Rapprochement de données pour la reconnaissance d'entités dans les documents océrisés

Nihel Kooli

Rapprochement de données pour la reconnaissance d'entités dans les documents océrisés

Nihel Kooli

UNIVERSITÉ DE LORRAINE

École doctorale IAEM Lorraine

Rapprochement de données pour la reconnaissance d'entités dans les documents océrisés

THÈSE

présentée et soutenue publiquement le 13 septembre 2016

pour l'obtention du

Doctorat de l'Université de Lorraine

(mention informatique)

par

Nihel KOOLI

Composition du jury

Président :	Claire GARDENT	Directrice de recherche, CNRS
Rapporteurs :	Jean Yves RAMEL Véronique EGLIN	Professeur, Université François-Rabelais, Maître de conférences - HDR, INSA de Lyon
Examinateur :	Claire GARDENT	Directrice de recherche, CNRS
Invité :	Vincent POULAIN D'ANDECY	Directeur de projets de recherche, ITESOFT-Yooz
Directeur de thèse :	Abdel BELAÏD	Professeur, Université de Lorraine

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thesul.

Remerciements

Je remercie mon directeur de thèse Abdel Belaïd de m'avoir intégré l'équipe READ. Je le remercie également de sa disponibilité et ses efforts.

Je tiens aussi à remercier nos collaborateurs industriels Vincent Poulain d'Andecy et Aurélie Joseph de leur suivi et leur accueil chaleureux pendant mes visites à Itesoft-YOOZ.

Je remercie également Yolande Belaïd de ses conseils pertinents et sa relecture.

J'exprime mes remerciements à l'ensemble des membres de mon jury : Claire Gardent, Jean Yves Ramel et Véronique Eglin. Je suis très honorée de l'intérêt qu'ils ont porté à mes travaux.

Un grand merci à mes amis rencontrés à Nancy, spécialement Abdessalem, Kaled, Mariem, Mondher et Sabra. Je les remercie de leur encouragement et les bons moments passés en leur compagnie.

Je remercie sincèrement tous mes amis et spécialement mes chères copines Hanene, Ines, Nessrine, Safa et Zeina. Je les remercie de leur soutien.

Un remerciement vif à mon très cher Nabil de m'avoir beaucoup aidé, motivé et supporté. Je le remercie pour le partage de tous les moments de la thèse.

Je suis enfin très reconnaissante à ma famille sans laquelle je n'en serais pas là aujourd'hui. Je remercie spécialement mes très chers parents de leur amour, leur tendresse et leur confiance. Je remercie ma très chère sœur Maha, mon très cher frère Achraf et mon beau-frère Firas. Je vous aime.

Je dédie cette thèse à
ma très chère mère
mon très cher père

Sommaire

Partie II Reconnaissance d'entités

Résumé

Cette thèse traite de la reconnaissance d'entités dans les documents océrisés guidée par une base de données. Une entité peut être, par exemple, une entreprise décrite par son nom, son adresse, son numéro de téléphone, son numéro TVA, etc. ou des méta-données d'un article scientifique tels que son titre, ses auteurs et leurs affiliations, le nom de son journal, etc. Disposant d'un ensemble d'entités structurées sous forme d'enregistrements dans une base de données et d'un document contenant une ou plusieurs de ces entités, nous cherchons à identifier les entités contenues dans le document en utilisant la base de données. Ce travail est motivé par une application industrielle qui vise l'automatisation du traitement des images de documents administratifs arrivant en flux continu.

Nous avons abordé ce problème comme un problème de rapprochement entre le contenu du document et celui de la base de données. Les difficultés de cette tâche sont dues à la variabilité de la représentation d'attributs d'entités dans la base et le document et à la présence d'attributs similaires dans des entités différentes. À cela s'ajoutent les redondances d'enregistrements et les erreurs de saisie dans la base de données et l'altération de la structure et du contenu du document, causée par l'OCR.

Devant ces problèmes, nous avons opté pour une démarche en deux étapes : la résolution d'entités et la reconnaissance d'entités. La première étape consiste à coupler les enregistrements se référant à une même entité et à les synthétiser dans un modèle entité. Pour ce faire, nous avons proposé une approche supervisée basée sur la combinaison de plusieurs mesures de similarité entre attributs. Ces mesures permettent de tolérer quelques erreurs sur les caractères et de tenir compte des permutations entre termes.

La deuxième étape vise à rapprocher les entités mentionnées dans un document avec le modèle entité obtenu. Nous avons procédé par deux manières différentes, l'une utilise le rapprochement par le contenu et l'autre intègre le rapprochement par la structure. Pour le rapprochement par le contenu, nous avons proposé deux méthodes : M-EROCS et ERBL. M-EROCS, une amélioration/adaptation d'une méthode de l'état de l'art, consiste à faire correspondre les blocs de l'OCR avec le modèle entité en se basant sur un score qui tolère les erreurs d'OCR et les variabilités d'attributs. ERBL consiste à étiqueter le document par les attributs d'entités et à regrouper ces labels en entités. Pour le rapprochement par les structures, il s'agit d'exploiter les relations structurelles entre les labels d'une entité pour corriger les erreurs d'étiquetage. La méthode proposée, nommée G-ELSE, consiste à utiliser le rapprochement inexact de graphes attribués modélisant des structures locales, avec un modèle structurel appris pour cet objectif.

Cette thèse étant effectuée en collaboration avec la société ITESOFT-Yooz, nous avons expérimenté toutes les étapes proposées sur deux corpus administratifs et un troisième corpus extrait du Web.

Mots-clés: Reconnaissance d'entités, Document océrisé, Base de données, Rapprochement d'entités, Résolution d'entités, Mesures de similarité, Rapprochement de graphes, Structure locale.

Abstract

This thesis focuses on entity recognition in documents recognized by OCR, driven by a database. An entity is a homogeneous group of attributes such as an enterprise in a business form described by the name, the address, the contact numbers, etc. or meta-data of a scientific paper representing the title, the authors and their affiliation, etc. Given a database which describes entities by its records and a document which contains one or more entities from this database, we are looking to identify entities in the document

using the database. This work is motivated by an industrial application which aims to automate the image document processing, arriving in a continuous stream.

We addressed this problem as a matching issue between the document and the database contents. The difficulties of this task are due to the variability of the entity attributes representation in the database and in the document and to the presence of similar attributes in different entities. Added to this are the record redundancy and typing errors in the database, and the alteration of the structure and the content of the document, caused by OCR.

To deal with these problems, we opted for a two-step approach : entity resolution and entity recognition. The first step is to link the records referring to the same entity and to synthesize them in an entity model. For this purpose, we proposed a supervised approach based on a combination of several similarity measures between attributes. These measures tolerate character mistakes and take into account the word permutation.

The second step aims to match the entities mentioned in documents with the resulting entity model. We proceeded by two different ways, one uses the content matching and the other integrates the structure matching. For the content matching, we proposed two methods : M-EROCS and ERBL. M-EROCS, an improvement / adaptation of a state of the art method, is to match OCR blocks with the entity model based on a score that tolerates the OCR errors and the attribute variability. ERBL is to label the document with the entity attributes and to group these labels into entities. The structure matching is to exploit the structural relationships between the entity labels to correct the mislabeling. The proposed method, called G-ELSE, is based on local structure graph matching with a structural model which is learned for this purpose.

This thesis being carried out in collaboration with the ITESOFT-Yooz society, we have experimented all the proposed steps on two administrative corpuses and a third one extracted from the web.

Keywords: Entity recognition, Ocred document, Database, Entity matching, Entity resolution, Similarity measure, Graph matching, Local structure.

Notations et définitions

e **:** une entité définie comme un objet, une réalité ou une chose concrète ou abstraite d'un domaine fonctionnel, identifiable par ses attributs et potentiellement en relation avec les autres entités du domaine.

BD **:** une base de données définie comme une structure de données qui organise des entités, composée de tables et des associations entre elles.

E **:** Le modèle entité composé d'un ensemble d'entités e.

T **:** une table dans une base de données composée d'un ensemble fini de champs (ou colonnes) et d'un ensemble fini d'enregistrements.

r **:** un enregistrement composé d'une suite d'attributs.

c **:** un champ qui renseigne sur la sémantique des attributs dans la table.

$e.c$ **:** l'attribut (une caractéristique) associé à l'entité e correspondant au champ c.

d **:** un segment d'un document représentant une séquence d'une ou plusieurs phrases consécutives dans un document, composée chacune par une suite de mots.

OCR **:** la reconnaissance optique de caractères (en anglais Optical Character Recognition) désigne les procédés informatiques pour la transformation d'images de textes imprimés en fichiers de texte.

D **:** un document océrisé défini comme un document image traduit en fichier texte par l'OCR.

b **:** un bloc représentant un ensemble de lignes qui forment un paragraphe dans un document océrisé.

$d \in D$ **:** un segment qui correspond à un bloc ou un ensemble de blocs contigus dans un document océrisé.

t **:** un terme (mot) dans une chaîne de caractères.

l **:** un label qui représente une expression étiquetée dans le document image définie par sa valeur textuelle v, son champ c et ses coordonnées physiques.

$conf$ **:** la confiance d'étiquetage d'un label l par un champ c.

$P(l \in e/e)$ **:** la probabilité d'appartenance d'un label l à une entité connue e.

$S(e)$ **:** une structure locale définie par un ensemble de labels d'une entité e proches physiquement dans une page.

$P_s(c)$ **:** la probabilité d'appartenance d'un label dont le champ est c à une structure locale quelconque.

$G = (N, A, \mu, \xi)$ **:** un graphe attribué défini par un ensemble de nœuds N, un ensemble d'arcs A et deux fonctions d'étiquetage de nœuds et d'arcs μ et ξ.

$n \in N$ **:** un nœud dans N défini par un vecteur de caractéristiques.

$a \in A$ **:** un arc dans A défini par un vecteur de caractéristiques.

f **:** une caractéristique associée à un nœud ou un arc.

$d_f(f_1, f_2)$ **:** la distance de deux caractéristiques f_1 et f_2 de type f.

nt **:** le nombre de termes d'un label.

nl **:** le nombre de lignes d'un label.

p **:** la police normalisée d'un label par rapport à la police moyenne du document.

vs **:** la séparation verticale en nombre de lignes entre deux labels.

hs **:** la séparation horizontale en nombre de caractères entre deux labels.

al **:** l'alignement (gauche, centré, droite) entre deux labels.

$\delta(G, G')$ **:** la fonction de mise en correspondance entre deux graphes G et G'.

$C(G, G', \delta)$ **:** le coût de la mise en correspondance δ.

$C(G, G')$ **:** le coût de rapprochement entre deux graphes G et G'.

$C_N(n, n')$ **:** le coût de rapprochement entre deux nœuds n et n'.

$C_A(a, a')$ **:** le coût de rapprochement entre deux arcs a et a'.

Chapitre 1

Introduction générale

Sommaire

1.1 Contexte et objectifs

Cette thèse s'inscrit dans le domaine du rapprochement de données venant de sources hétérogènes. Étant donné une base de données qui décrit des entités par des enregistrements et un document qui mentionne une ou plusieurs de ces entités, nous nous intéressons à l'identification des entités mentionnées dans ce document en étant guidés par la base de données. L'application concerne les documents numérisés reconnus par un OCR (documents océrisés).

Cette thèse a été réalisée dans le cadre du projet industriel DoD (Documents on Demand) en collaboration avec la société ITESOFT-Yooz [1]. ITESOFT-Yooz est un éditeur innovateur de logiciels spécialisés dans l'analyse et la reconnaissance de documents. Ses clients sont des administrations publiques ou privées (telles que CAF, EDF, les banques, etc.) qui lui envoient quotidiennement leurs documents en flux continu. Les documents peuvent être de différents types (électroniques, papiers, etc.) et de différentes classes (bons de commandes, factures, mails, etc.). Ils doivent être traités automatiquement de manière rapide et efficace. Pour ce faire, il est nécessaire d'identifier et d'extraire les informations qui y sont contenues et de les représenter de manière structurée. Comme les données qui décrivent des entités particulières dans les documents administratifs sont souvent sauvegardées dans des bases de données, nous nous sommes orientés vers une démarche de rapprochement entre la représentation d'entités dans le document et leurs définitions structurées dans la base de données (appelé rapprochement d'entités dans la suite). La finalité de ce processus est double : d'une part, enrichir le document par des méta-données à partir de la description des entités dans la base de données et d'autre part, compléter les enregistrements de la base à partir du contenu des documents, en y ajoutant les attributs manquants.

1.2 Difficultés

Le rapprochement d'entités consiste à identifier une entité mentionnée dans un document par un rassemblement d'un nombre suffisant de ses attributs (valeurs de champs), extraits en utilisant la base de

1. http ://www.yooz.fr

données. Cette identification doit se faire sans connaissance a priori sur le document traité. Une solution primitive consiste à maximiser l'intersection entre les termes du document et les attributs des entités de la base de données. Cette méthode est coûteuse, car elle nécessite une comparaison terme à terme entre le document et toutes les entités dans la base de données. De plus, elle est trop sensible aux ambiguïtés qui sont rencontrées lors de la sélection de la meilleure entité dans la base de données. Ces ambiguïtés peuvent être dues à des attributs partagés par plusieurs entités, ou qui apparaissent dans plusieurs champs de la base de données (Par exemple, l'attribut "Nancy" peut être à la fois un "prénom de personne" et un "nom de ville"). Un exemple est montré dans la Figure 1.1, où la Figure 1.1 (a) représente un document administratif qui mentionne une entité "personne" et la Figure 1.1 (b) représente deux entités dans la base de données des personnes. Ces deux entités possèdent des attributs en commun avec l'entité mentionnée dans le document et induisent ainsi une ambiguïté dans son rapprochement.

Les défis consistent à résoudre ces cas d'ambiguïté et à surmonter les variantes de représentation d'attributs dans la base de données et dans les documents. Ces différences sont causées par des variations textuelles telles que des abréviations (par exemple, "St Cecile" et "Saint Cecile" dans la Figure 1.1), des formats différents (par exemple pour la date, "14 juillet 1983" et "14/07/1983" dans la Figure 1.1), des surnoms, des fusions ou des permutations de mots, de l'absence de ponctuations, etc. Par ailleurs, comme nous traitons des documents reconnus par OCR, leurs contenus peuvent être bruités et leurs structures peuvent être altérées. À cela s'ajoute, la mauvaise qualité de la base de données qui peut contenir des redondances d'enregistrements, des informations incomplètes et des erreurs de saisie.

Le défi est d'autant plus important que l'état de l'art est quasiment inexistant dans le domaine de l'analyse de documents images. S'il est plus important dans le domaine de la fouille de données, les méthodes restent inadaptées au cas de documents océrisés.

Les approches existantes dans le domaine de la fouille de données traitent seulement des documents du Web et considèrent leur contenu comme une succession de phrases formant des paragraphes ou segments bien structurés. Elles se basent sur la comparaison entre les termes des segments et des entités dans la base de données. Ces approches sont inappropriées aux documents océrisés dont la structure d'origine est altérée par l'OCR. De plus, ces approches emploient des comparaisons strictes entre les termes qui échouent dans le cas d'un texte bruité reconnu par l'OCR. L'utilisation de mesures de similarité peut être intéressante pour résoudre ce problème, mais la question qui se pose est : quelle(s) mesure(s) utiliser, surtout que les erreurs d'OCR diffèrent des fautes d'orthographe courantes et nécessitent donc un traitement spécial ?

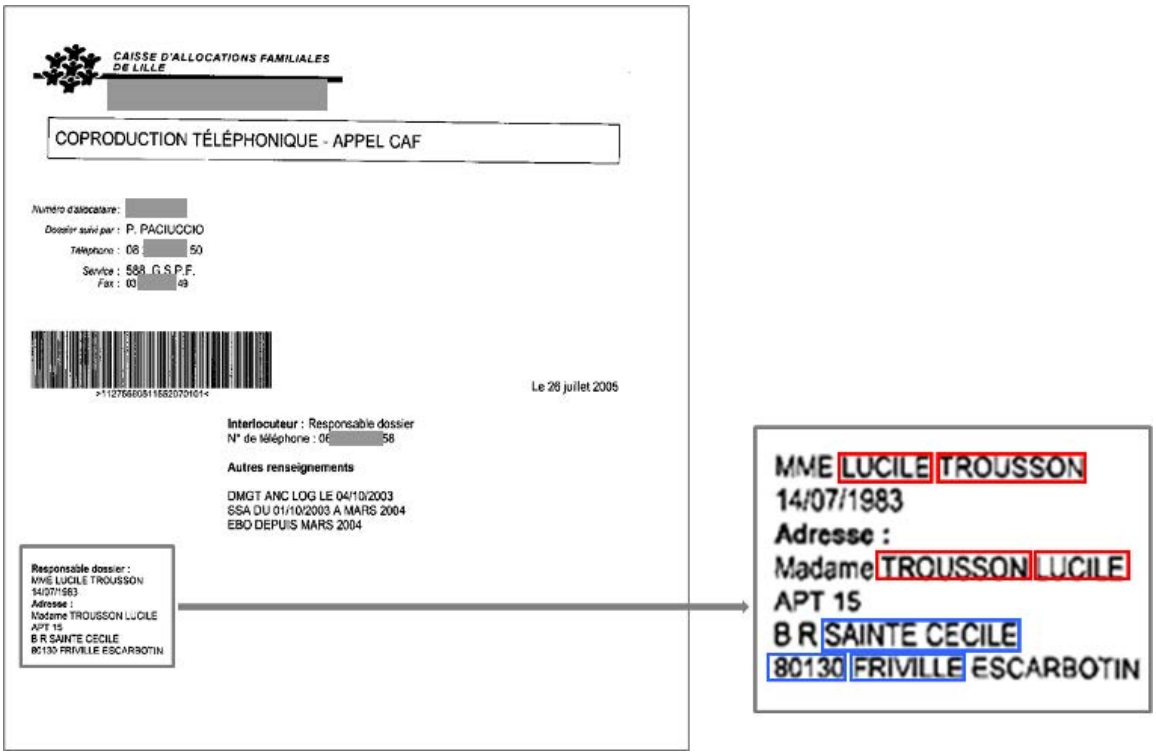
Des connaissances a priori sur la structure de documents et la représentation d'entités dans les documents pourraient faciliter la tâche de rapprochement d'entités. Ceci n'est pas possible dans notre cas car les documents arrivent en flux continu, leurs structures changent au cours du temps et la représentation d'entités varie.

Par ailleurs, les bases de données utilisées dans les approches existantes de reconnaissance d'entités sont supposées être fiables et complètes. Et même si des problèmes liés à la mauvaise qualité des bases de données ont été traités dans d'autres contextes, ils n'ont jamais été combinés, à nos connaissances, avec le problème de reconnaissance d'entités dans les documents.

À travers ces défis, apparaissent trois points qui doivent être pris en compte pour réussir le rapprochement de données. Ces points sont traités dans cette thèse :

— Un travail de normalisation et d'épuration des entités dans la base de données.
— Une modélisation de la structure des documents océrisés pour faciliter la recherche et l'extraction d'entités.
— Une prise en considération, lors du rapprochement, des cas d'ambiguïté, des erreurs de contenu générées par l'OCR et des variantes de représentation d'attributs dans le document et dans la base de données.

La figure 1.2 montre la démarche globale proposée pour réaliser le rapprochement. Cette démarche



(a) Exemple d'un document administratif

ID	Nom	Prénom	Date de naissance	Adresse	Code postal	Ville	Téléphone
5893486	Vanderlan	Nicolas	16 février 1789	Sainte Cecile	80130	Friville	0659145638
1258648	TROUSSON	LUCILE	14 juillet 1983	St Cecile	80000	Amiens	0593853215

(b) Échantillon d'une base de données décrivant des "personnes"

Figure 1.1 – Un exemple montrant une ambiguïté de rapprochement d'entités : les termes encadrés en rouge ou en bleu dans le document correspondent à des attributs (colorés en rouge en en bleu) de deux entités différentes dans la base de données.

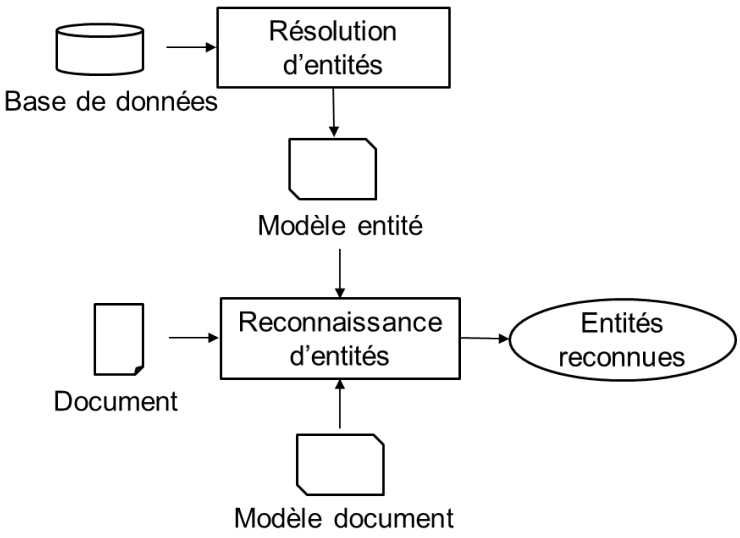


Figure 1.2 – Système global

comporte deux principales étapes : la résolution d'entités et la reconnaissance d'entités. La résolution d'entités consiste à coupler, dans une base de données, les enregistrements se référant à une même entité et de proposer pour chaque entité un modèle synthétisant ses variations. La reconnaissance d'entités a pour objectif d'identifier les entités mentionnées dans les documents en se servant de leurs modèles d'entité et d'un modèle document qui structure les entités telles qu'elles apparaissent dans le document.

1.3 Description des chapitres

Le reste de ce mémoire est organisé comme suit :

D'abord, nous commençons par décrire, dans le **Chapitre 2**, les corpus utilisés. Ensuite, nous présentons nos travaux de recherche en deux parties :

Dans la **Partie I**, nous détaillons l'étape de résolution d'entités dans une base de données. Cette partie est composée de deux chapitres :

Dans le **Chapitre 3**, nous présentons l'état de l'art sur les mesures de similarité entre des chaînes de caractères, qui serviront dans la comparaison d'attributs. Ensuite, nous donnons un aperçu sur les méthodes de couplage d'enregistrements.

Dans le **Chapitre 4**, nous décrivons une approche de résolution d'entités qui consiste à coupler les enregistrements se référant à une même entité et à les représenter par un modèle entité. Le couplage d'enregistrements se base sur la comparaison d'attributs en combinant plusieurs mesures de similarité. Deux méthodes de combinaison où l'une est supervisée et l'autre est non supervisée sont étudiées. L'approche de résolution d'entité est ensuite évaluée sur des bases de données réelles.

Dans la **Partie II**, nous détaillons l'étape de reconnaissance d'entités dans les documents océrisés. Cette partie est composée de trois chapitres :

Dans le **Chapitre 5**, nous étudions les méthodes de reconnaissance d'entités proposées dans la littérature en distinguant le cas des documents du Web et les documents océrisés. Nous mettons la lumière sur les traitements faits pour pallier les problèmes liés aux documents océrisés, notamment pour la correction des erreurs d'OCR.

Dans le **Chapitre 6**, nous proposons deux méthodes de reconnaissance d'entités basées sur le rapprochement par le contenu. La première, nommée M-EROCS, modélise les documents par des segments

(blocs de l'OCR) et propose de faire correspondre ces segments avec les entités dans le modèle entité. La deuxième méthode, nommée ERBL, se base sur l'étiquetage des attributs d'entités dans le document puis un regroupement des étiquettes (labels) en entités. Ces deux méthodes sont ensuite évaluées sur les corpus d'étude et comparées avec l'état de l'art.

Dans le **Chapitre 7**, nous résolvons le problème de reconnaissance d'entités à l'aide du rapprochement de graphes de structures. D'abord, nous présentons une étude théorique détaillée sur les méthodes de rapprochement de graphes dans la littérature. Ensuite, nous proposons une méthode de reconnaissance d'entités, nommée G-ELSE, qui emploie les relations structurelles pour la correction des erreurs d'étiquetage. Le chapitre se termine par des expérimentations qui feront ressortir l'intérêt de l'emploi des structures.

Enfin, le **Chapitre 8** présentera les conclusions sur le travail accompli dans cette thèse et donnera quelques perspectives de ce travail.

Chapitre 2

Description des corpus

Sommaire

2.1 Introduction

Pour diversifier les entités recherchées et montrer l'efficacité des méthodes proposées, nous avons utilisé trois types de corpus : corpus d'entreprises, corpus d'articles scientifiques et corpus de matériaux. Nous décrivons dans la suite chacun de ces corpus.

2.2 Corpus d'entreprises

C'est un corpus fourni par des clients de la société ITESOFT-Yooz, où les entités sont les clients et les fournisseurs de documents de type facture ou bon de commande. Ce corpus est composé de :

- 278 documents image multipages représentant :
  - 200 factures ;
  - 78 bons de commande ;
- une base de données qui contient 229338 enregistrements décrits chacun par :
  - des champs d'identifiants : identifiant, nom ;
  - des champs d'adresse : voie, complément d'adresse, code postal, ville, pays ;
  - des champs de données de contact : téléphone, fax, e-mail, site web ;
  - des champs de données juridiques : numéro SIREN, numéro SIRET, numéro TVA ;
- une table de vérité qui relie les documents avec leurs identifiants d'enregistrements dans la base de données (526 entités recherchées).

Chaque document image est traité par OCR et donné par un fichier XML (qui décrit la structure physique du document sous forme hiérarchique : blocs, lignes, mots et caractères avec leurs coordonnées dans le plan de l'image).

Nous montrons deux exemples de factures dans la Figure 2.1 où les éléments encadrés par la même couleur représentent les attributs d'une même entité. Nous remarquons que ces attributs sont regroupés

(dans des structures adresses, des structures de formulaires, etc.) dans différentes zones du document. La description des entités représentées dans ces documents dans la base de données est montrée dans le Tableau 2.1 où la première colonne représente les champs de la base d'entreprises et les colonnes suivantes représentent les valeurs de ces champs pour différents enregistrements.

Tableau 2.1 – Un échantillon de 4 enregistrements extraits de la base d'entreprises

	Enregistrement 1	Enregistrement 2	Enregistrement 3	Enregistrement 4
Id	37772	44181	44180	47460
Nom	Sas Jacoulot	Vecatel	Vecanord	Sarl Delomier
Adresse1	546 rue des Jacques	rue des Esselots	Rt Beaumont	(null)
Adresse2	(null)	Zac des Combottes	(null)	(null)
Code postal	71570	25700	44180	71800
Ville	Romaneche Thorins	Valentigney	Noyelles Godault	Varennes sous Dun
Téléphone	0385355185	0381361212	0321493820	0385280407
Fax	0385355556	0381345166	0321494029	(null)
E-mail	info@jacoulot.fr	trans@vecatel.fr	svecanord@-cassier.fr	(null)
Web	(null)	www.vecatel.fr	www.cassier.fr	(null)
Siren	401484183	300796554	409032059	382678092
Siret	40148418300017	30079655400032	40903205900035	(null)
Tva	FR69401484183	FR32300796554	FR17409032059	FR03382678092

Un autre exemple de document extrait du corpus d'entreprises, représentant une facture, est montré dans la Figure 2.2. Il contient des attributs communs partagés par des entités différentes (signalés par des couleurs spécifiques), comme les noms d'entreprises "TIMAC AGRO" et "FRANCE SECURITE".

La base d'entreprises est construite par plusieurs utilisateurs, de façon progressive et non contrôlée. Elle contient donc des redondances d'enregistrements se référant à une même entité. En se basant sur des statistiques faites sur cette base, nous comptons :

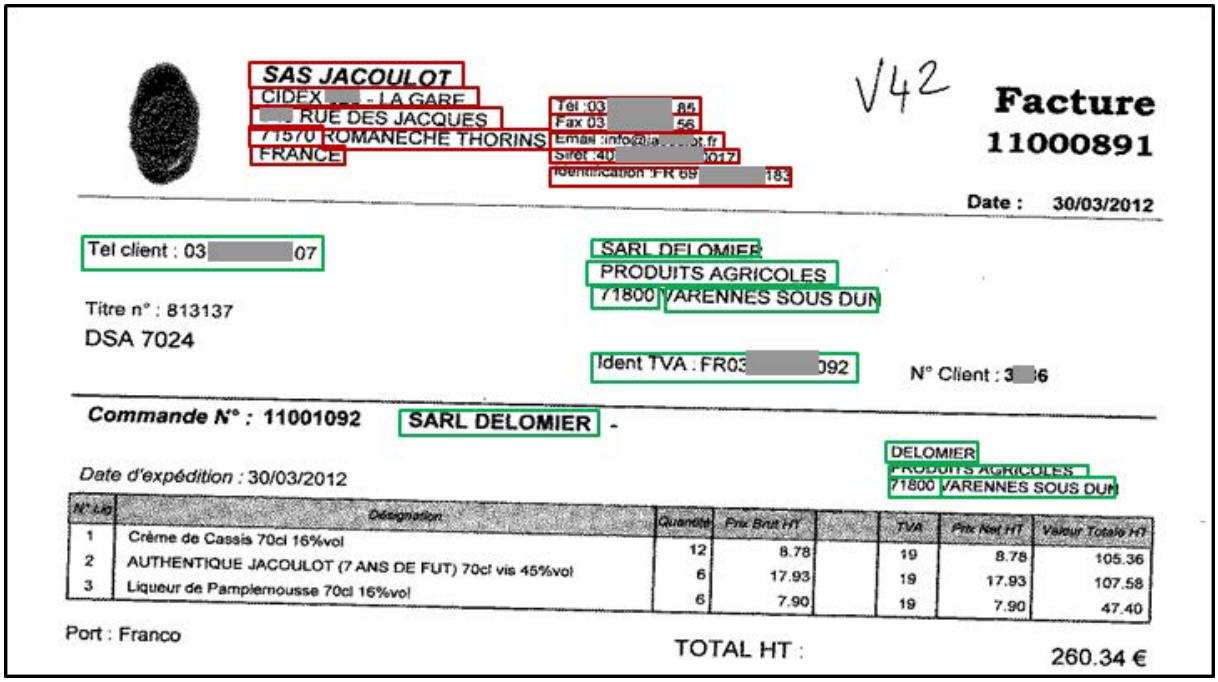
— 229338 enregistrements ;
— 129804 entités distinctes ;
— pourcentage de redondance : 43.4% ;
— nombre moyen d'enregistrements par entité : 1.77 ;
— nombre maximal d'enregistrements par entité : 181.

2.3 Corpus d'articles scientifiques

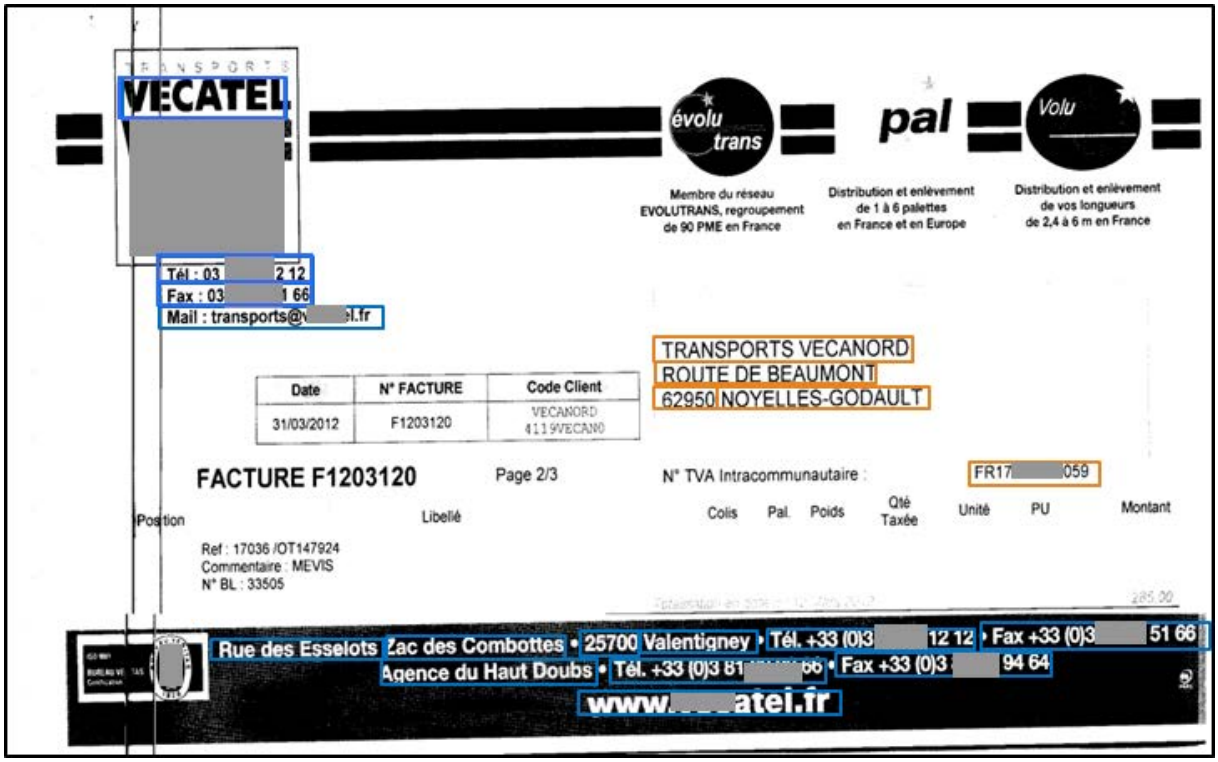
Le deuxième corpus est public. Il est extrait de sites d'archives d'articles scientifiques : HAL [2], ISTEX [3] et DBLP [4]. Il comprend des documents qui représentent les premières pages d'articles scientifiques. Nous avons aussi créé une base de données à partir de la description (disponible en fichiers XML ou CSV) des méta-données de ces articles. Les documents sont océrisés par l'OCR de la société ITESOFT-Yooz (ABBY Finereader paramétré par la société). Ce corpus contient :

— 252 images (premières pages des articles) représentant :
  — 50 revues dans des journaux (IJDAR, PRL, PR, etc.) ;
  — 140 actes de conférences (VLDB, KDD, ICPRAM, IEEE, etc.) ;
  — 48 manuscrits de thèse ;

2. https ://hal.archives-ouvertes.fr/
3. http ://www.istex.fr/
4. http ://dblp.uni-trier.de/



(a) Les attributs de l'entité en vert sont éparpillés dans différentes zones de la page du document. Les attributs de l'entité en rouge sont regroupés dans l'entête de la page et forment une structure d'adresse postale et une structure en formulaire.



(b) Les attributs de l'entité en bleu sont regroupés dans deux zones éloignées : l'entête et le pied de page.

Figure 2.1 – Exemples de deux documents, d'une page chacun, extraits du corpus d'entreprises. Les éléments encadrés par la même couleur représentent les attributs d'une même entité.

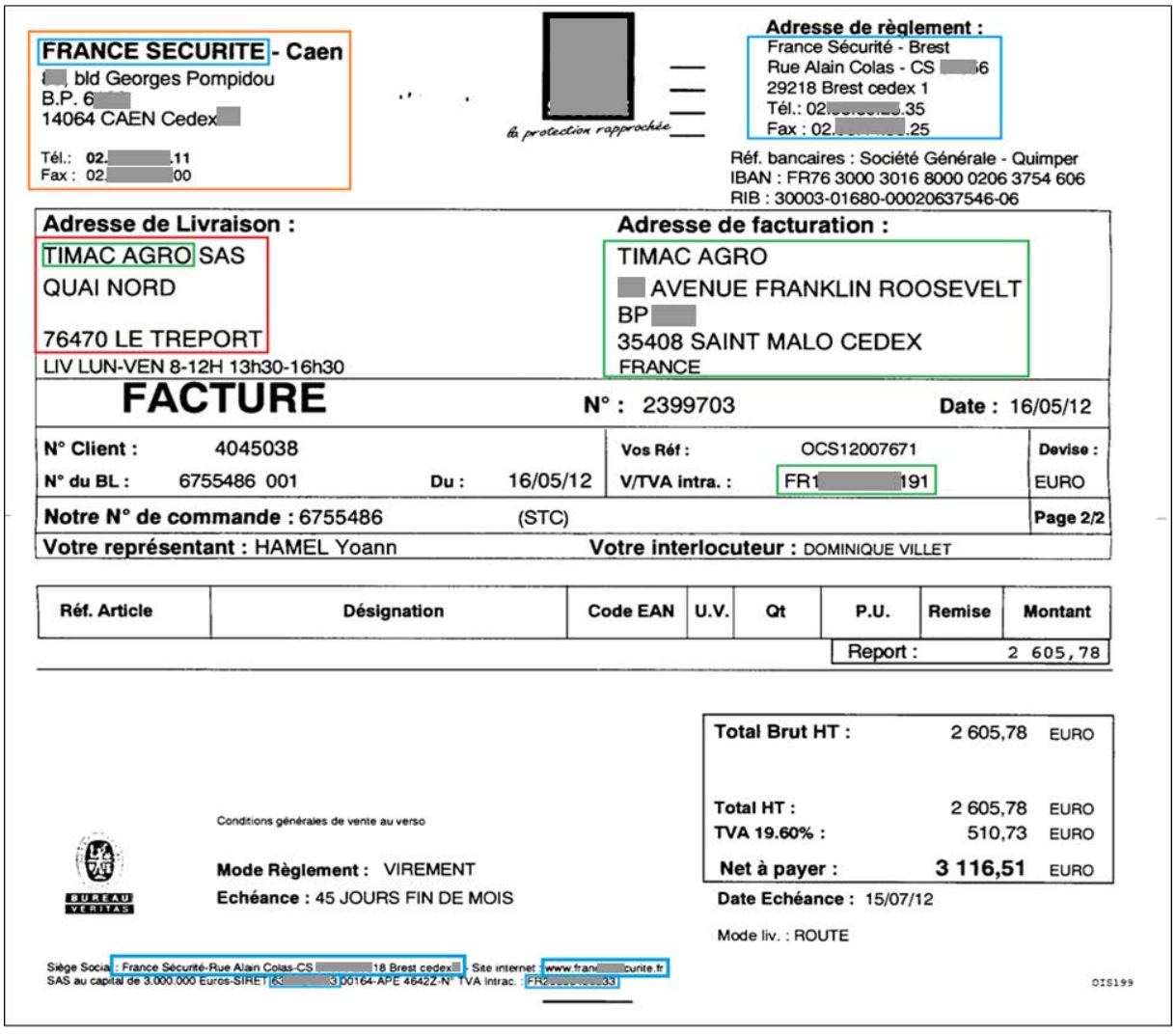


Figure 2.2 – Un document extrait du corpus d'entreprises montrant des entités différentes (représentées par des couleurs différentes) mais ayant des attributs communs. le nom d'entreprise "TIMAC AGRO" est commun entre l'entité en vert et l'entité en rouge ; le nom d'entreprise "FRANCE SECURITE" est commun entre l'entité en bleu et l'entité en orangé.

Tableau 2.2 – Un échantillon de 2 enregistrements extraits de la base de références bibliographiques

	Enregistrement 1	Enregistrement 2
Id	hal-00918671	conf/kdd/Lacoste-JulienPDKGG13
Type	Communication	(null)
Titre	SiGMa : Simple Greedy Matching for Aligning Large Knowledge Bases	SIGMa : simple greedy matching for aligning large knowledge bases
Auteurs	Simon Lacoste-Julien ; Konstantina Palla ; Alex Davies ; Gjergji Kasneci ; Thore Graepel ; Zoubin Ghahramani	S. Lacoste-Julien ; K. Palla ;A. Davies ; G. Kasneci ; T. Graepel ; Z. Ghahramani
Affiliations	INRIA ;University of Cambridge ;University of Cambridge ;Microsoft Research ; Microsoft Research ; University of Cambridge	(null)
Date production	2013-08-11	2013
Journal	(null)	(null)
Éditeur	(null)	(null)
Pages	pp. 572-580	572-580
Conférence	KDD 2013 - The 19th ACM SIGKDD International Conférence on Knowledge Discovery and Data Mining	KDD
Date conférence	2013-08-11	2013

  - 14 posters ;
- une base de données composée de 415500 enregistrements décrits par :
  - des champs de références : titre, sous-titre, type de la référence ;
  - des champs d'éditions : nom de conférence, nom de journal, volume, numéro, page, éditeur, date production, date acceptation, date publication ;
  - des champs d'auteurs : noms, prénoms, affiliations, adresses, téléphones, faxes, e-mails ;
- une table de vérité qui relie chaque document avec des identifiants d'enregistrements dans la base de données (252 entités recherchées).

Un exemple de première page d'un acte de conférence est montré dans la figure 2.3 où les éléments encadrés en vert représentent les méta-données sur cet article. Le Tableau 2.2 montre un extrait de la base de références bibliographiques où la première colonne représente les champs de la base de données et la deuxième, représente les attributs de deux enregistrements qui correspondent à l'entité représentée par la Figure 2.3. Nous constatons des variations dans certains attributs tels que l'abréviation dans le champ "conférence", la représentation des prénoms par leurs initiaux dans le champ "auteur" et le manque de certains attributs (représentés par "(null)"), etc.

D'autres exemples de premières pages d'articles scientifiques sont montrés dans la Figure 2.4, où la Figure 2.4 (a) correspond à un acte de conférence, la Figure 2.4 (b) correspond à une revue dans un journal, la Figure 2.4 (c) correspond à un poster et la Figure 2.4 (d) correspond à un manuscrit de thèse. En effet, ces documents montrent des variations dans les présentations d'attributs d'entités (attributs alignés à gauche, sur la même ligne, contigus, éparpillés, etc.), dans différentes zones dans le document (entête, pied de page, corps de la page, etc.).

Comme la base d'articles scientifiques est une fusion de données extraites de 3 différents sites, elle contient des redondances d'enregistrements, où nous comptons :

- 415500 enregistrements ;

SiGMa: Simple Greedy Matching for Aligning Large Knowledge Bases

Simon Lacoste-Julien
INRIA - Sierra project-team
Paris, France

Konstantina Palla
University of Cambridge
Cambridge, UK

Alex Davies
University of Cambridge
Cambridge, UK

Gjergji Kasneci
Microsoft Research
Cambridge, UK

Thore Graepel
Microsoft Research
Cambridge, UK

Zoubin Ghahramani
University of Cambridge
Cambridge, UK

ABSTRACT

The Internet has enabled the creation of a growing number of large-scale knowledge bases in a variety of domains containing complementary information. Tools for automatically aligning these knowledge bases would make it possible to unify many sources of structured knowledge and answer complex queries. However, the efficient alignment of *large-scale* knowledge bases still poses a considerable challenge. Here, we present Simple Greedy Matching (SiGMa), a simple algorithm for aligning knowledge bases with millions of entities and facts. SiGMa is an iterative propagation algorithm that leverages both the structural information from the relationship graph and flexible similarity measures between entity properties in a greedy local search, which makes it scalable. Despite its greedy nature, our experiments indicate that SiGMa can efficiently match some of the world's largest knowledge bases with high accuracy. We provide additional experiments on benchmark datasets which demonstrate that SiGMa can outperform state-of-the-art approaches both in accuracy and efficiency.

Categories and Subject Descriptors

D.2.12 [**Software Engineering**]: Interoperability—*Data mapping*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*Information networks*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Semantic networks*

Keywords

knowledge base; alignment; large-scale; entity; relationship; greedy algorithm

1. INTRODUCTION

In the last decade, a growing number of large-scale knowledge bases have been created online. Examples of domains include music, movies, publications, and biological data[1]. As these knowledge bases sometimes contain both overlapping and complementary information, there has been growing interest in attempting to merge them by *aligning* their common elements. This alignment could have important uses for information retrieval and question answering. For example, one could be interested in finding a scientist with expertise on certain related protein functions – information which could be obtained by aligning a biological database with a publication one. Unfortunately, this task is challenging to automate as different knowledge bases generally use different terms to represent their entities, and the space of possible matchings grows exponentially with the number of entities.

A significant amount of research has been done in this area – particularly under the umbrella term of *ontology matching* [7, 16, 12]. An ontology is a formal collection of world knowledge and can take different structured representations. In this paper, we will use the term *knowledge base* to emphasize that we assume very little structure about the ontology (to be specified in Section 2). Despite the large body of literature in this area, most of the work on ontology matching has been demonstrated only on fairly small datasets of the order of a few hundred entities. In particular, Shvaiko and Euzenat [26] identified *large-scale evaluation* as one of the ten challenges for the field of ontology matching.

In this paper, we consider the problem of aligning the *instances* in *large* knowledge bases, of the order of millions of entities and facts, where *aligning* means automatically identifying corresponding entities and interlinking them. Our starting point was the challenging task of aligning the movie database IMDb to the Wikipedia-based YAGO [28]. This provides another step towards the Semantic Web vision of interlinking different sources of knowledge as exemplified by the Linking Open Data Initiative[2] [4]. Initial attempts to match IMDb entities to YAGO entities by naively exploiting string and neighborhood information failed; we thus designed SiGMa (Simple Greedy Matching), a scalable greedy iterative algorithm that is able to exploit previous matching decisions as well as the relationship graph information between entities.

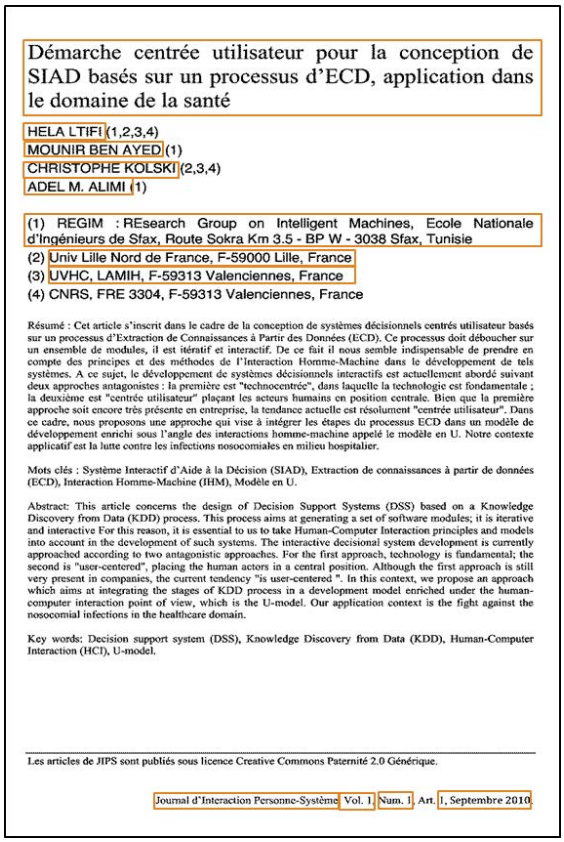
The design decisions behind SiGMa were both to be able to take advantage of the combinatorial structure of the match-

KDD'13, August 11–14, 2013, Chicago, Illinois, USA

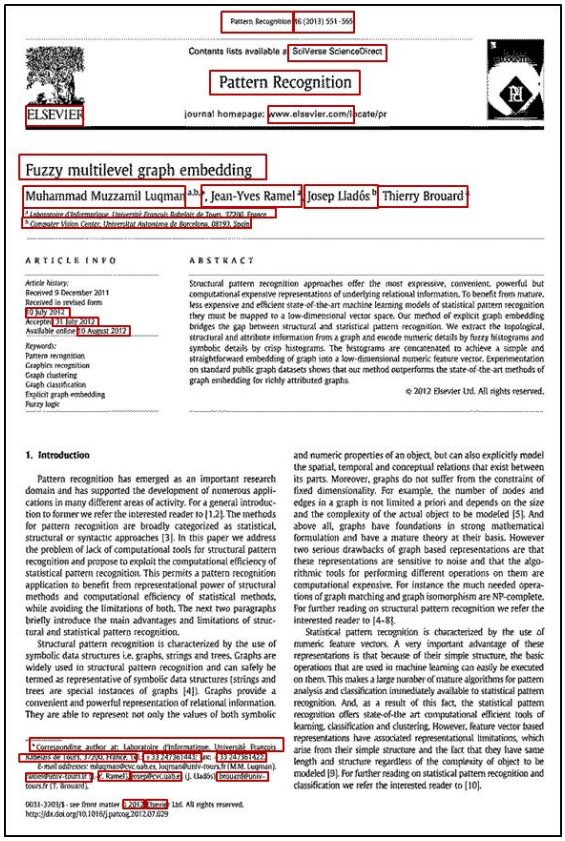
ACM 978-1-4503-2174-7/13/08 ...$15.00.

