ingénierie de caractéristiques, les réseaux de neurones profonds permettent au système de traiter les caractéristiques des tables plus efficacement, car l'étape d'ingénierie de caractéristiques est parfois difficile et longue à maintenir. Par exemple, Sherlock [19] est basé sur 1588 caractéristiques issues de colonnes. Pour remédier à ce problème, un apprentissage de bout en bout est préférable et de plus en plus utilisé, par exemple, la modélisation de KG à l'aide de plongements de graphes [17] et la modélisation de tables avec des modèles de type BERT[54]. Cependant, nous observons que les approches de modélisation de tables utilisant des modèles de langage ciblent toujours des tâches d'annotation de classes (CTA) ou de relations (CPA). La tâche d'annotation d'entités (CEA) n'a pas encore fait l'objet de travaux spécifiques de ce type, excepté TURL [16] proposant une matrice de visibilité pour décrire les connexions entre les éléments du tableau (par exemple, les cellules dans les mêmes colonnes, les cellules dans les mêmes lignes, etc.). En outre, de nombreux systèmes [49, 54, 60] simplifient la représentation des tableaux en ignorant l'ordre des lignes et des colonnes notamment.

Compromis entre l'efficacité et la précision. Les systèmes d'annotation sont généralement confrontés à un compromis entre efficacité et précision. TableMiner+ [59] introduit un appariement partiel dans lequel le calcul du CTA repose sur seulement huit lignes du tableau afin d'améliorer les performances. Cette stratégie rend en effet les systèmes plus rapides mais dégrade la précision. Par exemple, si l'on considère l'annotation d'une colonne contenant ["Joe Biden", "Donald Trump", "Barack Obama", "Abe Shinzo"], l'application de la correspondance partielle sur les trois premières cellules de la colonne produira "Présidents américains" comme type de cette colonne, alors que la réponse correcte est plus probablement "politiciens" puisque "Abe Shinzo" n'est pas un président américain mais un premier ministre japonais. Enfin, les systèmes dont le pipeline d'annotation comprend une étape de génération de candidats dépendront fortement du service de lookup d'entités utilisé. Cependant, les points d'accès publics imposent plusieurs limites à leur utilisation et l'obtention d'un ensemble de candidats avec une couverture souhaitable de la table cible peut prendre davantage de temps.

TABLE 2 – Top 3 des approches pour chaque jeu de données au regard du F1-score.