[1] Such as MusicBrainz, IMDb, DBLP, and UnitProt.
[2] http://linkeddata.org/

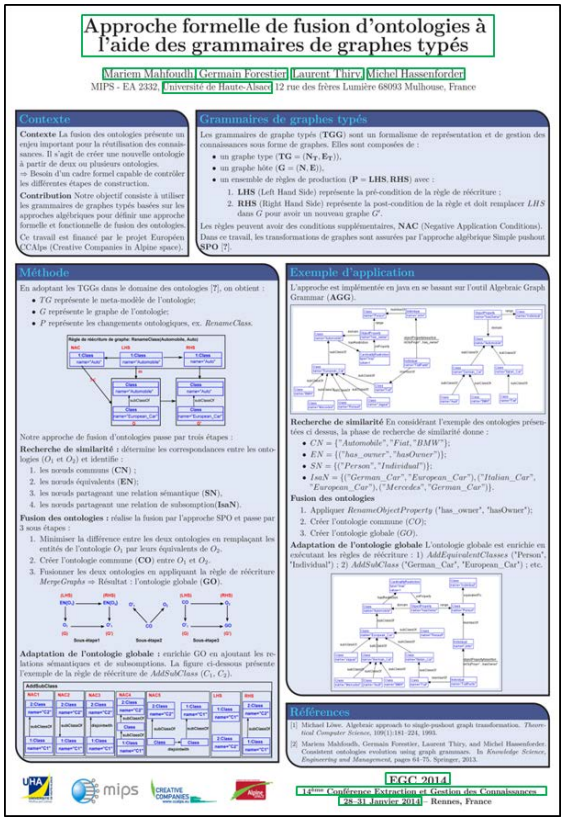
Figure 2.3 – Un document du corpus d'articles scientifiques, les éléments encadrés en vert représentent les méta-données sur cet article. Les attributs représentant le titre, les auteurs et leurs affiliations sont localisés dans le haut de la page. Les attributs qui décrivent la conférence sont localisés en bas de page.



(a) Un acte de conférence : attributs regroupés en haut ou en bas de page.



(b) Une revue dans un journal : attributs regroupés dans différentes zones de la page.



(c) Un poster : attributs regroupés dans l'entête ou le bas de page.

(d) Un manuscrit de thèse : attributs éparpillés dans la page.

Figure 2.4 – Exemples de documents extraits du corpus d'articles scientifiques (premières pages) montrant la variabilité de la structuration des attributs d'entités.

Tableau 2.3 – Un échantillon de 3 enregistrements extraits de la base de matériaux

	Enregistrement 1	Enregistrement 2	Enregistrement 3
Id	121	122	123
Référence	40-908-001	1-921-001-KIT1	41-200-044
Quantité	1	1	1
Description	Pxi chassis 8 slots	Pcivicia-Pxi control kit	Dmm 6.5 Smx2044
Num. série	8470NS3003	8102SC10	10073071
Prix	1766.00	1500.00	2063.00
Prix unit.	1766.00	1500.00	2063.00
Prix net	1766.00	1500.00	2063.00
Taxe	0	0	0

— 286695 entités distinctes ;
— pourcentage de redondance : 31% ;
— nombre moyen d'enregistrements par entité : 2.5 ;
— nombre maximal d'enregistrements par entité : 3.

2.4 Corpus de matériaux

C'est un corpus qui est fourni par des clients de la société ITESOFT-Yooz, où nous proposons d'extraire les matériaux commandés dans des factures. Ce corpus est composé de :

— 200 images multipages représentant 200 Factures ;
— une base de données composée de 86600 enregistrements décrits par :
  — des champs de matériel : référence, description, numéro de série, quantité, etc. ;
  — des champs de prix : prix unitaire, taxe, prix net, etc. ;
— une table de vérité qui relie chaque document avec des identifiants d'enregistrements dans la base de données (630 entités recherchées).

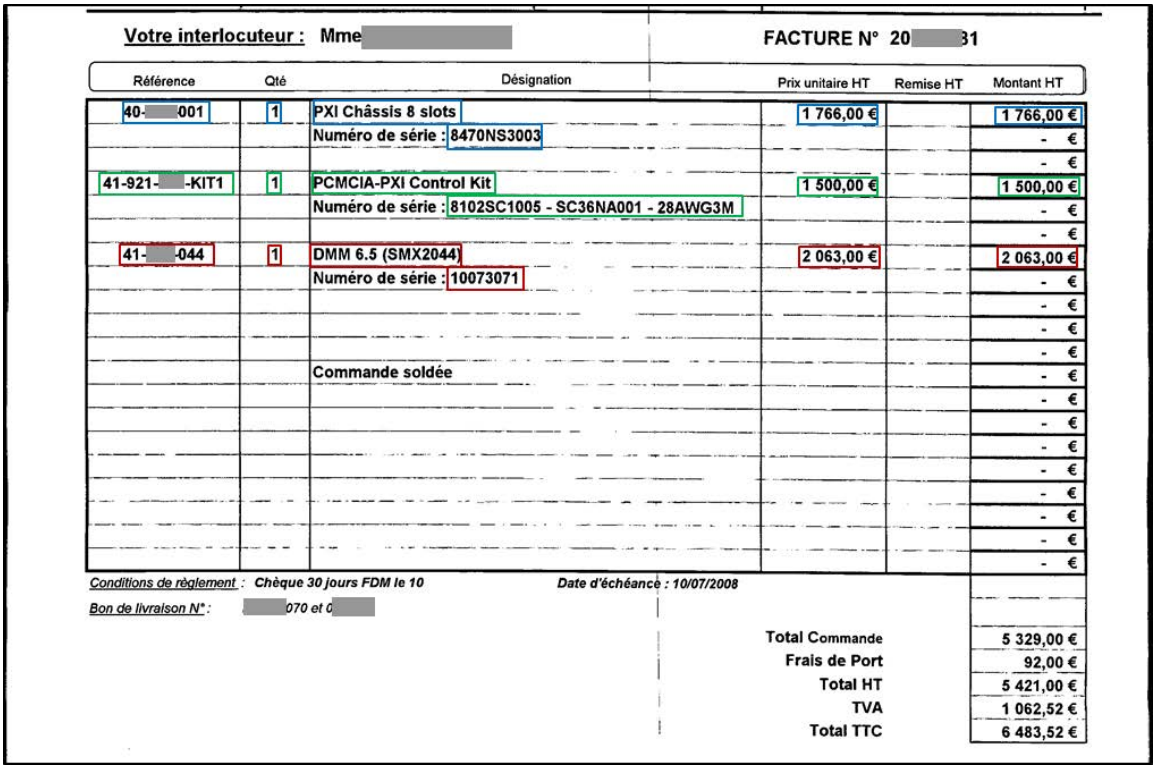
Pour chaque base de données, nous disposons d'une table de vérité, annotée manuellement, qui associe aux identifiants d'enregistrements se référant à la même entité, un même index.

Nous montrons deux extraits de factures dans la Figure 2.5 où les attributs des matériaux commandés, par chaque facture, sont décrits par les lignes d'un tableau. La description des matériaux commandés dans la facture de la Figure 2.5 (a), resp. Figure 2.5 (b) dans la base de matériaux est montrée par les colonnes du Tableau 2.3, resp. Tableau 2.4. Nous pouvons signaler la difficulté de rapprocher certains attributs à cause de variations dans leurs représentations dans le document et dans la base de données, A titre d'exemple, les attributs qui correspondent au champ "description" sont des chaînes de caractères longues et variables. De plus, certains attributs sont fusionnés tels que ceux qui correspondent aux champs "désignation" et "marque" (voir Tableau 2.4).

2.5 Conclusion

Nous avons décrit dans ce chapitre les corpus utilisés pour l'évaluation de nos méthodes.

D'après les exemples de documents montrés, nous constatons que les entités peuvent être structurées de différentes manières dans chacun des corpus (structures d'adresses, de formulaires, de lignes de tableau, etc.). De plus, les termes qui mentionnent les entités peuvent être rassemblés ou éparpillés. Ils peuvent figurer dans n'importe quelle zone de la page (dans l'entête, le corps de la page, le pied de page, etc.). Ceci va rendre complexe la tâche de reconnaissance d'entités dans ces documents.



Votre interlocuteur : Mme [redacted]

FACTURE N° 20[redacted]31

Référence	Qté	Désignation	Prix unitaire HT	Remise HT	Montant HT
40-[redacted]001	1	PXI Châssis 8 slots	1 766,00 €		1 766,00 €
		Numéro de série : 8470NS3003			- €
					- €
41-921-[redacted]-KIT1	1	PCMCIA-PXI Control Kit	1 500,00 €		1 500,00 €
		Numéro de série : 8102SC1005 - SC36NA001 - 28AWG3M			- €
					- €
41-[redacted]044	1	DMM 6.5 (SMX2044)	2 063,00 €		2 063,00 €
		Numéro de série : 10073071			- €
					- €
					- €
		Commande soldée			- €
					- €
					- €
					- €
					- €
					- €
					- €
					- €

Conditions de règlement : Chèque 30 jours FDM le 10 — Date d'échéance : 10/07/2008

Bon de livraison N° : [redacted]070 et 0[redacted]

Total Commande	5 329,00 €
Frais de Port	92,00 €
Total HT	5 421,00 €
TVA	1 062,52 €
Total TTC	6 483,52 €

(a) Facture 1

Item	Désignation	Marque	Référence	Qté	P.U. H.T.€	Total H.T. €
	Installation d'équipement Audiovisuel en salle de Communication écran en place 300 X 200					
	Choix du vidéoprojecteur en rapport à ces spécifications.					
	Vidéoprojecteur XGA 1024 X 768 DLP Vidéoprojecteur XGA 1024 X 768, **DMD 2600 Lumens** contraste 2500 : 1, 600 lignes vidéo, Zoom Focus manuel, Compatibilité Informatique: VGA - SXGA (640 X480 - 1280 X 1024). Lampe 5000 Hrs en mode éco, Pal, Sécam, NTSC 4,43, Entrées: 1Vidéo composite RCA, 1 YC, 1 SubD15 Informatique et Vidéo composante, DVI-D, compatible DVD/HDTV. **Garantie MITSUBISHI 3 ans, dont 1 an sur site (Voir Conditions MITSUBISHI)**	MITSUBISHI	XD 460 U	1	1 680,00	1 680,00
	Support de vidéoprojecteur + Potence Support incluant platine vidéoprojecteur et potence fixation	EUREX	SUPP	1	228,00	228,00
	Sélecteur de sources IR et contrôle de volume	EXTRON	MLC52IRVC	1	298,00	298,00
	Support sélecteur de source (support mural)	EXTRON	EWB2	1	48,00	48,00
	Forfait câblage, connectique, visserie, . (comprenant câbles, connectiques, et petites fournitures.)	IMATECH	DIV	1	80,00	80,00
	Forfait Main d'œuvre, installation, réglages, Formation.2 Tech. Démontage de la partie du faux plafond et de l'ensemble support et vidéoprojecteur DELL actuellement en place.Démontage faux plafond, Adaptation et Installation nouveau support et vidéoprojecteur en plafond, Déplacement et raccordement câbles secteurs, Vidéo, Informatique et pilotage du nouveau matériel.Remontage faux plafond, Installation raccordement et programmation Sélecteur de sources EXTRON. connectique, réglages vidéo, et informatique, Essais, contrôles, et formation utilisateurs.	IMATECH	MOT	1	840,00	840,00

Total Euros H.T.	3 174,00
T V A 19,6%	622,10
Total Euros T.T.C.	**3 796,10**

(b) Facture 2

Figure 2.5 – Deux factures extraites du corpus de matériaux, les attributs des matériaux commandés par chaque facture sont décrits par les lignes d'un tableau. Les éléments encadrés par une même couleur représentent les attributs d'une même entité.

Tableau 2.4 – Un échantillon de 6 enregistrements extraits de la base de matériaux

	Enregistrement 1	Enregistrement 2	Enregistrement 3	Enregistrement 4	Enregistrement 5	Enregistrement 6
Id	88	89	90	91	92	93
Référence	(null)	(null)	(null)	(null)	(null)	(null)
Quantité	1	1	1	1	1	1
Description	Videoprojecteur XGA $1024X768$ OLP Mitsubichi XD 4	Support de videoprojecteur potence Eurex supp sup	Selecteur de sources IR et contrôle de volume extr	Support selecteur de source support mural Extron E	Forfait cablage connectique visserie. Imatech DIV	Forfait main d'œuvre installation réglages format
Num. série	(null)	(null)	(null)	(null)	(null)	(null)
Prix	(null)	(null)	(null)	(null)	(null)	(null)
Prix unit.	1680.00	228	298	48	80	840
Prix net	1680.00	228	298	48	80	840
Taxe	0	0	0	0	0	0

D'après la description des bases de données, nous signalons que les deux premières bases (celle d'entreprises et celle d'articles scientifiques) souffrent du problème de redondance. Elles nécessitent donc un traitement de résolution d'entités.

Première partie

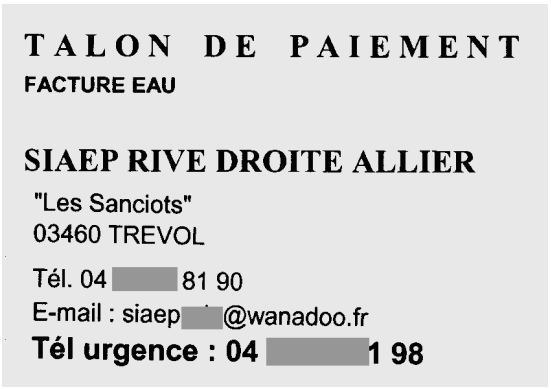
Résolution d’entités

Introduction

La base d'entités est souvent un dépôt de données fait de manière non contrôlée : elle résulte d'une fusion de différentes sources de données ; elle est mise à jour dynamiquement et elle est souvent gérée par plusieurs administrateurs. Ceci conduit à des données dupliquées, incomplètes et erronées. Les attributs d'entités peuvent être absents, nombreux ou différents, comprenant des écritures non standardisées et des erreurs typographiques.

Pour exploiter ces données, il est nécessaire de les épurer. La résolution d'entités joue ce rôle en faisant la synthèse de tous les enregistrements, en enlevant les redondances de définitions, en identifiant les différentes écritures d'un même attribut telles que les abréviations, les acronymes et en repérant les erreurs typographiques, voire les corrigeant quand elles sont identifiées.

La Figure 1 montre un talon de paiement extrait d'une facture et le Tableau 1 montre les 3 enregistrements qui décrivent l'entité "entreprise" dans la base de données. Nous notons des représentations variables de certains attributs (par exemple, dans "nom" et "ville"), des manques d'attributs dans certains enregistrements (par exemple, "e-mail" dans Enreg. 1 et Enreg. 2) ou des valeurs différentes pour certains attributs (par exemple, dans "téléphone").


TALON DE PAIEMENT
FACTURE EAU

SIAEP RIVE DROITE ALLIER
"Les Sanciots"
03460 TREVOL
Tél. 04 [redacted] 81 90
E-mail : siaep[redacted]@wanadoo.fr
Tél urgence : 04 [redacted] 1 98

Figure 1 – Talon de paiement extrait d'une facture

Cette partie est organisée en deux chapitres. Dans le Chapitre 3, nous présentons l'état de l'art sur les mesures de similarité et les méthodes de couplage d'enregistrements. Dans le Chapitre 4, nous proposons une approche de résolution d'entités et nous proposons une validation expérimentale de cette approche.

Tableau 1 – Un échantillon de 3 enregistrements extraits de la base des entreprises

	Enregistrement 1	Enregistrement 2	Enregistrement 3
Nom	Siaep rive droite allier	S.i.a.e.p. allier	Siaep rive droite
Adresse	Les sanciots	Les sanciots	Saint-Jean
Code postal	03460	03460	03400
Ville	Trevol	Trevol	Trevol auvergne
Téléphone	(null)	0470468191	0470468198
Fax	(null)	0470468190	(null)
E-mail	siaep@rda.fr	(null)	siaep.rda@wanadoo.fr

Chapitre 3

État de l'art

Sommaire

3.1 Introduction

Le problème de la résolution d'entités est plus connu dans la communauté des bases de données sous le nom de fusion (merge)/nettoyage (purge). Il consiste à relier les enregistrements se référant à la même entité, nommé couplage d'enregistrements, dans le but d'enlever les redondances. Le couplage d'enregistrements dans une base de données s'appuie souvent sur la comparaison d'attributs en utilisant des mesures de similarité qui tolèrent certaines variantes de représentations.

Dans ce chapitre, nous présentons dans la Section 3.2 une étude de plusieurs mesures de similarité proposées dans la littérature. Nous nous intéressons également aux techniques de comparaison et de combinaison de ces mesures. Ensuite, nous proposons dans la Section 3.3 une étude bibliographique des méthodes de couplage d'enregistrements dans une base de données.

3.2 Mesures de similarité

Pour comparer les attributs d'enregistrements d'une base de données, des mesures de similarité entre chaînes de caractères sont souvent utilisées dans le but de tolérer leurs variations. Des études de ces mesures sont présentées dans [Cohen03a, Cohen03b, Bilenko03a]. Les mesures peuvent être classées en trois catégories : les mesures basées sur les séquences de caractères, les mesures basées sur les sacs de mots et les mesures hybrides. Les 8 enregistrements extraits de la base d'entreprises, qui se réfèrent à une même entité et qui sont représentés dans le Tableau 3.1, serviront d'exemples.

Tableau 3.1 – Échantillon de 8 enregistrements de la table des entreprises

	Nom	Adresse	Code postal	Ville	Téléphone
Enregistrement 1	Xerox	24 rue de la chapelle	97122	Baie Mahault	0825082081
Enregistrement 2	Xerox0	(null)	(null)	(null)	(null)
Enregistrement 3	Xeroxfs	CDG	92000	Neuilly-sur-Seine	0825082090
Enregistrement 4	Xerox financial servicex	120 avenue Charles de gaulle	92202	Neuilly	0825082081
Enregistrement 5	Xerox finacial services	Charles de gaulle	92202	Neuilly	0825082081
Enregistrement 6	Xerox financial sces	120 av. Charles de gaulle	92000	Neuilly sur seine	0825082081
Enregistrement 7	Xerox financial services	120 av Charles de gaulle	92202	Neuilly	0825082090
Enregistrement 8	Financial services xerox	Av Charles de gaulle	92000	Neuilly sur seine	(null)

3.2.1 Mesures à base de séquences de caractères

Ces mesures définissent la similarité par la présence de caractères identiques à des positions similaires. Nous étudions ci-dessous les mesures de Levenshtein, de Jaro et de Jaro-Winkler.

3.2.1.1 Mesure de Levenshtein

C'est la mesure d'édition la plus utilisée pour comparer des chaînes de caractères [Levenshtein66]. Elle affecte des coûts pour les opérations d'insertion, de suppression ou de substitution et calcule le coût minimal de transformation d'une chaîne en une autre. Soient $S = s_1, s_2, \ldots, s_n$ et $T = t_1, t_2, \ldots, t_m$ deux chaînes de caractères. La mesure de Levenshtein entre S et T est donnée par d_{mn}, définie par la récurrence de l'équation (3.1) :

$$
\begin{array}{ll}
d_{0,0} = 0 \ ; & \\
d_{i,0} = \sum_{k=1}^{i} coût(t_k \rightarrow \epsilon) & \forall\, 1 \leq i \leq m; \\
d_{0,j} = \sum_{k=1}^{j} coût(\epsilon \rightarrow s_k) & \forall\, 1 \leq j \leq n; \\
\multicolumn{2}{l}{d_{i,j} = \left\{ \begin{array}{ll} d_{i-1,j-1} & \text{si } s_j = t_i \\ \min \left\{ \begin{array}{l} d_{i-1,j} + coût(t_i \rightarrow \epsilon) \\ d_{i,j-1} + coût(\epsilon \rightarrow s_j) \\ d_{i-1,j-1} + coût(s_i \rightarrow t_j) \end{array} \right. & \forall\, s_j \neq t_i \end{array} \right. \quad \forall\, 1 \leq i \leq m, 1 \leq j \leq n.}
\end{array}
\tag{3.1}
$$

où ϵ représente le vide, $coût(t_i \rightarrow \epsilon)$, $coût(\epsilon \rightarrow s_j)$ et $coût(s_i \rightarrow t_j)$ représentent respectivement le coût de suppression de t_i, le coût d'insertion de s_j et le coût de substitution de s_i par t_j.

Dans la version classique de la mesure de similarité de Levenshtein, des coûts unitaires sont utilisés (voir l'équation (3.2)).

$$
coût(s_i \rightarrow t_j) = \left\{ \begin{array}{ll} 0 & \text{si } s_i = t_j \\ 1 & \text{sinon.} \end{array} \right. \tag{3.2}
$$

La mesure de similarité de Levenshtein normalisée sur $[0, 1]$, est définie par l'équation (3.3).

$$\textit{Levenshtein}(S,T) = 1 - \frac{d_{mn}}{\max(|S|,|T|)} \tag{3.3}$$

où $|x|$ représente la longueur de la chaîne de caractères x.

Exemple 1. Soient les trois chaînes de caractères : S="*Xerox*", T ="*Xerox0*" et U="*Xeroxfs*". Les mesures de Levenshtein entre S, T et S, U sont données par :

$$\textit{Levenshtein}(S,T) = 1 - \frac{\textit{coût}(\epsilon \rightarrow \text{`0'})}{|T|} = 1 - \frac{1}{6} = 0.83$$

$$\textit{Levenshtein}(S,U) = 1 - \frac{\textit{coût}(\epsilon \rightarrow \text{`}f\text{'}) + \textit{coût}(\epsilon \rightarrow \text{`}s\text{'})}{|U|} = 1 - \frac{2}{7} = 0.71$$

3.2.1.2 Mesure de Jaro

La mesure de Jaro [Jaro95] se base sur le nombre et l'ordre des caractères communs entre les deux chaînes de caractères. Ainsi, la mesure de Jaro entre S et T est définie par l'équation (3.4).

$$\textit{Jaro}(S,T) = \frac{1}{3}.(\frac{c}{m} + \frac{c}{n} + \frac{c - Tr}{c}) \tag{3.4}$$

avec :

- c est le nombre de caractères de S communs avec T ; un caractère s_i de S est dit en commun avec T s'il existe t_j dans T tel que $s_i = t_j$ et $i - H \leq j \leq i + H$, où $H = \frac{1}{2}.\min(n, m)$;
- Tr est la moitié du nombre de transpositions de S et T, où une transposition de S et T est une position i tel que $s_i \neq t_i$.

Exemple 2. Soient les chaînes de caractères : S="*Xerox*" et T ="*Xerox0*". La mesure de Jaro entre S est donnée par :

$$Jaro(S,T) = \frac{1}{3}.(\frac{5}{6} + \frac{5}{5} + \frac{5 - 0.5}{5}) = 0.92$$

où la chaîne commune est "*Xerox*" et $H = \frac{5}{2}$.

3.2.1.3 Mesure de Jaro-Winkler

La mesure de Jaro-Winkler [Winkler99] est une variante de la mesure de Jaro. Elle est définie par l'équation (3.5). Elle donne des mesures plus précises que la mesure de Jaro puisqu'elle met en valeur les longs préfixes communs (dont la taille maximale est fixée à 4) entre les deux chaînes de caractères.

$$\textit{Jaro-Winkler}(S,T) = \textit{Jaro}(S,T) + \frac{P'}{10}.(1 - \textit{Jaro}(S,T)) \tag{3.5}$$

avec $P' = \min(4, P)$ où P est le plus long préfixe commun entre S et T.

Exemple 3. Soient les chaînes de caractères : S="*Xerox*" et T ="*Xerox0*". La mesure de Jaro-Winkler entre S et T est donnée par :

$$\textit{Jaro-Winkler}(S,T) = 0.92 + \frac{4}{10}.(1 - 0.92) = 0.95$$

avec $P = 5$ et $P' = \min(4, 5) = 4$.

3.2.2 Mesures à base de sacs de mots

Les mesures à base de sacs de mots (en anglais, token based), appelées aussi n-grammes de caractères proposent de traiter chaque chaîne de caractères comme un ensemble d'éléments (les mots ou les n-grammes) et de comparer le nombre d'éléments en commun entre les deux chaînes indépendamment de leurs positions. Nous distinguons les mesures de Jaccard et Tf-idf.

3.2.2.1 Mesure de Jaccard

Soient S et T deux ensembles de mots, la mesure de Jaccard [Jaccard66] est définie par le ratio entre l'intersection et l'union de S et T. Elle est donnée par l'équation (3.6). Cette mesure tolère la permutation de termes.

$$\textit{Jaccard}(S,T) = \frac{S \cap T}{S \cup T} \tag{3.6}$$

Exemple 4. Soient les chaînes de caractères S="Xerox financial services", T= "Financial services xerox", U="Xerox financial servicex"et V="Xerox finacial services". Les mesures de Jaccard entre S, T et U, V sont données par :

$$\textit{Jaccard}(S,T) = \frac{3}{3} = 1$$

$$\textit{Jaccard}(U,V) = \frac{1}{5} = 0.2$$

Il convient de constater que la valeur de la mesure de Jaccard entre U et V est faible (0.2) car elle utilise une comparaison stricte entre les mots et ne tolère donc pas les erreurs de saisie.

3.2.2.2 Mesure Tf-idf (ou Cosinus)

La mesure Tf-idf [Jones72] est très utilisée dans le domaine de recherche d'information et se base sur la fréquence et l'importance des termes en commun entre les deux chaînes relativement à un corpus ou une collection de mots. Soient S et T deux chaînes de caractères composées d'ensembles de mots. $Tf_{w,S}$ représente le nombre d'occurrences du terme w dans la chaîne de caractères S. Idf_w représente le nombre total d'enregistrements dans la base de données divisé par le nombre d'enregistrements qui contiennent au moins une occurrence de w. La mesure Tf-idf est donnée par l'équation (3.7).

$$\textit{Tf-idf}(S,T) = \sum_{w \in S \cap T} V(w,S).V(w,T) \tag{3.7}$$

avec $V(w,S) = V'(w,S)/\sqrt{\sum_{w \in S} V'(w,S)^2}$ et $V'(w,S) = \log(Tf_{w,S}+1).\log(Idf_w)$.

Exemple 5. Soient les chaînes de caractères S="Xerox", T= "Xerox financial servicex" et U="Xerox finacial services". Nous supposons que la base de données est composée seulement des 8 enregistrements de l'exemple dans le Tableau 3.1. La mesure Tf-idf entre S et T est donnée par :

$$\textit{Tf-idf}(S,T) = V(\text{“}Xerox\text{”},S).V(\text{“}Xerox\text{”},T) = 0.13$$

avec :

- $Tf_{\text{“}Xerox\text{”},S} = 1$ et $Idf_{\text{“}Xerox\text{”}} = \frac{8}{6} = 1.33$ d'où $V(\text{“}Xerox\text{”},S) = 1$;
- $Tf_{\text{“}financial\text{”},T} = 1$ et $Idf_{\text{“}financial\text{”}} = \frac{8}{4} = 2$ d'où $V'(\text{“}financial\text{”},T) = 0.09$;
- $Tf_{\text{“}servicex\text{”},T} = 1$ et $Idf_{\text{“}servicex\text{”}} = \frac{8}{1} = 8$ d'où $V'(\text{“}servicex\text{”},T) = 0.27$;

— $Tf_{\text{“}Xerox\text{”},T} = 1$ et $Idf_{\text{“}Xerox\text{”}} = \frac{8}{6} = 1.33$ d'où $V'(\text{“}Xerox\text{”}, T) = 0.04$ et $V(\text{“}Xerox\text{”}, T) = 0.13$.

En suivant le même calcul précédent, la mesure Tf-idf entre T et U est donnée par :

$$\textit{Tf-idf}(T, U) = V(\text{“}Xerox\text{”}, T).V(\text{“}Xerox\text{”}, U) = 0.13 * 0.12 = 0.02$$

Nous remarquons que cette mesure donne une valeur faible à cause des erreurs de saisie.

3.2.3 Mesures hybrides

Les mesures hybrides proposent de combiner les caractéristiques des deux précédents types de mesures afin de profiter de leurs avantages. Nous étudions les mesures de Monge-Elkan et Soft-tf-idf ci-dessous.

3.2.3.1 Mesure de Monge-Elkan

Soient les chaînes de caractères S et T composées chacune par un ensemble de mots. La mesure de Monge-Elkan, proposée dans [Monge96], cherche le meilleur appariement de chaque mot de S avec un mot de T. Cet appariement se fait l'aide d'une mesure basée sur les séquences de caractères permettant de tolérer les variations de certains caractères. Elle est présentée dans l'équation (3.8).

$$\textit{Monge-Elkan}(S, T) = \frac{1}{|S|}.\sum_{i=1}^{|S|} \max_{j=1,|T|} sim(S[i], T[j]) \tag{3.8}$$

où $|S|$, resp. $|T|$ représente le nombre de mots dans S, resp. T, $S[i]$, resp. $T[j]$ représente le $i$ème, resp. $j$ème mot dans S, resp. T et sim représente une mesure basée sur les séquences de caractères (par exemple la mesure de Levenshtein).

Exemple 6. Soient les chaînes de caractères S= “Xerox financial servicex” et T=“Xerox finacial services”. La mesure de Monge-Elkan entre S et T, en considérant la mesure de Levenshtein comme mesure secondaire, est donnée par :

$$\textit{Monge-Elkan}(S, T) = 0.92$$

avec :

— $Levenshtein(\text{“}Xerox\text{”}, \text{“}Xerox\text{”}) = 1$

— $Levenshtein(\text{“}financial\text{”}, \text{“}finacial\text{”}) = 0.89$;

— $Levenshtein(\text{“}servicex\text{”}, \text{“}services\text{”}) = 0.88$.

Cette mesure donne une valeur importante (0.92) par rapport à celle donnée par la mesure de Jaccard (0.2) grâce à l'utilisation de la mesure tolérante de Levenshtein pour la comparaison entre les mots au lieu d'une comparaison stricte.

3.2.3.2 Mesure Soft-Tf-idf

La mesure Soft-tf-idf [Cohen03a] est une variation de la mesure Tf-idf combinée avec une deuxième mesure basée sur les séquences de caractères. En effet, cette mesure étend l'ensemble de mots en commun à l'ensemble de mots similaires par rapport à la deuxième mesure. Soient S et T deux ensembles de mots, la mesure Soft-tf-idf est définie par l'équation (3.9).

$$\textit{Soft-Tf-idf}(S, T) = \sum_{w \in Close(\theta,S,T)} V(w, S).V(w, T).D(w, T) \tag{3.9}$$

avec :

— $Close(\theta, S, T)$ est l'ensemble des mots $w \in S$ où il existe $v \in T$ tel que $D(w, v) > \theta$;
— pour $w \in Close(\theta, S, T)$, $D(w, T) = \max_{v \in T} dist(w, v)$ telle que $dist$ est une mesure basée sur les séquences de caractères.

Exemple 7. Soient les chaînes de caractères S= "Xerox financial servicex" et T="Xerox finacial services". Nous supposons que la base de données est composée seulement des 8 enregistrements de l'exemple dans le Tableau 3.1 et nous considérons que $dist$ est la mesure de Jaro-Winkler et que $\theta = 0.9$. La mesure Soft-Tf-idf entre S et T est donnée par :

$$\begin{aligned}\textit{Soft-Tf-idf}(S,T) =& V(\text{“}Xerox\text{”}, S).V(\text{“}Xerox\text{”}, T).D(\text{“}Xerox\text{”}, T) \\ &+ V(\text{“}\textit{financial}\text{”}, S).V(\text{“}\textit{finacial}\text{”}, T).D(\text{“}\textit{financial}\text{”}, T) \\ &+ V(\text{“}servicex\text{”}, S).V(\text{“}services\text{”}, T).D(\text{“}servicex\text{”}, T) \\ =& 0.13 * 0.12 * 1 + 0.31 * 0.90 * 0.98 + 0.94 * 0.42 * 0.95 \\ =& 0.67\end{aligned}$$

avec :
— $D(\text{“}Xerox\text{”}, T) = \textit{Jaro-Winkler}(\text{“}Xerox\text{”}, \text{“}Xerox\text{”}) = 1$;
— $D(\text{“}\textit{financial}\text{”}, T) = \textit{Jaro-Winkler}(\text{“}\textit{financial}\text{”}, \text{“}\textit{finacial}\text{”}) = 0.98$;
— $D(\text{“}servicex\text{”}, T) = \textit{Jaro-Winkler}(\text{“}servicex\text{”}, \text{“}services\text{”}) = 0.95$.

En comparant avec la mesure Tf-idf qui a retourné une valeur faible (0.02), on a ici une valeur importante (0.67) avec la mesure Soft-tf-idf qui tolère une marge de variations dans les mots composant la chaîne de caractères.

Tableau 3.2 – Comparaison de mesures de similarité

Type	Mesure	Chaînes de caractères comparées	Valeur	Particularité
Mesures à base de séquences de caractères	Levenshtein [Levenshtein66]	“Xerox”, “Xeroxfs”	0.71	Considère les opérations de transformation d’une chaîne à l’autre.
		“Xerox services financial”, “Financial services xerox”	0.25	
	Jaro [Jaro95]	“Xerox”, “Xeroxfs”	0.90	Adaptée au traitement de chaînes courtes. Considère le nombre et la position des caractères communs.
		“Xerox services financial”, “Financial services xerox”	0.59	
	Jaro-Winkler [Winkler99]	“Xerox”, “Xeroxfs”	0.94	Met en avant les préfixes communs
		“Xerox services financial”, “Financial services xerox”	0.59	
Mesures à base de sacs de mots	Jaccard [Jaccard66]	“Xerox financial services”, “Financial services xerox”	1.00	Met en avant les mots en commun indépendamment de leur position.
		“Xerox financial servicex”, “Xerox finacial services”	0.20	
	Tf-idf [Jones72]	“Xerox”, “Xerox financial servicex’	0.20	Évalue l’importance des mots en commun dans les chaînes, relativement à la collection de chaînes dans la base de données.
		“Xerox financial servicex”, “Xerox finacial services”	0.04	
Mesures hybrides	Monge-Elkan [Monge96]	“Xerox”, “Xerox financial servicex”	0.33	Tolère une marge de variation (par la mesure de Levenshtein) des mots en commun.
		“Xerox financial servicex”, “Xerox finacial services”	0.92	
	Soft-Tf-idf [Cohen03a]	“Xerox”, “Xerox financial servicex”	0.20	Évalue l’importance des mots en commun dans les chaînes, relativement à la collection de chaînes dans la base de données, en tolérant leur variation.
		“Xerox financial servicex”, “Xerox finacial services”	0.83	

3.2.4 Comparaison et combinaison de mesures

Une comparaison entre les différentes mesures présentées est montrée dans le Tableau 3.2. Il convient de noter que chacune des mesures présentées ci-dessus est intéressante pour absorber des variations particulières dans la représentation d'attributs. En effet, les mesures à base de séquence de caractères sont clairement utiles pour remédier aux erreurs typographiques et aux erreurs d'OCR. Les mesures à base de sacs de mots sont intéressantes dans les cas où l'attribut est composé de plusieurs mots et permettent alors de mettre en valeur certains mots pertinents dans la comparaison de deux chaînes de caractères. Les mesures hybrides combinent les avantages des deux types précédents.

Il est donc intéressant d'utiliser plusieurs types de mesures en fonction de la nature du champ en question et le type de variation que nous souhaitons tolérer. Ceci peut être fait en combinant d'une façon ou d'une autre plusieurs mesures. Plusieurs travaux de la littérature se sont intéressés à la comparaison et la combinaison de mesures de similarité pour l'appariement de chaînes de caractères.

Une comparaison expérimentale d'exemples de mesures de similarité des deux catégories (celles basées sur les séquences de caractères et celles basées sur les sacs de mots) a été proposée dans [Cohen03a, Cohen03b] dans le cadre de couplage d'enregistrements dans une base de données. Ces expérimentations ont favorisé la mesure Tf-idf, resp. une variation pondérée de la mesure de Levenshtein proposée dans [Monge96] dans le cas des mesures basées sur les sacs de mots, resp. les séquences de caractères sur le corpus testé.

Les auteurs dans [Pouliquen06] ont proposé une approche de combinaison de plusieurs mesures de similarité à l'aide d'une simple moyenne sur trois mesures (Tf-idf sur les bi-grammes, tri-grammes sur les chaînes de caractères et les bi-grammes sur seulement les consonnes des chaînes de caractères) dans le but de la reconnaissance des noms de personnes dans des collections multilingues.

Certaines méthodes de combinaison se basent sur la classification supervisée des mesures de similarité qui offre une considération plus fine des caractéristiques des mesures et des données. Par exemple, la méthode donnée dans [Bilenko03a] propose d'apparier des attributs dans des bases de données en combinant différentes mesures à l'aide du classifieur linéaire SVM.

Les auteurs dans [Moreau08] ont proposé une combinaison de plusieurs mesures de similarité à l'aide d'une classification supervisée pour apparier des entités nommées en tolérant les variations dues aux translittérations. L'entrée du classifieur est un ensemble de vecteurs de paramètres créés pour chaque couple d'entités nommées. Ces paramètres représentent les valeurs données par les mesures de similarité et les caractéristiques (telles que les longueurs et les fréquences) d'un couple d'entités nommées. Deux classifieurs (Régression Logique et C4.5) ont été utilisés sur deux corpus composés de paires d'entités nommées (des noms de personnes, d'organisations et de lieux). Les expérimentations ont montré l'intérêt de la classification supervisée pour améliorer les résultats d'appariement d'entités nommées.

3.3 Couplage d'enregistrements

Le couplage d'enregistrements (en anglais, record linkage) se réfère à la tâche de trouver des enregistrements dans un ensemble de données qui font référence à la même entité à travers différentes sources de données (par exemple, les fichiers de données, livres, sites Web, bases de données).

Plusieurs travaux se sont intéressés au problème de couplage d'enregistrements dans la littérature. Une étude bibliographique détaillée est fournie dans [Christen12]. Sur le plan méthodologique, nous pouvons classer ces travaux suivant deux approches : les méthodes déterministes et les méthodes probabilistes. Nous détaillons chacune de ces méthodes dans la suite.

3.3.1 Méthodes déterministes

Les méthodes déterministes, telle que celle proposée dans [Lee00], se basent sur des règles définies par des experts déterminant les conditions de couplage d'une paire d'enregistrements. Ces règles sont généralement dépendantes d'un ensemble de champs pertinents (dits variables de couplage). Si les attributs d'une paire donnée d'enregistrements correspondant aux champs pertinents coïncident, alors cette paire est couplée. Des poids (par exemple, la fréquence du champ) peuvent être attribués à ces champs. Ainsi, si la somme pondérée du nombre d'attributs qui coïncident dépasse un seuil, alors le couplage est retenu. Souvent, les comparaisons sont de type exact et ne tolèrent pas les erreurs de saisie. Ces techniques sont coûteuses en temps car elles nécessitent une implication importante de l'utilisateur pour permettre la génération de transformations spécifiques au domaine et aux données. De plus, elles sont trop rigides (car elles sont dépendantes de la base de données) pour corriger les erreurs évoquées dans l'introduction de cette partie.

D'autres techniques [Church91, Bitton83, Kukich92] ramènent la comparaison à des mesures de similarité entre les attributs pour corriger les variations de représentation, comme la mesure de Levenshtein [Monge97]. En revanche, ces techniques obligent également l'utilisateur à intervenir pour fixer les seuils de mesures.

3.3.2 Méthodes probabilistes

Les méthodes probabilistes consistent à utiliser les statistiques sur les propriétés des variables en commun entre paires d'enregistrements pour calculer la probabilité qu'ils représentent la même entité. Elles peuvent être non supervisées ou supervisées.

3.3.2.1 Méthodes non supervisées

Ces méthodes sont intéressantes quand il n'existe pas de données annotées. Un modèle théorique de couplage d'enregistrements qui offre des méthodes statistiques pour estimer les paramètres de couplage et les taux d'erreurs a été proposé par Fellegi et Sunter dans [Fellegi69]. Ce modèle est décrit ci-après.

Modèle de Fellegi et Sunter

Définition. Le modèle définit les quantités suivantes pour fonctionner :

- M-probabilité : c'est la probabilité de couplage (matching) estimée pour qu'une paire d'attributs quelconque d'un champ donné soit similaire sachant que la paire d'enregistrements correspondante est un vrai couplage (i.e. les deux enregistrements correspondent en réalité) ;
- U-probabilité : c'est la probabilité de non couplage (unmatching) estimée pour qu'une paire d'attributs quelconque d'un champ donné soit similaire sachant que la paire d'enregistrements correspondante est un faux couplage, i.e. la probabilité que les deux enregistrements se couplent par hasard ;
- ratio de couplage d'attributs : le ratio de couplage d'une paire d'attributs correspondants à un champ ayant une probabilité de couplage M et une probabilité de non couplage U est calculé par :
  - pour un accord sur l'attribut : $log(\frac{M}{U})$;
  - pour un désaccord sur l'attribut : $log(\frac{1-M}{1-U})$;
- ratio de couplage d'enregistrements R : le ratio de couplage d'une paire d'enregistrements est calculé par la somme des ratios de couplage des différentes paires d'attributs qui le composent ;
- le couplage d'enregistrements se fait par rapport aux seuils T_λ et T_μ comme suit :

— si $R < T_\lambda$ alors non couplage (rejet) ;
— si $T_\mu < R < T_\lambda$ alors possible couplage (indécis) ;
— si $R > T_\mu$ alors couplage (acceptation).

Problèmes posés. Les problèmes posés par ce modèle se traduisent par 3 questions :
— Comment définir M et U ?
— Comment fixer les seuils T_λ et T_μ ?
— Comment déterminer les mesures de similarité entre champs ?

Interprétation. Pour l'évaluation, nous pouvons définir les vrais et les faux positifs (resp. négatifs) à l'aide de ces considérations de couplage (resp. non couplage) et de liaison (resp. non liaison) (voir Tableau 3.3). En effet, si nous avons décidé de coupler (resp. ne pas coupler) une paire d'enregistrements alors si cette paire est liée (resp. n'est pas liée) en réalité, alors il s'agit d'un vrai positif (resp. vrai négatif) sinon il s'agit d'un faux positif (resp. faux négatif).

Tableau 3.3 – Classification des résultats de couplage

	Couplage	Non couplage
Liaison	vrai positif (TP)	faux positif (FP)
Non liaison	faux négatif (FN)	vrai négatif (TN)

Plusieurs méthodes se sont basées sur le modèle de Fellegi et Sunter pour le couplage d'enregistrements. A titre d'exemple, Winkler a détaillé dans [Winkler99] une technique qui se base sur l'algorithme "espérance-maximisation" pour estimer les paramètres du modèle et optimiser les règles de couplage.

Dans le cas des bases de données volumineuses (par exemple, des dizaines de milliers d'enregistrements), le couplage d'enregistrements consiste à comparer tous les enregistrements deux à deux en calculant le produit cartésien, ce qui n'est pas efficace en temps et en mémoire. Pour résoudre ce problème, les auteurs dans [Hernandez95] proposent une amélioration du modèle de Fellegi et Sunter consistant à séparer la base de données en des groupements d'enregistrements (blocs ou fenêtres), en se basant sur de simples heuristiques qui aident à éliminer les paires d'enregistrements clairement non liés.

3.3.2.2 Méthodes supervisées

Dans ce type de méthodes, les auteurs traitent le problème de couplage d'enregistrements se référant à la même entité comme un problème de classification supervisée. En effet, ils proposent de représenter chaque paire d'enregistrements à l'aide d'un vecteur de caractéristiques décrivant les similarités entre les paires d'attributs. Ces caractéristiques peuvent être binaires (par exemple, les attributs "nom" correspondent), discrètes (par exemple, les n-premiers caractères du "prénom" correspondent) ou continues (par exemple, la mesure de Levenshtein entre les prénoms). Ainsi, la comparaison entre paires d'enregistrements conduit à les ranger dans les 3 classes : couplage, non couplage ou couplage possible. Les classifieurs utilisés déterminent, à partir des données annotées, les conditions de couplage entre les enregistrements, à savoir la détermination des champs pertinents et les seuils de décision pour la similarité entre les attributs.

Nous pouvons citer à titre d'exemple les travaux dans [Bilenko03b] qui proposent une méthode basée sur deux niveaux d'apprentissage supervisé employant un SVM. Le premier niveau représente la comparaison de paires d'attributs en utilisant un apprentissage avec la mesure de Levenshtein (similarité au niveau attribut) et le deuxième niveau représente la comparaison des paires d'enregistrements en utilisant un apprentissage sur la similarité (similarité au niveau enregistrement).

Les auteurs dans [Tejada02] utilisent un apprentissage actif pour sélectionner les paires d'enregistrements les plus informatives. Ainsi, l'utilisateur est sollicité pour annoter ces paires d'enregistrements informatives comme paires liées ou non liées afin d'entraîner les classifieurs. Ces derniers génèrent des règles de classification qui croisent des attributs et des mesures de similarité entre ces attributs.

Les auteurs dans [Churches02] proposent de normaliser certains attributs (ceux qui désignent des noms et des adresses postales) pour effectuer convenablement la comparaison de ces attributs dans le processus de couplage d'enregistrements. Cette normalisation est effectuée en se basant sur un résultat de segmentation des attributs (par exemple un attribut adresse est segmenté en : numéro de voie, type de voie, nom de voie). Pour la segmentation, un modèle de Markov caché est utilisé avec comme observations, les mots de l'attribut, et comme états cachés, les segments. Cette méthode nécessite suffisamment de données annotées sous forme de chaînes de caractères segmentées pour l'apprentissage.

3.4 Conclusion

Dans ce chapitre, nous avons étudié, dans un premier temps, quelques mesures de similarité. D'après cette étude, nous avons constaté que chaque type de mesure est intéressant pour absorber des variations particulières dans la représentation d'attributs. Nous avons cité quelques travaux qui ont étudié la comparaison et la combinaison de ces mesures. Ces travaux ont proclamé l'intérêt de la combinaison dans la comparaison de chaînes de caractères. Dans un second temps, nous avons présenté l'état de l'art sur les méthodes de couplage d'enregistrements. Ces méthodes se basent sur la comparaison d'attributs. Il est alors intéressant d'étudier les manières de combinaison de mesures de similarité pour le couplage d'enregistrements.

Chapitre 4

Méthode proposée

Sommaire

4.1 Introduction

Nous proposons dans ce chapitre une approche de résolution d'entités qui se base sur la comparaison d'attributs pour le couplage d'enregistrements se référant à une même entité. Pour la comparaison d'attributs, nous étudions différentes manières de combinaisons entre les mesures de similarité.

Dans la Section 4.2, nous détaillons notre approche de résolution d'entités. Nous étudions deux méthodes de couplage d'enregistrements dont l'une est supervisée et l'autre est non supervisée. Dans la Section 4.3, nous évaluons l'approche de résolution d'entités sur des bases de données réelles. Nous comparons notamment les deux méthodes de couplage d'enregistrements.

4.2 Méthode proposée

La résolution d'entités se base principalement sur le couplage d'enregistrements se référant à une même entité. Nous proposons d'intégrer, en plus, une phase préliminaire qui permet de réduire le nombre de comparaisons et une phase finale de structuration des enregistrements couplés en entités. Ceci permettra d'améliorer, par la suite, la phase de rapprochement d'entités avec les documents. Ainsi, notre approche est composée de trois étapes :

1. Pré-couplage qui prépare le couplage en regroupant les enregistrements proches dans des blocs.
2. Couplage qui inclut le couplage de paires d'attributs, puis le couplage de paires d'enregistrements.

3. Post-couplage qui structure les enregistrements liés dans un même modèle (modèle entité) et déduit un dictionnaire d'abréviations et de sigles.

Nous reviendrons dans la suite sur chacune de ces étapes du système.

4.2.1 Pré-couplage

Le pré-couplage consiste à segmenter la base de données en des blocs d'enregistrements à l'aide d'un regroupement des enregistrements qui ont des chances de représenter la même entité dans un même bloc. Ainsi, seuls les enregistrements d'un même bloc sont comparés entre eux dans la phase de couplage. Ce regroupement est réalisé à l'aide de clés de regroupement (par exemple, un champ donné, une combinaison de champs, etc.). Les enregistrements dont les attributs, qui correspondent à ces clés, coïncident sont regroupés. Comme exemples de clés, nous pouvons considérer les n-premiers caractères du champ "nom" d'une entreprise ou les n-premiers termes du champ "titre" d'une référence bibliographique.

4.2.2 Couplage d'enregistrements

Le couplage d'enregistrements se base sur la comparaison de leurs paires d'attributs. Un problème important consiste à sélectionner la mesure de similarité adéquate pour cette comparaison. Une fois la mesure sélectionnée, une méthode de prise de décision de couplage, qui tient compte des résultats de comparaison d'attributs, doit être utilisée. Pour ce faire, nous proposons deux méthodes de couplage d'enregistrements :

— la première, basée sur le modèle de Fellegi et Sunter, est non supervisée et propose de calculer un score probabiliste, entre paires d'enregistrements, défini en fonction de la similarité entre leurs paires d'attributs ;
— la deuxième est supervisée et propose d'utiliser deux niveaux de classification pour lier les enregistrements : niveau paire d'attributs et niveau paire d'enregistrements.

Nous commençons par comparer expérimentalement les mesures de similarité dans le but de choisir la plus adéquate. Ensuite, nous détaillons chacune de deux méthodes de couplage d'enregistrements proposées.

4.2.2.1 Étude de mesures de similarité

Nous proposons de comparer les différentes mesures présentées dans la Section 3.2 à l'aide de corpus annotés de paires d'attributs (les attributs de chaque paire correspondent à un même champ). Nous considérons ici tous les champs, de la même manière, comme des chaînes de caractères. Cette étude expérimentale a été réalisée sur deux corpus, où l'un est extrait de la base d'entreprises et l'autre est extrait de la base d'articles scientifiques. Comme le principe est similaire, nous nous contentons de présenter l'étude faite sur le corpus d'entreprises.

Corpus annoté de paires d'attributs. Ce corpus est composé de paires d'attributs extraits à partir de la base d'entreprises et annotées semi-automatiquement (classification préliminaire à l'aide de la mesure de Levenshtein puis une révision manuelle) en paires similaires ou non similaires. Les paires similaires sont les paires qui représentent un même attribut. En effet, nous avons sélectionné 1940 paires d'attributs, de sorte que les deux classes (similaire et non-similaire) soient équilibrées en nombre de paires. Ces paires d'attributs sont sélectionnées manuellement de sorte qu'elles soient représentatives de la base d'entreprises. Le corpus obtenu est décrit dans le Tableau 4.1.

Tableau 4.1 – Corpus annoté de paires d'attributs

Champ d'attributs	# paires d'attributs	#similaires	#non similaires
Nom	680	330	350
Siren	120	60	60
Adresse	270	130	140
Code postal	250	120	130
Ville	130	60	70
Téléphone	250	120	130
E-mail	130	60	70
Web	110	50	60
Total	1940	930	1010

Comparaison des mesures de similarité. Nous considérons les mesures de : Levenshtein, Jaro, Jaro-Winkler, Jaccard, Monge-Elkan, Tf-idf, Soft-tf-idf-lev (la mesure Soft-tf-idf qui utilise la mesure de Levenshtein pour comparer les mots) et Soft-tf-idf-jaro (la mesure Soft-tf-idf qui utilise la mesure de Jaro-Winkler pour comparer les mots). Ces mesures de similarité sont comparées en évaluant le couplage d'attributs, en considérant tous les champs de la même manière comme des chaînes de caractères. Le Rappel (R), la Précision (P) et la F-mesure (F) du couplage d'attributs sont définis respectivement par les équations (4.1), (4.2) et (4.3).

$$R = \frac{\#\ paires\ couplées\ correctes}{\#\ paires\ similaires} \tag{4.1}$$

$$P = \frac{\#\ paires\ couplées\ correctes}{\#\ paires\ couplées} \tag{4.2}$$

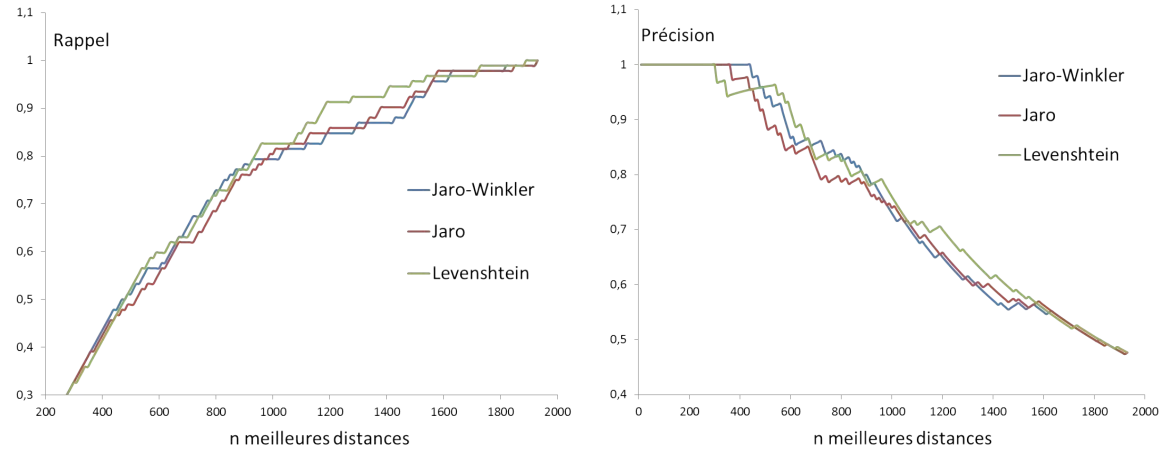
$$F = \frac{2.R.P}{R+P} \tag{4.3}$$

Pour évaluer chaque mesure, nous avons ordonné de manière descendante les paires selon leurs valeurs de similarités et nous avons évalué la Précision et le Rappel selon le nombre des n meilleures valeurs de similarités acceptées. La Figure 4.1 montre les courbes de variation de Rappel et de Précision pour les 8 mesures de similarités (présentées dans le Chapitre précédent) en distinguant leurs 3 catégories. Nous constatons d'après ces courbes qu'aucune de ces mesures ne se démarque. Ceci est expliqué par l'intérêt de chaque mesure pour un type spécifique de chaîne de caractères (chaîne longue, courte, contient un seul ou plusieurs mots, etc.) et pour tolérer des variations spécifiques de représentation d'attributs (substitution de caractères, ordre des mots dans une chaîne, etc.). Par exemple, la mesure de Jaro-Winkler est intéressante pour tolérer les variations de certains codes postaux ("92200" et "92202" dans le Tableau 3.1) et téléphones (par exemple "825082081" et "825082090" dans le Tableau 3.1) puisqu'elle souligne le plus long préfixe commun entre les chaînes de caractères.

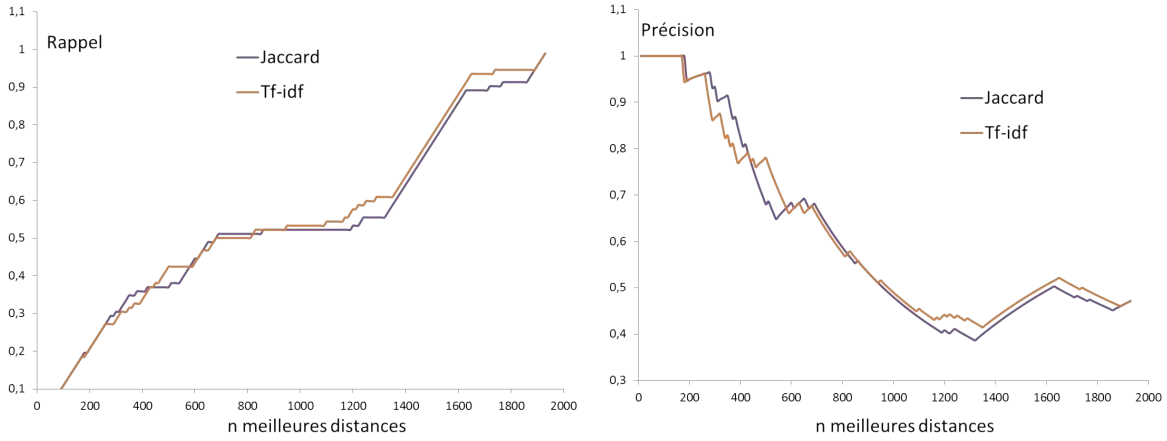
En conséquence, nous avons cherché à étudier les manières de combiner certaines de ces mesures, c'est l'objectif de la section suivante.

4.2.2.2 Couplage non supervisé d'enregistrements

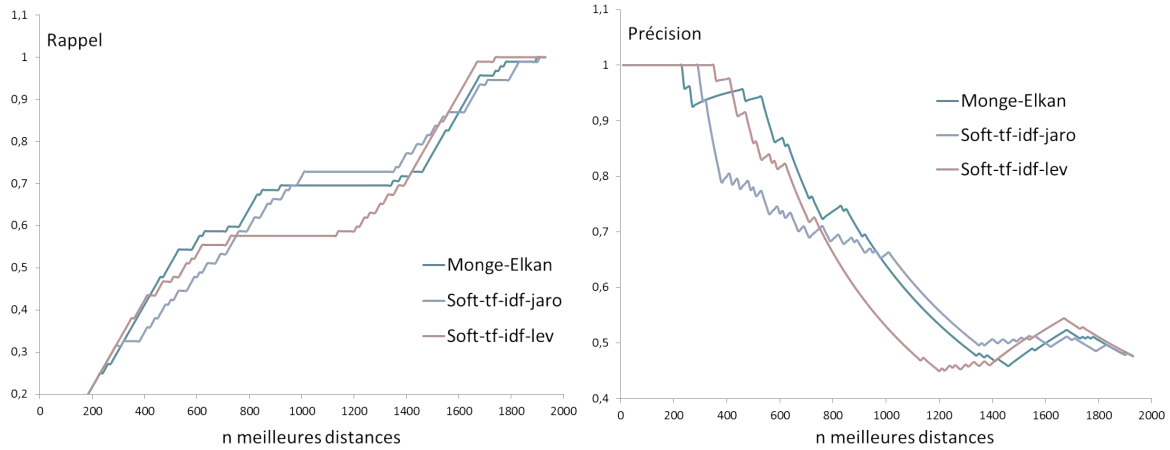
Nous avons choisi une approche probabiliste basée sur le modèle de Fellegi et Sunter [Fellegi69] en intégrant une combinaison de plusieurs mesures de similarité pour la comparaison d'attributs. Le choix de ce modèle est motivé par le fait qu'il s'agit d'une approche facile à mettre en œuvre et qui demande peu de paramètres à fixer. Nous proposons de comparer d'abord les paires d'attributs. Ensuite, nous



(a) Rappel et Précision des mesures basées sur les séquences de caractères



(b) Rappel et Précision des mesures basées sur les sacs de mots



(c) Rappel et Précision des mesures hybrides

Figure 4.1 – Comparaison des mesures de similarité

Tableau 4.2 – Comparaison non supervisée de mesures de similarité

Mesure	Seuil	Résultats		
		P (%)	R (%)	F (%)
Levenshtein	0.47	**82.60**	**78.35**	**80.42**
Jaro	0.75	79.35	75.26	77.25
Jaro-Winkler	0.85	76.09	82.35	79.10
Jaccard	0.20	51.09	63.51	56.63
Monge-Elkan	0.24	66.60	74.39	70.11
Tf-idf	0.37	50.00	66.67	57.14
Soft-tf-idf-lev	0.46	55.43	82.26	66.23
Soft-tf-idf-jaro	0.49	72.83	65.69	69.07

Tableau 4.3 – Combinaison non supervisée de mesures de similarité

Mesure	Seuil	Résultats		
		P (%)	R (%)	F (%)
Moyenne de toutes les mesures	0.48	64.13	81.94	71.95
Moyenne (Levenshtein,Jaro-Winkler)	0.6	83.70	77.78	80.63
Moyenne (Jaro-Winkler,Soft-tf-idf-lev)	0.43	83.70	62.10	71.30
Moyenne (Levenshtein,Jaro-Winkler,Soft-tf-idf-jaro)	0.21	72.83	73.63	73.22
Maximum de toutes les mesures	0.85	79.35	82.95	81.11
Maximum (Jaro-Winkler,Monge-Elkan,Soft-tf-idf-lev)	0.85	79.35	82.95	81.11
Maximum (Monge-Elkan,Soft-tf-idf-lev)	0.46	67.40	82.67	74.25
Maximum (Jaro-Winkler,Monge-Elkan)	0.85	76.09	82.35	79.10
Maximum (Jaro-Winkler,Soft-tf-idf-lev)	**0.85**	**79.35**	**82.95**	**81.11**
Minimum de toutes les mesures	0.19	50.00	63.01	55.76
Minimum (Levenshtein,Jaro-Winkler)	0.47	82.61	78.35	80.42
Minimum (Monge-Elkan,Jaro-Winkler,Soft-tf-idf-lev)	0.33	46.74	84.31	60.14
Minimum (Jaro-Winkler,Soft-tf-idf-lev)	0.46	54.35	81.97	65.36

couplons les paires d'enregistrements qui possèdent suffisamment d'attributs similaires en se basant sur un score de couplage.

Comparaison d'attributs. Pour la comparaison entre paires d'attributs, nous avons étudié expérimentalement les mesures de similarité sur le corpus présenté dans le Tableau 4.1. Nous avons comparé ces différentes mesures individuellement en utilisant le meilleur seuil (celui qui maximise la F-mesure), puis nous les avons combinées à l'aide des trois fonctions : Maximum, Minimum et Moyenne. Les résultats des comparaisons individuelles et des combinaisons les plus intéressantes (en terme de taux de classification) sont illustrés respectivement dans le Tableau 4.2 et le Tableau 4.3. D'après ces résultats, nous constatons que la comparaison individuelle favorise la mesure de Levenshtein. De plus, nous constatons l'intérêt de la combinaison et notamment le Maximum entre les mesures de Jaro-Winkler et Soft-tf-idf-lev. En effet, cette combinaison maximise la F-mesure en utilisant le nombre minimum de mesures. Cette combinaison a également donné les meilleurs résultats sur le corpus extrait de la base d'articles scientifiques. Elle est ainsi retenue pour cette méthode.

Comparaison d'enregistrements. La comparaison d'une paire d'enregistrements suit le modèle de Fellegi et Sunter. Elle est faite par rapport à un score probabiliste calculé en fonction de la similarité

entre les paires d'attributs et les probabilités de couplage estimées pour les champs de la base de données. Cette comparaison est décrite par l'Algorithme 1 où nous notons :

- E est la base de données définie par n enregistrements $\{r_i\}$ et m champs $\{c_j\}$; chaque enregistrement r_i est composé de m attributs, où l'attribut $r_i.c_j$ correspond au champ c_j ;
- q est le nombre de blocs $\{B_k\}$ regroupant les enregistrements de E ;
- $|B_k|$ est la taille (i.e. nombre d'enregistrements) du bloc B_k ;
- $sim(r_1.c, r_2.c)$ est la fonction Maximum des mesures de Jaro-Winkler et Soft-tf-idf-lev entre deux attributs $r_1.c$ et $r_2.c$;
- $M(c)$ et $U(c)$ représentent respectivement les probabilités M et U estimées pour un champ c ;
- $Ratio$ est le ratio de couplage calculé pour chaque paire d'enregistrements ;
- T_λ et T_μ représentent respectivement les seuils pour la mesure de similarité entre attributs sim et le ratio de couplage $Ratio$. Ces seuils sont fixés expérimentalement à l'aide d'une base de validation.

L'algorithme consiste à comparer deux à deux les paires d'enregistrements dans chaque bloc. Cette comparaison se base sur un ratio de couplage $Ratio$ défini en fonction de la similarité entre les paires d'attributs d'un même champ (si sim dépasse le seuil T_λ alors les attributs sont considérés comme similaires) et les probabilités M et U. Dans le cas où le ratio de couplage dépasse le seuil T_μ, les enregistrements sont considérés se référant à la même entité et donc liés.

Algorithme 1 Couplage d'enregistrements

Entrées: base de données : E, seuils : T_λ, T_μ
Sortie: base de données liée : E'

pour $k = 1 \rightarrow q$ **faire** // parcours des blocs
  pour $i = 1 \rightarrow |B_k| - 1$ **faire** // parcours des enregistrements du bloc courant
    pour $j = i + 1 \rightarrow |B_k|$ **faire** // parcours des enregistrements du bloc courant
      $Ratio = 0$
      pour $p = 1 \rightarrow m$ **faire** // parcours des champs de la paire d'enregistrements courante
        $s = sim(r_i.c_p, r_j.c_p)$
        si $s \geq T_\lambda$ **alors**
          $Ratio+ = \log(\frac{M(p)}{U(p)}) * s$
        sinon
          $Ratio+ = \log(\frac{1-M(p)}{1-U(p)})$
        fin si
      fin pour
      si $Ratio \geq T_\mu$ **alors**
        $E' = link(E, r_i, r_j)$
      fin si
    fin pour
  fin pour
fin pour

Estimation des probabilités. La probabilité de couplage M d'un champ donné de la base de données (appelée aussi sensibilité) est estimée comme :

$$M = 1 - taux\ d'erreurs$$

où $taux\ d'erreurs$ représente le pourcentage qu'une paire d'attributs soit en désaccord (non similaires) sachant que la paire d'enregistrements correspondante est un vrai couplage. Ce taux est fixé expérimentalement pour chaque base de données à l'aide d'une base de validation.

La probabilité de non couplage U d'un champ donné (appelée aussi spécificité) est estimée comme :

$$U = \frac{1}{\#valeurs\ distinctes}$$

où $\#valeurs\ distinctes$ représente le nombre de valeurs d'attributs distinctes d'un champ donné dans tous les enregistrements de la base de données.

Complexité. Soient n le nombre d'enregistrements dans la base de données et q le nombre de blocs extraits. Si nous supposons que le nombre d'enregistrements par bloc est $\frac{n}{q}$, la complexité de l'algorithme de couplage d'enregistrements est de l'ordre de $\frac{2^n}{2^q}.q.m$ (comparée à $2^n.m$, sans l'étape de pré-couplage) en nombre de comparaisons d'attributs. Nous remarquons l'intérêt du regroupement d'enregistrements en bloc à réduire considérablement la complexité du processus de couplage d'enregistrements (car $2^q >> q$ pour q suffisamment grand).

Exemple. Soient les trois enregistrements E1, E2 et E3 représentés dans la Figure 4.2 (a). Nous avons comparé les enregistrements E1, E2 (se référant à une même entité dans la vérité) et E1, E3 (ne se référant pas à une même entité dans la vérité) en utilisant l'Algorithme 1. Les résultats de cette comparaison sont montrés dans la Figure 4.2 (b), où le ratio de couplage donné par la comparaison entre E1 et E2, resp. E1 et E3 est supérieur, resp. inférieur au seuil (fixé à 10 ici). Ce ratio de couplage a été calculé en fonction de la similarité entre les paires d'attributs (le maximum entre les mesures Jaro-Winker et Soft-tf-idf-lev en utilisant le seuil 0.85) à l'aide des valeurs de probabilités M et U estimés pour les champs de la base d'entreprises. Les enregistrements E1 et E2, resp. E1 et E3 sont donc couplés, resp. non couplés.

Enreg.	Nom	Adresse	Code postal	Ville	Téléphone
E1	Xerox financial sces	120 av. Charles de gaulle	92200	Neuilly sur seine	0825082081
E2	Xerox financial services	120 av. Charles de gaulle	92202	Neuilly	0825082090
E3	Xerox	24 rue de la chapelle	97122	Baie Mahault	0825082081

(a) Échantillon de deux enregistrements de la base d'entreprises

	Sim (Nom)	Sim (Adresse)	Sim (Code postal)	Sim (Ville)	Sim (Téléphone)	Ratio	Résultat
(E1, E2)	0.96 (similaires)	1.00 (similaires)	0.92 (similaires)	0.88 (similaires)	0.87 (non-similaires)	15.49	couplage
(E1, E3)	0.85 (similaires)	0.68 (non-similaires)	0.64 (non-similaires)	0.44 (non-similaires)	1.00 (similaires)	5.29	non-couplage

(b) Résultat de couplage non supervisé d'enregistrements

Figure 4.2 – Exemple de couplage non supervisé d'enregistrements.

4.2.2.3 Couplage supervisé d'enregistrements

Cette deuxième méthode de couplage d'enregistrements fonctionne par apprentissage. Elle consiste à réaliser deux niveaux d'apprentissage, où le premier niveau concerne les attributs et le deuxième niveau concerne les enregistrements. Nous détaillons ces deux niveaux dans la suite.

Tableau 4.4 – Apprentissage supervisé de mesures de similarité

Mesures	Résultats								
	C4.5			SVM			Bayésien naïf		
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Levenshtein	76.40	75.80	76.10	**79.60**	**79.40**	**79.40**	78.90	78.90	78.90
Levenshtein + carac. intrinsèques	75.80	75.80	75.80	77.40	77.30	77.35	78.40	78.30	78.35
Jaro	74.80	74.70	74.75	77.40	77.30	77.35	75.80	75.30	75.55
Jaro + carac. intrinsèques	70.10	70.00	70.05	76.10	75.80	75.95	77.30	76.80	77.05
Jaro-winkler	79.40	79.40	79.40	69.50	66.00	67.70	70.80	66.50	68.58
Jaro-winkler + carac. intrinsèques	72.20	72.20	72.20	75.40	73.70	74.54	75.80	74.20	74.99
Jaccard	76.00	65.50	70.36	69.90	65.50	67.63	65.40	63.90	64.64
Jaccard + carac. intrinsèques	68.50	68.00	68.25	66.30	66.00	66.15	65.30	64.40	64.85
Monge-Elkan	80.70	76.30	78.44	80.40	75.80	78.03	78.80	75.80	77.27
Monge-Elkan + carac. intrinsèques	77.70	77.30	77.50	76.80	74.20	75.48	77.20	74.70	75.93
Tf-idf	75.40	64.40	69.47	61.50	61.30	61.40	64.10	63.40	63.75
Tf-idf + carac. intrinsèques	65.40	64.90	65.15	69.10	68.90	69.00	61.30	61.00	61.15
Soft-tf-idf-lev	79.80	72.20	75.81	67.90	67.50	67.70	71.70	70.60	71.15
Sotftfidf-lev + carac. intrinsèques	77.60	75.80	76.69	73.20	73.20	73.20	66.60	66.50	66.55
Soft-tf-idf-jaro	66.00	64.30	65.14	68.80	68.60	68.70	68.00	68.00	68.00
Soft-tf-idf-jaro + carac. intrinsèques	78.40	78.40	78.40	65.90	66.00	65.95	62.80	62.90	62.85

Comparaison d'attributs. Pour sélectionner les mesures de similarité et le classifeur qui seront utilisés pour la comparaison de paires d'attributs, nous nous appuyons sur une étude expérimentale des différentes possibilités. Pour ce faire, nous utilisons le même corpus présenté dans le Tableau 4.1 et nous utilisons comme descripteurs, en plus des similarités entre les attributs, les 3 caractéristiques intrinsèques suivantes : nombre de mots, nombre de caractères et le champ correspondant.

Nous comparons plusieurs combinaisons de ces descripteurs à l'aide des trois classifieurs : C4.5 [Quinlan93] qui propose une classification en arbre de décision, SVM qui propose une classification binaire et Bayésien naïf qui propose une classsification bayésienne. Nous avons réalisé les tests à l'aide du logiciel Weka [Witten05].

Les résultats de comparaison en Précision (P), Rappel (R) et F-mesure (F) des mesures de similarité considérées individuellement (seules et avec caractéristiques intrinsèques) sont montrés dans le Tableau 4.4. Nous montrons également dans le Tableau 4.5 les résultats des plus importantes combinaisons (sans et avec caractéristiques intrinsèques) de ces mesures.

Ces résultats montrent l'intérêt de la classification supervisée par rapport à la classification non supervisée (dans le Tableau 4.2 et le Tableau 4.3) pour la comparaison des attributs. De plus, nous constatons que le classifieur C4.5 est plus performant que les deux autres pour la plupart des descripteurs utilisés, que la combinaison des mesures est généralement plus intéressante que les différentes mesures indi-

Tableau 4.5 – Combinaison supervisée de mesures de similarité

Mesures	Résultats								
	C4.5			SVM			Bayésien naïf		
	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)	P (%)	R (%)	F (%)
Toutes les mesures + caractéristiques intrinsèques	83.50	83.50	83.50	82.30	82.00	82.15	76.00	75.80	75.90
Toutes les mesures	85.60	85.60	85.60	83.50	82.50	83.00	74.9	74.70	74.80
Toutes les mesures + noms de champs	83.00	83.00	83.00	83.50	83.00	83.25	73.9	73.70	73.80
Toutes les mesures (sauf Soft-tf-idf-lev)	85.60	85.60	85.60	83.90	83.00	83.45	75.80	75.80	75.80
Toutes les mesures (sauf Soft-tf-idf-jaro) + carac. intrinsèques	83.00	86.40	84.67	82.50	82.50	82.50	78.2	77.8	78.00
Toutes les mesures (sauf Soft-tf-idf-jaro)	86.80	86.60	86.70	82.90	80.90	81.89	78.70	78.40	78.55
Toutes les mesures (sauf Soft-tf-idf-jaro, Tf-idf)	**87.30**	**87.30**	**87.20**	81.10	79.40	80.24	77.10	76.80	76.95
Toutes les mesures (sauf Soft-tf-idf-jaro, Tf-idf) + carac. intrinsèques	82.50	82.50	82.50	81.50	81.40	81.45	77.60	77.30	77.45
Soft-tf-idf-jaro + Jaccard + Jaro-Winkler + carac. intrinsèques	77.40	77.30	77.35	74.00	73.70	73.85	70.80	70.10	70.45
Soft-tf-idf-jaro + Jaccard + Jaro-Winkler	77.90	77.80	77.85	69.70	69.10	69.40	69	68.60	68.80
Soft-tf-idf-lev + Jaccard + Jaro-Winkler + carac. intrinsèques	76.00	75.80	75.90	71.60	71.10	71.35	70.70	70.60	70.65
Soft-tf-idf-jaro + Jaccard + Jaro-Winkler	77.40	77.30	77.35	68.80	67.00	67.89	67.30	67.00	67.15
Soft-tf-idf-jaro + Jaccard + Levenshtein + carac. intrinsèques	82.80	82.50	82.65	80.40	80.40	80.40	79.90	79.90	79.90
Soft-tf-idf-jaro + Jaccard + Levenshtein	78.90	78.80	78.85	79.10	78.90	79.00	79.60	79.40	79.50
Monge-Elkan + Soft-tf-idf-jaro + Soft-tf-idf-lev + carac. intrinsèques	80.00	79.90	79.95	73.00	72.70	72.85	75.90	75.30	75.60
Monge-Elkan + Soft-tf-idf-jaro + Soft-tf-idf-lev	78.80	75.80	77.27	76.10	75.80	75.95	76.40	75.30	75.85

viduellement et que l'intégration des caractéristiques intrinsèques avec les mesures n'a pas beaucoup d'intérêt.

Comme la combinaison de toutes les mesures sauf Soft-tf-idf-jaro et Tf-idf à l'aide du classifieur C4.5 est la plus intéressante, nous la retenons pour la comparaison d'attributs dans la suite. Nous avons trouvé le même résultat sur le corpus extrait de la base d'articles scientifiques.

Comparaison d'enregistrements. Pour comparer une paire d'enregistrements (contenant chacun m attributs), nous la représentons par un vecteur de descripteurs de dimension $7 * m$, où chaque paire d'attributs est représentée par 7 descripteurs. Les 6 premiers correspondent aux valeurs des mesures de similarité retenues pour la comparaison d'attributs, et le dernier correspond au résultat de cette comparaison (1 si les attributs sont similaires et 0 sinon).

Nous utilisons à nouveau un classifeur C4.5 qui permet d'affecter, à une paire d'enregistrements, la classe 1 si ces enregistrements sont couplés et la classe 0 sinon.

Nous fixons les paramètres du classifeur C4.5 : « ConfidenceFactor » (qui contrôle le niveau d'élagage), « minNumObj » (qui représente le nombre minimum d'exemples par feuille) et « unpruned » (qui fait de l'élagage s'il est mis à faux) expérimentalement (en essayant plusieurs combinaisons de valeurs et en gardant la combinaison qui maxime les résultats de comparaison).

Nous appliquons la relation de transitivité dans le couplage d'enregistrements (si les enregistrements r_i et r_j sont liés et que r_j et r_k le sont aussi, alors r_i et r_k sont liés).

Exemple. Nous avons comparé les enregistrements E1, E2 et E1, E3 représentés dans la Figure 4.2 (a) en utilisant la méthode de couplage d'enregistrements supervisée. Les valeurs de descripteurs (les 6 mesures de similarités + résultat de comparaison d'attributs) fournies au classifieur sont montrées dans la Figure 4.3. Ce classifieur a donné un résultat correct en attribuant la classe 1 à la paire d'enregistrements (E1, E2) et la classe 0 à la paire (E1, E3).

	Desc1-7 (Nom)	Desc1-7 (Adresse)	Desc1-7 (Code postal)	Desc1-7 (Ville)	Desc1-7 (Téléphone)
(E1, E2)	(0.83, 0.93, 0.96, 0.50, 0.50, 0.04, 1)	(1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1)	(0.80, 0.87, 0.92, 0.00, 0.80, 0.80, 1)	(0.41, 0.80, 0.88, 0.33, 0.33, 0.49, 1)	(0.80, 0.87, 0.92, 0.00, 0.80, 0.00, 0)
(E1, E3)	(0.25, 0.75, 0.85, 0.33, 0.33, 0.03, 1)	(0.32, 0.68, 0.68, 0.11, 0.11, 0.03, 0)	(0.20, 0.60, 0.64, 0.00, 0.00, 0.00, 0)	(0.12, 0.44, 0.44, 0.00, 0.00, 0.00, 0)	(1.00, 1.00, 1.00, 1.00, 1.00, 1.00, 1)

Figure 4.3 – Exemple de couplage supervisé d'enregistrements.

4.2.3 Post-couplage

Après couplage des enregistrements se référant à la même entité, nous proposons de les structurer de manière factorisée dans un modèle entité et de construire un dictionnaire d'abréviations et de sigles.

4.2.3.1 Modèle entité

Le modèle entité est une structure de données qui organise de manière non redondante et standardisée les entités de la base de données. En effet, il est composé d'autant d'arbres que de nombre d'entités. Une entité est représentée par un arbre de 3 niveaux qui rassemble et factorise les différentes représentations de ses attributs (voir l'exemple de la Figure 4.4). Le premier niveau représente l'entité, le deuxième niveau représente les champs, et le troisième niveau représente les différentes variations d'attributs correspondant à chaque champ.

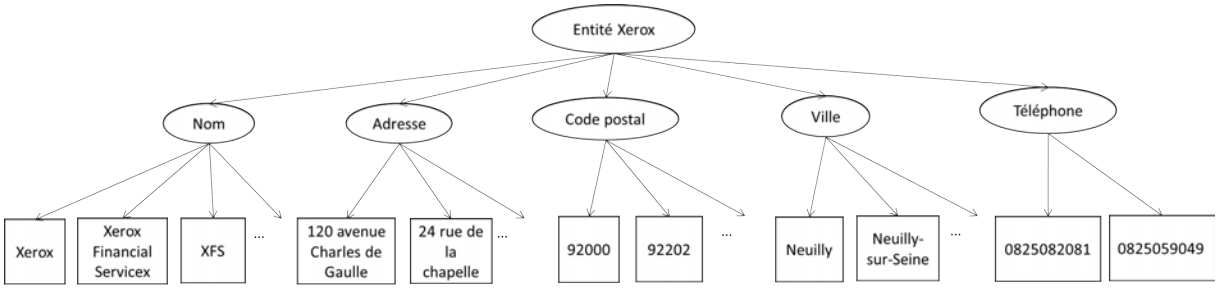


Figure 4.4 – Exemple de représentation d'enregistrements liés dans le modèle entité

4.2.3.2 Dictionnaire d'abréviations et de sigles

A partir des variantes de représentation d'attributs, nous proposons de créer un dictionnaire d'abréviations et de sigles. Ce dictionnaire est construit en se basant sur des règles syntaxiques appliquées sur les champs nominaux permettant d'extraire des noms équivalents. A titre d'exemple, nous utilisons des règles d'extraction de type de la voie à partir du champ "adresse" pour extraire les noms "Boulevard", "Bld", "Bd", etc.

Voici, par exemple, le dictionnaire créé à partir des 8 enregistrements se référant à l'entité montrée dans le Tableau 3.1 :

— avenue, av, av. ;
— Charles de Gaulle, CDG ;
— services, sces.

4.3 Évaluations

Les expérimentations sont faites sur deux bases de données : la base d'entreprises et la base d'articles scientifiques. La base des matériaux n'a pas été traitée car elle ne contient pas de redondances d'enregistrements.

Pour chacune des deux bases, nous disposons d'une table de vérité, annotée manuellement, qui associe aux identifiants d'enregistrements se référant à la même entité, un même index.

4.3.1 Pré-couplage

Pour l'étape de regroupement d'enregistrements en blocs, nous avons testé trois clés sur la base d'entreprises :

— clé 1 = le champ "nom" ;
— clé 2 = le premier terme du champ "nom" ;
— clé 3 = la concaténation du premier terme du champ "nom" avec le champ "code postal" (nous avons aussi considéré les enregistrements qui ont la même valeur du premier terme du champ "nom" où la valeur du champ "code postal" est vide).

Le Tableau 4.6 montre les résultats du regroupement en blocs. La première colonne montre le nombre de blocs obtenus en regroupant les enregistrements par la même clé. La première ligne ne fait pas de regroupement, elle considère l'ensemble des enregistrements (2300 enregistrements) dans un seul bloc. La deuxième ligne fait des regroupements en utilisant comme clé le champ "nom". Nous obtenons 953 blocs, ce qui correspond à une comparaison de 48772 paires d'enregistrements, parmi lesquels nous détectons 11599 enregistrements redondants. La troisième ligne, resp. la quatrième utilise la clé 2, resp. la clé 3 avec les nombres correspondants d'enregistrements comparés et redondants. Nous avons retenu la

Tableau 4.6 – Résultat du regroupement en blocs sur la base d'entreprises

	# blocs	# paires comparées	# paires couplées
Sans regroupement	1	2643850	11750
Clé 1 ("code postal")	953	48772	11599
Clé 2 (premier terme du "nom")	498	104182	11746
Clé 3 (concaténation de la clé 1 et la clé 2)	1114	92411	11745

troisième puisqu'elle réduit considérablement le nombre d'enregistrements comparés avec la deuxième, avec presque le même nombre d'enregistrements redondants, d'où une économie importante de temps d'exécution. Nous avons écarté la première car même si le nombre de paires comparées est moins important, le nombre de redondances détectées est beaucoup plus faible.

De la même façon, nous avons testé trois clés sur la base d'articles scientifiques :

— clé 1 = les 3 premiers termes du champ "titre" ;
— clé 2 = l'année du champ "date de publication" (extraction de l'année à partir de la date à l'aide d'expressions régulières) ;
— clé 3 = la concaténation des 3 premiers termes du champ "titre" et l'année du champ "date de publication".

En suivant la même logique des expérimentations sur les clés de la base d'entreprises, nous avons retenu la troisième clé.

Dans la suite de ces expérimentations, nous considérons comme clé de la base d'entreprises, resp. d'articles scientifiques, la concaténation du premier terme de l'attribut du champ "nom" avec l'attribut du champ "code postal", resp. la concaténation des 3 premiers termes du champ titre et l'année du champ "date de publication".

4.3.2 Couplage

Une paire d'enregistrements qui correspondent est définie par une paire d'enregistrements se référant à la même entité dans la vérité. Une paire d'enregistrements couplée est définie par une paire d'enregistrements qui a été couplée par notre système. Le Rappel, la Précision et la F-mesure sont définis respectivement par les équations (4.4), (4.5) et (4.6).

$$R = \frac{\#\ paires\ couplées\ correctes}{\#\ paires\ qui\ correspondent} \tag{4.4}$$

$$P = \frac{\#\ paires\ couplées\ correctes}{\#\ paires\ liées} \tag{4.5}$$

$$F = \frac{2.R.P}{R+P} \tag{4.6}$$

Nous proposons d'expérimenter et comparer en Rappel, Précision et F-mesure les deux méthodes de couplage d'enregistrements (couplage non supervisé et couplage supervisé) sur les deux bases de données d'étude.

4.3.2.1 Couplage non supervisé

Le seuil du ratio de couplage est appris sur une base de validation annotée. En effet, nous avons sélectionné aléatoirement 2300 enregistrements de la base d'entreprises et 3000 enregistrements de la base d'articles scientifiques puis testé notre système de résolution d'entités sur ces enregistrements.

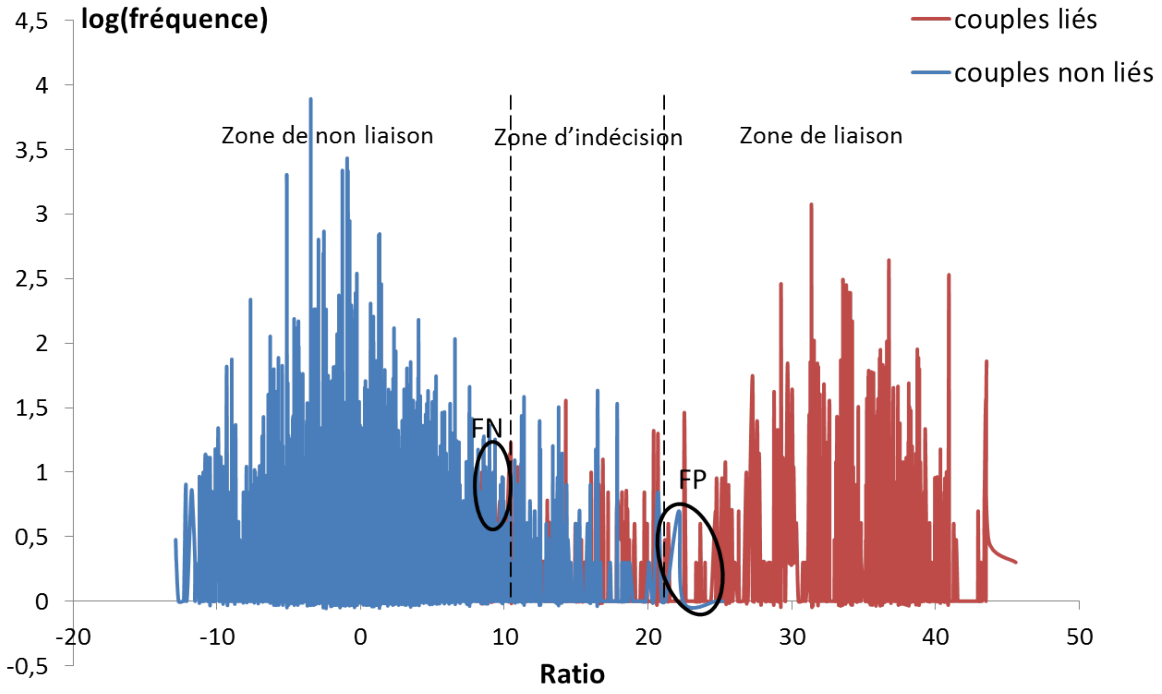


Figure 4.5 – Variation de la fréquence des paires d'enregistrements comparés en fonction du ratio de couplage sur la base d'entreprises ; les traits verticaux séparant les trois zones et les cercles en entourent, à gauche, les faux négatifs, et à droite, les faux positifs.

Nous avons mesuré la fréquence des paires d'enregistrements qui correspondent en fonction des différentes valeurs du ratio de couplage. Nous montrons, à titre d'exemple, dans la Figure 4.5 le résultat obtenu sur la base d'entreprises : la courbe en bleu montre les paires d'enregistrements liées et celle en rouge montre les paires d'enregistrements non liées. Cette figure montre les trois zones du modèle probabiliste : la zone de non liaison, la zone de liaison et la zone d'indécision. Nous remarquons la présence de faux positifs (courbe bleue encerclée dans la zone de liaison) dans la zone de liaison et de faux négatifs (courbe rouge encerclée dans la zone de non liaison) dans la zone de non liaison.

Dans la Figure 4.6, nous présentons la courbe de variation des valeurs de la Précision, du Rappel et de la F-mesure en fonction du seuil sur la base d'entreprises en utilisant les données de validation. Nous retenons la valeur du seuil (13 dans la Figure 4.6) qui maximise la F-mesure (environ 90% dans la Figure 4.6). Cette valeur correspond à une Précision d'environ 93% et un Rappel d'environ 87%. De la même façon, le ratio de couplage de la base d'articles scientifiques est fixé à 19.

Comme proposé dans la Section 4.2.2.2, nous avons estimé les probabilités M pour les champs de chaque base de données à partir des bases de validation contenant 2300, resp. 3000 enregistrements annotés dans la base d'entreprises, resp. d'articles scientifiques. Quand aux probabilités U, nous avons utilisé les bases du corpus d'évaluation pour les estimer automatiquement (inversement proportionnelles au nombre de valeurs distinctes de chaque champ).

Les résultats obtenus en utilisant ces probabilités estimées et ces ratios de couplage fixés sont montrés dans le Tableau 4.7. Ces résultats confirment l'intérêt du Maximum entre les mesures de Jaro-Winkler et Soft-tf-idf-lev pour mesurer la similarité entre les attributs (cette combinaison maximise la F-mesure en utilisant le nombre minimum de mesures). Le nombre d'entités trouvées en utilisant cette combinaison est d'environ 142310, resp. 260812 entités pour la base d'entreprises, resp. base d'articles scientifiques.

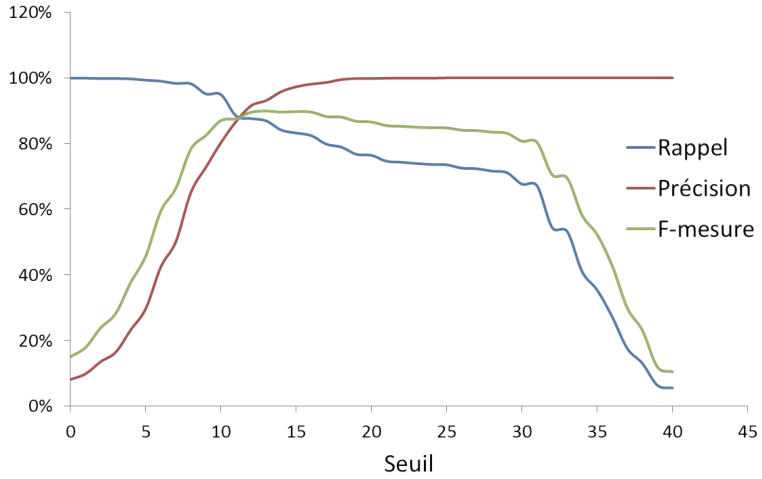


Figure 4.6 – Variation du Rappel, de la Précision et de la F-mesure en fonction du seuil de ratio de couplage sur la base d'entreprises.

Tableau 4.7 – Résultat de couplage non supervisé d'enregistrements

Similarité	Entreprises			Articles scientifiques		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
Levenshtein	81.50	85.95	83.67	87.30	89.10	88.19
Max toutes les mesures	85.67	88.31	86.97	90, 45	92, 20	91, 32
Max(JaroWinkler,SoftTfidfLev)	**85.67**	**88.31**	**86.97**	**90,45**	**92,20**	**91,32**

4.3.2.2 Couplage supervisé

Cette méthode de couplage d'enregistrements se base sur 2 niveaux d'apprentissage : niveau d'attributs et niveau d'enregistrements.

Pour la base d'entreprises, nous utilisons :

— pour l'apprentissage au niveau des attributs, le corpus annoté composé de 1940 paires d'attributs, représenté dans le Tableau 4.1 ;

— pour l'apprentissage au niveau des enregistrements, nous utilisons le même échantillon des 2300 enregistrements annotés (utilisé dans la section précédente).

Pour la base d'articles scientifiques, nous utilisons :

— pour l'apprentissage au niveau des attributs, un corpus annoté composé de 4000 paires d'attributs ;

— pour l'apprentissage au niveau des enregistrements, nous utilisons le même échantillon des 3000 enregistrements annotés (utilisé dans la section précédente).

Nous avons évalué notre système de couplage d'enregistrements sur les deux bases de données. Les résultats sont montrés dans le Tableau 4.8. Ces résultats confirment l'intérêt de la combinaison de toutes les mesures sauf Soft-tf-idf-jaro et Tf-idf en utilisant le classifieur C4.5 pour mesurer la similarité entre les attributs. En utilisant cette combinaison, le nombre d'entités trouvées est d'environ 135043, resp. 250490 entités pour la base d'entreprises, resp. base d'articles scientifiques.

Pour montrer l'intérêt de notre approche basée sur deux niveaux d'apprentissage (nommée : approche supervisée 3) comme détaillé dans la section 4.2.2.3, nous l'avons comparé avec :

— une approche de couplage (nommée : approche supervisée 1) avec un seul niveau d'apprentissage (c.à.d. niveau d'enregistrements) en représentant chaque paire d'enregistrements par un vecteur de descripteurs qui contient la liste des distances de similarité entre les paires d'attributs.

Tableau 4.8 – Résultat de couplage supervisé d'enregistrements

Similarité	Entreprises			Articles scientifiques		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
Levenshtein	90.32	92.10	91.20	93.30	92.95	93.12
Toutes	91.20	93.50	92.34	94.90	94.56	94.73
Toutes sauf Soft-tf-idf-jaro et Tf-idf	**91.20**	**93.50**	**92.34**	**94.90**	**94.56**	**94.73**

Tableau 4.9 – Résultat de couplage d'enregistrements

Approche	Entreprises			Articles scientifiques		
	R (%)	P (%)	F (%)	R (%)	P (%)	F (%)
non supervisée [Fellegi69]	85.67	88.31	86.97	90, 45	92, 20	91, 32
supervisée 1	89.30	91.10	90.19	91.40	92.75	92.07
supervisée 2	90.90	92.00	91.45	94.78	93.63	94.20
supervisée 3	**91.20**	**93.50**	**92.34**	**94.90**	**94.56**	**94.73**

— une approche de couplage (nommée : approche supervisée 2) à deux niveaux d'apprentissage en représentant chaque paire d'enregistrements par un vecteur de descripteurs qui contient uniquement la liste des résultats binaires de comparaisons de paires d'attributs.