Jeux de données		CEA / Correspondance ligne-instance			CTA [a] / Thématisation			CPA		
Limaye		TabEL	TabEAno [36]	T2K ++	T2K ++	Guo et al	MantisTable	Mulwad et al.	T2K ++	TableMiner+
		0.894	0.88	0.87	0.88	0.852	0.84	0.89	0.80	0.76
T2D		TabEAno	Zhang et al.	Kruit et al.	ColNet	Alobaid et al. [2]	MantisTable	T2K ++	Singh et al.	MantisTable
		0.91	0.90	0.89	0.976	0.96	0.95	0.91	0.71	0.51
SemTab 2019	R2	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi	CSV2KG	IDLab	Tabularisi
		0.911	0.883	0.826	1.414	1.376	1.099	0.881	0.877	0.790
	R3	MTab	CSV2KG	ADOG	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi
		0.970	0.962	0.912	1.956	1.864	1.702	0.844	0.841	0.827
	R4	MTab	MantisTable	CSV2KG	MTab	CSV2KG	Tabularisi	MTab	CSV2KG	Tabularisi
		0.983	0.973	0.907	2.012	1.846	1.716	0.832	0.830	0.823
SemTab 2020	R1	MTab	LinkingPark	MantisTable	JenTab	LinkingPark	MTab	MTab	LinkingPark	JenTab
		0.987	0.987	0.982	0.962	0.926	0.885	0.971	0.967	0.963
	R2	MTab	DAGOBAH	LinkingPark	LinkingPark	MTab	DAGOBAH	MTab	LinkingPark	DAGOBAH
		0.995	0.993	0.993	0.984	0.984	0.983	0.997	0.993	0.992
	R3	MTab	LinkingPark	DAGOBAH	LinkingPark	MTab	DAGOBAH	MTab	DAGOBAH	bbw [45]
		0.991	0.986	0.985	0.978	0.976	0.974	0.995	0.993	0.989
	R4	MTab	LinkingPark	DAGOBAH	MTab	bbw	DAGOBAH	MTab	bbw	DAGOBAH
		0.993	0.985	0.984	0.981	0.98	0.972	0.997	0.995	0.995
	2T	MTab	bbw	DAGOBAH	DAGOBAH	MTab	LinkingPark	-	-	-
		0.907	0.863	0.830	0.743	0.728	0.686	-	-	-
SemTab 2021	R1 (DBpedia)	DAGOBAH	GBMTab	JenTab	JenTab	DAGOBAH	Magic	-	-	-
		0.945	0.692	0.607	0.46	0.422	0.159	-	-	-
	R1 (WikiData)	DAGOBAH	MTab	AMALGAM [3]	DAGOBAH	MTab	JenTab	-	-	-
		0.923	0.907	0.658	0.832	0.728	0.697	-	-	-
	R2-Hard	MTab	DAGOBAH	MantisTable	MTab	DAGOBAH	MantisTable	MTab	JenTab	DAGOBAH
		0.985	0.975	0.968	0.977	0.976	0.955	0.997	0.996	0.996
	R2-Bio	DAGOBAH	MTab	MantisTable	MTab	Magic	DAGOBAH	MTab	DAGOBAH	JenTab
		0.970	0.964	0.93	0.956	0.916	0.916	0.947	0.899	0.899
	R3-Biodiv	JenTab	MTab	DAGOBAH	KEPLER-aSI [4]	DAGOBAH	MTab	-	-	-
		0.602	0.522	0.496	0.593	0.391	0.123	-	-	-
	R3-Hard	DAGOBAH	MTab	MantisTable	DAGOBAH	MTab	MantisTable	MTab	JenTab	DAGOBAH
		0.974	0.968	0.959	0.99	0.984	0.965	0.993	0.992	0.991
	R3-Git (DBp)	-	-	-	DAGOBAH	KEPLER-aSI	MantisTable	-	-	-
		-	-	-	0.07	0.041	0.037	-	-	-
	R3-Git (Sch)	-	-	-	MantisTable	DAGOBAH	-	-	-	-
		-	-	-	0.205	0.183	-	-	-	-

a. Le score AH est pris en compte pour SemTab 2019 tandis que le score AF1 est utilisé pour SemTab 2020 et 2021

KGs publiques VS KGs ad-hocs. De nombreuses approches s'appuient sur des KG encyclopédiques tels que Wikidata et DBpedia, qui fournissent des informations riches permettant de produire des annotations de qualité. Toutefois, une plus grande quantité d'informations entraîne également une plus grande ambiguïté, et les bases de connaissances sont généralement incomplètes. Les évolutions des KGs dans le temps sont également un défi pour les approches. Nous observons que certaines approches [16, 19, 30, 49, 54, 40] traitent uniquement le KG cible comme un dictionnaire de concepts. Toutefois, savoir comment injecter correctement la structure du KG cible dans un modèle statistique reste une question ouverte.

6 Conclusion et directions de recherche

Ces dernières années ont été marquées par une croissance significative du domaine de l'interprétation de données tabulaires, notamment sous l'impulsion d'initiatives telles que SemTab. Dans cette étude, nous avons fourni un ensemble de définitions autour des données tabulaires et des tâches d'intepretation pour structurer et unifier le domaine ainsi qu'une vue actualisée sur les approches de l'état de l'art. Ces dernières sont classées en trois familles, les approches heuristiques, l'ingénierie de caractéristiques et l'apprentissage profond. Nous avons également mis en évidence les systèmes de STI les plus performants pour chaque ensemble de données et avons identifié plusieurs défis à relever pour améliorer les systèmes STI. Bien que les travaux récents aient permis de réaliser des progrès significatifs dans le domaine du STI, les approches existantes présentent plusieurs limites que nous décrivons ensuite.