Les résultats de comparaison des 3 méthodes supervisées et de la méthode non supervisée (basée sur le modèle de Fellegi et Sunter [Fellegi69]), sur la base d'entreprises et la base d'articles scientifiques, sont illustrés dans le Tableau 4.9. Ces résultats montrent l'intérêt de l'apprentissage supervisé et notamment l'apprentissage supervisé à deux niveaux où nous intégrons les résultats de comparaisons d'attributs et nous intégrons également leurs distances de similarité dans la description des paires d'enregistrements.

Les cas d'erreurs sont essentiellement liés à : plusieurs attributs manquants, plusieurs erreurs de saisie dans l'enregistrement ou des erreurs de segmentation de la base de données en blocs. La dernière cause consiste à placer des enregistrements se référant à une même entité dans des blocs différents, ce qui écarte la possibilité de les comparer et de les coupler.

4.4 Conclusion

Dans ce chapitre, nous avons étudié l'étape de résolution d'entités qui a comme but de représenter les enregistrements d'une base de données dans un modèle entité factorisé. Cette étape se base sur le couplage d'enregistrements qui inclue la comparaison de paires d'attributs et la comparaison de paires d'enregistrements. Pour ce faire, nous avons proposé deux approches, où l'une est non supervisée et l'autre est supervisée. La première se base sur une approche de l'état de l'art en intégrant la combinaison de mesures de similarité dans la comparaison d'attributs. La deuxième se base sur l'apprentissage de la similarité entre attributs et entre enregistrements en combinant les mesures de similarité.

Nous avons expérimenté ces deux approches sur les bases d'entreprises et de méta-données d'articles scientifiques. Ces expérimentations ont favorisé l'approche supervisée et les résultats obtenus sont respectivement 93.50% et 94.56% en Rappel et 91.20% et 94.90% en Précision.

La résolution d'entités représente un pré-traitement primordial pour la reconnaissance d'entités dans les documents, guidée par le modèle entité construit.

Deuxième partie

Reconnaissance d’entités

Introduction

Le processus de Reconnaissance d'Entités (RE) inclue la tâche d'extraction d'entités nommées, où une entité nommée représente un ensemble de mots supposés apparaître regroupés dans le document et faisant référence à une entité particulière. De plus, ce processus consiste à regrouper ces entités nommées même lorsqu'elles sont éparpillées arbitrairement dans le document et à identifier l'entité pertinente, dans le modèle entité, qui les relie. Plusieurs entités pertinentes peuvent être mentionnées dans un document donné.

Cette partie est organisée en trois chapitres. Dans le Chapitre 5, nous présentons l'état de l'art sur les méthodes de RE. Dans le Chapitre 6, nous proposons deux méthodes de RE basées sur le rapprochement par le contenu. Dans le Chapitre 7, nous proposons une méthode de RE basée sur le rapprochement par les graphes de structures locales, qui combine le contenu avec la structure du document.

Chapitre 5

État de l'art

Sommaire

5.1 Introduction

Dans ce chapitre, nous explorons les méthodes de RE ainsi que des méthodes connexes proposées dans la littérature. Nous distinguons le cas de documents du Web (essentiellement du texte ASCII) et le cas de documents océrisés. Dans la Section 5.2, nous étudions la RE dans les documents du Web. Dans la Section 5.3, nous nous intéressons aux documents océrisés. Nous mettons également la lumière sur les traitements faits pour pallier les problèmes liés à ces documents, notamment la correction des erreurs d'OCR.

5.2 Reconnaissance d'entités dans les documents du Web

Dans la littérature, les méthodes de RE peuvent être catégorisées en quatre types : les méthodes à base de règles, les méthodes probabilistes, les méthodes hybrides et les méthodes à base de données. Nous détaillons ces méthodes dans ce qui suit.

5.2.1 Méthodes à base de règles

Appelées aussi méthodes à base de grammaire ou à base de contexte, ces méthodes sont très utilisées dans des applications de traitement automatique des langues. Elles nécessitent une reconnaissance grammaticale (et éventuellement sémantique) explicite du texte. Elles proposent d'analyser le texte et

d'étiqueter des mots ou des ensembles de mots par des entités nommées (telles que personne, organisation, date, etc.). Pour ce faire, des expressions régulières et des règles grammaticales (et éventuellement des ontologies) sont manuellement prédéfinies par des experts, comme par exemple, les règles suivantes :

— Prénom + Mot en majuscule = Personne
— Mot inconnu + « Inc. » = Organisation
— Deux chiffres + deux points + deux chiffres = Heure

Ces méthodes peuvent aussi utiliser des dictionnaires d'attributs d'entités pour accélérer le processus d'étiquetage. Ces dictionnaires sont manuellement prédéfinis par des experts et incluent, par exemple, des listes de pays, de villes, d'entreprises, de noms de personnes, de titres, etc.

Parmi les méthodes à base de règles, nous pouvons citer l'approche nommée FASTUS, proposée dans [Appelt93, Hobbs95] dans le contexte de la reconnaissance d'événements qui concernent le terrorisme en Amérique latine, dans des quotidiens. Cette approche se base sur l'analyse de texte à l'aide de grammaires à états finis (dites grammaires hors-contexte) manuellement préparées. Cette grammaire décrit des patrons d'événements à l'aide de mots-clés, comme par exemple le patron « <Perp> attacked <HumanTarget>'s » qui décrit un événement d'attaque en utilisant le mot clé "attacked". Le système est composé de quatre étapes principales : la recherche de mots-clés, la reconnaissance de phrases, la reconnaissance de patrons de chaque phrase et la fusion d'événements. D'abord, les mots-clés prédéfinis dans un dictionnaire sont identifiés dans le texte. Ensuite, les phrases sont reconnues en se basant sur l'identification de groupes nominaux et de groupes verbaux décrits par les grammaires hors-contexte. Les patrons d'événements qui correspondent à ces phrases sont reconnus en correspondant leurs mots clés. Ils sont ensuite utilisés pour extraire les informations d'événements. Enfin, les informations retrouvées dans des phrases consécutives, portant sur un même événement, sont fusionnées pour compléter sa description. FASTUS a été testée sur 100 documents et des résultats meilleurs (Rappel = 44%, Précision =55%) que ceux proposés dans la littérature ont été affirmés par les auteurs.

Dans [Wacholder97], les auteurs proposent une méthode de RE qui consiste à extraire, d'abord, les noms propres (par exemple, "Robert Jordan"), puis à les classer en entités (par exemple, personne). Un processus de désambiguïsation de ces noms propres (par exemple, "Ford" peut représenter une personne, une organisation, une marque de voiture ou une place) en utilisant leur contexte est ensuite réalisé. Enfin, les noms qui représentent des variations d'une même entité (par exemple, "Robert Jordan" et "Mr. Jordan" représentent la même Personne) sont regroupés. L'extraction de noms propres est faite par le marquage de mots dans le texte et la sélection des séquences de mots commençant par une majuscule. Des heuristiques lexicales sont ensuite appliquées pour le découpage de noms complexes (par exemple, "Mr Jordan of Steptoe Johnson" est découpé en "Mr Jordan" et "Steptoe Johnson"). La désambiguïsation des noms est faite en utilisant des heuristiques basées sur la capitalisation, la ponctuation, des mots contextuels (par exemple, "Mr" pour une personne et "Inc" pour une organisation) et l'emplacement des noms au sein de la phrase et du document. Le regroupement des noms propres qui représentent la même entité se base sur leur transformation en formes canoniques (par exemple, les abréviations sont éliminées). Les noms propres reconnus dans le corpus sont finalement collectés dans un dictionnaire ou une base de noms. Cette méthode a été évaluée sur un corpus de 88 documents extraits du quotidien de "Wall Street" et les taux de reconnaissance obtenus sont de 91% en Rappel et de 92% en Précision.

Dans [Hashemi03, Hashemi04], les auteurs proposent une approche de reconnaissance de descripteurs d'intérêt (tels que nom, âge, date, ville, etc.) dans les annonces d'évènements de décès. Cette approche se base sur des patrons d'événements qui décrivent les phrases dans l'annonce. Un patron est composé d'une série de marques (en anglais, tokens) qui décrivent grammaticalement et sémantiquement des mots ou des groupes de mots. Par exemple, la marque "DENM" représente un nom de décédé, elle est composée d'un mot ou d'un ensemble de mots composés de caractères alphabétiques et commençant par une lettre majuscule. Cette méthode prend en entrée une phrase d'annonce et propose de la découper en mots puis de regrouper ces mots en marques. Le patron de cette phrase est ensuite identifié

en correspondant la description grammaticale des marques. Les descripteurs d'intérêt sont enfin déduits à partir du patron identifié. La méthode a été expérimentée sur 866 annonces de décès extraites de 68 magazines électroniques. 173 annonces sont sélectionnées arbitrairement pour générer 50 patrons et les 693 annonces restantes sont utilisées pour les tests. Les résultats obtenus en Rappel et en Précision sont respectivement 97.6% et 97.4%.

Ces méthodes sont généralement plus performantes que les méthodes à base d'apprentissage dans les domaines restreints puisqu'elles sont capables de détecter les entités complexes. L'inconvénient majeur de ces méthodes est que les grammaires et les règles utilisées sont dépendantes du langage et du domaine. Il est alors nécessaire de redéfinir le système de reconnaissance pour chaque nouveau domaine ou langage. De plus, ces méthodes sont sensibles au texte bruité ou aux variations particulières (par exemple, les abréviations inusuelles).

5.2.2 Méthodes probabilistes

Ces méthodes se basent généralement sur un corpus annoté par des experts fournissant de nombreux exemples de données et sur un modèle d'apprentissage permettant d'identifier les entités dans le texte à partir de ce corpus. Le modèle d'apprentissage peut être des règles logiques, un arbre de décision, un modèle numérique, etc.

A titre d'exemple, nous pouvons citer le modèle numérique proposé dans [Malouf03] pour résoudre la tâche de RE indépendamment du langage. C'est un modèle de Markov caché où les observations sont les mots avec leurs contextes et les états cachés sont des séquences d'étiquettes, tels que début ou fin de nom de personne, d'organisation, de localisations, etc. Ce modèle est dépendant du corpus d'entraînement fourni. Des expérimentations faites sur des corpus de différents langages ont données des taux de reconnaissance de 74.55% en Précision et de 70.45% en Rappel.

Une approche de reconnaissance d'entités nommées, basée sur l'apprentissage de règles d'association à partir d'un corpus d'entrainement, a été proposée dans [Budi03]. Cette méthode utilise trois types de règles d'association permettant d'attribuer une classe d'entité (parmi nom de personne, localisation, etc. ou pas-de-nom s'il ne s'agit pas d'une entité nommée) à un mot. La première règle, nommée règle de dictionnaire, consiste à associer une classe à un mot (par exemple, la classe nom de personne est associée au mot « Hasibuan »). La deuxième, nommée règle de bigrammes, associe une classe à deux mots (par exemple, la classe "nom de personne" est associée à « Prof., Hasibuan »). La troisième règle, nommée règle de descripteur, associe une classe à un mot et un descripteur (par exemple, la classe "nom de personne" est associée à « Prof., un mot qui commence avec une majuscule »). Un facteur d'appui et un facteur de confiance, qui dépendent du nombre d'occurrences des mots ou des descripteurs dans la classe, sont attribués à chaque règle d'association. Cette approche propose de chercher l'ensemble de règles qui correspondent à chaque paire de mots dans le texte. La règle qui maximise les valeurs des facteurs d'appui et de confiance est retenue. Si ces valeurs dépassent un seuil alors la classe de la règle est attribuée aux mots. Autrement, la classe "pas-de-nom" est attribuée à ces mots. L'approche est évaluée sur deux corpus. Le premier corpus est extrait de la base MUC-7 [Chinchor99] où 200 articles sont utilisés pour la phase d'apprentissage des règles d'association et 100 articles sont utilisés pour les tests. Le deuxième corpus est extrait à partir de 55 articles d'actualité rédigés en langue indonésienne. Les deux corpus ont été évalués pour différentes combinaisons des trois règles et ont été comparés à chaque fois avec une méthode de la littérature [Chieu02] qui utilise l'entropie maximale. Les résultats obtenus montrent un important gain en Précision et un niveau concurrentiel en Rappel.

Une approche bayésienne d'identification des lignes d'adresses postales américaines a été proposée dans [Talburt00]. Cette approche a pour objectif d'analyser et d'associer des lignes d'adresses à leurs composantes sémantiques (ex : associer la première ligne à un nom de particulier et la deuxième ligne à l'adresse de la rue). Cette méthode utilise un corpus d'adresses annoté et propose de classifier chaque

ligne dans des fonctions de lignes en maximisant la probabilité d'appartenance d'une ligne donnée à une parmi 7 fonctions de lignes. Le calcul de la probabilité se base sur une analyse de chaque ligne en des mots séparés et un calcul d'occurrences des mots et leurs positions dans les fonctions de lignes. Des expérimentations ont été faites sur 100000 adresses annotées et 23 millions d'adresses de test. Les auteurs affirment l'augmentation de 7% de la Précision d'identification comparée au système d'identification à base de règles. Cette approche est flexible et permet l'apprentissage de nouveaux résultats après validation par un utilisateur. En revanche, le texte traité est purement structuré et la méthode est limitée aux adresses postales américaines.

Une approche plus générique inspirée de cette dernière a été proposée dans [Osesina12] pour l'extraction d'entités nommées dans des notices nécrologiques. Elle s'appuie sur une base de connaissances qui contient des exemples d'entités annotées, dont chacune est décrite par sa classe et ses propriétés. La classe informe sur le type d'entité nommée, tels que nom du décédé, âge, parent, épouse, etc. Les propriétés sont de deux types : contextuelles (le contexte qui précède/ suit l'entité nommée, la position de l'entité nommée dans le document) et intrinsèques (nombre de caractères, nombre de termes, type numérique ou alphabétique de l'entité nommée). Cette méthode propose d'extraire les propriétés d'un mot ou d'un groupe de mots dans un document, et les compare avec les propriétés des entités de la base de connaissances pour déduire la classe de l'entité nommé correspondante. Cette comparaison est réalisée à l'aide d'un modèle de décision construit à partir de la base de connaissances. Cette approche a été évaluée sur 100 notices nécrologiques extraits du Web et a été testé à l'aide de trois types de modèles de décision : un modèle bayésien, un classifieur $C4.5$ et un modèle de dispersion. La méthode bayésienne a donné des meilleurs résultats en Précision $(53,6\%)$ et en Rappel $(30,5\%)$. L'avantage de cette méthode est qu'elle est indépendante des langages et des ontologies. De plus, le modèle d'extraction ne doit pas être modifié lorsqu'un nouvel exemple est ajouté. En revanche, elle est dépendante de l'étape d'annotation manuelle du corpus et nécessite un grand volume d'exemples annotés.

5.2.3 Méthodes hybrides

Les approches hybrides de RE proposent de combiner les méthodes à base de règles et les méthodes probabilistes. Ces méthodes se basent sur un apprentissage de règles à partir d'exemples annotés puis une révision par un expert, ou inversement, consistent à élaborer des règles par un expert puis à les étendre (semi-) automatiquement par un moteur d'inférence.

Comme exemple de méthode hybride, nous pouvons citer l'approche proposée dans [Rohini00] qui combine une approche à base de règles, un modèle de maximum d'entropie et un modèle de Markov caché pour la reconnaissance d'entités nommées. D'abord, les expressions temporelles (tels que date, fréquence, âge, etc.), les expressions numériques (tels que montant, pourcentage, poids, etc.) et les expressions de contact (tels que adresse, email, etc.) sont étiquetées dans le texte en utilisant des règles grammaticales prédéfinies. Ensuite, les noms de personnes et de localisations sont étiquetés à l'aide d'un modèle probabiliste basé sur le maximum d'entropie. Ce modèle utilise des dictionnaires de noms de personnes et de localisations (dites gazetiers). Il consiste à classer un mot dans le document (dans une classe parmi les trois classes : personne, localisation et autre) en le représentant par des descripteurs qui informent sur son apparition dans les gazetiers et des descripteurs contextuels. Enfin, un modèle de Markov caché contraint basé sur les bi-grammes est utilisé pour corriger les erreurs de segmentation des entités étiquetées lors des deux premières phases. Les contraintes de ce modèle sont fournies sous forme de règles de correction des ambiguïtés de prépositions, des erreurs de capitalisation et des fausses séquences d'entités. Ce système a été testé sur des articles du quotidien américain "New York Times" extraits de la base MUC-7, composés de 22000 mots et le résultat obtenu est de 93.39% en F-mesure.

Bien que les méthodes hybrides profitent des avantages des deux méthodes précédentes, elles héritent aussi de leurs inconvénients. Par exemple, les règles élaborées manuellement sont dépendantes du

domaine et du langage.

5.2.4 Méthodes guidées par une base de données

Ces méthodes s'appuient sur une base de données relationnelle, qui contient les entités recherchées, pour la RE. Dans ce type d'approches, le problème de RE revient à un problème de rapprochement entre le contenu des documents et celui de la base de données. C'est ce que nous cherchons à faire dans cette thèse.

Un système, nommé EROCS (Entity Recognition in Context of Structured data), a été proposé dans [Chakaravarthy06] pour lier un document textuel à des entités décrites dans une base de données relationnelle. EROCS propose d'identifier les entités qui correspondent le mieux avec un document donné en cherchant les liens entre les enregistrements de la base et des segments dans le document. Un segment est assimilé à 8 phrases consécutives dans le texte du document. Cette méthode propose de maximiser un score de rapprochement qui emploie la fréquence et l'importance des termes en commun entre un segment et une entité. Cette méthode a été évaluée à l'aide d'une base de données de films du Web qui contient 401660 entités et 50 documents qui représentent des avis sur les films. Le taux de reconnaissance obtenu est d'environ 80%. Comme elle traite une problématique proche de la nôtre, nous détaillons d'avantage cette méthode dans la section suivante.

Dans [Bhattacharya08], les auteurs proposent d'employer la classification de documents au profit de la reconnaissance d'entités et inversement. Ils se basent sur un modèle probabiliste qui relie la distribution de mots dans le document avec les attributs d'entités contenues dans la base de données. La notion de classification consiste, ici, à catégoriser les documents en se basant sur un vocabulaire de mots pertinents contenus dans chaque document (par exemple, la classe "Action" est déduite du vocabulaire : "adventurer", "quest", "justice", etc.). En effet, cette méthode consiste à rechercher dans la base de données une colonne-type qui renseigne sur la classe du document (par exemple, le champ "genre" du film) puis limiter la recherche d'entités uniquement dans les enregistrements dont l'attribut de la colonne-type correspond à la classe donnée du document (par exemple, si la classe du document est "Action" alors seules les entités dont l'attribut du champ "genre" est "Action" sont considérées). Cette méthode a été évaluée à l'aide d'une base de données de 26250 films de 25 genres différents et 12500 documents qui représentent des commentaires sur les films. Les performances de la RE ont augmenté de 38.5% à 40.8% après emploi de la classification. De plus, la moyenne de confusion pour la prise de décision est diminuée de 2.359 à 0.067. Cette méthode propose une idée intéressante pour guider la recherche d'entités et résoudre certaines ambiguïtés. Cependant, il n'est pas toujours évident de catégoriser les documents et nous ne trouvons pas toujours une colonne-type qui peut informer sur la classe du document. De plus, la tâche de classification des documents peut compliquer le système et alourdir la tâche de RE.

Un problème important relatif aux méthodes guidées par une base de données est le problème de désambiguisation d'entités dans le cas où plusieurs entités candidates ont été identifiées par le système.

Parmi les méthodes de désambiguisation d'entités, nous pouvons citer les travaux proposés par [Talburt07a, Talburt07b, Talburt07c] dans le contexte du modèle RDBMS (Relational Database Management Systems). Le principe de cette approche est modélisé dans la Figure 5.1 et consiste à extraire des fragments (où un fragment représente un mot ou une phrase) du document, puis à sélectionner de la base de données, la liste d'entités candidates qui correspondent avec chaque fragment. Les entités candidates qui partagent un ou plusieurs attributs (E2 et E6 dans la Figure 5.1) sont les plus probables de représenter des entités correctes pour leurs fragments et sont alors sélectionnées par le système. Par exemple, dans le cas de deux personnes mentionnées dans deux fragments d'un document, ceux qui possèdent la même adresse sont les entités retenues par le système. Les expérimentations ont été faites sur 111 docu-

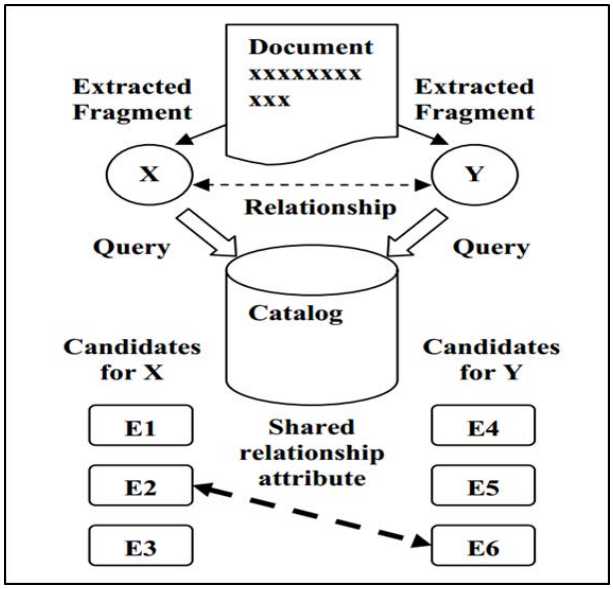


Figure 5.1 – Désambiguisation d'entités par des relations partagées [Talburt07c]

ments qui représentent des notices nécrologiques et un catalogue en ligne [5] qui maintient les informations pour des individus et des organisations aux États-Unis. Les performances de RE ont augmenté de 20% à $67,8\%$ après intégration de cette méthode de désambiguïsation. C'est une méthode intéressante qui permet de réduire les fausses détections d'entités associées à des fragments dans le document mais ce n'est pas toujours évident de retrouver des relations partagées entre les entités référencées dans un même document.

5.2.5 Description détaillée de la méthode EROCS

Le système global de la méthode EROCS est montré dans la Figure 5.2. Le document est filtré à l'aide d'un tagueur en ne gardant que les mots nominaux. Il est ensuite modélisé par des segments où chaque segment est représenté par un sac de mots. Chaque entité dans la base de données est considérée avec son contexte qui est fourni par un modèle d'entités. Ce dernier est représenté sous forme d'arbre où les nœuds sont indexés chacun par une table de la base de données, et les arêtes représentent les éventuelles relations par des clés étrangères. Le nœud racine représente la table pivot du modèle. EROCS propose ainsi de rapprocher ces entités avec des segments dans le document puis les incorpore dans le document par des relations entre les entités identifiées dans la base de données et les termes dans le document, représentant les attributs d'entités.

Le modèle de rapprochement entité-document définit les quantités suivantes pour fonctionner :

— le poids $w(t)$ d'un terme t est défini par l'équation 5.1.

$$w(t) = \begin{cases} \log(N/n(t)) & \text{si } n(t) > 0; \\ 0 & \text{sinon.} \end{cases} \tag{5.1}$$

où N est le nombre d'entités distinctes et $n(t)$ est le nombre d'entités distinctes qui contiennent le terme t dans leur contexte ;

5. zabasearch.com

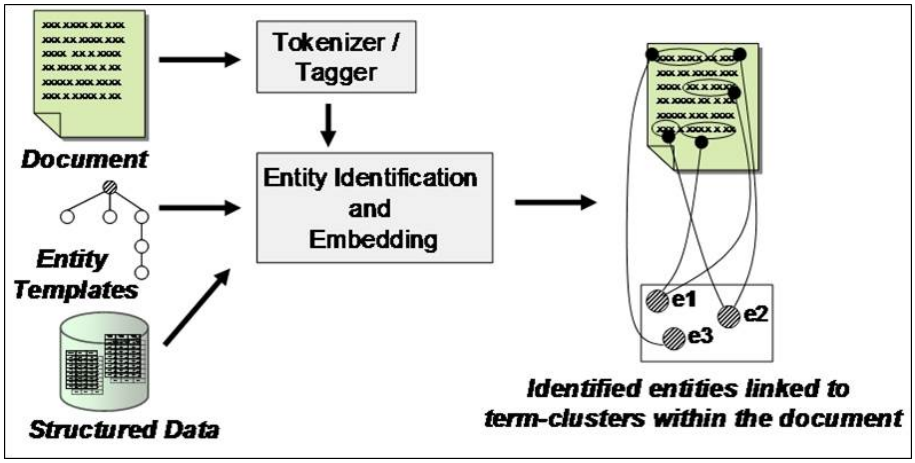


Figure 5.2 – Système global d'EROCS [Chakaravarthy06]

— le score d'une entité e, dans la base de données E, par rapport à un segment d est défini par l'équation 5.2.

$$score(e,d) = \sum_{t \in T(e,d)} tf(t,d).w(t) \tag{5.2}$$

avec $tf(t,d)$ la fréquence du terme t dans le segment d ;

— l'annotation (S,F) d'un document D représente une association entre des segments et des entités, où S est un ensemble de segments dans D n'ayant pas de chevauchement et F une fonction qui associe chaque segment $d \in S$ à une entité $F(d) \in E$, avec :

$$score(S,F) = \sum_{d \in S} score(F(d),d) - \lambda \tag{5.3}$$

où $\lambda \geq 0$ est un paramètre introduit pour éliminer les segments non pertinents (i.e. possédant un score inférieur à λ).

L'algorithme proposé d'EROCS est explicité dans l'Algorithme 2. Il est basé sur la programmation dynamique et permet de retrouver la meilleure annotation d'un document en se basant sur le calcul des meilleures entités qui correspondent à chaque segment. Pour réduire la complexité en temps d'exécution et en accès aux données de la base de données, l'espace de recherche des contextes d'entités est limité en utilisant un cache qui indexe des relations (t,e) pour les termes t contenus dans le contexte d'entités e. Ces relations sont ensuite extraites directement du cache en utilisant les fonctions $getEntities(t)$ (recherche les entités qui contiennent le terme t dans leurs contextes) et $getTerms(e)$ (recherche les termes contenus dans le contexte de e).

5.3 Reconnaissance d'entités dans les documents océrisés

Dans le cas de documents océrisés, la tâche de RE devient plus difficile à cause d'altérations de leurs contenus et leurs structures par l'OCR. Il existe peu de méthodes dans la littérature consacrées à la RE dans les documents océrisés. Dans cette section, nous présentons des exemples de ces méthodes, ainsi que d'autres méthodes proposées dans des contextes connexes traitant des documents océrisés. Deux catégories de méthodes peuvent être distinguées : les méthodes orientées contenu et les méthodes orientées structure.

Algorithme 2 EROCS

Entrées: document : D
Sortie: meilleure annotation : $(S_{|D|}, F_{|D|})$, score : $r_{|D|}$

pour $i = 1 \rightarrow |D|$ **faire** // parcours des phrases du document
  pour $j = i \rightarrow |D|$ **faire**
    $e_{i,j} = \arg\max_{e \in E} score(e, D_{i,j})$ // sélection de la meilleure entité pour le segment $|D_{i,j}|$ allant de la phrase i à la phrase j
    $s_{i,j} = score(e_{i,j}, D_{i,j})$
  fin pour
fin pour
$S_0 = \emptyset$
$r_0 = 0$
pour $k = 1 \rightarrow |D|$ **faire** // recherche de la meilleure annotation
  $j = \arg\max_{0 \leq j \leq k-1}(r_j + s_{j+1,k} - \lambda)$
  $S_k = S_j \cup \{D_{j+1,k}\}$
  $r_k = r_j + s_{j+1,k} - \lambda$
fin pour
pour chaque $d \in S_{|D|}$ **faire** // parcours des segments de l'annotation $S_{|D|}$
  $F_{|D|}(d) = \arg\max_{e \in E} score(e, d)$ // sélection de la meilleure entité pour la meilleure annotation
fin pour
retourner $((S_{|D|}, F_{|D|}), r_{|D|})$

5.3.1 Méthodes orientées contenu

Les méthodes orientées contenu proposent de corriger le contenu textuel bruité des documents océrisés afin de l'exploiter pour la tâche de RE.

Parmi ces méthodes, nous pouvons citer l'approche proposée dans [Miller00] qui consiste à employer un modèle de Markov caché pour l'extraction d'entités nommées. Dans ce modèle, les observations sont les séquences de mots reconnus par l'OCR, et les états cachés sont des étiquettes de type d'entités sur ces mots (par exemple, nom, organisation, date, etc.) ou bien une étiquette non-entité. Ce modèle a été appris à la fois sur des documents corrects et d'autres bruités contenant les mêmes types d'erreurs. Des expérimentations faites sur des quotidiens américains ont montré une dégradation des résultats d'extraction (par 8% en F-mesure) causée par des erreurs dans certains mots reconnus par l'OCR (15% des mots sont erronés).

Une approche plus récente a été proposée dans [Grover08]. Elle consiste à utiliser un système à base de règles pour reconnaître des noms de personnes et des lieux dans des documents historiques (du 17 ème au 19 ème siècle). Cette méthode propose de réaliser un pré-traitement de suppression de bruit et de séparation entre le corps du texte et des notes marginales à l'aide de règles grammaticales. Les entités nommées sont ensuite étiquetées dans le texte reconnu par l'OCR en s'appuyant sur des règles grammaticales spécifiques et des dictionnaires contenant des noms et des prénoms de personnes et des noms de lieux. Des évaluations sur des quotidiens américains ont donné des résultats qui varient entre 70.35% et 76.34% en F-mesure.

Une comparaison expérimentale de méthodes de reconnaissance de noms de personnes dans des documents océrisés bruités a été proposée dans [Packer10]. Cette comparaison a montré une corrélation entre le taux d'erreurs des mots reconnus par l'OCR et la qualité de la reconnaissance d'entités nommées pour les différentes méthodes. De plus, les auteurs ont proclamé que l'impact des erreurs de contexte et

d'ordre de mots sur la reconnaissance est plus important que celui des erreurs de caractères.

Une méthode d'extraction d'entités nommées a été testée, dans [Rodriquez12], sur des documents océrisés bruités et sur les mêmes documents après correction d'erreurs d'OCR. Les résultats ont confirmé une amélioration des taux de reconnaissance après la correction d'erreurs d'OCR pour des entités variées (noms de personnes, de localisations et d'organisations).

Comme le contenu de documents océrisés est souvent bruité, il est nécessaire d'avoir recours à des techniques de tolérance et de correction d'erreurs d'OCR pour une meilleure exploitation de ce contenu. Nous nous intéressons dans la suite de cette section aux approches permettant de corriger ce type d'erreurs.

Nous pouvons classer les méthodes de correction d'erreurs OCR en deux catégories : les méthodes de correction des erreurs d'orthographe et les méthodes de correction des erreurs spécifiques à l'OCR. Nous détaillons ces deux catégories ci-dessous.

5.3.1.1 Correction des erreurs d'orthographe

Ces méthodes utilisent généralement des dictionnaires qui contiennent les mots corrects. Nous détaillons trois types de méthodes basées sur les erreurs d'orthographe ci-dessous.

Méthodes basées sur les distances. Ces méthodes proposent de corriger les erreurs d'OCR dans une chaîne de caractères en la comparant à un ensemble de mots corrects à l'aide de distances de similarité. A titre d'exemple, la distance de Levenshtein est utilisée dans [Damerau64].

L'approche de [Farag09] propose de corriger les erreurs d'OCR en se basant sur les n-grammes. Traditionnellement, la proportion de n-grammes en commun de deux mots par rapport à leurs n-grammes uniques est utilisée comme une mesure de similarité. Cette méthode propose une variation de cette mesure qui consiste à définir une fenêtre de n-grammes permettant de comparer seulement ceux qui se trouvent dans des positions proches. Pour corriger un mot erroné, le mot le plus similaire, au sens de la nouvelle mesure des n-grammes, est sélectionné à partir de ressources lexicales et de dictionnaires.

Méthodes basées sur le contexte. Elles utilisent les informations contextuelles pour détecter et corriger les erreurs d'OCR.

A titre d'exemple, un algorithme de correction nommé WSD (Word Sense Disambiguation) est proposé par [Ginter04] dans le domaine biologique. Il cherche à corriger les mots utilisés dans un faux contexte par leurs sens, où le sens d'un mot est un autre mot qui possède une orthographe proche mais une signification différente. Par exemple, les sens du mot "desert" sont "dessert" et "deserter". Cet algorithme se base sur un classifieur qui apprend le sens d'un mot en considérant son contexte (un ensemble de n mots qui le précèdent et le suivent). Ce classifieur propose de retenir le sens qui maximise une fonction de décision définie en fonction du nombre d'occurrences des mots de contexte dans ce sens.

Une autre méthode, qui exploite le thème du document pour la correction, est proposée dans [Wick07]. Elle favorise les mots liés sémantiquement au thème en question parmi une liste de mots candidats pour corriger les termes erronés. Cette méthode construit un modèle de thèmes et un modèle d'OCR. Le modèle de thèmes décrit chaque thème par un vocabulaire. Il est initialement appris à partir d'une collection de documents. Le modèle d'OCR représente les probabilités des différentes altérations de caractères. Il est construit statistiquement à partir de documents corrigés. Cette méthode prend en entrée un document océrisé et une liste de ses mots erronés. Elle génère pour chaque mot erroné w, la liste des mots qui diffèrent de w par aucun, un ou deux caractères. Un score est ensuite attribué à chaque mot de cette liste. Il est calculé en fonction de la probabilité d'apparition du mot dans les thèmes et des probabilités de confusion de ses caractères avec les caractères du mot erroné. Ainsi, le mot qui possède le score le plus important est retenu.

Une méthode de correction et de normalisation d'entités nommées dans des documents océrisés est proposée dans [Sagot14]. Cette méthode propose de corriger des formules chimiques, des dates et des adresses en se basant sur des règles grammaticales définies pour chaque type d'entité nommée. Par exemple, la date erronée "23 avril i908" est corrigée par "23 avril 1908" en se basant sur une règle qui remplace un 'i' qui apparaît dans une année par un '1'. Un autre exemple de normalisation d'une formule chimique est la correction de "MgA1204" par "$MgAl_2O_4$" en remplaçant le chiffre '1', resp. '0' par un 'l', resp. 'O' et en mettant les chiffres '2' et '4' en exposant. L'inconvénient majeur de cette méthode est la nécessité de prédéfinir toutes les règles de correction.

Une méthode qui utilise les suggestions d'orthographe données par un corpus gigantesque, publié par « Google », est proposée dans [Bassil12]. Ce corpus contient des millions de mots et des séquences de n-grammes (allant des uni-grammes jusqu'aux 5-grammes) de mots. Cette méthode procède, d'abord, par la détection des erreurs à l'aide d'une vérification orthographique par rapport au corpus de « Google ». Les mots qui ne sont pas retrouvés dans ce corpus sont considérés erronés. Les candidats de chaque mot erroné sont ensuite générés à l'aide de la comparaison de bi-grammes de mots. En effet, chaque mot du corpus contenant au moins un bi-gramme commun avec le mot erroné est considéré comme un candidat. Enfin, les 5-grammes de mots composés des séquences des quatre mots qui précède le mot erroné dans le texte et un parmi les mots candidats sont générés et comparés aux 5-grammes du corpus de « Google ». Le mot candidat qui correspond au 5-gramme ayant la plus importante fréquence est retenu.

Le problème avec les méthodes de correction des erreurs d'orthographe est qu'elles exigent des dictionnaires spécifiques au langage et au lexique du corpus utilisé. A cela s'ajoute qu'elles sont dépendantes de la qualité (complet, représentatif du corpus, correct, etc.) du dictionnaire utilisé. Mais, il est parfois difficile de trouver ces dictionnaires (par exemple, dans le cas des corpus historiques).

5.3.1.2 Correction des erreurs spécifiques à l'OCR

Ces méthodes exploitent les erreurs spécifiques à l'OCR pour proposer des corrections des caractères erronés en se basant sur l'apprentissage à partir d'un corpus annoté ou sur une classification préalable des caractères. Nous donnons ci-dessous quelques-unes de ces méthodes.

Méthodes probabilistes. Les méthodes probabilistes estiment statistiquement à partir d'un corpus annoté la probabilité de correction d'un mot altéré par un autre mot correct.

A titre d'exemple, nous citons la méthode, proposée dans [Collins-Thompson01], qui se base sur des règles de confusion d'un n-gramme s dans le mot erroné par un n-gramme R. Ces règles sont apprises et des probabilités (notés $p(s|R)$) y sont associées (voir les exemples dans le Tableau 5.1). Cette méthode propose de générer les différentes partitions de n-grammes d'un mot erroné (par exemple {"am","end","me","nt"} et {"a","men","d","me","nt"} représentent deux partitions du mot "amendment"). Elle se base ensuite sur un algorithme de programmation dynamique qui cherche à minimiser le coût total de transformation de ce mot en un autre, en appliquant les transformations possibles de n-grammes sur toutes ses partitions. Si le coût minimal ne dépasse pas un seuil, fixé manuellement, alors le mot qui correspond à ce coût est retenu.

Une autre méthode de correction probabiliste qui utilise les automates à états finis est proposée dans [Kolak03]. Cet automate modélise les séquences de transformations possibles des caractères d'un mot erroné, pondérées par leurs probabilités. Ces transformations incluent les opérations d'insertion, de suppression et de substitution de caractères.

Bien que ces méthodes n'aient pas besoin de dictionnaires linguistiques, elles nécessitent un corpus annoté pour l'apprentissage.

Tableau 5.1 – Exemples de confusions de n-grammes de caractères [Collins-Thompson01]

$s \rightarrow R$	$-\log p(R\|s)$ (coût d'édition)
am $\rightarrow$ arn	1.074
en $\rightarrow$ ea	0.956
en $\rightarrow$ e,n	4.400
nt $\rightarrow$ at	1.013
end $\rightarrow$ ead	0.708
end $\rightarrow$ eud	2.508
men $\rightarrow$ rnea	0.858
me $\rightarrow$ me,	1.211

Tableau 5.2 – Classification de caractères selon leur forme [Niklas10]

Classe	Caractères	Cardinalité
i	f, i, j, k, l, r, t, B, D, E, F, I, J, K, L, P, R, T, 1, !	1
	n, h, u, H, N, U	2
	m, M	3
o	a, b, d, g, o, p, q, O, Q, 6, 9, 0	1
c	e, c, C, G	1
v	v, x, y, V, Y, X	1
	w, W	2
s	s, S, 5	1
z	z, Z	1
a	A	1

Méthodes à base de classification des caractères. Ces méthodes se basent sur la similarité de formes des caractères pour la correction.

La méthode de [Niklas10, Tahmasebi13] propose de calculer, d'abord, une clé, appelée OCR-key, pour chaque mot. Cette clé est déterminée en fonction de la classe c et la cardinalité n (décrites dans le Tableau5.2) en remplaçant chaque caractère par cn. Par exemple, OCR-key ("tiine") = OCR-key ("time") = i5c1. Ensuite, pour un terme erroné, des termes candidats qui possèdent la même valeur de clé sont extraits à partir d'un dictionnaire préétabli. Enfin, ces termes candidats sont ordonnés selon leur distance de Levenshtein, leur fréquence dans le corpus et la différence de cardinalité avec le mot erroné.

C'est une méthode générique qui profite des erreurs pertinentes de l'OCR et qui ne nécessite pas de données de vérité. En revanche, certaines classes doivent être adaptées pour d'autres types d'OCR. Par exemple, le 'h' et le 'b' sont parfois confondus. De même, pour le 'o' et le 'c'. Aussi, il parait étrange de classer le 'B' et le 'D' dans la classe 'i'.

5.3.2 Méthodes orientées structure

Les méthodes orientées structure proposent d'exploiter la structure physique du document pour localiser et extraire les attributs d'entités. Nous nous inspirons ici des méthodes proposées dans le contexte de l'étiquetage logique du document image qui consiste à modéliser la structure de document de façon à guider l'extraction des étiquettes logiques.

Une étude bibliographique des méthodes d'étiquetage logique est fournie dans [Mao03, Dengel14]. Ces méthodes peuvent être classées dans deux types : les méthodes partielles et les méthodes globales.

Nous détaillons ces deux types de méthodes dans la suite.

5.3.2.1 Méthodes partielles

Les méthodes partielles proposent d'étiqueter individuellement chaque zone en se basant sur des caractéristiques sur ces zones. Elles peuvent être basées sur des règles prédéfinies ou sur un modèle probabiliste pour l'étiquetage des zones individuellement. Nous détaillons ces deux catégories séparément.

Méthodes à base de règles. Elles se basent sur des règles structurelles et/ou textuelles prédéfinies par des experts du domaine pour étiqueter logiquement des blocs ou des expressions textuelles dans le document. Plusieurs travaux, tels que dans [Kim01, Tan03], utilisent de telles règles.

L'approche proposée dans [Kim01] définit 120 règles conditionnelles sur trois niveaux (zone, ligne et caractère) pour l'étiquetage d'articles scientifiques. Les règles utilisées emploient des caractéristiques de mise en forme telles que la position, la taille, l'espacement, la police, etc. pour étiqueter le titre, l'auteur, l'affiliation et le résumé. Cette approche a été évaluée sur 11000 articles dans des journaux biomédicaux et le taux d'extraction obtenu est de l'ordre de 96%.

L'inconvénient de ce genre de méthodes est que les règles employées sont spécifiques et restreintes à des domaines particuliers.

Méthodes probabilistes. Dans ce genre de méthodes, c'est le système qui apprend la fonction d'étiquetage statistiquement sur un corpus annoté à l'aide d'un classifieur. Plusieurs travaux d'étiquetage logique supervisé [Staelin07, Medvet11] ont été proposés dans la littérature.

L'approche nommée Biblio, proposée dans [Staelin07], se base sur une combinaison d'un SVM avec des réseaux de neurones. Elle utilise des descripteurs textuels et des descripteurs de mise en page (dictionnaires de méta-données, boîtes englobantes des éléments, taille de police, etc.). Ces descripteurs sont utilisés pour entraîner une cascade de classifieurs et un vote majoritaire est ensuite employé pour attribuer les étiquettes logiques. Le système a été validé sur trois corpus et les résultats varient largement selon le type de méta-données.

Ces méthodes ne demandent pas de connaissances ou des règles a priori mais exigent un corpus annoté qui couvre toutes les variabilités.

5.3.2.2 Méthodes globales

Les méthodes globales intègrent des contraintes globales entre les zones telles que la position relative pour déterminer leurs étiquettes logiques. Elles peuvent être classées en trois catégories : les méthodes ascendantes, les méthodes descendantes et les méthodes hybrides qui combinent les deux. Nous détaillons chacune de ces catégories.

Méthodes ascendantes. Ce sont des méthodes d'inférence qui se basent sur un étiquetage des sous-éléments pour déduire l'étiquette de l'élément composite. Ces méthodes se basent généralement sur des règles syntaxiques pour déduire la structure logique du document. Elles se basent sur des techniques de traitement automatique des langues appliquées sur le résultat ASCII de l'OCR, telles que les méthodes proposées dans [Belaid01, Aiello02, Lin06].

L'approche de reconnaissance de tables de matières proposée par [Belaid01] consiste à identifier le titre et les auteurs dans les journaux scientifiques en se basant sur leurs caractéristiques linguistiques (par exemple, la liste des auteurs est une suite de noms de personnes). Cette approche est composée de trois étapes : un étiquetage linguistique du texte donné par l'OCR, une analyse morphologique et une désambiguïsation syntaxique. L'étiquetage linguistique consiste à marquer les composantes physiques (par

exemple, début de ligne, tabulation, item, etc.) et morphologiques (par exemple, nom propre, préposition, article, etc.) du texte à l'aide de dictionnaires et de règles spécifiques. L'analyse morphologique consiste à regrouper les étiquettes consécutives en attributs (titre ou auteur) en utilisant des règles linguistiques de regroupent, de continuité et de réduction. La désambiguïsation syntaxique propose d'intégrer les mots non étiquetés (à cause des erreurs d'OCR) avec les attributs en utilisant des règles grammaticales ou des modèles de structure physique générés à partir des journaux correctement étiquetés. Cette méthode a été évaluée sur 38 journaux scientifiques contenant 2020 articles. Le taux de segmentation obtenu est de 96.3% et le taux d'extraction d'attributs est de 93%. L'inconvénient de cette méthode est qu'elle utilise des règles spécifiques et restreintes au domaine d'application.

Méthodes descendantes. Elles se basent sur la reconnaissance de la structure physique du document pour l'étiquetage de ses composantes logiques. Certaines méthodes se basent sur un parcours arborescent de la page en utilisant des règles de mise en forme [Tsujimoto92, Wang01] ou des règles syntaxiques [Nagy92, Krishnamoorthy93]. D'autres méthodes se basent sur une classification supervisée à partir d'exemples de documents annotés appartenant à une classe déterminée, telles que les méthodes proposées dans [Cesarini03, Mao04, Zou10] ou sur une classification non supervisée des blocs dans le document, telle que la méthode proposée dans [Bres02]. Ces méthodes sont généralement rapides mais nécessitent des connaissances a priori sur la mise en page du document.

Nous détaillons, à titre d'exemple, la méthode proposée dans [Klink01] qui est composée de plusieurs phases. D'abord, le document scanné est segmenté en blocs puis reconnu par un OCR. Ensuite, les structures usuelles sont reconnues (par exemple : entête, pied de page, tableau, liste, etc.) afin de raffiner le résultat donné par l'OCR. Enfin, une phase d'étiquetage logique à base de règles logiques est effectuée sur les blocs textuels identifiés. Ces règles se basent sur des critères géométriques et textuels sur les blocs (dimension, position, nombre de lignes et leur alignement, police, mots clés) et sur les relations entre eux (position relative, mots en commun). Ces règles sont employées pour attribuer une étiquette logique à chaque segment de la page. Une évaluation a été réalisée sur 89 lettres administratives et 979 pages de journaux diverses et a donné des résultats qui varient entre 36% et 100% en Précision et en Rappel selon le type de l'étiquette.

Dans [Nagy92], les auteurs proposent de transformer une image représentant une page d'article scientifique en un arbre qui représente hiérarchiquement la structure logique de cette page. Cette transformation se base sur une analyse syntaxique à base de grammaires hors-contexte. Cette grammaire est définie sur les blocs physiques de la page en utilisant le profil de projection. Cette approche a été expérimentée sur seulement 24 pages de journaux scientifiques dont 16 ont été parfaitement étiquetées, 6 contiennent des erreurs mineures et 2 contiennent des erreurs majeures.

L'approche, proposée dans [Mao04], se base sur les modèles de Markov cachés pour l'étiquetage des articles scientifiques. Ces modèles sont utilisés pour représenter la mise en page en se basant sur le profil de projection des boîtes englobantes des caractères dans les régions physiques du document. Les observations de ce modèle représentent des caractéristiques des régions physiques telles que la police, la taille, les mots clés, etc. Les états cachés sont les étiquettes sur les régions tels que titre, auteur, affiliation, etc. L'approche a été évaluée sur 69 pages d'articles scientifiques et comparée à d'autres modèles. Le taux d'extraction obtenu est de 91% comparé à 70% pour un modèle à base d'heuristiques.

Méthodes hybrides. Elles proposent de combiner les deux approches précédentes afin de profiter de la rapidité des méthodes descendantes et de la robustesse des méthodes ascendantes.

Dans [Liang02a, Liang02b, Liang03], les auteurs proposent une approche d'étiquetage logique d'articles scientifiques, basée sur la modélisation de leur structure physique du document par un graphe attribué. Les nœuds de ce graphe représentent les blocs reconnus par l'OCR et les arcs représentent les

relations spatiales entre eux. Ce graphe est comparé à un ensemble de graphes modèles, où chacun est appris pour une classe de document. Le résultat de la comparaison est utilisé pour attribuer les étiquettes logiques aux blocs du document. Cette approche est dépendante de la classe de documents. Malheureusement, aucune évaluation quantitative de cette méthode n'a été donnée.

Dans le même contexte d'étiquetage logique d'articles scientifiques, une approche avec raisonnement à partir de cas est proposée dans [Beusekom07]. Pour un document en entrée, cette approche propose de rechercher le cas le plus proche dans une base de cas qui représente un ensemble de documents étiquetés. Le document qui correspond au cas le plus proche est utilisé pour étiqueter les blocs du document d'entrée. Pour la recherche dans la base de cas, une mesure de similarité de mise en page de documents, qui combine des critères structurels et textuels au niveau bloc, est employée. L'évaluation de cette approche sur la base publique MARG [6] a donné des taux d'extraction qui varient entre $94,8\%$ et $99,6\%$.

Une approche récente, proposée dans [Santosh13a, Santosh13b, Santosh13c, Santosh15], consiste à identifier les lignes de tableau à partir de motifs (champs textuels ou attributs) sélectionnés par le client. Cette approche est basée sur une technique d'isomorphisme tolérant à la substitution entre un graphe construit des motifs et un graphe modèle appris. Les nœuds de ce graphe représentent les noms de champs et les arcs représentent les positions relatives entre eux. Les graphes traités sont linéaires et la technique de rapprochement de graphe utilisé ne tolère pas les opérations d'insertion et de suppression. Cette approche a été évaluée sur des documents administratifs et les résultats obtenus sont de 86.64% en Précision et de 90.80% en Rappel.

5.4 Conclusion

Dans ce chapitre, nous avons présenté une étude bibliographique détaillée sur les méthodes de RE en distinguant le cas des documents du Web et le cas de document océrisés. D'après cette étude, nous constatons que les méthodes proposées pour le rapprochement d'entités dans les documents du Web sont insuffisantes dans notre cas. En effet, elles considèrent le document comme une suite de phrases et se basent sur une comparaison stricte entre les mots. Ces méthodes sont donc inappropriées pour exploiter la structure physique du document et sont sensibles au contenu bruité. De plus, nous constatons une faiblesse de l'état de l'art sur la RE dans les documents océrisés. Nous avons étudié des méthodes proposées dans des contextes connexes, notamment la correction d'erreurs d'OCR et l'étiquetage logique de documents. Nous proposons de nous inspirer de certaines de ces méthodes pour tolérer le contenu bruité des documents océrisés et pour exploiter la structure physique du document pour localiser les attributs d'entités.

6. http ://marg.nlm.nih.gov/index2.asp

Chapitre 6

Rapprochement par le contenu

Sommaire

6.1 Introduction

Nous proposons dans ce chapitre deux méthodes de rapprochement d'entités par le contenu. Nous détaillons la première méthode, nommée M-EROCS, dans la Section 6.2 et la deuxième méthode, nommée ERBL, dans la Section 6.3. M-EROCS modélise les documents par des segments (blocs de l'OCR) qui sont représentés par des sacs de mots et propose de faire correspondre ces segments aux entités, dans le modèle entité, représentées également par des sacs de mots. ERBL consiste à étiqueter le document par les attributs d'entités contenus dans le modèle entité puis à regrouper ces labels en entités. Ces deux méthodes sont ensuite évaluées dans la Section 6.4 sur les trois corpus et comparées avec une méthode de l'état de l'art.

6.2 Méthode M-EROCS

Cette méthode de RE, appelée M-EROCS (Modified-Entity RecOgnition in Context of Structured data), est une modification de la méthode EROCS présentée dans le chapitre précédent afin de l'adapter aux documents océrisés.

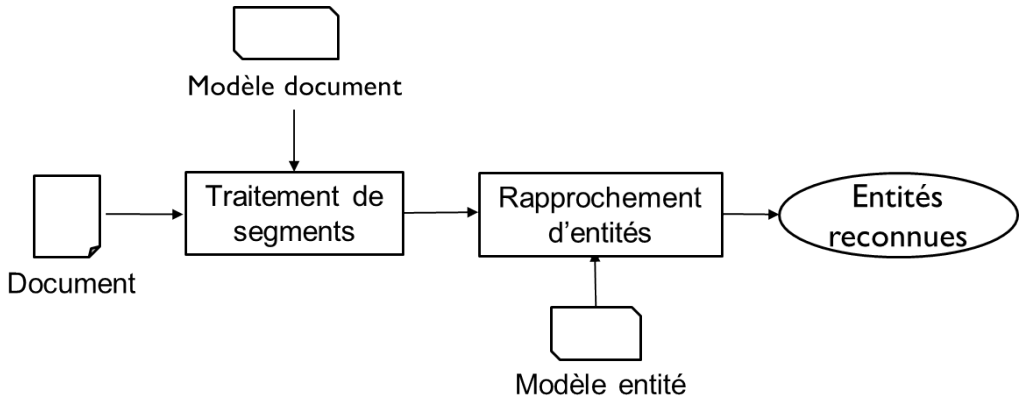


Figure 6.1 – Système global de M-EROCS

Les adaptations et les améliorations proposées dans M-EROCS par rapport à la méthode EROCS sont les suivantes :

— Le modèle d'entité (entités avec leur contexte dans EROCS) représente les différentes variations des attributs d'entités (résultat du processus de résolution d'entités).
— Le modèle document est représenté par une suite de segments, où un segment (suite de 8 phrases dans EROCS) est assimilé à un bloc d'OCR.
— Une étape de filtrage des segments est introduite pour guider la recherche d'entités.
— Une étape de reconstruction physique de segments est introduite pour remédier à leur destruction par l'OCR.
— Le score de rapprochement est amélioré afin de tolérer les erreurs textuelles introduites par l'OCR ainsi que les variations de représentation d'attributs entre le document et le modèle entité.
— Un seuil de score est introduit pour rejeter les segments qui ne mentionnent pas d'entités.

Le système global de M-EROCS est montré dans la Figure 6.1. Le modèle physique (modèle document) facilite la recherche des entités. Ce modèle est composé d'un ensemble de segments initialisés aux blocs données par l'OCR. Ces segments sont ensuite filtrés puis reconstruits. Les segments reconstruits sont rapprochés avec les entités dans le modèle d'entité en se basant sur un score de rapprochement tolérant les erreurs textuelles d'OCR.

Nous commençons par présenter le modèle document. Nous détaillons ensuite les différents modules de notre système.

6.2.1 Modèle document

Un document océrisé est représenté par une hiérarchie de blocs composés de lignes où chaque ligne est composée de mots. Souvent l'OCR procure les coordonnées des boîtes englobantes de chacun de ces éléments. La Figure 6.2 (a) montre un exemple de document image où les blocs obtenus par l'OCR sont encadrés. La Figure 6.2 (b) montre le résultat d'OCR du bloc encadré en rouge.

Le modèle document consiste à représenter le document image par un ensemble de segments, où chaque segment est représenté par un sac de mots (voir Figure 6.3). Dans un premier temps, nous assimilons un segment du document à un bloc donné par l'OCR.

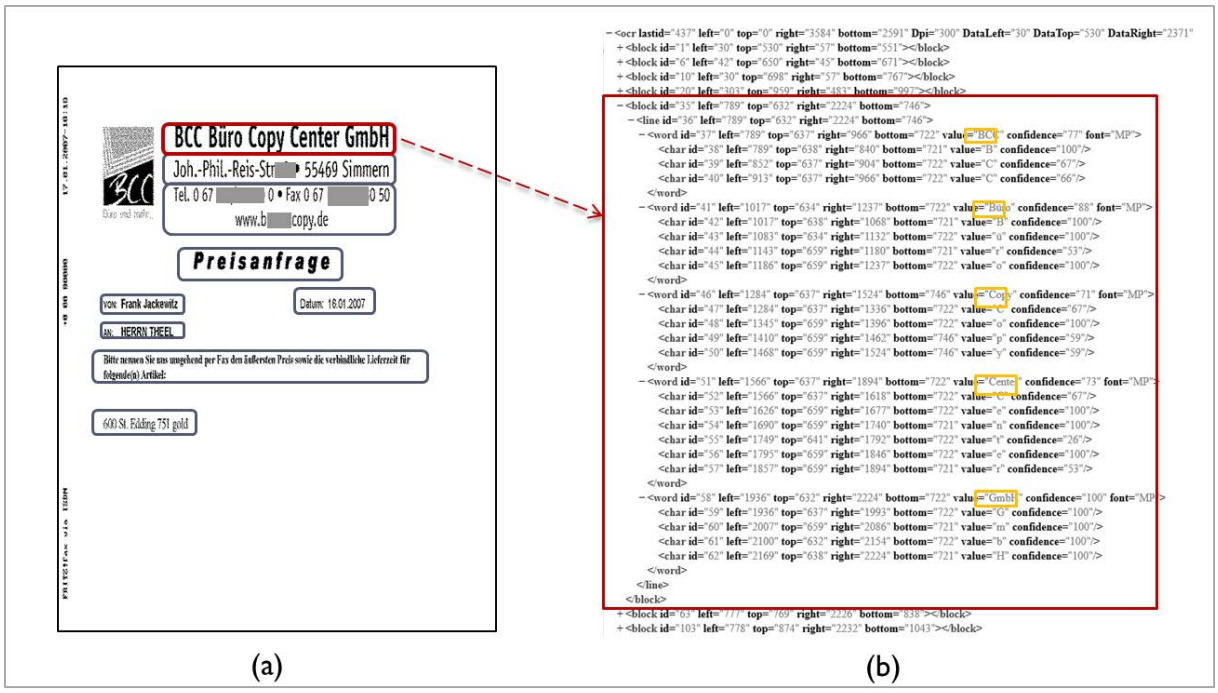


Figure 6.2 – (a) Exemple de document image où les blocs obtenus par l'OCR sont encadrés, (b) résultat d'OCR du bloc encadré en rouge.

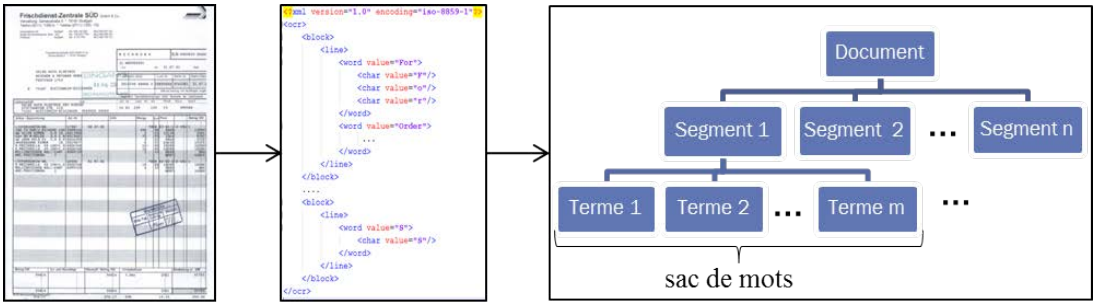


Figure 6.3 – Représentation des documents images par un modèle document.

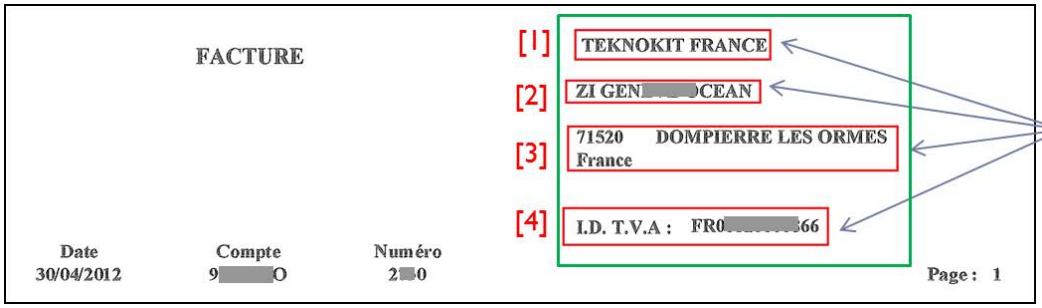


Figure 6.4 – Entité éclatée par l'OCR en 4 segments consécutifs (les segments de numéros 1, 2, 3 et 4), l'entité est encadrée en vert et les blocs donnés par l'OCR sont encadrés en rouge.

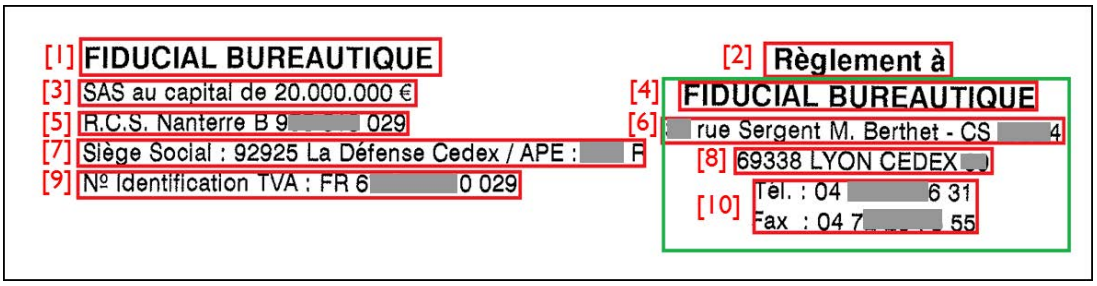


Figure 6.5 – Entité, encadrée en vert, éclatée par l'OCR en 4 segments non consécutifs (les segments de numéros 4, 6, 8 et 10), l'entité est encadrée en vert et les blocs donnés par l'OCR sont encadrés en rouge.

6.2.2 Traitement de segments

Comme la comparaison des termes de tous les segments avec les termes de toutes les entités est coûteuse, il est intéressant de cibler les segments pertinents. Ainsi, nous proposons de rechercher des contextes d'entités dans les documents. Ces contextes nous aideront à localiser et à analyser des segments d'intérêt. A titre d'exemple, pour identifier l'adresse d'une entreprise, nous proposons de rechercher le code postal. Pour ce faire, des expressions régulières sont construites pour les champs de format standard (par exemple, code postal, téléphone, montant, date, etc.). Voici un exemple d'une expression régulière des codes postaux de la France : $\hat{}(F-)?[0-9]2\backslash s?[0-9]\{3\}\$$.

L'entité n'est pas toujours contenue dans un seul segment. En effet, elle peut être éclatée en plusieurs segments consécutifs ou non consécutifs (voir Figure 6.4 et Figure 6.5). Nous proposons alors d'utiliser les coordonnées des blocs pour restructurer les segments en regroupant les segments contigus.

La restructuration des segments permet de reconstruire le document en formant des paragraphes bien formés considérés comme segments physiques. Ceci permettra de retrouver les attributs de l'entité en se basant sur la particularité de leur regroupement.

6.2.3 Reconnaissance d'entités

Nous proposons de redéfinir le score d'EROCS afin de tolérer les erreurs textuelles générées par l'OCR.

Le score d'une entité e, dans le modèle entité E, par rapport à un segment d, est défini par l'équation (6.1).

$$score(e,d) = \sum_{t_1 \in close(\theta, T(d), T(e))} tf(t_1,d).idf(t_2) \tag{6.1}$$

avec $tf(t,d)$ la fréquence du terme t dans le segment d, $idf(t)$ l'importance du terme t dans les entités de E et $close(\theta, T(d), T(e))$ l'ensemble de termes t_1 dans d tel qu'il existe un terme t_2 dans e où t_1 et t_2 sont similaires par rapport à une mesure de similarité. Pour mesurer cette similarité, nous introduisons une nouvelle distance de correction d'erreurs OCR dans la section suivante. Par la suite, nous étudions expérimentalement la combinaison de cette distance avec les distances de comparaison de chaînes de caractères présentées dans le Chapitre 3 de la Partie I.

Nous proposons de rapprocher chaque segment d avec l'entité e_{max} qui maximise le score, comme défini dans l'équation (6.2).

$$e_{max} = \arg\max_{e \in E} score(e, d) \tag{6.2}$$

Nous avons introduit un seuil T permettant d'accepter seulement les entités e_{max} tel que :

$$score(e_{max}, d) \geq T$$

Ce seuil est fixé empiriquement à l'aide d'un corpus de validation.

Algorithme 3 M-EROCS

Entrées: Modèle entité : E, document : D
Sortie: Ensemble d'entités reconnues : $matchE$

1: $matchE = \emptyset$
2: $segSet = segFilter(D)$ // filtrage de segments
3: **pour chaque** $d \in segSet$ **faire**
4: $score = \max_{e \in E'} score(e, d)$
5: i=0
6: **tant que** $i < \lambda$ **faire**
7: $d = addClosestSeg(d)$ // fusionner d avec son plus proche segment physiquement
8: $e_{max} = \arg\max_{e \in E'} score(e, d)$
9: $i++$
10: **si** $score(e_{max}, d) \geq score$ **alors**
11: $i = 0$
12: $score = score(e_{max}, d)$
13: **fin si**
14: **fin tant que**
15: $d = removeClosestSeg(d, \lambda)$ // retirer de d les λ derniers segments rajoutés
16: **si** $score \geq T$ **alors**
17: $matchE = addEntity(matchE, e_{max}, score)$
18: $segSet = removeSegments(d)$ // retirer de $segSet$ les segments mis en correspondance
19: **fin si**
20: **fin pour**

L'algorithme de RE est donné par l'Algorithme 3. Ce dernier propose de fusionner progressivement les segments contigus en se basant sur un calcul itératif du score de rapprochement. D'abord, les segments contenant des données étiquetées sont identifiés (ligne 2 de l'algorithme). Ensuite, pour chaque segment identifié, la zone de recherche est agrandie en lui fusionnant les segments les plus proches physiquement tant que le score augmente durant λ fusions successives (lignes $6-14$ de l'algorithme). Lorsque la fusion est arrêtée, les λ derniers segments rajoutés sont retirés (ligne 15 de l'algorithme). λ est fixée expérimentalement à 2. Enfin, l'entité mise en correspondance avec le segment résultat est acceptée si son score dépasse le seuil T.

Tableau 6.1 – Classification de caractères selon leur forme en se basant sur des statistiques sur nos corpus

Classe	Caractères	Cardinalité
i	1, i, j, I, J, t, T, f, l, L, !, /, \, 7	1
	n, r, u, ü, û, ù, v, U, V, h, H, x, y, X, Y	2
	m, N	3
	M, w, W	4
a	a, à, â, @, p, q, b, d, g, e, é, è, ê, ë, o, ô, ö, ò, c, 9, 6	1
A	A, 4	1
O	(,)	1
	O, Ö, Ô, Ò, Û, Ü, Ù, 0, D, B, G, C, Q	2
p	E, p, P, F, R, k, K	1
s	s, S, 5, 2, 8, 3	1
z	z, Z	1

6.2.4 Similarité de termes

6.2.4.1 Mesure de tolérance d'erreurs d'OCR

Nous proposons une méthode de tolérance d'erreurs d'OCR inspirée de la méthode OCR-key définie dans le contexte de la post-correction d'erreurs OCR (voir Section 5.3.1.2).

Nous adaptons les classes à notre OCR en réalisant une étude statistique sur les erreurs fréquentes dans nos corpus d'étude. Par exemple, on range le 'h' et le 'b' dans une même classe. De même, pour le 'o' et le 'c'. De plus, nous complétons les classes par des caractères accentués, des caractères non alphanumériques ou des ponctuations. Par exemple, mettre le '@' dans la même classe que 'a' et le '/' dans la même classe que 'I'. Le tableau de classification de caractères adopté est montré dans le Tableau 6.1.

Soit $key(s)$ la forme canonique de la chaîne de caractères s qui remplace chacun de ses caractères par m fois sa classe, avec m sa cardinalité. Par exemple, $key(\text{A1IVIARGUES}) = \textit{AiiiiiApOipps} = key(\text{AIMARGUES})$. Nous constatons que "A1IVIARGUES" et "AIMARGUES" possèdent la même clé puisque '1' et 'I' possèdent la même forme et peuvent ainsi être confondus par l'OCR (similairement 'M' et "IVI").

Nous proposons une distance entre deux chaînes de caractères s et w, appelée OCR-key, qui représente la moyenne de la distance de Levenshtein entre les clés de chaînes et celle entre les chaînes elles-mêmes. Cette distance est définie dans l'équation (6.3).

$$\textit{OCR-key}(s,w) = \frac{\textit{Levenshtein}(key(s), key(w)) + \textit{Levenshtein}(s,w))}{2} \tag{6.3}$$

6.2.4.2 Combinaison de mesures de similarité

Nous proposons de comparer la distance OCR-key avec d'autres mesures de similarités (présentées dans le Chapitre 3 de la Partie I). Pour ce faire, nous utilisons un corpus annoté de paires d'attributs, où le premier attribut est extrait du modèle entité et le deuxième du document. Ce corpus est extrait de la base d'entreprises. Il est décrit dans le Tableau 6.2.

Pour évaluer chaque mesure, nous avons ordonné de manière descendante les paires selon leurs distances et nous avons évalué la Précision et le Rappel selon le nombre des n meilleures distances acceptées. La Figure 6.6 montre les courbes de variation du Rappel et de la Précision, où nous constatons qu'aucune des mesures ne se démarque. Ceci est expliqué par l'intérêt de chaque mesure pour tolérer

Tableau 6.2 – Corpus annoté de paires d'attributs extrait de la base d'entreprises

Champ d'attributs	# paires d'attributs	#similaires	#non similaires
Nom	294	130	164
tva	335	165	170
Siren	370	220	150
Adresse	200	90	110
Code postal	410	220	190
Ville	81	35	46
Téléphone	998	540	458
E-mail	305	160	145
Web	240	110	130
Total	3233	1670	1563

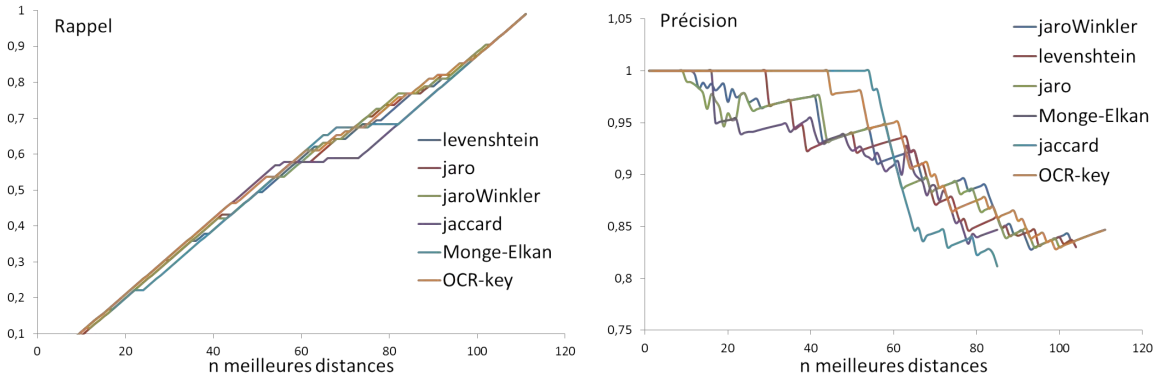


Figure 6.6 – Comparaison de la mesure OCR-key avec d'autres mesures de similarité en Rappel et Précision.

Tableau 6.3 – Exemples de paires d'attributs

Champ de label	Valeur dans le modèle entité	Valeur dans le document	Causes
Adresse	**Av** de l'Europe	**Avenue** de l'Europe	abréviation
Nom	**Tps** Digoinnais	**Transports** Digoinnais	abréviation
Nom	Transports Alain Cassier	Cassier Alain Transports	permutation de termes
Téléphone	0**1**45**6**23**0**78	0**i**45**b**23**O**78	erreurs d'OCR
Code postal	3()6**/**	3**0**6**7**	erreurs d'OCR
Date	$\xi 1/03/12$	**2**$1/03/12$	erreurs d'OCR

Tableau 6.4 – Comparaison et combinaison de mesures de similarité

Mesures et combinaisons	Résultats		
	P (%)	R (%)	F (%)
Levenshtein	49.50	97.15	65.58
Jaro	67.16	86.20	75.50
Jaro-Winkler	69.14	89.72	78.10
OCR-key	69.45	84.43	76.21
Jaccard	37.26	94.44	53.44
Monge-Elkan	44.34	97.65	60.99
Maximum (Levenshtein,OCR-key,Jaccard,Monge-Elkan,Jaro-Winkler)	94.09	80.20	86.59
Maximum (OCR-key,Jaccard,Monge-Elkan,Jaro-Winkler)	94.09	80.20	86.59
Maximum (OCR-key,Monge-Elkan,Jaro-Winkler)	**94.07**	**80.45**	**86.73**
Maximum (OCR-key,Monge-Elkan)	81.66	89.21	85.27
Moyenne (Levenshtein,OCR-key,Jaccard,Monge-Elkan,Jaro-Winkler)	74.33	94.21	83.10
Moyenne (OCR-key,Monge-Elkan,Jaro-Winkler)	70.65	95.23	81.12
Minimum (Levenshtein,OCR-key,Jaccard,Monge-Elkan,Jaro-Winkler)	29.82	96.23	45.53

des variations spécifiques de représentation d'attributs dans la base de données et dans le document (permutation de termes, erreurs d'OCR, erreurs de saisie, etc.). Des exemples de paires d'attributs montrant certaines de ces variations sont montrés dans le Tableau 6.3.

En conséquence, il sera intéressant d'étudier les manières de combiner certaines de ces mesures de façon similaire à l'étude présentée dans la Section 4.2.2.2. Pour la comparaison entre paires d'attributs, nous avons étudié expérimentalement les mesures de similarité sur le même corpus présenté dans le Tableau 6.2. En effet, nous avons comparé ces différentes mesures individuellement puis nous les avons combinées à l'aide des trois fonctions : Maximum, Minimum et Moyenne. Les résultats des comparaisons individuelles et des combinaisons les plus intéressantes (en terme de taux de classification) sont illustrés dans le Tableau 6.4. Nous constatons l'intérêt de la combinaison et notamment le Maximum entre les distances de OCR-key, Monge-Elkan et Jaro-Winkler. Cette combinaison est ainsi retenue pour cette méthode.

Il convient de noter que les résultats de combinaison de mesures de similarité obtenus ici diffèrent des résultats obtenus dans le contexte de couplage d'enregistrements (dans la Section 4.2.2.2). Ceci est expliqué par les erreurs d'OCR affectant les attributs extraits des documents.

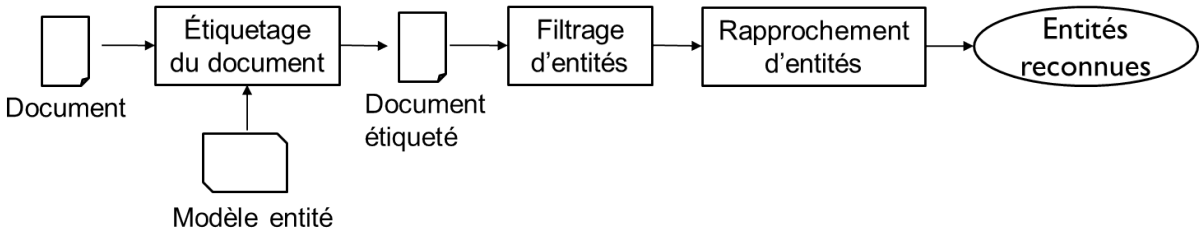


Figure 6.7 – Système global de ERBL

6.3 Méthode ERBL

Nous proposons une deuxième méthode de RE appelée ERBL (Entity Recognition Based on Labeling). Cette méthode se base sur un étiquetage des attributs d'entités dans le document puis sur un regroupement des labels obtenus en entités.

Le système global de la méthode proposée est représenté dans la Figure 6.7. D'abord, le document est étiqueté en utilisant les attributs d'entités dans le modèle entité. Ensuite, le document étiqueté est utilisé pour filtrer les entités candidates afin de réduire l'espace de recherche. Enfin, les labels sont regroupés et rapprochés avec les entités candidates.

6.3.1 Étiquetage du document

Les attributs du modèle-entité servent à étiqueter automatiquement les mots ayant les mêmes valeurs dans le document à l'aide du logiciel FullText fourni par la société ITESOFT-Yooz. Ce logiciel prend en entrée un document image et une description de données textuelles, l'océrise et recherche ensuite les données dans les lignes du document. La description des données textuelles peut être donnée sous forme de dictionnaires de valeurs ou d'expressions régulières.

Pour réaliser l'étiquetage, nous proposons de générer à partir du modèle entité :

— Des expressions régulières pour les champs de format standard tel que "code postale", "téléphone", "e-mail", "date", "montant", etc.

— Des dictionnaires d'attributs pour les champs nominaux tel que "nom", "ville", "titre", "description", etc. La création de ces dictionnaires emploie les dictionnaires d'abréviations et de sigles pour tolérer les différentes variations.

— Des dictionnaires de n-grammes de termes pour les attributs longs qui sont généralement représentés dans les documents sur plus qu'une seule ligne. Par exemple, le titre d'une référence bibliographique. Une fois étiquetés, ces n-grammes seront regroupés en un seul label.

— Des dictionnaires de contextes tels que "Tél. :", "Fax :", "TVA :", etc.

Pour tolérer les variations et les erreurs, la distance de Levenshtein est utilisée dans la comparaison des termes entre les dictionnaires et le document. De plus, les labels de contexte sont utilisés pour extraire les labels manqués et résoudre les ambiguïtés (par exemple, désambiguïser entre un téléphone et un fax qui possèdent la même expression régulière).

Un label est défini par :

$$l_i = (c_i, conf_i, v_i, sim_i)$$

avec : c_i est le champ de la base de données qui correspond au label, $conf_i$ est la confiance d'étiquetage du label l_i comme un champ c_i (i.e. l'importance du label l_i dans le champ c_i) définie par le rapport entre le nombre d'apparition du label dans le champ c_i et le nombre total de son apparition dans tous les

champs de la base de données, $v_i = \{t_k\}$ est la valeur du label représentée par un ensemble de termes et sim_i est la distance de Levenshtein entre v_i et $e.c_i$ (l'attribut de e qui correspond au champ c_i).

Chaque document en entrée est défini par un ensemble de labels comme : $D = \{l_i\}$.

6.3.2 Filtrage d'entités candidates

Comme la comparaison des labels dans le document avec toutes les entités dans le modèle entité est coûteuse, nous proposons de ne garder que les entités qui ont des chances d'être reconnues dans le document. Cela consiste à filtrer les entités à l'aide de labels en gardant seulement celles qui ont au moins un attribut en commun.

Nous avons évalué le pourcentage d'entités candidates par document par rapport au nombre total d'entités dans le modèle entité après la phase de filtrage. La Figure 6.8 représente l'histogramme des taux d'entités filtrées par document sur un échantillon de 107 documents extraits du corpus d'entreprises. La moyenne d'entités candidates par document sur cet échantillon est de seulement $4,57\%$. D'où l'intérêt du filtrage à réduire considérablement le nombre d'entités à considérer dans le rapprochement.

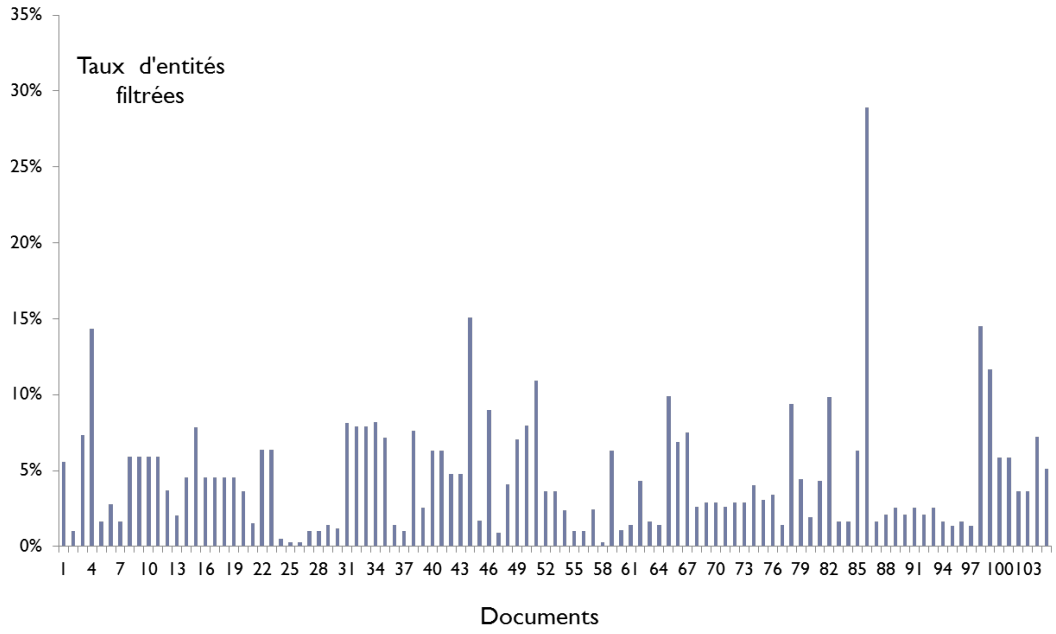


Figure 6.8 – Histogramme des taux d'entités filtrées par document sur un échantillon de 107 documents extraits du corpus d'entreprises

6.3.3 Rapprochement d'entités

Similairement à la méthode M-EROCS, nous définissons ici un score de rapprochement entre un document D défini par ses labels et une entité e dans le modèle entité E définie par ses attributs. Ce score est montré dans l'équation (6.4).

$$score(e, D) = \sum_{l_i \in F(e,D)} conf_i.sim_i \tag{6.4}$$

où $F(e, D)$ est l'ensemble des labels appartenant à D et correspondant à l'entité e ; i.e. $l_i \in F(e, D) \equiv (l_i \in d \text{ and } v_i \simeq e.c_i)$ avec $v_i \simeq e.c_i$ signifie que v_i et $e.c_i$ sont considérés similaires par rapport à sim_i.

L'ensemble de labels D est alors mis en correspondance avec l'entité e_{max} qui maximise le score (voir l'équation (6.2)). Cette entité est acceptée si son score dépasse un seuil T fixé empiriquement à l'aide d'un corpus de validation.

L'algorithme de rapprochement d'entités est décrit par l'Algorithme 4. Il propose de rapprocher un ensemble de labels D avec un ensemble d'entités candidates $setE$ en se basant sur la maximisation du score de rapprochement défini dans l'équation (6.4). D est initialisé à l'ensemble des labels dans le document. Il est ensuite mis à jour au fur et à mesure en supprimant les labels utilisés dans le rapprochement. $setE$ est initialisé à l'ensemble d'entités dans le modèle entité puis actualisé à l'ensemble d'entités filtrées. L'algorithme de rapprochement s'arrête lorsque l'un de d ou $setE$ devient vide. Il retourne ainsi l'ensemble des entités mises en correspondance.

Cet algorithme est moins complexe que celui de la méthode M-EROCS grâce à la réduction des termes traités dans le document par l'étiquetage et au filtrage d'entités candidates. De plus, la comparaison d'un label avec les attributs des entités candidates est limitée aux attributs dont les champs correspondent au champ de ce label.

Algorithme 4 ERBL

Entrées: Modèle entité : E, document : D
Sortie: Ensemble d'entités reconnues $matchE$

$matchE = \emptyset$
$setE = filter(E, D)$ // filtrage d'entités
tant que $(setE \neq \emptyset \ \& \ D \neq \emptyset)$ **faire**
    $setE = \{e \in setE \mid score(e, D) \geq T\}$
    $e_{max} = \arg\max_{e \in setE} score(e, D)$
    $D = \{l \in D \mid v \notin e_{max}.c\}$
    $setE = setE \backslash \{e_{max}\}$
    $matchE = addEntity(matchE, e_{max}, score(e_{max}, D))$
fin tant que
retourner $matchE$

6.4 Évaluations

Dans cette section, nous proposons d'évaluer les deux méthodes de RE proposées et de les comparer avec la méthode de la littérature EROCS.

Nous commençons par donner un rappel sur les corpus utilisés pour les évaluations :

1. Corpus d'entreprises
   — 278 documents ;
   — 135043 entités dans le modèle entité ;
   — 526 entités recherchées (entreprises : clients + fournisseurs).
2. Corpus d'articles scientifiques
   — 252 premières pages de documents ;
   — 250490 entités dans le modèle entité ;
   — 252 entités recherchées (d'articles scientifiques).
3. Corpus de matériaux :
   — 200 documents ;
   — 86600 entités dans le modèle entité ;
   — 630 entités recherchées (matériaux).

Tableau 6.5 – Résultats de M-EROCS sur le corpus d'entreprises ; T(s/d) est le temps de traitement d'un document par seconde

Comparaison de termes	R (%)	P (%)	F (%)	T (s/d)
Comparaison stricte	69.39	68.87	69.13	4.1
Levenshtein	73.38	71.48	72.42	4.4
OCR-key	79.47	76.00	77.70	4.5
Maximum(OCR-key,Monge-Elkan,Jaro-Winkler)	81.37	77.12	79.19	4.7

Tableau 6.6 – Résultats de M-EROCS sur le corpus d'articles scientifiques

Comparaison de termes	R (%)	P (%)	F (%)	T (s/d)
Comparaison stricte	71.43	90.00	79.65	2.9
Levenshtein	67.46	90.91	77.45	3.0
OCR-key	75.79	90.52	82.51	3.1
Maximum(OCR-key,Monge-Elkan,Jaro-Winkler)	85.32	91.49	88.30	3.3

Une entité pertinente pour un document donné est une entité mentionnée dans ce document et contenue aussi bien dans le modèle entité. Le Rappel, la Précision et la F-mesure sont définis respectivement par les équations (6.5), (6.6) et (6.7).

$$R = \frac{\#\ entit\acute{e}s\ pertinentes}{\#\ entit\acute{e}s\ correctes} \tag{6.5}$$

$$P = \frac{\#\ entit\acute{e}s\ pertinentes}{\#\ entit\acute{e}s\ retrouv\acute{e}es} \tag{6.6}$$

$$F = \frac{2.R.P}{R+P} \tag{6.7}$$

6.4.1 M-EROCS

Les résultats donnés par M-EROCS sur chacun des trois corpus sont montrés dans les Tableaux 6.5, 6.6 et 6.7. Ces résultats sont satisfaisants en terme de taux de reconnaissance (la F-mesure varie entre 79.19% et 88.3%) mais moins satisfaisants en terme de temps d'exécution (le traitement d'un document dépasse 3.3 secondes). De plus, les tableaux montrent l'intérêt de la combinaison des mesures OCR-key, Monge-Elkan et Jaro-Winkler en utilisant la fonction Maximum dans la comparaison des termes. En effet, cette combinaison donne des taux de reconnaissance (par exemple, Précision=81.31% et Rappel=77.12% dans le corpus d'entreprises) meilleurs que la mesure OCR-key seule (Précision=79.47% et Rappel=76%), la mesure Levenshtein seule (Précision=73.38% et Rappel=71.48%) ou en utilisant la comparaison stricte de termes (Précision=69.39% et Rappel=68.87%) même si le temps d'exécution augmente légèrement.

Intérêt du traitement de segments. Pour valider l'intérêt du processus de traitement de segments, nous avons comparé la RE sans et avec intégration de ce processus dans M-EROCS. Les résultats obtenus sur les trois corpus sont montrés dans le Tableau 6.8. Ils prouvent que ce processus améliore les taux de reconnaissance grâce à l'élargissement de l'espace de recherche dans le document par la fusion des segments contigus. De plus, les temps d'exécution sont maintenus presque constants. Cela est dû au fait que malgré la réduction du nombre de segments comparés par le processus de filtrage, le rapprochement est alourdit par la fusion de segments.

Tableau 6.7 – Résultats de M-EROCS sur le corpus de matériaux

Comparaison de termes	R (%)	P (%)	F (%)	T (s/d)
Comparaison stricte	79.37	78.74	79.05	3.3
Levenshtein	76.98	79.12	78.04	3.5
OCR-key	79.84	80.22	80.03	3.6
Maximum(OCR-key,Monge-Elkan,Jaro-Winkler)	80.79	80.92	80.86	3.8

Tableau 6.8 – Résultats montrant l'intérêt du traitement des segments dans M-EROCS

Corpus	Sans traitement des segments				Avec traitement des segments			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
Entreprises	77.95	73.35	75.58	4.8	81.37	77.12	79.19	4.7
Articles scientifiques	81.35	90.71	85.77	3.3	85.32	91.49	88.30	3.3
Matériaux	69.84	80.59	74.83	3.7	80.79	80.92	80.86	3.8

Seuil d'acceptation du score. Le seuil du score de RE est déterminé à l'aide de corpus de validation étiquetés manuellement. En effet, nous proposons de varier le seuil et d'évaluer pour chaque valeur les taux de Rappel et de Précision. Le seuil retenu est celui qui maximise la valeur de la F-mesure (la moyenne harmonique de la Précision et le Rappel). A titre d'exemple, nous montrons dans la Figure 6.9 la courbe de variation du Rappel, Précision et F-mesure de M-EROCS en fonction du seuil sur la base d'entreprises. La valeur retenue du seuil d'acceptation dans ce cas est 15.

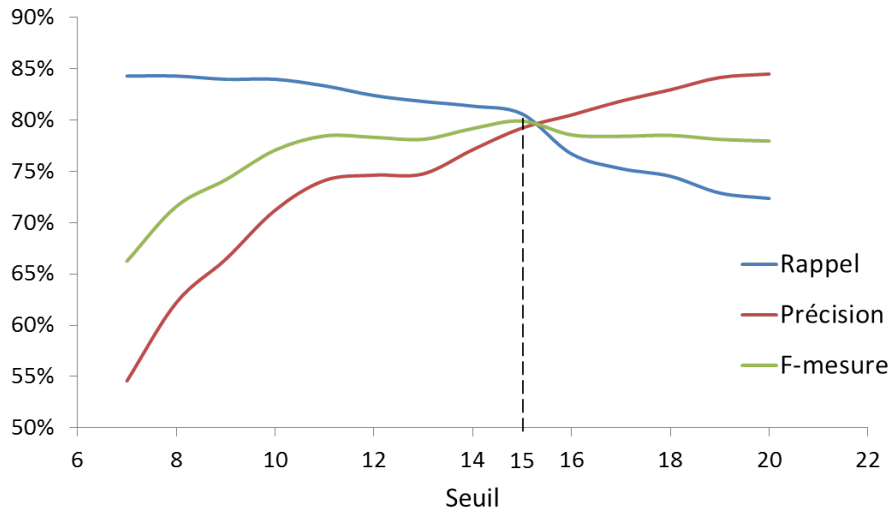


Figure 6.9 – Variation du Rappel, de la Précision et de la F-mesure de M-EROCS en fonction du seuil d'acceptation du score sur le corpus d'entreprises (valeur retenue est 15).

Intérêt de la résolution d'entités. Pour valider l'intérêt du pré-traitement de résolution d'entités dans la base de données pour la RE dans le document, nous avons évalué la reconnaissance sans et avec intégration de ce pré-traitement. Les résultats d'évaluation sur les corpus d'entreprises et d'articles scientifiques (le corpus de matériaux n'a pas été traité car sa base de données ne souffre pas du problème de la redondance d'enregistrements) sont montrés dans le Tableau 6.9. Ces résultats prouvent que la résolution

Tableau 6.9 – Résultats montrant l'intérêt de la résolution d'entités dans M-EROCS

Corpus	Avant résolution d'entités				Après résolution d'entités			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
Entreprises	79.85	74.34	76.99	6.2	81.37	77.12	79.19	4.7
Articles scientifiques	82.94	85.31	84.10	5.0	85.32	91.49	88.30	3.3

d'entités améliore considérablement les taux de reconnaissance et réduit le temps de réponse. En effet, la structuration factorisée d'entités dans le modèle entité :

— augmente la chance de retrouver l'entité dans le document grâce aux différentes variations d'attributs ;
— réduit les cas d'ambiguïté dans le rapprochement causés par l'incomplétude d'enregistrements ;
— réduit le nombre de comparaisons grâce à l'élimination des duplicatas.

Cas d'erreurs. Nous avons analysé les cas d'erreurs sur les trois corpus. Les résultats approximatifs de non détection d'entités sont montrés dans le Tableau 6.10.

Il convient de constater que les cas d'erreurs sont principalement dus à la non-contiguïté de certains éléments de l'entité dans le document (par exemple dans une facture, les données de contact d'une entreprise se retrouvent en haut de la page et les données juridiques se retrouvent en pied de page) ou encore à la sous-segmentation du document. Ce dernier cas consiste à avoir deux entités ou plus dans un même segment ce qui permet de retourner seulement l'entité qui possède le score le plus important (voir l'exemple extrait du corpus de matériaux dans la Figure 6.10). De plus, nous remarquons que le cas d'erreurs liées à la non-standardisation d'attributs est plus important dans le corpus de matériaux. Cela peut être expliqué par l'absence de l'étape de résolution d'entités dans la base de matériaux. Enfin, les autres cas d'erreurs sont liés soit à des erreurs d'OCR irrécupérables (très mauvaise qualité du document) ou une incomplétude de la base de données (plusieurs attributs non renseignés).

Tableau 6.10 – Les cas de non détection d'entités par la méthode M-EROCS

Corpus / Causes	Entreprises	Articles scientifiques	Matériaux
Erreurs d'OCR irrécupérables (%)	2	3	2
Éléments d'entité non contigus (%)	9	3	6
Sous-segmentation (%)	2	3	6
Non-standardisation (%)	-	4	5
Entités incomplètes (%)	1	2	-

6.4.2 ERBL

Les résultats de la méthode ERBL sur les trois corpus sont montrés dans le Tableau 6.11. Ces résultats montrent que le Rappel est moins important (varie entre 77.14% et 79.13%) que la Précision (varie entre 91.74% et 92.93%). Cette baisse de Rappel est expliquée par les erreurs d'étiquetage dans le document. Néanmoins, la suppression de labels mis en correspondance dans l'algorithme de ERBL a conduit à des Précisions importantes. De plus, le tableau de résultats montre l'intérêt de l'intégration du processus de filtrage d'entités dans la méthode ERBL à réduire considérablement les temps d'exécution (par exemple de 19 secondes à 0.9 secondes dans le corpus d'articles scientifiques) en maintenant les taux de reconnaissance.

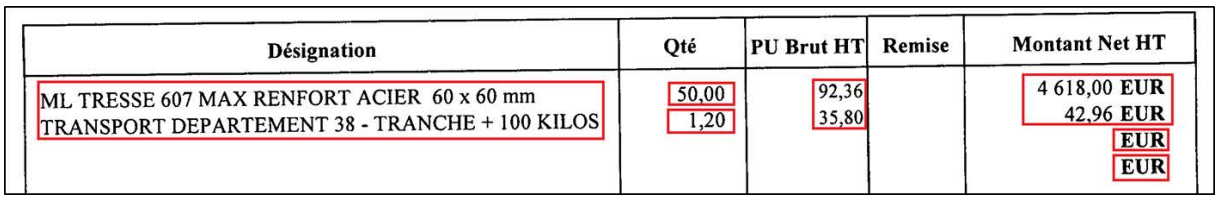


Figure 6.10 – Illustration du problème de sous-segmentation sur un exemple extrait du corpus de matériaux où les blocs donnés par l'OCR sont encadrés en rouge : les deux attributs qui correspondent à la "désignation" de deux entités différentes sont mis dans un même bloc, de même pour les attributs qui correspondent au prix unitaire et au prix net, ce qui a engendré la reconnaissance du deuxième matériau seulement.

Tableau 6.11 – Résultats de ERBL

Corpus	Sans filtrage d'entités				Avec filtrage d'entités			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
Entreprises	79.09	92.04	85.07	13.00	79.09	92.04	85.07	0.84
Articles scientifiques	79.37	91.74	85.11	19.00	79.37	91.74	85.11	0.90
Matériaux	77.14	92.93	84.30	6.00	77.14	92.93	84.30	0.85

La détermination du seuil de score de reconnaissance est fait de la même façon que celle dans la méthode M-EROCS.