Tout d'abord, la plupart des approches se concentrent sur des tables à sujet monocellulaire dans des tables relationnelles ou d'entités et font de fortes suppositions quant à la mise en forme utilisée pour la présentation des tables. En outre, les approches actuelles tiennent peu compte des spécificités de certaines tables relationnelles telles que les sujets cachés ou les sujets composés. Pour combler cette lacune et stimuler la recherche de nouvelles solutions, nous pensons qu'il est important d'élargir le spectre des complexités trouvé dans les corpus. À cette fin, nous recommandons de créer de nouveaux ensembles de données avec des structures de tableaux multiples et des contenus complexes afin d'aborder toute la diversité des données du monde réel. Nous estimons que la complexité du contenu ne devrait pas se limiter au bruit ajouté aux mentions, que ce soit de manière synthétique ou manuelle, car ce type de complexité peut être géré sans difficulté par la plupart des approches comme le démontre les résultats des derniers challenges SemTab.

Deuxièmement, les approches existantes supposent que le KG cible est complet et exempt d'erreurs. Par conséquent, une annotation (instance, type ou relation) peut toujours être générée même si le résultat correct ne se trouve pas dans le KG. Cette situation peut être préjudiciable, no-

tamment parce qu'elle peut propager l'erreur d'une annotation à l'ensemble de la colonne, voire à l'ensemble du tableau. Supposons par exemple un tableau avec une colonne contenant les noms de famille d'écrivains et une autre colonne contenant les titres de livres (pour les besoins de l'exemple, nous supposons que la majorité de ces livres ont été adaptés au cinéma). Si le KG cible couvre largement les films mais seulement quelques œuvres littéraires (ou est moins précis pour cette deuxième catégorie), le processus d'annotation pourrait typer la deuxième colonne comme "film", ce qui pourrait conduire à mal désambiguïser les mentions dans la première colonne (si certains acteurs apparentés ont des noms de famille similaires par exemple). En conséquence, ce tableau sera interprété comme un élément "acteurs-films" au lieu de la cible correcte "écrivains-livres". Certains mécanismes existants, tels que l'attribution d'un score de confiance à chaque candidat, peuvent aider à filtrer davantage les annotations incorrectes mais restent insuffisant. Enfin, nous soulignons que les approches futures devraient également envisager de s'attaquer à des domaines dans lesquels il n'existe que des KG naissants, l'objectif étant d'utiliser la STI pour augmenter ces KG dans une boucle vertueuse.

Troisièmement, nous observons que de nombreuses approches n'exploitent que partiellement les éléments du tableau (table 1), bien que les approches les plus récentes tendent à inverser cette tendance. Nous pensons que l'exploitation du plus grand nombre d'éléments possible devrait améliorer la précision en ajoutant davantage d'informations contextuelles. Ainsi, les modèles de langage pourraient être utilisés. En effet, on peut considérer un tableau comme un moyen de structurer le langage : dans le cas le plus simple, une ligne du tableau peut être considérée comme une phrase décrivant un sujet avec quelques attributs. Il en va de même pour le sous-graphe correspondant dans le KG cible. La représentation des phrases pourrait donc être utilisée pour calculer les similitudes. Néanmoins, la spécificité des données tabulaires et des KG doit être prise en compte, ce qui implique d'adapter les mécanismes d'attention à cette structure. Nous remarquons également que la plupart des approches traitent les tableaux de manière indépendante. Cependant, certaines tables sont liées les unes aux autres puisqu'elles peuvent être générées avec le même template, faire partie d'un corpus cohérent de tables ou être liées par des identifiants communs (e.g. bases de données SQL). Nous pensons que les systèmes de STI pourraient tirer un avantage significatif de la combinaison d'éléments de tableaux avec des connexions entre tableaux, qui peuvent être considérées comme une autre couche de contexte ajoutée pour capturer des informations plus riches sur les données à traiter.