Cas d'erreurs. Nous avons analysé les cas d'erreurs sur les différents corpus. Les résultats approximatifs de non détection d'entités sont montrés dans le Tableau 6.12. Ce tableau montre que les erreurs sont principalement dues à des erreurs d'étiquetage et à des labels redondants. Nous expliquons ces deux cas ci-dessous.

Tableau 6.12 – Les cas de non détection d'entités par la méthode ERBL

Causes \ Corpus	Entreprises	Articles scientifiques	Matériaux
Erreurs d'étiquetage (%)	7	13	17
Labels ambigus (%)	6	4	5
Entités incomplètes (%)	1	2	-

Erreurs d'étiquetage. Nous avons évalué l'étiquetage en utilisant un corpus annoté de 100 documents extraits aléatoirement des trois corpus. Les erreurs d'étiquetage sont réparties comme suit :

- 11% des labels sont manqués par l'étiqueteur (voir des exemples dans le Tableau 6.3) à cause des :
  - erreurs d'OCR, dans 46.33% de cas ;
  - variantes de représentation d'attributs (abréviations, erreurs de saisie, ponctuation, permutation de termes, etc.), dans 53.66% de cas.
- 5% des labels possèdent des champs erronés à cause d'expressions régulières proches (par exemple "téléphone" et "fax" ou "montant" et "code postal").
- 4% des labels sont supplémentaires (par exemple, un numéro de 4 chiffres étiqueté comme une "date").

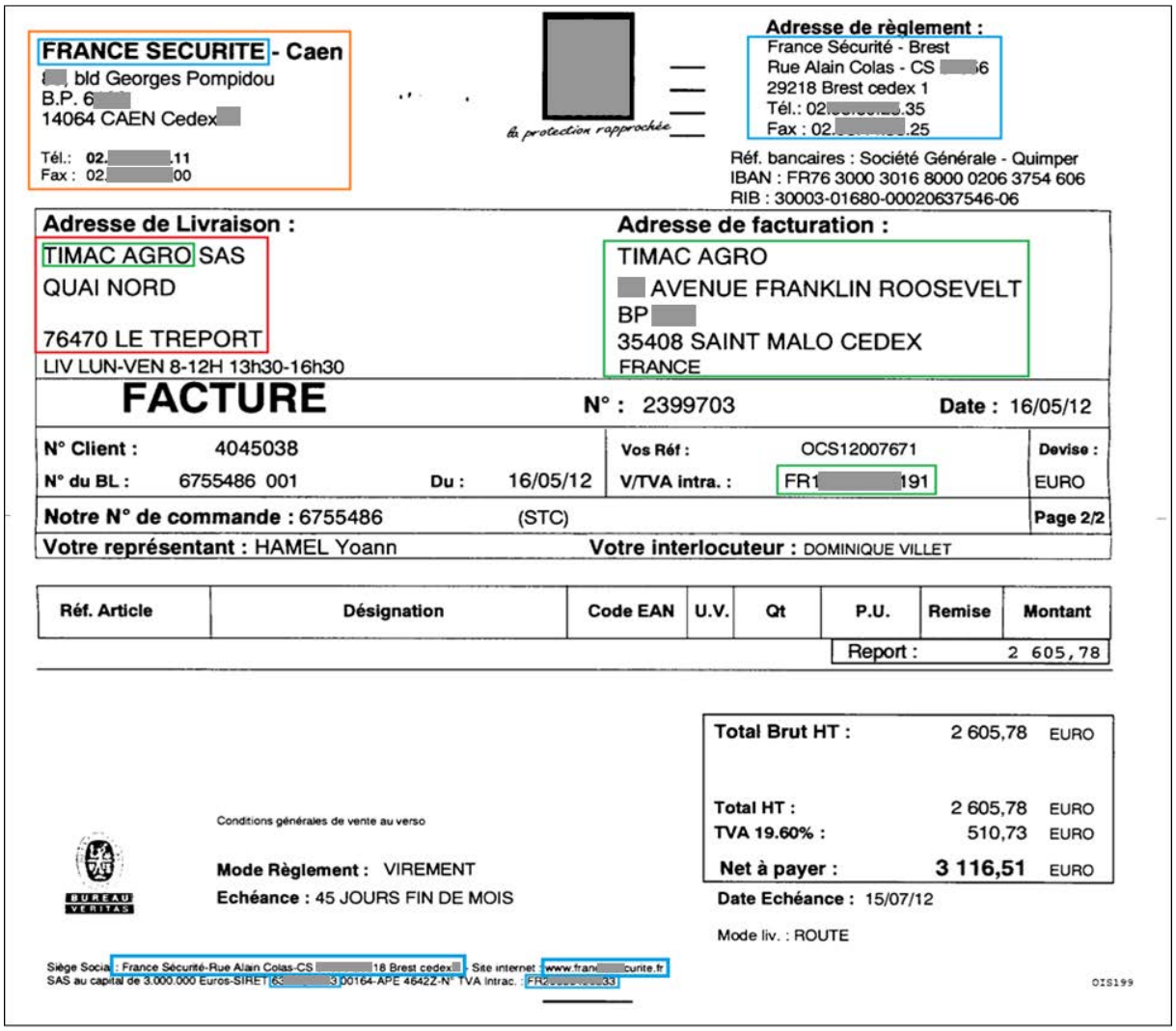


Figure 6.11 – Illustration du problème de redondance de labels sur une facture extraite de la base d'entreprises : le label de valeur "*TIMAC AGRO*", resp. "*FRANCE SECURITE*" qui se réfère à l'entité encadrée en rouge, resp. orangé est mis en correspondance par erreur avec l'entité encadrée en vert, resp. bleu.

Labels ambigus. Certaines valeurs de labels sont partagées par différentes entités dans le document, ce qui engendre des ambiguïtés dans le rapprochement. Par exemple, dans la Figure 6.11, les deux labels de valeur "TIMAC AGRO", resp. "FRANCE SECURITE" sont mis en correspondance avec l'entité encadrée en vert, resp. en bleu, ce qui peut engendrer la non reconnaissance de l'entité encadrée en rouge, resp. en orangé à cause d'une insuffisance de score. Nous constatons ici que le label d'une entité particulière est souvent proche à d'autres labels de la même entité, d'où l'intérêt de l'emploi de la structure physique dans la RE.

6.4.3 Comparaison

Nous avons comparé les deux méthodes de RE avec la méthode de l'état de l'art EROCS sur les trois corpus. Les résultats obtenus sont présentés dans le Tableau 6.13. Ces comparaisons montrent que les résultats obtenus par ERBL et M-EROCS sont beaucoup plus importants que ceux obtenus par EROCS.

Tableau 6.13 – Comparaison des méthodes de reconnaissance d'entités

Méthode	Corpus d'entreprises				Corpus d'articles scientifiques				Corpus de matériaux			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
EROCS	67.68	54.10	60.14	69.00	67.46	77.27	72.03	42.00	76.19	77.42	76.80	37.00
M-EROCS	81.37	77.12	79.19	4.70	85.32	91.49	88.30	3.30	80.79	80.92	80.86	3.80
ERBL	79.09	92.04	85.07	0.84	79.37	91.74	85.11	0.90	77.14	92.93	84.30	0.85

Cette dernière s'est avérée lente et non adaptée pour les documents océrisés puisqu'elle utilise une comparaison stricte de termes et traite le texte comme une séquence de phrases. De plus, aucune des méthodes M-EROCS et ERBL ne se démarque de l'autre. En effet, le Rappel obtenu par EROCS est plus important que celui obtenu par ERBL grâce à la comparaison de termes à l'aide de combinaison de trois mesures de similarité. Néanmoins, la Précision obtenue par ERBL est plus importante grâce à la limitation des comparaisons des termes aux attributs qui correspondent à un même champ. Enfin, le temps d'exécution de ERBL est moins important que celui de M-EROCS grâce au processus d'étiquetage qui cible la recherche dans le document et au processus de filtrage qui diminue considérablement le nombre de comparaisons.

6.5 Conclusion

Nous avons présenté deux méthodes de RE dans les documents océrisés basées sur le contenu. M-EROCS propose une comparaison intéressante entre les termes permettant de tolérer les variations et de surpasser les erreurs d'OCR. Mais cette méthode s'avère coûteuse en terme de temps d'exécution. ERBL se base sur une étape d'étiquetage du contenu de documents permettant une réduction considérable du temps d'exécution puis propose de regrouper les labels obtenus en entités. L'inconvénient de cette méthode est sa sensibilité aux erreurs d'étiquetage. L'étude des cas d'erreurs a montré l'intérêt d'employer les relations structurelles entre les labels pour corriger les erreurs d'étiquetage.

Chapitre 7

Rapprochement par les graphes de structures

Sommaire

7.1 Introduction

Pour modéliser les entités représentées par leurs labels dans les images de documents, nous utilisons les graphes. Ce choix est motivé par la représentation pratique et générique que les graphes offrent pour représenter les différents types de caractéristiques des labels et leur capacité à représenter des entités de type quelconque. Les labels peuvent être organisés de différentes manières dans la page du document : regroupés horizontalement, alignés verticalement, etc. ou peuvent présenter des relations structurelles plus complexes (voir Chapitre 2). Nous employons les graphes de structures pour modéliser ces différentes relations structurelles entre les labels d'une même entité. Comme nous travaillons avec des documents

altérés et nous traitons des données non standardisées, l'étiquetage du document n'est pas très fiable. Nous proposons d'assouplir les techniques de rapprochement existantes dans la littérature pour tolérer les opérations d'insertion, de substitution et de suppression, dans le cas de rapprochement inexact, pour corriger les erreurs d'étiquetage. Ce rapprochement de graphes est réalisé entre un graphe construit à partir d'un document étiqueté et une base de graphes donnée. Un autre problème est alors le risque d'accroissement du nombre des comparaisons du graphe construit avec tous les graphes de la base, surtout dans le cas de grandes bases. Pour pallier ce problème, une solution consiste à classifier les graphes dans le but de regrouper les graphes similaires dans un même graphe modèle.

Une étude des techniques de rapprochement de graphes et de classification non supervisée de graphes a été réalisée. Nous présentons dans la Section 7.2 les méthodes qui nous semblent les plus adaptées à notre problématique. La contribution de ce chapitre est détaillée dans la Section 7.3. Elle consiste en la proposition d'une approche de rapprochement de graphes de structures pour la reconnaissance d'entités. Une étude expérimentale a été menée sur nos trois corpus où chacun traite différents types de structures locales d'entités. Nous détaillons cette étude dans la Section 7.4.

7.2 État de l'art sur les graphes

Dans le domaine de la Reconnaissance de Formes (RF), les graphes sont fortement utilisés [Jrt94, Sebastian04, Shearer01, Tao08] pour offrir des représentations significatives des objets et leurs relations.

Nous présentons ci-dessous quelques rappels et définitions sur les graphes. Ensuite, nous détaillons les plus importantes méthodes de rapprochement inexact de graphes et de classification d'une base de graphes.

7.2.1 Définitions

Définition 1 : Graphe orienté étiqueté. Un graphe orienté étiqueté G est défini par le 4-uplet $G = (N, A, \mu, \xi)$, avec :

- N est un ensemble fini de nœuds ;
- $A \subseteq N \times N$ est un ensemble d'arcs ;
- $\mu : N \rightarrow L_N$ est une fonction qui assigne un label à un nœud, L_N étant un ensemble fini d'étiquettes pour les nœuds ;
- $\xi : A \rightarrow L_A$ est une fonction qui assigne un label à un arc, L_A étant un ensemble d'étiquettes pour les arcs.

Définition 2 : Graphe attribué. Un graphe $G = (N, A, \mu, \xi)$ est dit attribué ou relationnel attribué « Attributed Relational Graph » (ARG) lorsque G est un graphe étiqueté et :

- chaque nœud $n \in N$ est associé à un vecteur d'attributs par la fonction μ ;
- chaque arc $a \in A$ est associé à un vecteur d'attributs par la fonction ξ.

Ces attributs (dits attributs du graphe) peuvent être de type numérique, nominal, vectoriel, etc. Un graphe attribué relationnel est un cas particulier d'un graphe étiqueté.

Définition 3 : Sous-graphe. Un sous-graphe G_S de G est défini par $G_S = (N_S, A_S, \mu_S, \xi_S)$ tel que :

- $N_S \subseteq N$;
- $A_S \subseteq A \cap (N_S \times N_S)$;
- μ_S et ξ_S sont des restrictions de μ et ξ, i.e. $\forall n \in A_S$,$\forall a \in N_S$:
  - $\mu_S(n) = \mu(n)$;
  - $\xi_S(a) = \xi(a)$.

7.2.2 Types de rapprochement

Il existe deux types de rapprochement de graphes proposés dans la littérature : rapprochement exact et rapprochement inexact. Nous allons les décrire ci-après.

7.2.2.1 Rapprochement exact

Le rapprochement exact de graphes est un isomorphisme de graphes. De même, le rapprochement exact de sous-graphes est défini comme un isomorphisme de sous-graphes.

Définition 4 : Isomorphisme de graphes. Deux graphes attribués $G = (N, A, \mu, \xi)$ et $G' = (N', A', \mu', \xi')$ sont isomorphes si et seulement si il existe une correspondance un-un entre chaque nœud (respectivement chaque arc) de G et chaque nœud (respectivement chaque arc) de G'. i.e. il existe une fonction bijective $f : N \to N'$, dite isomorphisme de graphes de G vers G', telle que :

— $\forall n \in N$, $f(n) = n' \in N'$, $f^{-1}(n') = n$ et $\mu(n) = \mu'(f(n))$;
— $\forall a = (n_1, n_2) \in A$, $\exists a' = (f(n_1), f(n_2)) \in A'$ tel que $\xi(a) = \xi'(a')$;
— $\forall a' = (n'_1, n'_2) \in A'$, $\exists a = (f^{-1}(n'_1), f^{-1}(n'_2)) \in A$ tel que $\xi(a) = \xi'(a')$.

Définition 5 : Isomorphisme de sous-graphes. Il existe un isomorphisme de graphes entre G et un sous-graphe G'_S de G'. i.e. il existe une fonction injective $f : N \to N'$ telle que f soit un isomorphisme de graphes de G vers $G'_S = (N'_S, A'_S, \mu'_S, \xi'_S)$, c-à-d :

— $\forall n \in N$, $f(n) = n' \in N'_S$, $f^{-1}(n') = n$ et $\mu(n) = \mu'_S(f(n))$;
— $\forall a = (n_1, n_2) \in A$, $\exists a' = (f(n_1), f(n_2)) \in A'_S$ tel que $\xi(a) = \xi'_S(a')$;
— $\forall a' = (n'_1, n'_2) \in A'_S$, $\exists a = (f^{-1}(n'_1), f^{-1}(n'_2)) \in A$ tel que $\xi(a) = \xi'_S(a')$.

7.2.2.2 Rapprochement inexact

Le rapprochement inexact de graphes est le type de rapprochement le plus utilisé dans le domaine de la RF. Il peut être appelé aussi : rapprochement à tolérance d'erreurs. Le rapprochement inexact de graphes traite généralement des graphes étiquetés et propose de faire correspondre un graphe candidat à un graphe modèle en tolérant les cas :

— de manques de nœuds ou d'arcs (causés, par exemple, par une sous-segmentation) ;
— de surplus de nœuds ou d'arcs (causés, par exemple, par une sur-segmentation) ;
— de substitutions d'étiquettes de nœuds ou d'arcs.

De même, le rapprochement inexact de sous-graphes consiste à rechercher le meilleur rapprochement de sous-graphes en tolérant les erreurs.

Nous détaillons, dans la suite, les manières possibles pour faire cet rapprochement.

7.2.3 Méthodes de rapprochement de graphes

Des études sur les techniques de rapprochement inexact de graphes ont été proposées dans [Conte04, Foggia14]. Les plus importantes d'entre elles peuvent être classées en trois types : les méthodes de recherche arborescente, les méthodes aléatoires et les méthodes spectrales. Nous allons détailler ces méthodes ci-après.

7.2.3.1 Méthodes de recherche arborescente

Les méthodes de recherche arborescente sont les plus utilisées dans la littérature. Elles proposent de faire des parcours d'arbre avec retour en arrière en délaissant les chemins qui ne représentent pas d'appariements possibles en s'appuyant sur des heuristiques.

L'idée fondamentale ici est qu'une correspondance partielle (initialement vide) entre les nœuds du graphe est complétée successivement en y rajoutant de nouvelles paires de nœuds appariés. Le choix d'une telle paire est guidé par le coût du rapprochement de la correspondance obtenue jusqu'à cet instant et une estimation heuristique du coût de rapprochement des nœuds restants. Finalement, soit une correspondance complète est atteinte, soit un point où la correspondance actuelle ne peut pas être complétée à cause des contraintes d'appariement est atteint. Dans le deuxième cas, un retour en arrière est engendré, ensuite, la recherche est poursuivie.

Les heuristiques peuvent être utilisées pour élaguer des chemins infructueux à l'aide d'un algorithme par séparation et évaluation (branch and bound), tel que celui adopté dans [Ghahraman80, Wong90], ou encore pour déterminer l'ordre dans lequel l'arbre de recherche doit être traversé, comme dans l'algorithme A* adopté dans [Dumay92]. Plusieurs heuristiques, telles que celles proposées dans [Berretti01, Cordella98, Gregory02, Serratosa00], ont été utilisées dans le domaine de la RF pour optimiser la recherche.

Ces méthodes garantissent de trouver la meilleure solution, mais nécessitent un temps exponentiel dans le pire des cas. L'utilisation d'heuristiques joue un rôle important dans la réduction du temps de recherche.

7.2.3.2 Méthodes aléatoires

Les méthodes aléatoires proposent de transformer le problème du rapprochement de graphes d'un problème d'optimisation discret à un problème d'optimisation continu et non linéaire.

Plusieurs algorithmes d'optimisation ont été utilisés dans ce contexte. Certains emploient des techniques de relaxation probabilistes ou statistiques, comme proposé dans [Wilson97, Chevalier07, Sanroma12, Sole-Ribalta11]. D'autres utilisent des techniques d'optimisation telles que les réseaux de neurones et les algorithmes génétiques, comme proposé dans [Auwatanamongkol07].

Les algorithmes utilisés sont polynomiaux en temps de calcul (déterministes) mais risquent de ne pas trouver la solution optimale car ils ne traitent pas tous les cas. De plus, la solution trouvée doit être reconvertie du domaine continu au problème discret initial en utilisant un processus qui pourra nécessiter des nouvelles approximations.

7.2.3.3 Méthodes spectrales

Les méthodes spectrales sont basées sur l'observation suivante : les valeurs propres d'une matrice sont inéchangeables si les lignes et les colonnes sont permutées. Ainsi, étant donné les représentations matricielles de deux graphes isomorphes (par exemple, leurs matrices d'adjacence), ils ont les mêmes valeurs.

Ces méthodes, telles que proposées dans [Caelli05, Carcassoni01], représentent les graphes sous formes matricielles puis utilisent alors le calcul et la comparaison des valeurs propres (ou vecteurs propres) de ces matrices pour faire le rapprochement.

Comme le problème de calcul de valeurs propres (ou vecteurs propres) d'une matrice est un problème bien étudié et qui peut être résolu en un temps polynomial, les méthodes de rapprochement de graphes spectrales sont alors rapides en temps de calcul. Par contre, comme la réciproque de l'observation ci-dessus est fausse, les techniques spectrales ne garantissent pas alors l'optimalité de la solution trouvée.

Une limite importante de ces techniques est qu'elles sont purement structurelles car la représentation matricielle proposée ne modélise que la structure des nœuds et des arcs et n'exploite pas leurs attributs. Pour pallier cette limite, certaines techniques spectrales, comme celle proposée dans [Xu01], traitent seulement les poids réels attribués aux arcs en utilisant une matrice d'adjacence avec des éléments à valeurs réelles.

La plupart des méthodes de rapprochement de graphes que nous venons de présenter, se basent sur le calcul de similarité dans le but de favoriser une correspondance entre les nœuds de deux graphes par rapport à d'autres correspondances possibles. Nous présentons ci-dessous les mesures de similarité utilisées dans la littérature.

7.2.4 Mesures de similarité entre graphes

Plusieurs mesures de similarité entre graphes ont été proposées dans la littérature. Nous détaillons les plus importantes ci-dessous.

7.2.4.1 Distance d'édition de graphes

La distance d'édition de graphes « Graph Edit Distance » (GED) a été introduite par Sanfeliu et Fu dans [Sanfeliu83], comme étant une reformulation de la distance d'édition de chaînes de caractères, telle que la distance de Levenshtein [Levenshtein66], dans le domaine des graphes. La distance GED est utilisée dans le cas d'appariement inexact de graphes en tolérant le bruit et les altérations. Elle est définie par la séquence des opérations d'éditions des nœuds et des arcs pour transformer un graphe $G = (N, A, \mu, \xi)$ en un graphe $G' = (N', A', \mu', \xi')$. Pour simplifier, nous présentons seulement les opérations pour les nœuds :

- Substitution : $\mu(n) \rightarrow \mu'(n')$ avec $n \in N$ et $n' \in N'$. C'est la substitution d'attribut d'un nœud. Dans le cas de la substitution identique, c.-à-d. $\mu(n) = \mu'(n')$, l'opération n'est pas comptée.
- Suppression : $n \rightarrow \epsilon$, où ϵ représente le vide.
- Insertion : $\epsilon \rightarrow n'$.

Soit δ un chemin complet de transformation de G à G'. Il est dénoté par une séquence d'opérations d'édition $\delta = (\delta_1, \ldots, \delta_i) = (n \rightarrow n', \ldots, n \rightarrow \epsilon)$. A chaque opération d'édition δ_i, nous associons un coût d'édition $c(\delta_i)$. La distance d'édition de graphes est définie alors par l'équation (7.1).

$$d_{GED}(G, G') = min_{\delta \in \Delta} \sum_i c(\delta_i) \tag{7.1}$$

avec Δ l'ensemble fini des chemins d'éditions de G vers G'.

7.2.4.2 Distances basées sur mcs et MCS

Avant de présenter ces distances, nous commençons par définir les concepts qui étudient les parties communes entre les graphes.

Définition 6 : Sous-graphe commun. Un graphe G est un sous-graphe commun entre deux graphes $G1$ et $G2$ s'il existe des isomorphismes de sous-graphes de G vers $G1$ et de G vers $G2$.

Définition 7 : Plus grand sous-graphe commun. Un sous-graphe G commun à $G1$ et $G2$ est dit plus grand sous-graphe commun « maximum common subgraph » (mcs) s'il n'existe pas d'autre sous-graphe commun G' à $G1$ et $G2$ qui a plus de nœuds que G. Le mcs de deux graphes donnés $G1$ et $G2$ n'est pas nécessairement unique.

Définition 8 : Super-graphe commun. Un graphe G est un super graphe commun à deux graphes $G1$ et $G2$ s'il existe des isomorphismes de sous-graphes de $G1$ vers G et de $G2$ vers G.

Définition 9 : Plus petit super-graphe commun. Un super graphe G commun à $G1$ et $G2$ est dit plus petit super graphe commun « Minimum Commun Supergraph » (MCS) s'il n'existe pas d'autre super graphe commun G' à $G1$ et $G2$ qui a moins de nœuds que G. Le MCS de deux graphes donnés $G1$ et $G2$ n'est pas nécessairement unique.

Nous présentons, ci-après, des exemples de distances utilisées dans la littérature et qui emploient les concepts de mcs et MCS. Ces distances ont été comparées dans [Schenker03] dans le cadre de la classification non supervisée d'une collection de documents modélisés par des graphes.

— $d_1(G, G') = 1 - \frac{|mcs(G,G')|}{max(|G|,|G'|)}$;
où $|G|$ dénote le nombre de nœuds dans le graphe G. Cette distance a été proposée par Bunke, dans [Bunke98], comme alternative à la distance d'édition de graphes.

— $d_2(G, G') = 1 - \frac{|mcs(G,G')|}{|G|+|G'|-|mcs(G,G')|}$;
Cette distance [Wallis01] considère la taille du mcs de deux graphes G et G' par rapport à la taille de leur union.

— $d_3(G, G') = |G| + |G'| - 2|mcs(G, G')|$;
Cette distance, proposée dans [Bunke97], calcule la différence entre les tailles du mcs et du MCS de deux graphes G et G'.
Bunke a analysé, dans [Messmer94] et [Bunke97], la relation entre le GED et le mcs. Il a démontré que, pour une affectation appropriée des coûts d'opérations d'éditions, le calcul de la distance d'édition de graphes est équivalent à résoudre le problème du plus grand commun sous-graphe. En effet, en assignant un coût égal à 1 aux opérations de suppression des nœuds et des arcs et un coût égal à 0 aux autres opérations, la distance d'édition de graphes devient équivalente à la distance d_3.

— $d_4(G, G') = |MCS(G, G')| - |mcs(G, G')|$;
Cette distance, proposée dans [Fernandez01], calcule la différence entre les tailles du mcs et du MCS de deux graphes G et G'.
Les distances d_3 et d_4 sont non normalisées dans l'intervalle $[0, 1]$.

— $d_5(G, G') = 1 - \frac{|mcs(G,G')|}{|MCS(G,G')|}$;
Cette distance est une version normalisée de la distance d_4.

7.2.4.3 Sondage de graphes

Le sondage de graphe, proposé dans [Lopresti03], consiste à représenter les nœuds d'un graphe et les structures des arcs reliant ces nœuds sous forme de vecteurs et étudier ainsi la différence des vecteurs pour comparer deux graphes donnés. Il est applicable dans les cas des graphes non orientés et non étiquetés ainsi que les graphes orientés et étiquetés.

Le sondage de graphes est une mesure de similarité entre graphes qui n'est pas considérée comme distance puisqu'elle n'obéit pas à la propriété d'unicité. La complexité de comparaison entre graphes en utilisant le sondage de graphes est linéaire en nombre d'arcs et de nœuds. Dans le cas des graphes étiquetés, le sondage de graphes suppose que les étiquettes sont des labels nominaux et ne peut pas être adaptée dans le cas des graphes attribués.

7.2.4.4 Mesures de similarité entre graphes attribués

Le problème avec les mesures de similarité que nous avons présentées est qu'elles ne sont pas facilement adaptées lorsqu'il n'existe pas de correspondance exacte entre les étiquettes des nœuds et/ou arcs des graphes. C'est le cas, par exemple, des graphes où les étiquettes des nœuds et des arcs représentent un vecteur d'attributs résultant d'une étape d'extraction de caractéristiques et sont alors très exposés au bruit. Une première solution pour faire face à ce problème se base sur une discrétisation ou une classification pour transformer le vecteur d'attributs numériques en une étiquette nominale, puis comparer directement ces étiquettes nominales. L'inconvénient majeur de ces approches est leur sensibilité aux effets de la frontière de la discrétisation ou à la mauvaise classification. Un algorithme d'appariement pourrait alors échouer par la suite. Une deuxième solution consiste à apparier les nœuds et les arcs en comparant les distances entre les valeurs de leurs étiquettes. Le principal inconvénient de ces approches est leur dépendance des seuils. Une dernière façon consiste à utiliser une procédure de rapprochement de graphes qui intègre les distances entre les attributs lors de l'appariement.

Dans le contexte de ce type de graphes attribués, un algorithme d'isomorphisme de sous-graphes qui tolère les erreurs de substitution a été proposé dans [LeBodic12]. Cet algorithme propose de rechercher un isomorphisme de sous-graphes entre un graphe candidat $G_S = (N_S, A_S, \mu_S, \xi_S)$ et un graphe modèle $G = (N, A, \mu, \xi)$ en reformulant le problème d'appariement en un formalisme de programmation linéaire en nombres entiers et en minimisant la fonction globale de coût, définie par l'équation (7.2).

$$min_{x,y}(\sum_{i \in N_S} \sum_{k \in N} c_N(i,k).x_{i,k} + \sum_{ij \in A_S} \sum_{kl \in A} c_A(i,k).y_{ij,kl}) \tag{7.2}$$

Avec $c_N : N_S \times N \to \mathbb{R}^+$ et $c_A : A \times A_S \to \mathbb{R}^+$ deux fonctions de coût données respectivement pour les nœuds et pour les arcs. Pour deux nœuds $i \in A_S$ et $k \in A$, $x_{i,k} = 1$ si i et k correspondent (leur correspondance coûte $c_N(i,k)$) et $x_{i,k} = 0$ sinon ; de même pour deux arcs $ij \in A_S$ et $kl \in A$.

Les insuffisances de cet algorithme est qu'il suppose une correspondance exacte entre les nœuds et les arcs et ne permet pas les opérations de suppression dans le graphe candidat.

7.2.5 Organisation d'une base de graphes

Dans le cas des modèles sous forme de bases de graphes, il est nécessaire d'appliquer un appariement du graphe candidat avec tous les graphes modèles, ce qui est linéairement dépendant du nombre de graphes dans la base de graphes. Ceci peut être lourd dans le cas de grandes bases. Il est alors intéressant d'utiliser des caractéristiques spécifiques et facilement calculables du graphe candidat pour filtrer la base de graphes, comme montré dans [Irniger03], et limiter ainsi les graphes modèles comparés. Pour pallier ce problème, une première solution consiste à une organisation hiérarchique de la base de graphes permettant de guider la recherche par la suite. Une deuxième solution consiste à la classification non supervisée (en anglais, clustering) de la base de graphes afin regrouper les graphes similaires puis à la représentation de chaque classe par un seul graphe représentatif (graphe modèle). Nous citons des méthodes de la littérature proposées dans le cadre de ces deux solutions.

7.2.5.1 Organisation hiérarchique d'une base de graphes

Une organisation hiérarchique de la base de graphe permettra de réduire considérablement la complexité de la comparaison entre le graphe candidat et la base de graphes. Nous détaillons, ci-dessous, quelques méthodes proposées dans ce contexte.

Dans [Seong93], les auteurs ont proposé d'organiser la base de graphes hiérarchiquement en regroupant de manière continue les graphes modèles dans des classes de similarité. En effet, pour une nouvelle

instance de graphe, une des 5 opérations : recherche, création, fusion, découpage et terminaison est effectuée selon le résultat d'un test. Ce dernier consiste à étudier les manières possibles de l'intégration de la nouvelle instance dans la hiérarchie déjà construite. Dans ce cas, un graphe candidat est initialement apparié avec le premier niveau de la hiérarchie pour choisir la classe de graphes modèles la plus proche de ce graphe candidat. Ensuite, le même processus est répété avec les graphes modèles de cette classe jusqu'à se retrouver avec un nombre limité de graphes modèles. Mais il n'est pas toujours possible de retrouver une hiérarchie claire entre les graphes modèles de la base.

Dans [Messmer99], Messmer et Bunke ont proposé de construire un arbre de décision qui organise les graphes de la base hiérarchiquement. Pour ce faire, chaque graphe est représenté par une matrice d'adjacences et toutes les permutations possibles de chaque matrice sont générées. Ensuite, chaque matrice est représentée par un ensemble de vecteurs qui seront, par la suite, organisés dans l'arbre de décision. Chaque vecteur est représenté une seule fois dans l'arbre de décision s'il est partagé par plusieurs graphes. Les feuilles de cet arbre représentent l'ensemble des graphes de la base. Le problème avec cette méthode est qu'elle exige une représentation matricielle du graphe ce qui n'est pas toujours possible surtout dans le cas de graphes attribués. De plus, la construction de l'arbre de décision se fait au préalable et ne se modifie pas dans l'étape de comparaison de la base avec les graphes candidats.

Les auteurs ont proposé dans [Messmer98] une méthode proche qui consiste à découper les graphes de la base en un ensemble de sous graphes jusqu'à ce que les sous-graphes soient des nœuds singuliers, chaque sous-graphe étant représenté une seule fois dans la base. Cette décomposition de la base réduit considérablement le temps d'appariement puisque les sous-graphes qui sont partagés par plusieurs graphes dans le modèle vont être comparés une seule fois avec le graphe candidat. L'inconvénient de cette méthode est qu'elle suppose qu'il existe des sous-graphes exactement identiques en commun entre les graphes modèles, ce qui n'est pas toujours le cas surtout pour les graphes attribués.

7.2.5.2 Classification non supervisée d'une base de graphes

Pour mieux organiser une base de graphes, il est intéressant de regrouper les graphes similaires dans un seul graphe représentatif. Pour ce faire, il faut classifier les graphes dans la base puis représenter les graphes exemples de chaque classe par un graphe modèle. Nous présentons dans cette section, un aperçu sur les méthodes de l'état de l'art proposées pour la classification non supervisée de données. Ensuite, nous nous intéressons à la représentation d'une classe dans le cas de base de graphes.

Méthodes de classification non supervisée. La classification non supervisée consiste à regrouper les données en des ensembles homogènes, dites classes ou clusters, sans avoir besoin de données étiquetées. Plusieurs méthodes de classification non supervisée ont été proposées dans la littérature. Ces méthodes ont été décrites dans [Berkhin06]. Nous reprenons quelques-unes ci-dessous.

Les méthodes de classification non supervisée peuvent être rangées en deux catégories : les méthodes de partitionnement et les méthodes hiérarchiques.

Les méthodes de partitionnement sont des algorithmes, dits k-prototypes, qui divisent les données en un nombre prédéterminé k de clusters. L'algorithme le plus reconnu est le k-means qui s'applique sur des données numériques et qui se base sur les distances (par exemple la distance euclidienne ou cosinus) pour mesurer la similarité entre ces données et assigner chacune à son plus proche cluster. Ces méthodes sont de complexité linéaire $\mathcal{O}(n)$ en temps mais elles exigent de prévoir le nombre de clusters k, ce qui n'est pas toujours possible.

Les méthodes hiérarchiques peuvent être de division ou agglomératives. Les méthodes de division (descendantes) commencent par regrouper les données en un seul cluster, puis proposent de construire une hiérarchie en divisant successivement les données en des blocs plus petits. Les méthodes agglomératives (ascendantes) considèrent au début des clusters singletons puis construisent la hiérarchie en les

fusionnant successivement. L'inconvénient de ces méthodes est qu'elles nécessitent une grande capacité de mémoire et une complexité carrée $\mathcal{O}(n^2)$ en temps.

Certaines méthodes hybrides proposent de combiner différents types de méthodes de classification non supervisée. C'est le cas des méthodes proposées dans [Srinivas10] et [Tanaseichuk15].

Méthodes de classification non supervisée incrémentale. Dans le cas de flux continu de données, il est nécessaire de prévoir un algorithme de classification non supervisée qui soit capable de gérer des nouvelles données arrivantes appartenant à des classes existantes ou représentant de nouvelles classes. Nous présentons les principaux algorithmes proposés dans le contexte de la classification non supervisée incrémentale.

Plusieurs versions incrémentales du fameux algorithme k-means ont été proposées dans la littérature (par exemple les travaux dans [Khy06, Aaron14]). Une revue de ces versions prouvant leur efficacité et leur rapidité par rapport à la version classique est présentée dans [Yadav15]. De plus, certains algorithmes n'exigent plus la spécification préalable du nombre de clusters k.

Un algorithme de classification non supervisée incrémentale, basé sur les réseaux de neurones, appelé IGNG « Incremental Growing Neural Gas », est proposé dans [Prudent05]. Cet algorithme propose d'initialiser l'espace avec un ou plusieurs neurones puis de créer des neurones au fur et à mesure de l'apparition des données. Le regroupement des neurones en classes est basé sur la minimisation de la distance qui sépare un nouveau neurone des représentatifs des classes existantes en utilisant un seuil global pour permettre la création de nouvelles classes. L'avantage de cet algorithme est qu'il permet la gestion statistique de l'apparition des neurones pour supprimer des classes existantes ou rajouter des nouvelles classes.

Une amélioration de cet algorithme qui permet d'adapter le seuil en fonction des données arrivantes, appelé I2GNG « Incremental Growing Neural Gas », est proposée dans [Hamza08].

Ces algorithmes sont bien adaptés à classer les grandes bases mais sont dépendants des paramètres initiaux et risquent de devenir lents si l'espace se remplit de neurones.

Un algorithme de partitionnement incrémental simple, appelée algorithme des dirigeants, a été proposé dans [Vijaya04] (voir Algorithme 5). Cet algorithme propose de représenter chaque classe (cluster dans l'Algorithme 5) par un dirigeant et d'attribuer au fur et à mesure les données aux classes existantes ou de créer des nouvelles classes.

Algorithme 5 Algorithme des dirigeants [Vijaya04]

```
1: sélectionner le seuil
2: initialiser un dirigeant, le rajouter à la liste des dirigeants et mettre le compteur des dirigeants L = 1
3: pour chaque patron ; i = 2 → n faire
4:     calculer la distance avec tous les dirigeants
5:     rechercher le dirigeant le plus proche
6:     si distance avec le dirigeant le plus proche < seuil alors
7:         l'attribuer au dirigeant le plus proche
8:         marquer le numéro du cluster
9:         le rajouter à la liste des membres de ce cluster
10:        incrémenter le compteur des membres de ce cluster
11:    sinon
12:        le rajouter à la liste des dirigeants
13:        incrémenter le compteur des dirigeants L = L + 1
14:    fin si
15: fin pour
```

Cet algorithme est non coûteux en temps de calcul car il nécessite un seul parcours des données et ne nécessite pas beaucoup de mémoire car il représente chaque classe par son dirigeant. Le principal inconvénient de cet algorithme est qu'il est dépendant de l'ordre d'arrivée de données.

Représentant dans le cas d'une base de graphes. Dans le domaine de la RF, certaines méthodes, telles que celles proposées dans [Shapiro82, Bunke03, Gunter02, Sanfeliu02, Luo02], se sont intéressées au problème de la représentation d'un ensemble de graphes exemples d'une classe par un seul graphe représentatif.

La méthode proposée dans [Shapiro82] dans le contexte de l'analyse de scènes suppose que tous les graphes de la base possèdent n nœuds et représente chaque graphe par un vecteur de n^2 éléments représentant les possibles arcs (la valeur de l'élément est 1 si un arc existe et 0 sinon). Elle propose de classifier la base de graphes en se basant sur une simple distance définie entre deux graphes. Si la distance entre deux graphes est inférieure à un seuil prédéfini alors ces graphes appartiennent à une même classe. Cette méthode propose trois différentes manières pour définir un graphe représentant de chaque classe : le meilleur graphe, le graphe moyen et le graphe médian. Le meilleur graphe est défini par le graphe qui minimise la somme des distances par rapport à tous les graphes dans la classe. Le graphe moyen est défini par les arcs communs à tous les graphes de la base. Le graphe médian est défini par les arcs communs à, au moins, la moitié des graphes de la base de graphes. Les expérimentations ont montré qu'aucun de ces représentants ne se démarque.

Dans [Bunke03], les auteurs proposent d'utiliser le concept de plus grand super graphe commun pondéré « Weighted Minimum Common Supergraph (WMCS) ». Cette représentation favorise les données partagées, les distingue du bruit et des éventuelles distorsions et permet une représentation complète de la structure locale mise en jeu. Nous détaillons ci-dessous la construction du WMCS.

WMCS. Un graphe pondéré est défini par un 6-uplet $G = (N, A, \mu, \xi, \alpha, \beta)$, avec :
— N est un ensemble fini de nœuds.
— $A \subseteq N \times N$ est un ensemble d'arcs, $a_{ij} \in A$ liant le couple de nœuds (n_i, n_j) avec $n_i, n_j \in N$.
— $\mu : N \rightarrow L_N$ est une fonction qui attribue un label à un nœud, L_N étant un ensemble fini d'étiquettes pour les nœuds.
— $\xi : A \rightarrow L_A$ est une fonction qui attribue un label à un arc, L_A étant un ensemble d'étiquettes pour les arcs.
— $\alpha : N \rightarrow \mathbb{N}^+$ est une fonction qui attribue un poids positif à un nœud.
— $\beta : A \rightarrow \mathbb{N}^+$ est une fonction qui attribue un poids positif à un arc.

Un super graphe commun pondéré d'un ensemble de graphes $S_G = \{G_1, \ldots G_n\}$ est un graphe pondéré $G = (N, A, \mu, \xi, \alpha, \beta)$ tel que il existe un isomorphisme de sous-graphes de G_i dans G, $\forall i \in \{1, \ldots, n\}$.

G est dit être le plus grand super graphe commun pondéré de S_G, WMCS(S_G), s'il n'existe aucun autre super graphe commun de S_G qui a moins de nœuds que G.

Le calcul de WMCS d'un ensemble de graphes est un problème NP-complet. Bunke [Bunke03] a proposé une approximation basée sur le calcul de WMCS de deux graphes. Ceci consiste à initialiser le WMCS à un graphe de l'ensemble en fixant les poids des arcs et des nœuds à 1 puis à parcourir les graphes restants en calculant à chaque fois le WMCS entre le WMCS courant et le graphe étudié. Le calcul du WMCS de deux graphes consiste à comparer leurs arcs et leurs nœuds et à incrémenter les poids des parties communes. Nous avons alors les poids de tous les arcs et les nœuds (voir l'exemple dans la Figure 7.1). Ces poids renseignent sur la fiabilité des éléments dans le graphe modèle. Ainsi, seuls les nœuds et les arcs qui ont des poids importants (supérieurs à un poids donné) sont retenus. Les autres sont considérés comme du bruit et sont alors supprimés du graphe résultat.

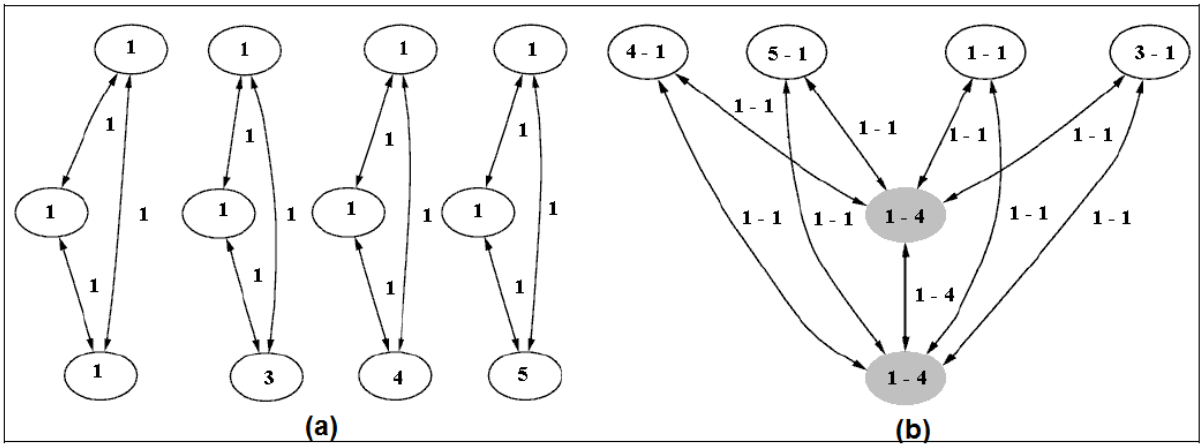


Figure 7.1 – Exemple de WMCS, extrait de [Bunke03] : (a) ensemble de graphes d'une classe, (b) le WMCS correspondant. Sur chaque nœud (arc) du WMCS, il existe une deuxième étiquette pour le poids représentant son occurrence dans la classe.

Synthèse

A partir de cette étude théorique, nous adoptons les choix suivants dans notre démarche de RE à base de rapprochement de graphes de structures :

- Pour la représentation des labels d'entités, nous utilisons les graphes attribués permettant de décrire les caractéristiques du contenu et de la structure.
- Nous traitons le problème de rapprochement inexact de graphes afin de tolérer les erreurs d'étiquetage par les opérations d'insertion, de substitution et de suppression. Plus particulièrement, nous nous intéressons aux méthodes arborescentes en utilisant des règles qui intègrent les attributs de graphes (attributs de nœuds et des arcs). Ces règles permettront d'optimiser la recherche de la meilleure solution.
- Pour la mesure de similarité de graphes, nous nous basons sur la distance d'édition de graphes. Nous proposons une fonction de coût appropriée qui intègre la dissimilarité entre les attributs de nœuds et d'arcs.
- Pour la classification de notre base de graphes, nous nous appuyons sur l'algorithme des dirigeants. Ce choix est motivé par le fait qu'il s'agit d'un algorithme incrémental simple qui nécessite un seul balayage de la base. De plus, une étude expérimentale sur les données de nos corpus a montré leur indépendance de l'ordre d'arrivée. Pour représenter les graphes modèles, nous utilisons le concept de WMCS permettant d'éliminer les données bruitées.
- Pour le filtrage des graphes de la base, nous utilisons des heuristiques sémantiques et structurelles spécifiques à nos graphes de structure.

Dans la section suivante, nous détaillons notre approche de RE qui se base sur ces choix.

7.3 Méthode proposée : G-ELSE

Le schéma global de la méthode proposée pour la RE, appelée G-ELSE (Graph matching for Entity Local Structure Extraction), est montré dans la Figure 7.2. Il prend en entrée un document étiqueté et un modèle entité. Les labels du document proches physiquement dans le document étiqueté sont représentés par un graphe, dit graphe de structure locale. Ce graphe est comparé (rapproché) au modèle structurel, représenté aussi par des graphes de structures locales. Le modèle structurel est initialement appris à partir

d'un échantillon représentatif du corpus, puis mis à jour au fur et à mesure. Le résultat du rapprochement est utilisé pour corriger les erreurs d'étiquetage éventuelles et ensuite valider la RE.

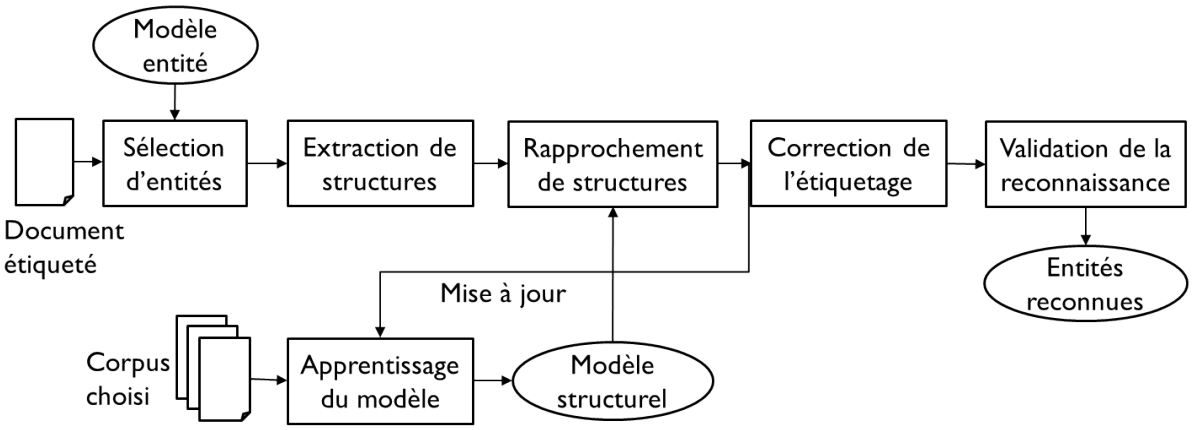


Figure 7.2 – Schéma global de la méthode G-ELSE

7.3.1 Sélection d'entités

Les labels du document sont regroupés en entités en maximisant le même score de rapprochement défini dans l'équation (6.4). Les entités dont le score dépasse un seuil T fixé empiriquement, sont considérées comme des entités candidates.

7.3.2 Extraction de structures locales

Une structure locale est définie par un ensemble de labels d'une même entité, proches physiquement dans une page. La proximité physique entre les labels est étudiée à l'aide de deux seuils : un seuil vertical et un seuil horizontal. Ces deux seuils sont déterminés à l'issue d'une observation sur la distance qui sépare les labels appartenant à une même structure locale sur chaque corpus. Des exemples de regroupement des labels d'une même entité en structures locales dans des documents de nos corpus sont présentés dans la Figure 7.3 (corpus d'entreprises), la Figure 7.4 (corpus d'articles scientifiques) et la Figure 7.5 (corpus de matériaux). Les labels encadrés par la même couleur appartiennent à une même structure locale.

Nous constatons, d'après ces exemples, que les structures locales d'entités peuvent présenter différents types de structures : structure d'adresse postale, structure de données de contact en formulaire, structure d'entête d'un article scientifique, structure de ligne d'un tableau, etc. Pour modéliser ces structures locales, nous utilisons les graphes attribués afin de décrire différents types de caractéristiques des labels : le contenu textuel, la mise en page et les relations géométriques entre eux. De plus, comme les erreurs d'étiquetage sont fréquentes, nous cherchons à profiter des opérations de substitution, d'insertion et de suppression dans le rapprochement inexact de graphes pour corriger ces erreurs (champs de labels erronés, labels non trouvés, labels supplémentaires).

Une structure locale est modélisée par le graphe attribué $G = (N, A, \mu, \xi)$, où N est un ensemble fini de nœuds correspondant aux labels du document, $A \subseteq N \times N$ est un ensemble fini d'arcs représentant les relations géométriques entre les labels, $\mu : N \rightarrow L_N$ et $\xi : A \rightarrow L_A$ sont deux fonctions qui affectent des vecteurs d'attributs respectivement aux nœuds et aux arcs. L_N et L_A sont des ensembles discrets de vecteurs d'attributs. Un arc $a_{ij} \in A$ reliant les nœuds n_i et n_j est représenté par $n_i n_j$.

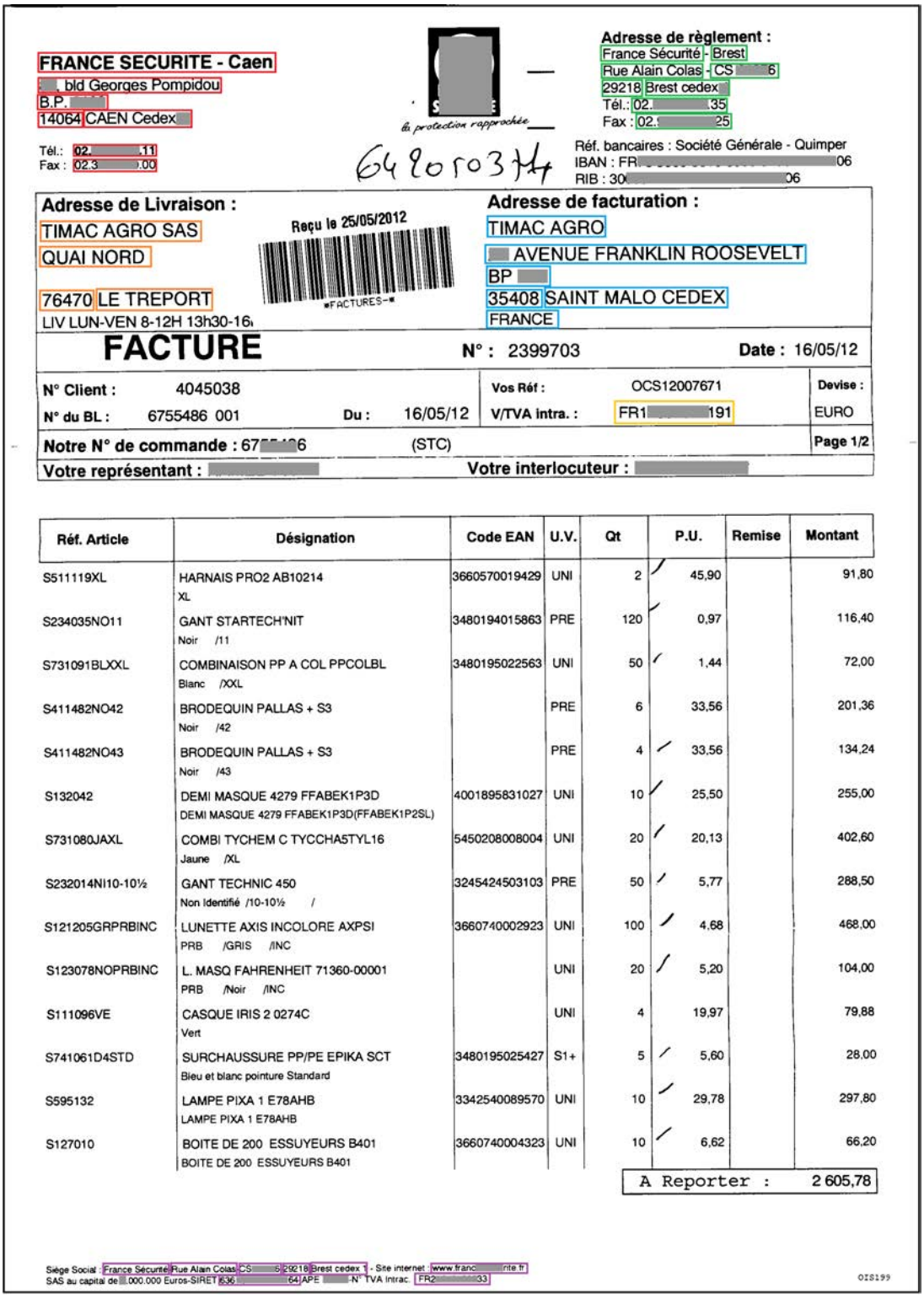


Figure 7.3 – Exemple d'une facture extraite du corpus d'entreprises montrant le regroupement des labels en des structures locales. Les labels encadrés par la même couleur appartiennent à une même structure locale (structure d'adresse postale, structure d'informations fiscales, structure mixte, etc.).

Pattern Recognition 46 (2013) 551–565

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Fuzzy multilevel graph embedding

Muhammad Muzzamil Luqman [a,b,*], Jean-Yves Ramel [a], Josep Lladós [b], Thierry Brouard [a]

[a] Laboratoire d'Informatique, Université François Rabelais de Tours, 37200, France
[b] Computer Vision Center, Universitat Autònoma de Barcelona, 08193, Spain

ARTICLE INFO

Article history:
Received 9 December 2011
Received in revised form
10 July 2012
Accepted 31 July 2012
Available online 10 August 2012

Keywords:
Pattern recognition
Graphics recognition
Graph clustering
Graph classification
Explicit graph embedding
Fuzzy logic

ABSTRACT

Structural pattern recognition approaches offer the most expressive, convenient, powerful but computational expensive representations of underlying relational information. To benefit from mature, less expensive and efficient state-of-the-art machine learning models of statistical pattern recognition they must be mapped to a low-dimensional vector space. Our method of explicit graph embedding bridges the gap between structural and statistical pattern recognition. We extract the topological, structural and attribute information from a graph and encode numeric details by fuzzy histograms and symbolic details by crisp histograms. The histograms are concatenated to achieve a simple and straightforward embedding of graph into a low-dimensional numeric feature vector. Experimentation on standard public graph datasets shows that our method outperforms the state-of-the-art methods of graph embedding for richly attributed graphs.

1. Introduction

Pattern recognition has emerged as an important research domain and has supported the development of numerous applications in many different areas of activity. For a general introduction to former we refer the interested reader to [1,2]. The methods for pattern recognition are broadly categorized as statistical, structural or syntactic approaches [3]. In this paper we address the problem of lack of computational tools for structural pattern recognition and propose to exploit the computational efficiency of statistical pattern recognition. This permits a pattern recognition application to benefit from representational power of structural methods and computational efficiency of statistical methods, while avoiding the limitations of both. The next two paragraphs briefly introduce the main advantages and limitations of structural and statistical pattern recognition.

Structural pattern recognition is characterized by the use of symbolic data structures i.e. graphs, strings and trees. Graphs are widely used in structural pattern recognition and can safely be termed as representative of symbolic data structures (strings and trees are special instances of graphs [4]). Graphs provide a convenient and powerful representation of relational information. They are able to represent not only the values of both symbolic and numeric properties of an object, but can also explicitly model the spatial, temporal and conceptual relations that exist between its parts. Moreover, graphs do not suffer from the constraint of fixed dimensionality. For example, the number of nodes and edges in a graph is not limited a priori and depends on the size and the complexity of the actual object to be modeled [5]. And above all, graphs have foundations in strong mathematical formulation and have a mature theory at their basis. However two serious drawbacks of graph based representations are that these representations are sensitive to noise and that the algorithmic tools for performing different operations on them are computational expensive. For instance the much needed operations of graph matching and graph isomorphism are NP-complete. For further reading on structural pattern recognition we refer the interested reader to [4–8].

Statistical pattern recognition is characterized by the use of numeric feature vectors. A very important advantage of these representations is that because of their simple structure, the basic operations that are used in machine learning can easily be executed on them. This makes a large number of mature algorithms for pattern analysis and classification immediately available to statistical pattern recognition. And, as a result of this fact, the statistical pattern recognition offers state-of-the art computational efficient tools of learning, classification and clustering. However, feature vector based representations have associated representational limitations, which arise from their simple structure and the fact that they have same length and structure regardless of the complexity of object to be modeled [9]. For further reading on statistical pattern recognition and classification we refer the interested reader to [10].

* Corresponding author at: Laboratoire d'Informatique, Université François Rabelais de Tours, 37200, France. Tel.: +33 247361443; fax: +33 247361422.
E-mail addresses: mluqman@cvc.uab.es, luqman@univ-tours.fr (M.M. Luqman), ramel@univ-tours.fr (J.-Y. Ramel), josep@cvc.uab.es (J. Lladós), brouard@univ-tours.fr (T. Brouard).

0031-3203/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2012.07.029

Figure 7.4 – Exemple d'un article scientifique montrant le regroupement des labels en des structures locales. Les labels encadrés par la même couleur appartiennent à une même structure locale (structure d'informations sur la référence, structure d'informations sur l'édition, structure d'informations sur les auteurs, etc.).

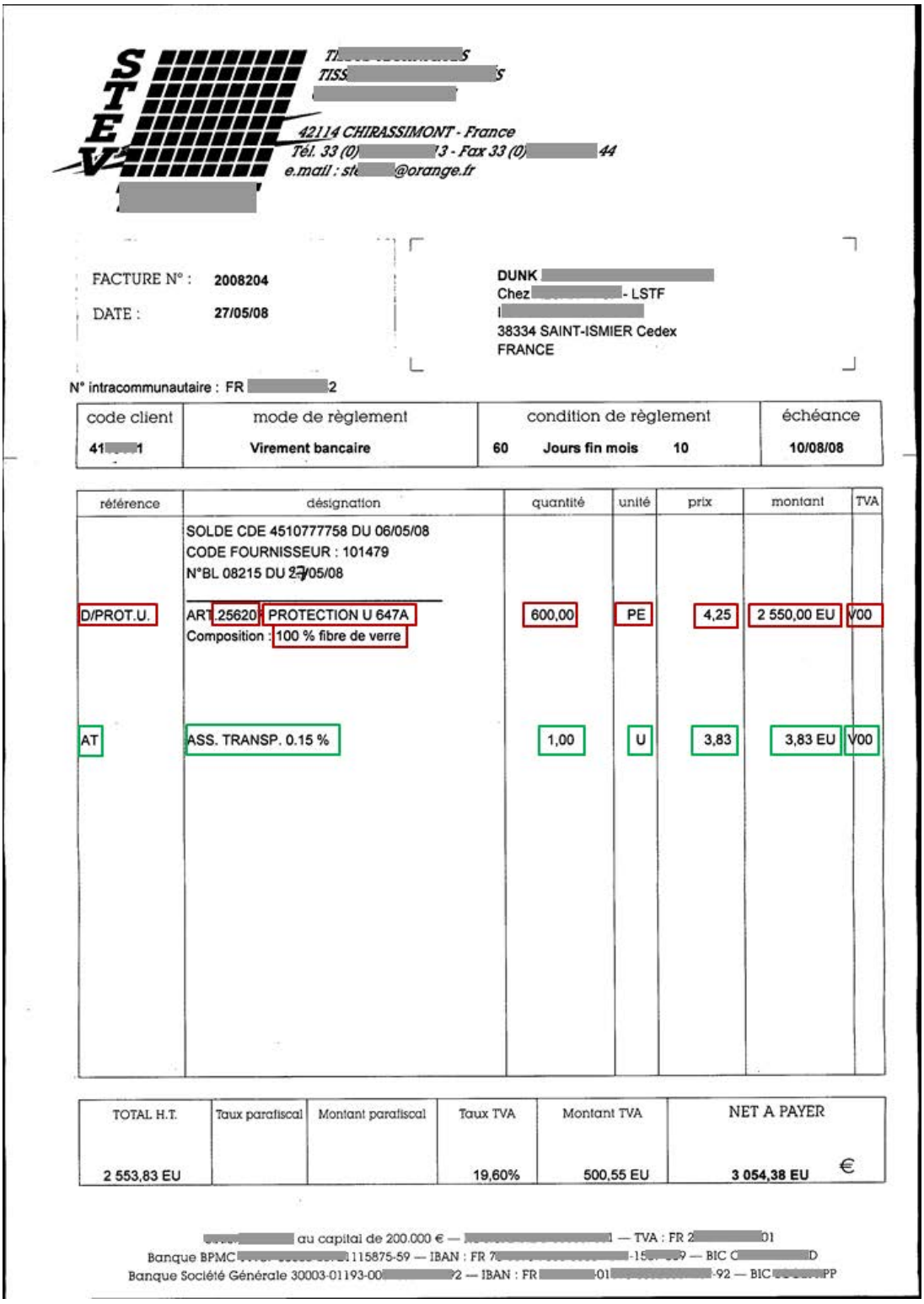


Figure 7.5 – Exemple d'une facture extraite du corpus de matériaux montrant un regroupement horizontal des labels en des structures locales. Les labels encadrés par la même couleur appartiennent à la même ligne du tableau et représentent des structures de lignes d'un tableau.

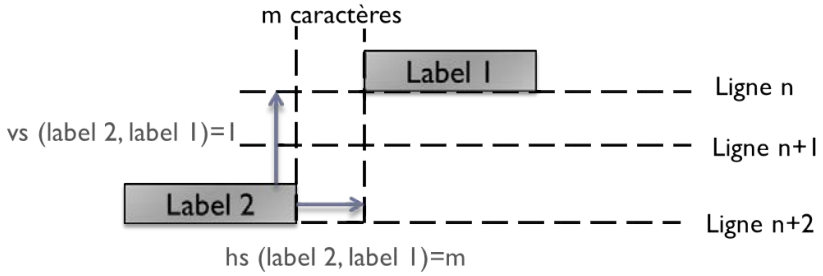
Un nœud n_i qui correspond à un label $l_i = (c_i, conf_i, v_i, sim_i)$ est défini par le 5-uplet $n_i = (c_i, conf_i, nt_i, nl_i, p_i)$, où :

— c_i est le champ de la base de données qui correspond au label l_i ;
— $conf_i$ est le score de confiance de l'étiquetage ;
— nt_i est le nombre de termes de v_i ;
— nl_i est le nombre de lignes de v_i ;
— p_i est la police normalisée par rapport à la police moyenne du document (petite : 0, moyenne : $\frac{1}{2}$, grande : 1).

Un arc a_{ij} reliant les nœuds n_i et n_j est défini par $a_{ij} = (vs_{ij}, hs_{ij}, al_{ij})$, où :

— vs_{ij} est la séparation verticale en nombre de lignes entre les labels l_i et l_j. vs_{ij} est une valeur signée pour informer sur la position verticale relative entre deux labels (au-dessus, au-dessous) ;
— hs_{ij} est la séparation horizontale en nombre de caractères entre les labels l_i et l_j. hs_{ij} est une valeur signée pour informer sur la direction relative entre deux labels (à gauche, à droite) ;
— $al_{ij} = (lJust_{ij}, cent_{ij}, rJust_{ij})$ est l'alignement (justification gauche, centrage, justification droite) entre l_i et l_j.

Une illustration des attributs des relations géométriques se trouve dans la Figure 7.6.



(a) Illustration de la séparation horizontale (hs) et la séparation verticale (vs)

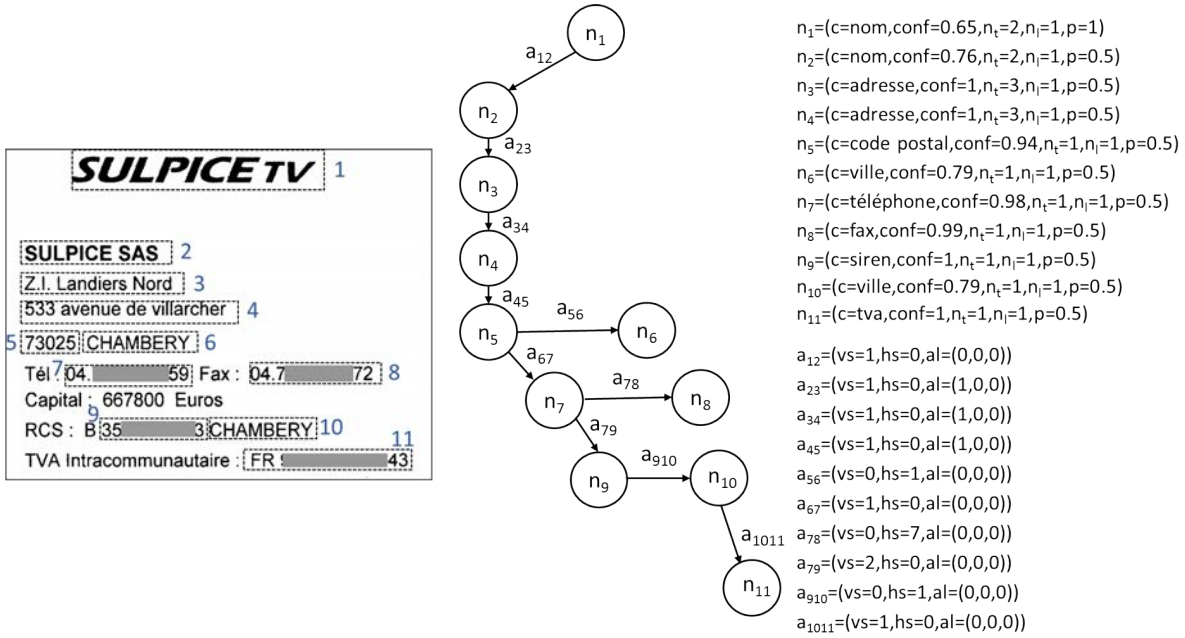
(b) Illustration des 5 configurations d'alignement (al) possibles

Figure 7.6 – Illustration des attributs des arcs

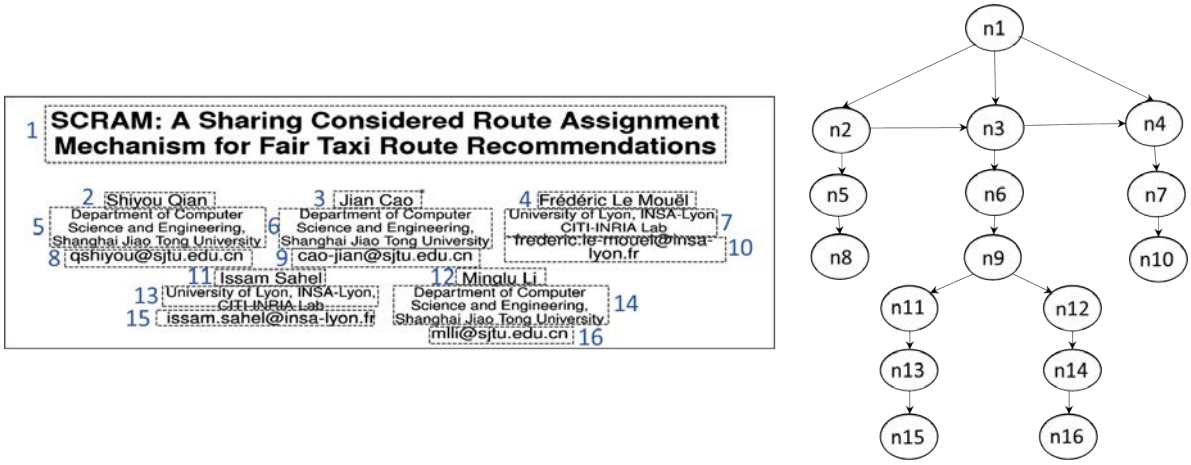
Nous représentons dans la Figure 7.7 des exemples de représentations de graphes de structures locales dans des documents réels (Pour la simplicité de la présentation, nous ne montrons pas tous les arcs dans les graphes). La Figure 7.7 (a) représente une structure d'adresse avec des informations de contact et des données juridiques d'une entreprise dans une facture. La Figure 7.7 (b) représente une structure d'entête d'un article scientifique. Les nœuds décrivent les éléments entourés par des rectangles (les labels).

7.3.3 Rapprochement de structures locales

Pour le rapprochement de structures locales, nous utilisons le rapprochement inexact de graphes qui cherche à apparier un graphe candidat, construit à partir d'un document, avec un graphe modèle, dans le modèle structurel. Nous tolérons les altérations suivantes dans le graphe candidat par rapport au graphe



(a) Structure d'une adresse postale, des informations de contact et des informations fiscales



(b) Structure d'une entête de référence bibliographique

Figure 7.7 – Exemples de représentation de structures locales d'entités

modèle :

- la modification des attributs d'un nœud ou d'un arc, comme par exemple des variations dans les distances hs des arcs ;
- le surplus d'un nœud ou d'un arc, causé par des données sur-étiquetées dans le document, comme par exemple une suite de chiffres confondue avec un code postal ;
- l'absence d'un nœud ou d'un arc, causée par des labels non trouvés dans le document, ceci peut être provoqué par des erreurs d'OCR ou par la variabilité de la représentation d'attributs d'entités.

7.3.3.1 Méthode de rapprochement

Pour retrouver la meilleure correspondance entre les nœuds d'un graphe candidat G et les nœuds d'un graphe modèle M, nous utilisons une version simplifiée de l'algorithme de séparation et évaluation (branch and bound) qui se base sur une recherche arborescente avec élagage des chemins défectueux. La stratégie de recherche suivie est la stratégie du meilleur-d'abord (best-first), proposée par l'algorithme A* [Hart68]. La méthode de rapprochement est explicitée par l'Algorithme 6, inspiré de [Riesen07, Abu-Aisheh15]. Cet algorithme consiste à représenter l'espace de recherche par un arbre ordonnée, où les nœuds représentent les chemins de correspondance partiels et les feuilles représentent les chemins de correspondance complets. L'arbre est construit dynamiquement durant l'exécution de l'algorithme. La détermination du chemin de coût minimal courant (δ_{new}) dans l'arbre de recherche courant se base sur la minimisation d'une fonction de coût (cette fonction sera explicitée dans la section suivante). La complétion de ce chemin se fait par l'ajout des opérations de substitution, de suppression et d'insertion de nœuds de G. Cet ajout est conditionné par la vérification d'un ensemble de règles de correspondance (à l'aide de la fonction $checkRules$) dans le but d'éliminer les chemins défectueux.

Nous proposons d'employer des règles syntaxiques et/ou sémantiques et des règles physiques. Certaines règles sont génériques et ne dépendent pas du corpus traité, parmi lesquelles nous trouvons :

- on ne met pas en correspondance un nœud représentant un label dont le champ est de type alphabétique (par exemple "nom", "nom de ville", "titre", "journal", etc.) avec un nœud représentant un label dont le champ est de type numérique ou alphanumérique (par exemple : "code postal", "numéro de téléphone", "année", "montant", etc.) ;
- les chemins qui mettent en correspondance plus de N labels ayant des champs différents sont éliminés ; N est fixé expérimentalement.

D'autres règles sont spécifiques aux corpus étudiés, parmi lesquelles, nous trouvons :

- on ne place pas un "auteur" au-dessus du "titre" dans une référence bibliographique ;
- les chemins qui placent à droite d'un "code postal" autre qu'un "nom de ville" sont éliminés.

Il est à noter qu'il est facile d'intégrer d'autres règles dans notre système.

7.3.3.2 Coût de rapprochement

Soient $G = (N_G, A_G, \mu_G, \xi_G)$ un graphe candidat et $M = (N_M, A_M, \mu_M, \xi_M)$ un graphe modèle. Pour le rapprochement entre G et M, nous définissons une fonction $\delta : G \rightarrow M \cup \{\epsilon\}$ de mise en correspondance un-un entre les nœuds de G et ceux de M.
L'opération de suppression d'un nœud ou d'un arc dans le graphe candidat consiste à faire correspondre le nœud ou l'arc de G à ϵ.

La fonction de coût qui correspond à un appariement δ est définie par l'équation (7.3).

$$C(G, M, \delta) = \frac{\alpha}{|N_G|}. \sum_{n \in N_G} C_N(n, \delta(n)) \; + \frac{1-\alpha}{|A_G|}. \sum_{n \in N_G} \sum_{n' \in N_G} C_A(nn', \delta(n)\delta(n')) \qquad (7.3)$$

Algorithme 6 Rapprochement de graphes attribués

Entrées: Graphe candidat : $G = (N_G, A_G, \mu_G, \xi_G)$, graphe modèle : $M = (N_M, A_M, \mu_M, \xi_M)$
Sortie: Chemin d'édition de coût minimal : $\delta_{min} = \{n_i \rightarrow n'_p, n_j \rightarrow \epsilon\ , \epsilon \rightarrow n'_q, \ldots\}$

$\Delta \leftarrow \{\emptyset\}$ // l'ensemble des chemins d'éditions
$\delta_{min} \leftarrow \emptyset$ // le chemin de coût minimal
pour chaque nœud $n' \in N_M$ **faire** // traitement du premier nœud n_1 de G
  $\delta_{new} \leftarrow \delta_{min} \cup \{n_1 \rightarrow n'\}$ // insertion des opérations de substitution
  si $checkRules(\delta_{new})$ **alors** // vérification des règles et élagage des chemins défectueux
    $\Delta \leftarrow \Delta \cup \{\delta_{new}\}$
  fin si
fin pour
$\Delta \leftarrow \Delta \cup \{n_1 \rightarrow \epsilon\}$ // ajout des opérations de suppression
tant que vrai **faire**
  $\delta_{min} \leftarrow \arg\min_{\delta \in \Delta} co\hat{u}t(\delta)$ // sélection du chemin de coût minimal
  $\Delta \leftarrow \Delta \setminus \delta_{min}$
  si δ_{min} est un chemin complet **alors**
    retourner δ_{min}
  sinon
    Soit $\delta_{min} \leftarrow \{n_1 \rightarrow n'_{i_1}, ..., n_k \rightarrow n'_{i_k}\}$
    si $k < |N_G|$ **alors**
      pour chaque $n' \in N_M \setminus \{n'_{i_1}, ..., n'_{i_k}\}$ **faire** // complétion du chemin de coût minimal par les opérations de substitution
        $\delta_{new} \leftarrow \delta_{min} \cup \{n_{k+1} \rightarrow n'\}$
        si $checkRules(\delta_{new})$ **alors**
          $\Delta \leftarrow \Delta \cup \{\delta_{new}\}$
        fin si
      fin pour
      $\delta_{new} \leftarrow \delta_{min} \cup \{n_{k+1} \rightarrow \epsilon\}$ // complétion du chemin de coût minimal par l'opération de suppression
      si $checkRules(\delta_{new})$ **alors**
        $\Delta \leftarrow \Delta \cup \{\delta_{new}\}$
      fin si
    sinon
      $\delta_{new} \leftarrow \delta_{min} \bigcup_{n' \in N_M \setminus \{n'_{i_1}, ..., n'_{i_k}\}} \{\epsilon \rightarrow n'\}$ // complétion du chemin de coût minimal par les opérations d'insertion
      $\Delta \leftarrow \Delta \cup \{\delta_{new}\}$
    fin si
  fin si
fin tant que

avec $\alpha \in [0,1]$ est un paramètre fixé expérimentalement ; $C_N : N_G \times N_M \to \mathbb{R}^+$ et $C_A : A_G \times A_M \to \mathbb{R}^+$ des fonctions de coûts respectivement pour les nœuds et les arcs. Ces fonctions de coût sont définies par la somme pondérée des mesures de dissimilarité entre les attributs. Ces mesures seront définies dans la suite de cette section.

Le coût d'appariement entre les graphes G et M est alors défini par l'équation (7.4).

$$C(G,M) = \min_{\delta \in \Delta} C(G,M,\delta) \tag{7.4}$$

avec Δ l'ensemble des chemins de mise en correspondance entre les nœuds de G et de M.

Essayons maintenant de définir les fonctions de coûts C_N et C_A. Soient $F_N = \{vs, hs, al\}$ et $F_A = \{nt, nl, p\}$ deux ensembles d'attributs respectivement pour les nœuds et les arcs.

La fonction de coût C_N entre un nœud $n_1 = (c_1, conf_1, nt_1, nl_1, p_1) \in N_G$ et un nœud $n_2 = (c_2, conf_2, nt_2, nl_2, p_2) \in N_M$ est définie par l'équation (7.5).

$$C_N(n_1,n_2) = \begin{cases} 1 - conf_1.conf_2 & \text{si } c_1 = c_2; \\ \frac{1}{F_N}.\sum_{f \in F_N} \lambda_{f_2}.d_f(f_1,f_2) & \text{sinon.} \end{cases} \tag{7.5}$$

La fonction de coût C_A entre deux arcs $a_1 = (vs_1, hs_1, al_1) \in A_G$ et $a_2 = (vs_2, hs_2, al_2) \in A_M$ est définie par l'équation (7.6).

$$C_A(a_1,a_2) = \frac{1}{F_A}.\sum_{f \in F_A} \lambda_{f_2}.d_f(f_1,f_2) \tag{7.6}$$

où f_1, f_2 sont les valeurs de l'attribut (caractéristique) f dans les nœuds n_1, n_2 ou dans les arcs a_1, a_2 ; $\lambda_{f_2} \in [0,1]$ est un facteur de pondération pour indiquer l'importance de l'attribut f dans le nœud n_2 ou dans l'arc a_2 du graphe modèle M. Le calcul de ce facteur sera expliqué dans la section 7.3.

Les mesures de dissimilarité entre les attributs sont définies par l'équation (7.7) et l'équation (7.8).

$$d_f(f_1,f_2) = |f_1^N - f_2^N| \quad \forall f \in \{vs, hs, nt, nl, p\} \tag{7.7}$$

$$d_{al}(al_1,al_2) = \begin{cases} 0 & \text{si } al_1 \times al_2 \neq 0; \\ 1 & \text{sinon.} \end{cases} \tag{7.8}$$

f_1^N (respectivement f_2^N) est la valeur normalisée de f_1 (respectivement f_2). f_1^N (de manière équivalente f_2^N) est définie dans l'équation (7.9).

$$f_1^N = \frac{f_1 - \max(f_1)}{\max(f_1) - \min(f_1)} \tag{7.9}$$

avec $\max(f)$ et $\min(f)$ sont dépendants du corpus.

Dans le cas de la suppression d'un nœud n_1, resp. d'un arc a_1 de G, le coût de la suppression $C_N(n_1,\epsilon)$, resp. $C_A(a_1,\epsilon)$ est calculé de la même façon en considérant des attributs de valeurs nulles dans le graphe modèle M. De même, dans le cas de l'insertion d'un nœud n_1, resp. d'un arc a_1 de G, le coût de l'insertion $C_N(\epsilon,n_1)$, resp. $C_A(\epsilon,a_1)$ est calculé en considérant des valeurs nulles dans les attributs de G.

Notre mesure de rapprochement inexact de graphes entre un graphe candidat et un graphe modèle est adaptée aux graphes extraits de données réelles pouvant être altérées par du bruit. Elle représente une formulation de la distance GED pour qu'elle soit adaptée aux graphes attribués en intégrant les dissimilarités entre les attributs dans la fonction de coût de rapprochement.

7.3.3.3 Filtrage du modèle structurel

Nous proposons de filtrer le modèle structurel pour limiter le nombre d'appariements entre le graphe candidat et chacun des graphes modèles dans le modèle structurel. Pour ceci, nous utilisons des critères sémantiques et structurels. Parmi ces critères, nous trouvons :

— le nombre de nœuds ayant un champ de label commun ;
— le nombre de lignes dans la structure locale ;
— l'ordre logique des champs dans une ligne ou dans une colonne dans la structure locale.

Pour rapprocher un graphe candidat G avec un graphe dans le modèle structurel $S_M = \{M_1, \ldots, M_n\}$, nous retenons le graphe modèle M_{min} qui minimise le coût de rapprochement de graphes :

$$M_{min} = \arg\min_{M_i \in S_M} C(G, M_i) \tag{7.10}$$

Nous validons le graphe modèle retenu si son coût de rapprochement est inférieur à un seuil bien déterminé. Ce seuil est fixé empiriquement pour chaque corpus à l'aide d'un corpus de validation.

7.3.4 Correction des erreurs d'étiquetage

Un graphe modèle rapproché avec un graphe candidat est utilisé pour corriger les trois types d'erreurs d'étiquetage :

— des labels non trouvés ;
— des labels dont le champ est erroné ;
— des labels supplémentaires.

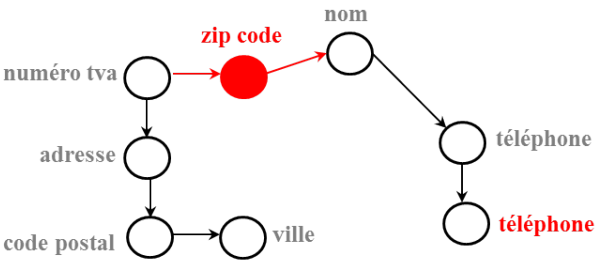
D'abord, les nœuds insérés dans le graphe candidat peuvent correspondre à des labels non trouvés dans la structure locale. Les relations géométriques fournies par les arcs reliés avec ces nœuds insérés sont utilisés pour localiser et extraire les labels non trouvés dans le document. Ensuite, les attributs de nœuds, qui correspondent au champ du label, substitués dans le graphe candidat sont utilisées pour corriger les champs des labels dans le document. Enfin, les labels qui correspondent à des nœuds supprimés dans le graphe candidat sont considérés comme des labels supplémentaires et sont donc supprimés dans le document.

La figure 7.8 montre un exemple de correction des erreurs d'étiquetage dans une structure locale en utilisant le rapprochement de graphes. Dans la Figure 7.8 (a), la valeur « 3091Z » a été étiqueté par un code postal à cause d'une confusion faite par l'OCR du caractère « Z » avec le caractère « 7 ». De plus, l'email n'a pas été trouvé par l'étiqueteur à cause des erreurs OCR et le fax a été étiqueté comme un téléphone du fait qu'ils possèdent la même syntaxe. Le graphe candidat construit pour cette structure locale est présenté dans la Figure 7.8 (b). Le graphe modèle résultant du rapprochement de graphes est montré dans la Figure 7.8 (c). Ce graphe modèle est utilisé pour extraire l'email non trouvé, corriger le label téléphone erroné par un label fax et supprimer le code postal supplémentaire dans le document, tel que montré dans la Figure 7.8 (d).

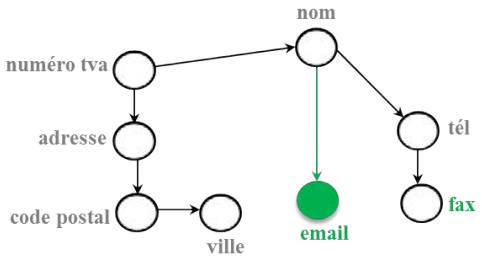
7.3.5 Validation de la reconnaissance d'entités

Après correction des erreurs d'étiquetage, nous proposons de vérifier la correspondance des valeurs des labels de la structure locale retenue avec les attributs de l'entité candidate dans la base de données. Nous nous intéressons essentiellement à ceux dont l'extraction a échoué dans la phase d'étiquetage et qui sont extraits du document en utilisant les relations géométriques du graphe dans la phase de correction. En effet, nous proposons d'utiliser une combinaison de mesures de similarité, tel que présentée la Section 6.2.4. Différents types de combinaison ont été testés, notamment le minimum, le maximum et la

(a) Structure locale contenant des erreurs d'étiquetage



(b) Le graphe candidat correspondant



(c) Le graphe modèle rapproché

(d) Structure locale corrigée

Figure 7.8 – Un exemple de correction des erreurs d'étiquetage en utilisant le rapprochement de graphes, ce qui est en rouge représente les erreurs d'étiquetage et ce qui est en vert représente les corrections.

moyenne entre plusieurs partitions de l'ensemble des mesures de similarité. Ces tests ont montré que le maximum entre les distances de Jaro-Winkler, OCR-key et Monge-Elkan donne les meilleurs résultats.

Enfin, pour valider la reconnaissance d'une entité candidate, un score de reconnaissance est calculé en prenant en compte les corrections. Ce score est une redéfinition de celui utilisé dans la phase de sélection d'entités, tel que montré dans l'équation (7.11).

$$score(e, D) = \sum_{l_i \in F(e,D)} P(l_i \in e|e).conf_i.sim_i \tag{7.11}$$

où $P(l_i \in e/e)$ est la probabilité d'appartenance du label l_i à une entité candidate connue e. Cette probabilité est dépendante de l'appartenance du label l_i à une structure locale de l'entité e, notée $S(e)$ et le coût de rapprochement entre le graphe candidat G représentant cette structure locale avec son graphe modèle correspondant M. Elle est définie par l'équation (7.12).

$$P(l_i \in e/e) = \begin{cases} 1 - C(G, M) & \text{si } l_i \in S(e); \\ 1 - P_s(c_i) & \text{sinon.} \end{cases} \tag{7.12}$$

avec : $P_s(c_i)$ est la probabilité d'appartenance d'un label, dont le champ est c_i, à une structure locale quelconque. Cette probabilité est calculée comme étant le nombre d'apparitions de ce label dans une structure locale quelconque divisée par le nombre total d'apparitions de ce label. Cette probabilité est calculée statistiquement sur le corpus. S est la structure locale modélisée par le graphe candidat G qui a été apparié avec le graphe modèle M.

7.3.6 Apprentissage du modèle structurel

Le modèle structurel est initialement appris en utilisant un échantillon représentatif sélectionné du corpus. En effet, une base de graphes, extraits de cet échantillon, est d'abord constituée et regroupée en classes (par un algorithme de classification non supervisée). Chaque classe est ensuite représentée par un graphe modèle qui correspond à une structure locale. Nous verrons plus loin comment un tel graphe est construit à partir des graphes exemples de sa classe

Le modèle structurel est mis à jour au fur et à mesure du traitement du flux continu de documents pour pouvoir :

— ajouter de nouveaux graphes modèles qui apparaissent ;
— adapter les graphes modèles existants aux nouvelles données arrivantes ;
— supprimer les graphes modèles qui deviennent, au bout d'un certain temps, obsolètes.

7.3.6.1 Initialisation

Nous effectuons une classification non supervisée de la base de graphes en utilisant l'Algorithme 7. Ce dernier est une amélioration de l'algorithme des dirigeants, proposé dans [Vijaya04], en introduisant la mise à jour continue du centroïde pour qu'il soit plus représentatif des graphes exemples d'une classe. Cet algorithme est adapté à notre cas puisque c'est un algorithme de classification non supervisée simple et rapide. De plus, des expérimentations sur nos corpus ont montré que les performances de la classification non supervisée sont indépendantes de l'ordre d'arrivée des données dans le flux, car les classes sont suffisamment éloignées.

Pour l'apprentissage d'un graphe modèle (centroïde) à partir des graphes exemples, il faut apprendre les nœuds et les arcs du graphe ainsi que leurs attributs et leurs facteurs de pondération. Nous détaillons l'apprentissage du graphe modèle dans ce qui suit.

Algorithme 7 Classification non supervisée de la base de graphes

Entrées: Base de graphes : G, seuil : T
Sortie: Ensemble de clusters : CL

1: $CL = \emptyset$ // un cluster $c \in CL$ est défini par son centroide $c.centoid$
2: sélectionner un graphe $g_1 \in G$
3: $CL = \{c_1\}$ où $c_1.centoid = g_1$
4: **pour chaque** $g_i \in G\,; i = 2 \rightarrow |G|$ **faire**
5: $c_{min} = \arg\min_{c \in CL} C(g_i, c.centroid)$ // $C(g_i, c.centroid)$ est le coût de rapprochement de graphes entre g_i et $c.centroid$
6: **si** $C(g_i, c_{min}.centroid) < T$ **alors**
7: recalculer $c_{min}.centroid$
8: **sinon**
9: $CL = CL \cup \{c\}$ où $c.centoid = g_i$
10: **fin si**
11: **fin pour**

Apprentissage des nœuds et des arcs. Nous regroupons les graphes exemples en un seul représentant du modèle à l'aide du concept de WMCS, tel que proposé dans [Bunke03]. Le calcul du WMCS se base sur la comparaison de nœuds et d'arcs des graphes exemples pour la recherche des éléments partagés entre eux. Dans cette comparaison, nous considérons seulement, comme attributs de nœuds, les champs de labels et, comme attributs d'arcs, les positions relatives entre les labels (à gauche, à droite, au-dessous, au-dessus, au-dessus et à droite, au-dessus et à gauche, au-dessous et à droite, au-dessous et à gauche).

Apprentissage des attributs et des facteurs de pondération. Une fois les nœuds et les arcs du graphe modèle construits, il ne reste plus que la détermination de leurs attributs ainsi que leurs facteurs de pondération qui vont servir dans la phase de son appariement avec un graphe candidat.

Les attributs des nœuds et d'arcs partagés entre les graphes exemples sont fusionnés pour obtenir les attributs du graphe modèle. Pour les attributs quantitatives (nombre de mots, nombre de lignes, séparation verticale et séparation horizontale), nous utilisons la moyenne. Pour l'alignement et la police, nous utilisons la valeur dominante (qui se répète le maximum de fois dans les graphes exemples).

Les facteurs de pondération des attributs de nœuds et d'arcs dans les graphes modèles sont inversement proportionnels aux écarts types des valeurs d'attributs dans les exemples. Plus les valeurs d'un attribut sont variables, moins cet attribut est pertinent et plus son facteur de pondération est faible. Soit une classe composée de q graphes exemples. Le facteur de pondération d'un attribut f dans un nœud ou dans un arc du graphe modèle, représentatif de cette classe, est défini dans l'équation (7.13).

$$\lambda_f = \frac{\sqrt{q}}{\sqrt{\sum_{i=1}^{q} f_i^2 - \sum_{i=1}^{q} f_i}} \tag{7.13}$$

7.3.6.2 Mise à jour

Au cours du traitement du flux de documents, à chaque arrivée d'un nouveau graphe candidat, deux cas peuvent se présenter : le graphe candidat correspond à un graphe modèle existant ou il ne correspond pas à un graphe modèle existant.

Cas où le graphe candidat correspond à un graphe modèle existant. Le graphe modèle est remplacé par le WMCS du graphe modèle existant et du graphe candidat traité. Les attributs des arcs et des nœuds

Tableau 7.1 – Classification de la base de graphes

Données \ Corpus	Entreprises	Articles scientifiques	Matériaux
# graphes dans la base de graphes	290	200	210
# documents dans l'échantillon utilisé	87	70	45
# graphes modèles obtenus	41	28	11

et les facteurs de pondération du graphe modèle sont recalculés en prenant en considération le graphe candidat traité. À chaque fois, seuls les nœuds et les arcs qui possèdent des fréquences importantes sont retenus dans le graphe modèle et seront utilisés dans l'appariement avec les nouveaux graphes candidats.

Pour permettre la suppression des graphes modèles qui deviennent obsolètes au cours du temps, nous utilisons une variable qui informe sur l'âge de ce graphe. En effet, à chaque fois qu'un graphe modèle est mis en correspondance avec un graphe candidat, son âge est réinitialisé à 0. Les âges des autres graphes modèles sont incrémentés. Si l'âge d'un graphe modèle dépasse un seuil (fixé statistiquement à 100) alors il est supprimé de la base de graphes.

Cas où le graphe candidat ne correspond à aucun graphe modèle existant. Nous créons un nouveau graphe modèle (graphe embryon) dans la base de graphes, initialisé par le graphe candidat. Ce graphe doit être validé (devient un graphe mature) statistiquement, i.e. lorsque le nombre d'apparitions de ce graphe dans le flux de documents dépasse un seuil (fixé manuellement à 10), avant d'être réutilisé dans la base de graphes. Ce processus est inspiré de la méthode IGNG (Incremental Growing Neural Gas) proposée par [Prudent05] dans le cadre de l'apprentissage incrémental.

7.4 Évaluations

Dans cette section, nous proposons d'évaluer la méthode G-ELSE sur les trois corpus étudiés. Pour ce faire, nous proposons plusieurs études de :

— l'apprentissage du modèle structurel ;
— le rapprochement de graphes de structures locales ;
— la correction des erreurs d'étiquetage ;
— la RE, objectif principal de notre approche.

7.4.1 Apprentissage du modèle structurel

Pour évaluer l'apprentissage du modèle structurel, nous proposons d'évaluer la classification non supervisée de la base de graphes utilisée pour l'apprentissage.

Le nombre de graphes dans la base de graphes, le nombre de documents dans l'échantillon utilisé et le nombre de graphes modèles (clusters) obtenus pour chaque corpus sont montrés dans le Tableau 7.1.

Pour l'évaluation, nous utilisons l'indice de Dunn D, généralement utilisé dans la littérature pour évaluer des algorithmes de classification non supervisée, défini par :

$$D = \frac{\min_{1 \leq i < j \leq n} d(i,j)}{\max_{1 \leq k \leq n} d'(k)}$$

avec $d(i,j)$ la distance inter-clusters entre les clusters i et j (la distance entre les centroïdes) et $d(k)$ la distance intra-cluster du cluster k (la distance maximale entre toute paire d'éléments dans le cluster k).

Tableau 7.2 – Comparaison des algorithmes de classification non supervisées

Méthode	Corpus d'entreprises			Corpus d'article scientifiques			Corpus de matériaux		
	Seuil	# Cluster	Dunn	Seuil	# Cluster	Dunn	Seuil	# Cluster	Dunn
Dirigeants [Vijaya04]	0.1	42	0.71	0.11	31	0.78	0.15	10	0.74
Notre méthode	0.1	41	**0.74**	0.12	28	**0.80**	0.15	11	**0.77**

Le Tableau 7.2 présente des comparaisons de résultats de classification non supervisée de l'algorithme proposé par rapport à celui des dirigeants. Il montre que l'algorithme proposé est plus performant que celui des dirigeants. En effet, dans le corpus d'entreprises, resp. d'articles scientifiques, resp. de matériaux, nous obtenons un indice de Dunn de 0.74, resp. 0.80, resp. 0.77 comparé à 0.71, resp. 0.78, resp. 0.74 avec l'algorithme des dirigeants [Vijaya04]. Ces améliorations dans les performances sont dues à la mise à jour progressive des centroïdes de chaque classe.

7.4.2 Rapprochement de graphes de structures locales

Pour expérimenter notre approche de rapprochement de graphes de structures locales, nous avons utilisé des jeux de données (composés de graphes candidats et de graphes modèles) extraits des trois corpus. Pour chaque corpus, nous distinguons différents types de structures locales. Ces jeux de données sont comme suit :

1. Corpus d'entreprises :
   — modèle structurel : 41 graphes modèles ;
   — structures locales : 525 graphes candidats ;
      — 135 structures d'adresses postales ;
      — 100 structures d'informations de contact ;
      — 175 structures d'informations fiscales ;
      — 115 structures mixtes.
2. Corpus d'articles scientifiques :
   — modèle structurel : 28 graphes modèles ;
   — structures locales : 200 graphes candidats ;
      — 50 structures d'informations sur les références ;
      — 40 structures d'informations d'éditions ;
      — 40 structures d'informations sur les auteurs ;
      — 70 structures mixtes.
3. Corpus de matériaux :
   — modèle structurel : 11 graphes modèles ;
   — structures locales : 320 graphes candidats ;
      — 320 structures de lignes de tableau.