Références

[1] Nora Abdelmageed and Sirko Schindler. JenTab Meets SemTab 2021's New Challenges. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[2] Ahmad Alobaid and Oscar Corcho. Balancing coverage and specificity for semantic labelling of subject columns. *Knowledge-Based Systems*, page 108092, 2022.

[3] Rabia Azzi and Gayo Diallo. AMALGAM : making tabular dataset explicit with knowledge graph. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 9–16, 2020.

[4] Wiem Baazouzi, Marouen Kachroudi, and Sami Faiz. KEPLER-asi at SemTab 2021. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[5] Chandra Sekhar Bhagavatula, Thanapon Noraset, and Doug Downey. Tabel : Entity linking in web tables. In *14^{th} International Semantic Web Conference*, pages 425–441. Springer, 2015.

[6] Russa Biswas, Rima Türker, Farshad Bakhshandegan Moghaddam, Maria Koutraki, and Harald Sack. Wikipedia Infobox Type Prediction Using Embeddings. In *DL4KGS@ ESWC*, pages 46–55, 2018.

[7] Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. DBpedia-A crystallization point for the Web of Data. *Journal of web semantics*, 7(3) :154–165, 2009.

[8] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase : a collaboratively created graph database for structuring human knowledge. In *ACM SIGMOD International Conference on Management of Data*, 2008.

[9] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013.

[10] Yoan Chabot, Thomas Labbe, Jixiong Liu, and Raphaël Troncy. DAGOBAH : an end-to-end context-free tabular data semantic annotation system. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching*, pages 41–48, 2019.

[11] Jiaoyan Chen, Ernesto Jimenez-Ruiz, Ian Horrocks, and Charles Sutton. ColNet : Embedding the Semantics of Web Tables for Column Type Prediction. In *33^{rd} AAAI International Conference on Artificial Intelligence*, 2018.

[12] Shuang Chen, Alperen Karaoglu, Carina Negreanu, Tingting Ma, Jin-Ge Yao, Jack Williams, Andy Gordon, and Chin-Yew Lin. Linkingpark : An integrated approach for semantic table interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[13] Marco Cremaschi, Roberto Avogadro, Andrea Barazzetti, and David Chieregato. MantisTable SE : an Efficient Approach for the Semantic Table Interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[14] Marco Cremaschi, Roberto Avogadro, and David Chieregato. MantisTable : an Automatic Approach for the Semantic Table Interpretation. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 15–24, 2019.

[15] Dong Deng, Yu Jiang, Guoliang Li, Jian Li, and Cong Yu. Scalable column concept determination for web tables using large knowledge bases. In *PVLDB*, pages 1606–1617. VLDB Endowment, 2013.

[16] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. TURL : Table Understanding through Representation Learning. arXiv :2006.14806, 2020.

[17] Vasilis Efthymiou, Oktie Hassanzadeh, Mariano Rodriguez-Muro, and Vassilis Christophides. Matching web tables with knowledge base entities : from entity lookups to entity embeddings. In *16^{th} International Semantic Web Conference (ISWC)*, pages 260–277. Springer, 2017.

[18] Tong Guo, Derong Shen, Tiezheng Nie, and Yue Kou. Web table column type detection using deep learning and probability graph model. In *International Conference on Web Information Systems and Applications*, pages 401–414. Springer, 2020.

[19] Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. Sherlock : A deep learning approach to semantic data type detection. In *25^{th} ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, pages 1500–1508, 2019.

[20] Viet-Phi Huynh, Yoan Chabot, Thomas Labbé, Jixiong Liu, and Raphaël Troncy. From Heuristics to Language Models : A Journey Through the Universe of Semantic Table Interpretation with DAGOBAH. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2022.

[21] Viet-Phi Huynh, Jixiong Liu, Yoan Chabot, Frédéric Deuzé, Thomas Labbé, Pierre Monnin, and Raphaël Troncy. DAGOBAH : Table and Graph Contexts for Efficient Semantic Annotation of Tabular Data. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[22] Viet-Phi Huynh, Jixiong Liu, Yoan Chabot, Thomas Labbé, Pierre Monnin, and Raphaël Troncy. DAGOBAH : Enhanced Scoring Algorithms for Scalable Annotations of Tabular Data. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[23] Ernesto Jiménez-Ruiz, Oktie Hassanzadeh, Vasilis Efthymiou, Jiaoyan Chen, and Kavitha Srinivas. SemTab 2019 : Resources to Benchmark Tabular Data to Knowledge Graph Matching Systems. In *European Semantic Web Conference (ESWC)*, pages 514–530. Springer, 2020.

[24] Emilia Kacprzak, José M Giménez-García, Alessandro Piscopo, Laura Koesten, Luis-Daniel Ibáñez, Jeni Tennison, and Elena Simperl. Making sense of numerical data-semantic labelling of web tables. In *European Knowledge Acquisition Workshop*, pages 163–178. Springer, 2018.

[25] Udayan Khurana and Sainyam Galhotra. Semantic annotation for tabular data, 2019.

[26] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. A large public corpus of web tables containing time and context metadata. In *25^{th} International Conference Companion on World Wide Web*, pages 75–76, 2016.

[27] Girija Limaye, Sunita Sarawagi, and Soumen Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proceedings of the VLDB Endowment*, 3(1-2) :1338–1347, 2010.

[28] Jixiong Liu, Yoan Chabot, Raphaël Troncy, Viet-Phi Huynh, Thomas Labbé, and Pierre Monnin. From tabular data to knowledge graphs : A survey of semantic table interpretation tasks and methods. *Journal of Web Semantics*, page 100761, 2022.

[29] Jixiong Liu, Viet-Phi Huynh, Yoan Chabot, and Raphaël Troncy. Radar Station : Using KG Embeddings for Semantic Table Interpretation and Entity Disambiguation. In *21^{st} International Semantic Web Conference (ISWC)*, 2022.

[30] Pham Minh, Alse Suresh, A. Knoblock Craig, and Szekelyèle Pedro. Semantic Labeling : A Domain-Independent Approach. In *15^{th} International Semantic Web Conference (ISWC)*, pages 446—462, 2016.

[31] Hiroaki Morikawa. Semantic Table Interpretation using LOD4ALL. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, pages 49–56, 2019.

[32] Varish Mulwad, Tim Finin, and Anupam Joshi. Semantic message passing for generating linked data from tables. In *12^{th} International Semantic Web Conference (ISWC)*, pages 363–378. Springer, 2013.

[33] Varish Mulwad, Tim Finin, Zareen Syed, Anupam Joshi, et al. Using linked data to interpret tables. In *1^{st} International Workshop on Consuming Linked Data (COLD)*, 2010.

[34] Sebastian Neumaier, Jürgen Umbrich, Josiane Xavier Parreira, and Axel Polleres. Multi-level semantic labelling of numerical values. In *15^{th} International Semantic Web Conference (ISWC)*, pages 428–445, 2016.

[35] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. MTab : Matching Tabular Data to Knowledge Graph using Probability Models. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[36] Phuc Nguyen, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. TabEAno : table to knowledge graph entity annotation. arXiv :2010.01829, 2020.

[37] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. Mtab4wikidata at semtab 2020 : Tabular data annotation with wikidata. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2020.

[38] Phuc Nguyen, Ikuya Yamada, Natthawut Kertkeidkachorn, Ryutaro Ichise, and Hideaki Takeda. SemTab 2021 : Tabular Data Annotation with MTab Tool. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[39] Daniela Oliveira and Mathieu d'Aquin. Adog-annotating data with ontologies and graphs. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[40] S Krishnamurthy Ramnandan, Amol Mittal, Craig A Knoblock, and Pedro Szekely. Assigning semantic labels to data sources. In *European Semantic Web Conference (ESWC)*, pages 403–417. Springer, 2015.

[41] Dominique Ritze and C. Bizer. Matching Web Tables To DBpedia - A Feature Utility Study. In *International Conference on Extending Database Technology (EDBT)*, pages 210—-221, 2017.

[42] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching html tables to dbpedia. In 5^{th} *International Conference on Web Intelligence, Mining and Semantics*, pages 1–6, 2015.

[43] Dominique Ritze, Oliver Lehmberg, and Christian Bizer. Matching HTML Tables to DBpedia. In 5^{th} *International Conference on Web Intelligence, Mining and Semantics (WIMS)*, pages 1–6, 2015.

[44] Yoones A Sekhavat, Francesco Di Paolo, Denilson Barbosa, and Paolo Merialdo. Knowledge base augmentation using tabular data. In *LDOW*, 2014.

[45] Renat Shigapov, Philipp Zumstein, Jan Kamlah, Lars Oberländer, Jörg Mechnich, and Irene Schumm. bbw : Matching CSV to Wikidata via Meta-lookup. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, volume 2775, pages 17–26. RWTH, 2020.

[46] Gaurav Singh, Siffi Singh, Joshua Wong, and Amir Saffari. Relation Extraction from Tables using Artificially Generated Metadata. arXiv :2108.10750, 2021.

[47] Bram Steenwinckel, Filip De Turck, and Femke Ongeane. MAGIC : Mining an Augmented Graph using INK, starting from a CSV. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2021.

[48] Bram Steenwinckel, Gilles Vandewiele, Filip De Turck, and Femke Ongenae. Csv2kg : Transforming tabular data into semantic knowledge. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, 2019.

[49] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. Annotating Columns with Pre-trained Language Models. arXiv :2104.01785, 2021.

[50] Thomas Pellissier Tanon, Gerhard Weikum, and Fabian Suchanek. YAGO 4 : A Reason-able Knowledge Base. In *European Semantic Web Conference (ESWC)*, pages 583–596. Springer, 2020.

[51] Avijit Thawani, Minda Hu, Erdong Hu, Husain Zafar, Naren Teja Divvala, Amandeep Singh, Ehsan Qasemi, Pedro A Szekely, and Jay Pujara. Entity Linking to Knowledge Graphs to Infer Column Types and Properties. In *Semantic Web Challenge on Tabular Data to Knowledge Graph Matching (SemTab)*, volume 2019, pages 25–32, 2019.

[52] Petros Venetis, Alon Y Halevy, Jayant Madhavan, Marius Pasca, Warren Shen, Fei Wu, and Gengxin Miao. Recovering semantics of tables on the web. *PVLDB*, 4(9) :528–538, 2011.

[53] Denny Vrandečić and Markus Krötzsch. Wikidata : a free collaborative knowledge base. *Communications of the ACM*, 57(10) :78–85, 2014.

[54] Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. Tcn : Table convolutional network for web table interpretation. arXiv :2102.09460, 2021.

[55] Jingjing Wang, Haixun Wang, Zhongyuan Wang, and Kenny Q Zhu. Understanding tables on the web. In *International Conference on Conceptual Modeling*, pages 141–155. Springer, 2012.

[56] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *AAAI Conference on Artificial Intelligence*, 2014.

[57] Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. Sato : Contextual Semantic Type Detection in Tables, 2019.

[58] Shuo Zhang, Edgar Meij, Krisztian Balog, and Ridho Reinanda. Novel entity discovery from web tables. In *The Web Conference*, pages 1298–1308, 2020.

[59] Ziqi Zhang. Effective and efficient semantic table interpretation using tableminer+. *Semantic Web*, 8(6) :921–957, 2017.

[60] Yiwei Zhou, Siffi Singh, and Christos Christodoulopoulos. Tabular Data Concept Type Detection Using Star-Transformers. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3677–3681, 2021.

[61] Stefan Zwicklbauer, Christoph Einsiedler, Michael Granitzer, and Christin Seifert. Towards Disambiguating Web Tables. In *International Semantic Web Conference (Posters & Demos)*, pages 205–208, 2013.