Pour l'évaluation, nous avons utilisé une table de vérité qui associe chaque graphe candidat au graphe modèle correspondant. Cette table a été préparée manuellement.

Un graphe modèle mis en correspondance avec un graphe candidat donné est dit correct lorsqu'il correspond dans la vérité à ce graphe candidat. Le Rappel, la Précision et la F-mesure sont définis respectivement par les équations (7.14), (7.15) et (7.16).

$$R = \frac{\#\ graphes\ retournés\ corrects}{\#\ graphes\ corrects} \tag{7.14}$$

$$P = \frac{\#\ graphes\ retournés\ corrects}{\#\ graphes\ retournés} \tag{7.15}$$

$$F = \frac{2.R.P}{R+P} \tag{7.16}$$

Les résultats d'évaluation des trois corpus sont présentés dans le Tableau 7.3, le Tableau 7.4 et le Tableau 7.5. Ces tableaux montrent que notre approche de rapprochement de graphes fonctionne bien pour toutes les structures dans les différents corpus. En effet, le Rappel varie entre 88.50% et 90.86% et la Précision varie entre 89.53% et 95.78%.

Certaines structures sont plus faciles à rapprocher que d'autres car elles possèdent moins de variabilités. Par exemple, dans le corpus d'entreprises, les structures d'informations fiscales (Rappel = 94.86%, Précision = 98.81%) sont plus faciles à reconnaitre que les structures d'informations de contact (Rappel = 86.00%, Précision = 90.53%) car elles représentent généralement des structures linéaires contenant peu de labels. Les structures d'informations de contact, quant à elles, peuvent être structurées en formulaire, sur une même ligne, sur plusieurs colonnes, etc. Dans le corpus d'articles scientifiques, les structures d'informations d'auteurs (Rappel = 82.50%, Précision = 82.93%) sont les plus difficiles à extraire car elles présentent une grande variabilité en fonction du nombre d'auteurs dans l'article. Dans le corpus de matériaux, les problèmes de rapprochement de structures sont liés à la présence de certains attributs sur plusieurs lignes (notamment les attributs du champ "désignation") et à la présence de structures en zigzag.

Tableau 7.3 – Résultats de rapprochement de graphes de structures locales sur le corpus d'entreprises

Structure	R (%)	P (%)	F (%)
Adresses postales (135 graphes)	92.59	96.15	94.34
Informations de contact (100 graphes)	86.00	90.53	88.21
Informations fiscales (175 graphes)	94.86	98.81	96.79
Structures mixtes (115 graphes)	86.96	95.24	90.91
Total (525 graphes)	90.86	95.78	93.26

Tableau 7.4 – Résultats de rapprochement de graphes de structures locales sur le corpus d'articles scientifiques

Structure	R (%)	P (%)	F (%)
Informations références (50 graphes)	96.00	90.56	93.20
Informations éditions (40 graphes)	90.00	90.00	90.00
Informations auteurs (40 graphes)	82.50	82.93	82.71
Structures mixtes (70 graphes)	85.71	92.30	88.88
Total (200 graphes)	88.50	89.53	89.01

Tableau 7.5 – Résultats de rapprochement de graphes de structures locales sur le corpus de matériaux

Structure	R(%)	P(%)	F (%)
Lignes de tableau (320 graphes)	89.69	95.67	92.58

Nous avons comparé notre méthode de rapprochement de graphes avec des méthodes existantes. Les résultats obtenus sur le corpus d'entreprises sont montrés dans le Tableau 7.3. Ces résultats montrent que notre méthode, qui intègre la dissimilarité d'attributs dans la fonction de coût, est plus performante que

Tableau 7.6 – Comparaison des méthodes de rapprochement de graphes sur le corpus d'entreprises

	Méthode	R (%)	P (%)	F (%)
Tolérance à la substitution	Discrétisation de labels [Sanfeliu83]	86.10	94.76	90.22
	Seuil de décision [Sanfeliu83]	86.67	95.39	90.82
	Optimisation [LeBodic12]	88.00	95.45	91.58
	Notre méthode	87.81	96.04	91.74
Tolérance à l'insertion, la suppression et la substitution	Discrétisation de labels [Sanfeliu83]	87.81	94.66	91.11
	Seuil de décision [Sanfeliu83]	84.76	92.96	88.67
	Notre méthode	**90.86**	**95.78**	**93.26**

celles qui utilisent la discrétisation de labels ou un seuil de décision, que ce soit dans le cas de la tolérance à la substitution d'attributs seulement ou dans le cas de la tolérance à l'insertion et la suppression de nœuds et d'arcs et à la substitution d'attributs. De plus, dans le cas de la tolérance à la substitution, notre méthode donne des meilleures Précision et F-mesure que la méthode d'optimisation proposée dans [LeBodic12] même si le Rappel est moins important. Ceci est expliqué par l'exclusion de certaines mauvaises solutions grâce aux heuristiques employées dans l'algorithme de séparation et évaluation.

7.4.3 Correction des erreurs d'étiquetage

Dans cette section, nous proposons d'évaluer le processus de correction des erreurs d'étiquetage effectué en considérant le résultat du processus de rapprochement de structures locales. En effet, nous proposons d'évaluer individuellement chacun des trois types de correction (identification des labels non trouvés, correction des champs de labels erronés et suppression des labels supplémentaires) sur différents types de labels dans les corpus d'étude en utilisant des vérités terrains préparées manuellement.

Les taux d'identification des labels non trouvés sur les trois corpus sont montrés dans les Tableaux 7.7, 7.8 et 7.9. Ces tableaux montrent que les taux d'identification obtenus sont importants pour tous les types de labels. En effet, le Rappel varie entre 72.00% et 100.00% et la Précision varie entre 84.28% et 100.00%. Nous constatons que les taux d'identification des codes postaux, dans le corpus d'entreprises, sont très importants (Rappel = 96%, Précision = 100%) grâce à la précision de leur position (un "code postal" est toujours situé à gauche d'une "ville"). De même, les titres sont parfaitement identifiés dans le corpus d'articles scientifiques (un "titre" se trouve toujours au-dessus des "auteurs").

Tableau 7.7 – Taux d'identification de labels non trouvés sur le corpus d'entreprises

Champ	# total	# trouvés	# corrects	R(%)	P (%)	F (%)
Nom	80	70	59	73.75	84.28	78.67
Adresse	80	73	65	81.25	89.04	84.97
Code postal	100	96	96.00	96.00	100.00	97.96
Ville	100	91	89	89.00	97.80	93.19
Pays	50	45	41	82.00	91.11	86.32
Téléphone	70	62	60	85.71	96.77	90.91
Fax	70	60	60	85.71	100.00	92.31
E-mail	50	44	44	88.00	100.00	93.62
Site web	50	43	42	84.00	97.67	90.32
Numéro TVA	50	42	39	78.00	92.86	84.78
Numéro SIRET	50	41	37	74.00	90.24	81.32
Numéro SIREN	50	44	37	74.00	84.09	78.72

Tableau 7.8 – Taux d'identification de labels non trouvés sur le corpus d'articles scientifiques

Champ	# total	# trouvés	# corrects	R(%)	P (%)	F (%)
Conférence	100	77	72	72.00	93.51	81.36
Journal	100	81	77	77.00	95.06	85.08
Affiliation	80	82	73	91.25	89.02	90.12
Auteur	100	98	87	87.00	88.78	87.88
Titre	50	50	50	100.00	100.00	100.00
Volume	50	49	45	90.00	91.84	90.91
Numéro	50	49	41	82.00	83.67	82.83

Tableau 7.9 – Taux d'identification de labels non trouvés sur le corpus de matériaux

Champ	# total	# trouvés	# corrects	R(%)	P (%)	F (%)
Désignation	100	70	64	64.00	91.43	75.29
Référence	100	92	87	87.00	94.57	90.63
Num. série	80	82	73	91.25	89.02	90.12
Prix	50	45	42	84.00	93.33	88.42
Quantité	50	45	43	86.00	95.56	90.53

La correction des champs de labels erronés est évaluée de la même façon. Les taux de correction des champs de labels erronés sur les trois corpus sont montrés dans les Tableaux 7.10, 7.11 et 7.12. Ces tableaux montrent que les résultats obtenus sont importants. En effet, le Rappel varie entre 75% et 96% et la Précision varie entre $85,37\%$ et 100%. Nous remarquons, encore une fois, que les taux de correction des labels dont la position dans les structures locales est précise (par exemple, le "code postal" dans le corpus d'entreprises) sont plus importants que ceux des labels dont la position peut varier (par exemple, le "nom").

Tableau 7.10 – Taux de correction de champs de labels erronés sur le corpus d'entreprises

Champ	# total	# substitués	# corrects	R(%)	P (%)	F (%)
Code postal	100	96	96.00	96.00	100.00	97.96
Nom	100	81	75	75.00	92.59	82.87
Ville	100	91	88	88.00	96.70	92.15
Téléphone	50	44	40	80.00	90.91	85.11
Fax	50	47	45	90.00	95.74	92.78

Tableau 7.11 – Taux de correction de champs de labels erronés sur le corpus d'articles scientifiques

Champ	# total	# substitués	# corrects	R(%)	P (%)	F (%)
Date de publication	40	41	35	87.50	85.37	86.42
Date de conférence	40	42	37	92.50	88.10	90.24
Pages	40	37	35	87.50	94.59	90.91

Les taux de suppression des labels supplémentaires sur les trois corpus sont montrés dans les Tableaux 7.13, 7.14 et 7.15. Les résultats obtenus sont importants et le Rappel varie entre $77,14\%$ et $97,50\%$ et la Précision varie entre $90,20\%$ et 100%.

Les erreurs de correction sont principalement causées par des erreurs de rapprochement de structures locales.

Tableau 7.12 – Taux de correction de champs de labels erronés sur le corpus de matériaux

Champ	# total	# substitués	# corrects	R(%)	P (%)	F (%)
Prix	70	65	58	82.86	89.23	85.93
Prix unitaire	70	71	63	90.00	88.73	89.36
Quantité	70	68	62	88.57	91.18	89.86
Taxe	70	66	60	85.71	90.91	88.24

Tableau 7.13 – Taux de suppression de labels supplémentaires sur le corpus d'entreprises

Champ	# total	# supprimés	# corrects	R(%)	P (%)	F (%)
Code postal	80	78	78	97.50	100.00	98.73
Nom	100	82	78	78.00	95.12	85.71
Ville	100	87	85	85.00	97.07	90.91
Téléphone	70	59	54	77.14	91.53	83.72
Numéro SIRET	50	46	45	90.00	97.83	93.75
Numéro SIREN	50	47	44	88.00	93.62	90.72

Tableau 7.14 – Taux de suppression de labels supplémentaires sur le corpus d'articles scientifiques

Champ	# total	# supprimés	# corrects	R(%)	P (%)	F (%)
Date publication	40	42	38	95.00	90.48	92.68
Volume	90	90	87	96.67	96.67	96.67
Numéro	80	78	73	91.25	93.59	92.41
Pages	50	51	46	92.00	90.20	91.09

Tableau 7.15 – Taux de suppression de labels supplémentaires sur le corpus de matériaux

Champ	# total	# supprimés	# corrects	R(%)	P (%)	F (%)
Num. Série	40	41	37	92.50	90.24	91.36
Prix	100	93	89	89.00	95.70	92.23
Prix unitaire	100	95	90	90.00	94.74	92.31
Quantité	100	97	92	92.00	94.85	93.40
Taxe	100	100	95	95.00	95.00	95.00

7.4.4 Reconnaissance d'entités par rapprochement de structures locales

Dans cette dernière partie des expérimentations, nous évaluons l'apport du rapprochement de graphes de structures locales dans la RE qui représente notre objectif principal.

Nous avons utilisé le même corpus décrit dans la Section 6.4 du Chapitre 6.

Les résultats de RE sur le corpus d'entreprises, resp. d'articles scientifiques, resp. de matériaux sont montrés dans le Tableau 7.16, resp. Tableau 7.17, resp. Tableau 7.18. Ces tableaux montrent l'intérêt de la combinaison des mesures de OCR-key, Monge-Elkan et Jaro-Winkler en utilisant la fonction Maximum dans l'étape de validation de la correspondance de valeurs des labels trouvés avec les attributs d'entités candidates. Une légère augmentation dans le temps de réponse est constatée.

Tableau 7.16 – Résultats de RE après correction des erreurs d'étiquetage sur le corpus d'entreprises

Distance	R (%)	P (%)	F (%)	T (s/d)
OCR-key	93.05	94.06	93.55	1.2
Monge-Elkan	94.04	93.71	93.87	1.2
Levenshtein	93.89	94.32	94.10	1.1
Maximum(OCR-key,Monge-Elkan,Jaro-winkler)	**95.06**	**94.70**	**94.88**	1.4
Maximum(OCR-key,Monge-Elkan)	94.68	94.68	94.68	1.3

Tableau 7.17 – Résultats de RE après correction des erreurs d'étiquetage sur le corpus d'articles scientifiques

Distance	R(%)	P (%)	F (%)	T (s/d)
OCR-key	83.33	95.45	88.98	1.1
Monge-Elkan	86.11	93.93	89.85	1.1
Levenshtein	85.71	95.57	90.37	1
Maximum(OCR-key,Monge-Elkan,Jaro-winkler)	**88.31**	**95.74**	**91.87**	1.2
Maximum(OCR-key,Monge-Elkan)	87.30	95.25	91.10	1.2

Tableau 7.18 – Résultats de RE après correction des erreurs d'étiquetage sur le corpus de matériaux

Distance	R(%)	P (%)	F (%)	T (s/d)
OCR-key	89.37	91.54	90.44	1.1
Monge-Elkan	90.32	92.37	91.33	1
Levenshtein	90.63	93.45	92.02	1
Maximum(OCR-key,Monge-Elkan,Jaro-winkler)	**92.06**	**94.31**	**93.17**	1.2
Maximum(OCR-key,Monge-Elkan)	91.27	92.89	92.07	1.1

Les résultats de comparaison des taux de RE avant et après l'emploi des structures sont montrés dans le Tableau 7.19. Ces résultats montrent l'apport considérable de l'emploi des structures pour la RE. En revanche, une légère augmentation du temps de réponse est engendrée par l'étape de rapprochement de structures.

Pour montrer l'intérêt de la mise à jour du modèle structurel dans la RE, le taux de reconnaissance est évalué en utilisant un modèle structurel appris incrémentalement sans être initialisé (pas de correction d'erreurs d'étiquetage au début). La Figure 7.9 représente les résultats obtenus sur 3 flux aléatoires de documents extraits du corpus d'entreprises. Il est à noter que les courbes de la figure 7.9 sont sensiblement les mêmes dans les 3 flux. Ceci prouve que les performances de la classification de la base des

Tableau 7.19 – Résultats de RE

Corpus	Sélection d'entités				G-ELSE			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
Entreprises	86.69	56.86	68.67	0.84	95.06	94.70	94.88	1.4
Articles scientifiques	80.51	70.41	75.13	0.90	88.49	95.30	91.77	1.2
Matériaux	77.62	74.31	75.93	0.85	92.06	94.30	93.17	1.2

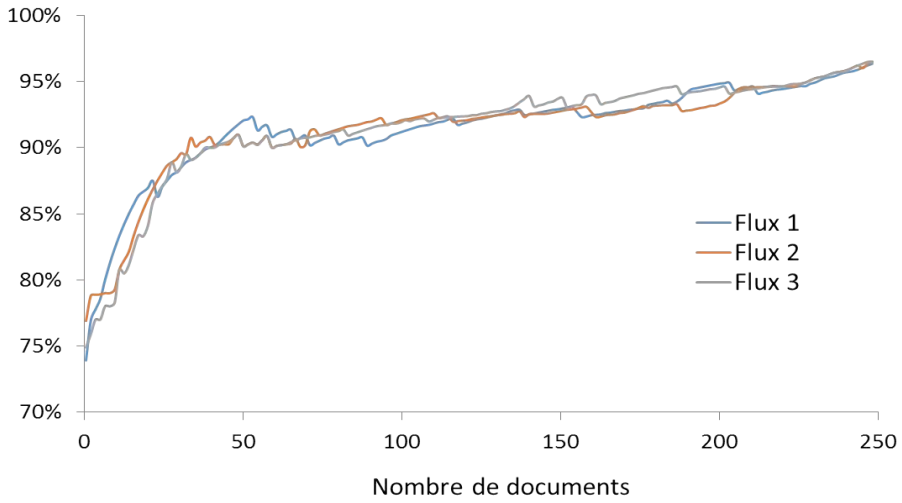


Figure 7.9 – Évolution des taux de RE pour trois flux aléatoires de documents sur la base d'entreprises.

graphes et par conséquent le taux de RE sont indépendants de l'ordre d'arrivée des documents. De plus, le taux de reconnaissance tend rapidement vers les valeurs obtenues en utilisant un modèle initialement appris.

Seuil d'acceptation. Les seuils utilisés dans les phases de sélection d'entités et de validation de la RE sont déterminés à l'aide de corpus de validation étiquetés manuellement. A titre d'exemple, nous montrons dans la Figure 7.10 la courbe de variation du Rappel, Précision et F-mesure en fonction du seuil sur la base d'articles scientifiques dans la phase de validation de la RE. La valeur retenue du seuil d'acceptation dans ce cas est 20.

Cas d'erreurs. Nous avons analysé les cas d'erreurs sur les trois corpus. Les causes de non détection d'entités sont montrées dans le Tableau 7.20. Certaines de ces causes sont dues à des erreurs de rap-

Tableau 7.20 – Les causes de non détection d'entités

Corpus / Causes	Entreprises	Articles scientifiques	Matériaux
Erreurs de segmentation en structures locales (%)	1.52	2.78	0.95
Erreurs de rapprochement de structures locales (%)	0.38	0.79	0.48
Erreurs d'OCR irrécupérables (%)	1.90	2.38	1.90
Non standardisation (%)	0.19	3.57	4.29
Entités incomplètes (%)	0.95	1.98	0.32

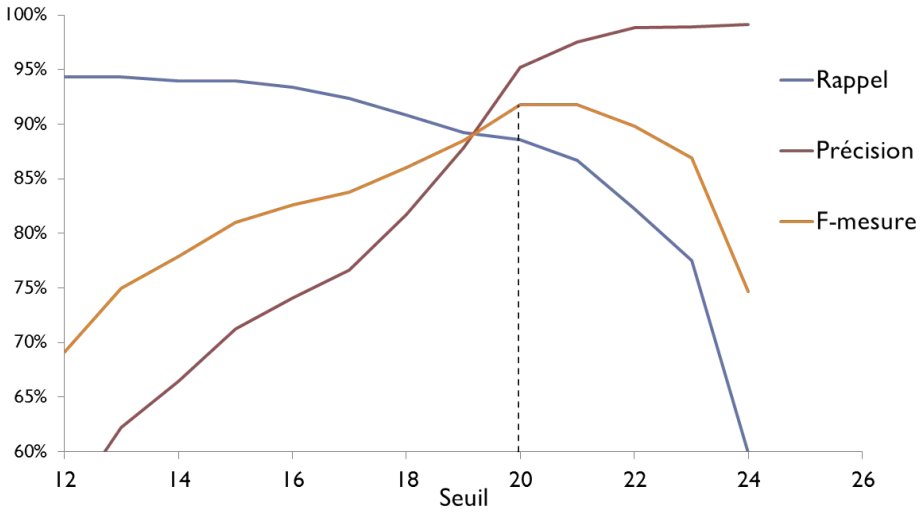


Figure 7.10 – Variation du Rappel, de la Précision et de la F-mesure en fonction du seuil de score de RE sur le corpus d'articles scientifiques (valeur retenue est 20).

prochement de structures locales ou de segmentation du document en structures locales. Ces dernières incluent les erreurs de sur-segmentation ou des erreurs de sous-segmentation. La sous-segmentation consiste à rassembler deux structures ou plus en une seule. La sur-segmentation consiste à éclater les labels sur plusieurs structures locales. Un exemple extrait du corpus d'entreprises est montré dans la Figure 7.11 où la structure encadrée en rouge est éclatée en 6 structures à cause de l'éloignement physique des labels dans la page. D'autres erreurs de non détection sont dues au problème de la non-standardisation

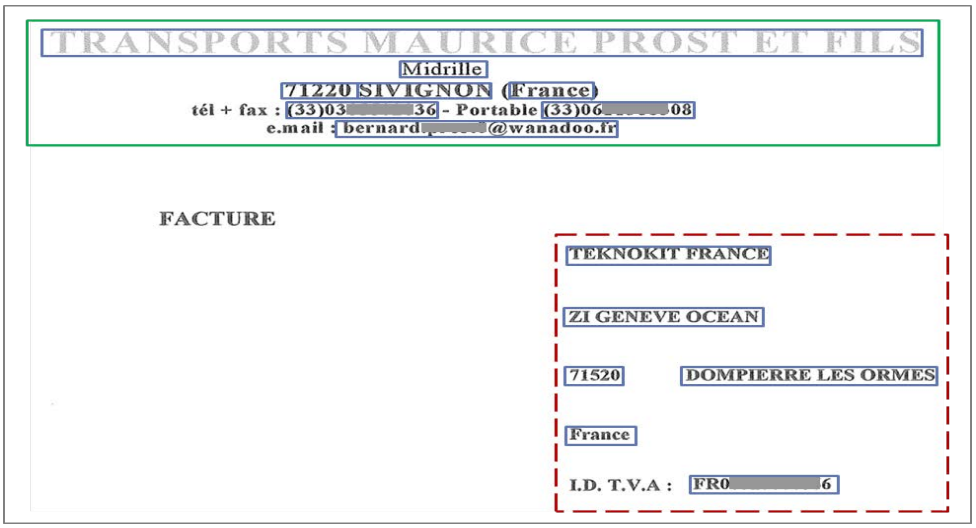


Figure 7.11 – Exemple extrait du corpus d'entreprises illustrant le problème de sur-segmentation de structure locale. La structure encadrée en rouge est éclatée en 6 structures à cause de l'éloignement physique des labels dans la page.

malgré le traitement de résolution d'entités dans la base de données et la combinaison de mesures de similarité pour la comparaison d'attributs. Ces erreurs sont plus importantes dans le corpus d'articles scientifiques à cause de la grande variabilité de certains attributs, notamment les noms de journaux et

Tableau 7.21 – Comparaison des méthodes de reconnaissance d'entités

Method	Corpus d'entreprises				Corpus d'articles scientifiques				Corpus de matériaux			
	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)	R (%)	P (%)	F (%)	T (s/d)
EROCS	67.68	54.10	60.14	69.00	67.46	77.27	72.03	42	76.19	77.42	76.80	37
M-EROCS	81.37	77.12	79.19	4.70	85.32	91.49	88.30	3.30	80.79	80.92	80.86	3.8
G-ELSE	95.06	94.70	94.88	1.40	88.49	95.30	91.77	1.20	92.06	94.30	93.17	1.2

des conférences, et le corpus de matériaux à cause de l'absence du traitement de résolution d'entités. Enfin, nous retrouvons des erreurs de RE dues à des erreurs d'OCR irrécupérables (mauvaise qualité du document numérisé) ou des entités incomplètes dans la base de données.

Les erreurs de fausses détections sont généralement causées par des erreurs de rapprochement de structures locales engendrant la non correction des erreurs de sur-étiquetage.

Comparaison. Nous avons comparé la méthode G-ELSE avec la méthode M_EROCS (proposée dans le chapitre précédent) et la méthode de l'état de l'art EROCS [Chakaravarthy06] pour les trois corpus. Les résultats obtenus sont présentés dans le Tableau 7.21. Ces comparaisons montrent que la Précision et le Rappel obtenus par G-ELSE sont beaucoup plus importants que ceux obtenus par M-EROCS et EROCS grâce à l'emploi des structures. De plus, le temps de réponse est plus rapide grâce au processus d'étiquetage qui réduit considérablement la recherche dans le document.

7.5 Conclusion

Dans ce chapitre, nous avons proposé une méthode de RE qui représente les labels d'entités, proches physiquement dans la page, par des graphes attribués et qui utilise le rapprochement inexact de graphes pour corriger les erreurs d'étiquetage. Les expérimentations de cette méthode sur les corpus d'entreprises, d'articles scientifiques et de matériaux ont montré une amélioration des taux de RE respectivement de 15.97%, 9.12% et 14.92% en Rappel et de 2.66%, 3.56% et 1.37% en Précision par rapport à la méthode ERBL. Ces expérimentations ont confirmé l'intérêt de l'emploi des structures et ont prouvé l'efficacité et la généricité de notre approche.

Chapitre 8

Conclusion et travaux futurs

Sommaire

8.1 Conclusion

Nous avons abordé dans cette thèse le problème de la reconnaissance d'entités dans des documents océrisés, guidée par une base de données prédéfinie. Le travail réalisé concerne le rapprochement entre le contenu de la base de données et celui des documents. Une première difficulté de ce travail réside dans la mauvaise qualité de la base de données. Cette dernière souffre de redondances d'enregistrements se référant à une même entité, d'incomplétude de données et d'erreurs de saisie. Une deuxième difficulté vient du fait que l'OCR peut altérer le contenu et la structure des documents images. Enfin, une autre difficulté se trouve dans la présence de variations de représentation d'attributs d'entités dans la base de données et le document.

Une étude des méthodes de l'état de l'art nous a permis de constater l'inexistence d'approches de rapprochement d'entités dans des documents océrisés. Les quelques méthodes existantes traitent des documents du Web et ne proposent pas de solutions aux difficultés mentionnées ci-dessus.

Pour résoudre ces difficultés, nous avons proposé une démarche en deux modules : la résolution d'entités et la reconnaissance d'entités. Le premier a pour but d'épurer la base de données en synthétisant, dans un modèle entité, les enregistrements se référant à une même entité. Pour ce faire, nous avons utilisé une approche de couplage d'enregistrements, basée sur la comparaison d'attributs à l'aide d'une combinaison de plusieurs mesures de similarité.

Nous avons expérimenté notre approche sur deux bases de données, une décrivant des entreprises et l'autre, des méta-données d'articles scientifiques. Les résultats obtenus sont respectivement 93.50% et 94.56% en Rappel et 91.20% et 94.90% en Précision.

Le deuxième module vise à rapprocher des entités mentionnées dans un document avec le modèle entité issu du premier module. Nous avons procédé par deux manières différentes, l'une utilise le rapprochement par le contenu et l'autre intègre le rapprochement par la structure. Pour le rapprochement par le contenu, nous avons proposé deux méthodes : M-EROCS et ERBL. M-EROCS consiste à faire correspondre les blocs de l'OCR avec les entités dans le modèle entité en se basant sur un score qui

tient compte des erreurs d'OCR et des variations d'attributs. ERBL, quant à elle, consiste à étiqueter le document par les attributs d'entités, puis à regrouper ces labels en entités.

Les expérimentations de ces deux approches sur les corpus d'entreprises, d'articles scientifiques et de matériaux ont donné des résultats encourageants qui dépassent l'état de l'art. Cependant, elles ont révélé un temps d'exécution élevé pour M-EROCS et une sensibilité aux erreurs d'étiquetage pour ERBL.

Le rapprochement par les structures permet d'exploiter les relations structurelles entre les labels d'une entité pour corriger les erreurs d'étiquetage. La méthode proposée, nommée G-ELSE, emploie le rapprochement inexact de graphes attribués modélisant des structures locales, avec un modèle structurel appris pour cet objectif.

L'expérimentation de cette méthode sur les mêmes corpus a permis d'améliorer les taux de reconnaissance respectivement de 15.97%, 9.12% et 14.92% en Rappel et de 2.66%, 3.56% et 1.37% en Précision. Ces résultats prouvent l'efficacité et la généricité de notre approche.

8.2 Travaux futurs

Dans le futur, nous envisageons d'apporter les améliorations suivantes.

8.2.1 A court terme

A court terme, nous prévoyons d'élargir les corpus que nous avons utilisés et d'expérimenter notre système sur de nouveaux corpus traitant d'autres natures d'entités et d'autres types de structures locales. Comme par exemple, des articles dans des magazines ou des quotidiens, des courriers électroniques, etc. De plus, nous planifions de traiter des bases de données relationnelles, où les entités sont décrites par plusieurs tables reliées par des clés étrangères.

Une deuxième perspective consiste à étudier les positions absolues et relatives des structures locales dans les documents, permettant de guider et d'accélérer la recherche d'entités.

Une limitation de la méthode proposée est que les seuils utilisés sont fixés au début, de manière définitive, ce qui ne garantit pas la stabilité des performances du système dans le cas d'un flux continu de documents. Une solution envisageable consiste à proposer des seuils adaptatifs qui s'ajustent, au cours du temps, en fonction des nouvelles données arrivantes.

Un autre aspect à approfondir concerne la mise à jour dynamique du modèle entité (représentant la base de données) durant le processus de reconnaissance d'entités. Ceci consiste à exploiter les résultats de rapprochement d'entités pour compléter les attributs manquants, rajouter des nouvelles variations et corriger les erreurs de saisie.

8.2.2 A long terme

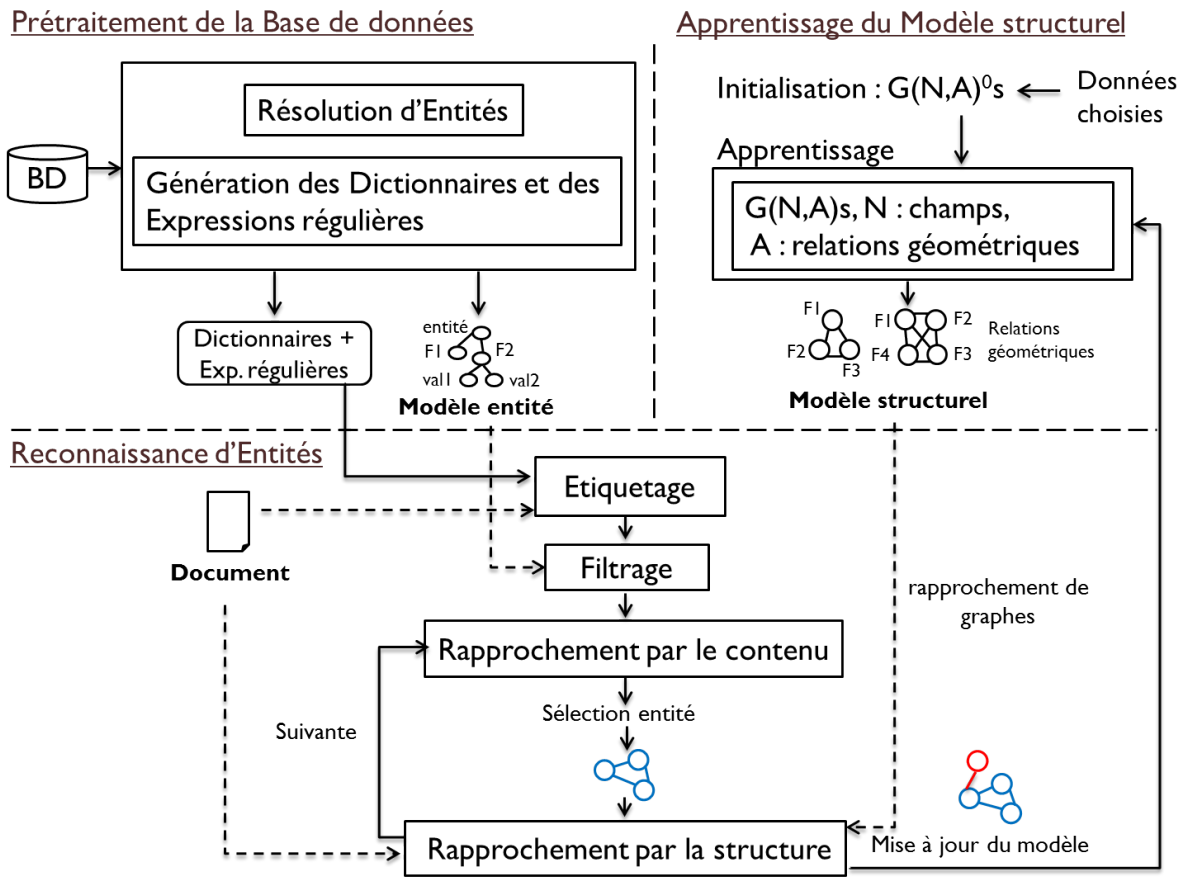
A long terme, nous envisageons d'intégrer le système dans des applications réelles permettant de profiter du résultat de rapprochement d'entités pour annoter les documents par des méta-données à partir de la base de données et pour compléter et/ou corriger les attributs de la base de données en utilisant le contenu de documents.

Une autre perspective est de profiter de l'aspect de couplage d'informations de contenu et de structure dans d'autres applications dans le domaine d'analyse de documents, tels que la classification de documents et la détection de régions d'intérêt.

Enfin, nous allons investir l'intégration d'autres sources de données telles que des ontologies de domaine contenant des données sémantiques ou des moteurs de recherche contenant des données en ligne.

Annexe A

Schéma global du système


Prétraitement de la Base de données
Apprentissage du Modèle structurel
Résolution d'Entités
BD
Génération des Dictionnaires et des Expressions régulières
Dictionnaires + Exp. régulières
entité
F1
F2
val1
val2
Modèle entité
Initialisation : G(N,A)0s
Données choisies
Apprentissage
G(N,A)s, N : champs,
A : relations géométriques
F1
F2
F3
F4
Relations géométriques
Modèle structurel
Reconnaissance d'Entités
Etiquetage
Document
Filtrage
rapprochement de graphes
Rapprochement par le contenu
Sélection entité
Suivante
Rapprochement par la structure
Mise à jour du modèle

Annexe B

Publications de l'auteur

— N. Kooli and A. Belaïd. Entity Matching in OCRed Documents with Structured Databases. *In Proceedings of the International Conference on Pattern Recognition Applications and Methods - ICPRAM*, pages 165–172, 2015.

— N. Kooli and A. Belaïd. Semantic Label and Structure Model based Approach for Entity Recognition in Database Context. *In Proceedings of the International Conference on Document Analysis and Recognition - ICDAR*, pages 301–305, 2015.

— N. Kooli, A. Belaïd, A. Joseph and V. Poulain D'Andecy. Entity Local Structure Graph Matching for Mislabeling Correction. *In Proceedings of the International Workshop on Document Analysis Systems - DAS*, 2016, pages 257–262, 2016.

— N. Kooli and A. Belaïd. Inexact Graph Matching for Entity Recognition in OCRed Documents. *To appear in the International Conference on Pattern Recognition - ICPR*, 2016.

Bibliographie

[Aaron14] B. Aaron, D. E. Tamir, N. D. Rishe, and A. Kandel. Dynamic incremental k-means clustering. In *Proceedings of the International Conference on Computational Science and Computational Intelligence*, pages 308–313, 2014.

[Abu-Aisheh15] Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. An exact graph edit distance algorithm for solving pattern recognition problems. In *International Conference on Pattern Recognition Applications and Methods, Volume 1*, pages 271–278, 2015.

[Aiello02] M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, 5(1) :1–16, 2002.

[Appelt93] D. E. Appelt, J. R. Hobbs, J. Bear, D. Israel, and M. Tyson. Fastus : A finite state processor for information extraction from real-world text. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1172–1178, 1993.

[Auwatanamongkol07] S. Auwatanamongkol. Inexact graph matching using a genetic algorithm for image recognition. *Pattern Recognition Letters*, 28(12) :1428–1437, 2007.

[Bassil12] Y. Bassil and M. Alwani. Context-sensitive spelling correction using google web 1t 5-gram information. *Computer and Information Science*, 5(3) :37–48, 2012.

[Belaid01] A. Belaïd. Recognition of table of contents for electronic library consulting. *International Journal on Document Analysis and Recognition*, 4(1) :35–45, 2001.

[Berkhin06] P. Berkhin. *Grouping Multidimensional Data : Recent Advances in Clustering*, chapter A Survey of Clustering Data Mining Techniques, pages 25–71. Springer Berlin Heidelberg, 2006.

[Berretti01] S. Berretti, A. Del Bimbo, and E. Vicario. Efficient matching and indexing of graph models in content-based retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(10) :1089–1105, 2001.

[Beusekom07] J. Beusekom, D. Keysers, F. Shafait, and T. M. Breuel. Example-based logical labeling of document title page images. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 919–923, 2007.

[Bhattacharya08] I. Bhattacharya, S. Godbole, and S. Joshi. Structured entity identification and document categorization : two tasks with one joint model. In *Proceeding of the 14th International Conference on Knowledge Discovery and Data Mining*, pages 25–33. ACM, 2008.

[Bilenko03a] M. Bilenko, R. J. Mooney, W. W. Cohen, P. Ravikumar, and S. E. Fienberg. Adaptive name matching in information integration. *IEEE Intelligent Systems*, 18(5) :16–23, 2003.

[Bilenko03b] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39–48, 2003.

[Bitton83] D. Bitton and D. J. DeWitt. Duplicate record elimination in large data files. *ACM Transactions on Database Systems*, 8(2) :255–265, 1983.

[Bres02] Stéphane Bres, Véronique Eglin, and Antoine Gagneux. Unsupervised clustering of text entities in heterogeneous grey level documents. In *16th International Conference on Pattern Recognition, ICPR 2002, Quebec, Canada, August 11-15, 2002.*, pages 224–227, 2002.

[Budi03] I. Budi and S. Bressan. Association rules mining for name entity recognition. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, pages 325–328, 2003.

[Bunke03] H. Bunke, P. Foggia, C. Guidobaldi, and M. Vento. Graph clustering using the weighted minimum common supergraph. In *Proceedings of the 4th IAPR international conference on Graph based representations in pattern recognition*, pages 235–246, 2003.

[Bunke97] H. Bunke. On a relation between graph edit distance and maximum common subgraph. *Pattern Recognition Letters*, 18(8) :689–694, 1997.

[Bunke98] H. Bunke and K. Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4) :255–259, 1998.

[Caelli05] T. Caelli and T. Caetano. Graphical models for graph matching : Approximate models and optimal algorithms. *Pattern Recognition Letters*, 26(3) :339–346, 2005.

[Carcassoni01] M. Carcassoni and E. R. Hancock. Weighted graph-matching using modal clusters. In *Proceedings of the 3rd IAPR-TC15 Workshop Graph-Based Representations in Pattern Recognition*, pages 260–269, 2001.

[Cesarini03] F. Cesarini, E. Francesconi, M. Gori, and G. Soda. Analysis and understanding of multiclass invoices. *International Journal on Document Analysis and Recognition*, 6(2) :102–114, 2003.

[Chakaravarthy06] V. T. Chakaravarthy, H. Gupta, P. Roy, and M. Mohania. Efficiently linking text documents with relevant structured information. In *Proceedings of the 32nd international conference on Very large data bases*, pages 667–678, 2006.

[Chevalier07] F. Chevalier, J. p. Domenger, J. Benois-Pineau, and M. Delest. Retrieval of objects in video by similarity based on graph matching. *Pattern Recognition Letters*, 28(8) :939–949, 2007.

[Chieu02] H. L. Chieu and T. N. Hwee. Named entity recognition : A maximum entropy approach using global information. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 190–196, 2002.

[Chinchor99] N. Chinchor and R. Patty B. Erica, F. Lisa. Named entity recognition task definition version 1.4. Technical report, MITRE Corporation and SAIC, 1999.

[Christen12] P. Christen. Data matching - concepts and techniques for record linkage, entity resolution, and duplicate detection. *Data-Centric Systems and Applications Description*, pages 1–270, 2012.

[Church91] K. W. Church and W. A. Gale. Probability scoring for spelling correction. *Statistics and Computing*, 1(2) :93–103, 1991.

[Churches02] T. Churches, P. Christen, K. Lim, and J. X. Zhu. Preparation of name and address data for record linkage using hidden Markov models. *Medical Informatics and Decision Making*, 2(9) :1–16, 2002.

[Cohen03a] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proceedings of the IJCAI Workshop on Information Integration on the Web*, pages 73–78, 2003.

[Cohen03b] W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string metrics for matching names and records. In *Proceedings of the KDD-2003 Workshop on Data Cleaning, Record Linkage and Object Consolidation*, pages 73–78, 2003.

[Collins-Thompson01] K. Collins-Thompson, C. Schweizer, and S. Dumais. Improved string matching under noisy channel conditions. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 357–364, 2001.

[Conte04] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(3) :265–298, 2004.

[Cordella98] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. *Graph Based Representations in Pattern Recognition*, chapter Subgraph Transformations for the Inexact Matching of Attributed Relational Graphs, pages 43–52. Springer Vienna, 1998.

[Damerau64] F. J. Damerau. A technique for computer detection and correction. *Communications of the ACM*, 7(3) :171–176, 1964.

[Dengel14] Dengel Andreas and Shafait Faisal. *Handbook of Document Image Processing and Recognition*, chapter Analysis of the Logical Layout of Documents, pages 177–222. Springer London, 2014.

[Dumay92] A. C. M. Dumay, R. J. van der Geest, J. J. Gerbrands, E. Jansen, and J. H. C. Reiber. Consistent inexact graph matching applied to labelling coronary segments in arteriograms. In *Proceedings of the International Conference on Pattern Recognition*, pages 439–442, 1992.

[Farag09] A. Farag, E. W. De Luca, and A. Nurnberger. Revised n-gram based automatic spelling correction tool to improve retrieval effectiveness. *Polibits*, 40 :39–48, 2009.

[Fellegi69] I. Fellegi and A. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64 :1183–1210, 1969.

[Fernandez01] M. L. Fernández and G. Valiente. A graph distance metric combining maximum common subgraph and minimum common supergraph. *Pattern Recognition Letters*, 22(6-7) :753–758, 2001.

[Foggia14] P. Foggia, G. Percannella, and M. Vento. Graph matching and learning in pattern recognition in the last 10 years. *International Journal of Pattern Recognition and Artificial Intelligence*, 28(1) :1450001–1450040, 2014.

[Ghahraman80] D. E. Ghahraman, A. K. C. Wong, and T. Au. Graph optimal monomorphism algorithms. *IEEE Transactions on System, Man and Cybernetics*, 10(4) :181–188, 1980.

[Ginter04] F. Ginter, J. Boberg, J. Jarvinen, T. Salakoski, and W. Cohen. New techniques for disambiguation in natural language and their application to biological text. *Journal of Machine Learning Research*, 5 :605–621, 2004.

[Gregory02] Lee Gregory and Josef Kittler. *Structural, Syntactic, and Statistical Pattern Recognition*, chapter Using Graph Search Techniques for Contextual Colour Retrieval, pages 186–194. Springer Berlin Heidelberg, 2002.

[Grover08] C. Grover, S. Givon, R. Tobin, and J. Ball. Named entity recognition for digitised historical texts. In *2008*. Proceedings of the Sixth International Conference on Language Resources and Evaluation, 2008.

[Gunter02] S. Gunter and H. Bunke. Self-organizing map for clustering in the graph domain. *Pattern Recognition Letters*, 23(4) :405–417, 2002.

[Hamza08] H. Hamza, Y. Belaïd, A. Belaïd, and B. B. Chaudhuri. Incremental classification of invoice documents. In *Proceedings of 19th International Conference on Recognition*, pages 1–4, 2008.

[Hart68] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Systems Science and Cybernetics*, 4(2) :100–107, 1968.

[Hashemi03] R. R. Hashemi, C. Ford, A. Bansal, S. D. Sieloff, and J. R. Talburt. Building semantic-rich patterns for extracting features from events of an on-line newspaper. In *Proceedings of the International Conference WWW/Internet*, pages 627–634, 2003.

[Hashemi04] R. R. Hashemi and J. R. Talburt. A scheme for approximate matching event announcements to a customer database. In *Proceedings of the IADIS International Conference on WWW/Internet*, pages 123–130, 2004.

[Hernandez95] M. Hernandez and S. Stolfo. The merge/purge problem for large databases. In *Proceedings of the 1995 ACM SIGMOD*, pages 127–138. ACM, 1995.

[Hobbs95] J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, A. Kehler, D. Martin, K. Myers, and M. Tyson. Sri international fastus system muc-6 test results and analysis. In *Proceedings of the Sixth Message Understanding Conference*, pages 237–248, 1995.

[Irniger03] C. Irniger and H. Bunke. Theoretical analysis and experimental comparison of graph matching algorithms for database filtering. In *Proceedings of the 4th IAPR international conference on Graph based representations in pattern recognition*, pages 118–129, 2003.

[Jaccard66] P. Jaccard. Bulletin de la société vaudoise des sciences naturelles. *37*, page 241–272, 1966.

[Jaro95] M. A. Jaro. Probabilistic linkage of large public health data files. *Statistics in Medicine*, 14 :491–498, 1995.

[Jones72] K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28 :11–21, 1972.

[Jrt94] J. Rocha and T. Pavlidis. A shape analysis model with applications to a character recognition system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(4) :393–404, 1994.

[Khy06] S. Khy, Y. Ishikawa, and H. Kitagawa. Novelty-based incremental document clustering for on-line documents. In *Proceedings of 22nd International Conference on Data Engineering Workshops*, page 40, 2006.

[Kim01] J. Kim, D. X. Le, and G. R. Thoma. Automated labeling in document images. In *Proceedings of SPIE Document Recognition and Retrieval VIII, vol. 4307*, pages 111–122, 2001.

[Klink01] S. Klink and T. Kieninger. Rule-based document structure understanding with a fuzzy combination of layout and textual features. *International Journal of Document Analysais and Recognition*, 4(1) :18–26, 2001.

[Kolak03] O. Kolak, W. Byrne, and P. Resnik. A generative probabilistic ocr model for nlp applications. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 55–62, 2003.

[Krishnamoorthy93] M. Krishnamoorthy, G. Nagy, S. Seth, and M. Viswanathan. Syntactic segmentation and labeling of digitized pages from technical journals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(7) :737–747, 1993.

[Kukich92] K. Kukich. Techniques for automatically correcting words in text. *ACM Computing Surveys*, 24(4) :377–439, 1992.

[LeBodic12] P. Le Bodic, P. Héroux, S. Adam, and Y. Lecourtier. An integer linear program for substitution tolerant subgraph isomorphism and its use for symbol spotting in technical drawings. *Pattern Recognition*, 45(12) :4214–4224, 2012.

[Lee00] M. L. Lee, T. W. Ling, and W. L. Low. Intelliclean : A knowledge-based intelligent data cleaner. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining*, pages 290–294, 2000.

[Levenshtein66] V. I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *English translation in Soviet Physics Doklady*, 10(8) :707–710, 1966.

[Liang02a] J. Liang and D. S. Doermann. Logical labeling of document images using layout graph matching with adaptive learning. In *Proceedings of the 5th International Workshop on Document Analysis Systems*, pages 224–235, 2002.

[Liang02b] J. Liang, D. S. Doermann, M. Y. Ma, and J. K. Guo. Page classification through logical labelling. In *Proceedings of the 16th International Conference on Pattern Recognition*, pages 477–480, 2002.

[Liang03] J. Liang and D. S. Doermann. Content features for logical document labeling. In *Proceedings of the International Conference on Document Recognition and Retrieval*, pages 189–196, 2003.

[Lin06] X. Lin and Y. Xiong. Detection and analysis of table of contents based on content association. *International Journal on Document Analysis and Recognition*, 8(2) :132–143, 2006.

[Lopresti03] D. P. Lopresti and G. T. Wilfong. A fast technique for comparing graph representations with applications to performance evaluation. *International Journal on Document Analysis and Recognition*, 6(4) :219–229, 2003.

[Luo02] B. Luo, R. C. Wilson, and E. R. Hancock. Spectral feature vectors for graph clustering. *Proceeding of the Joint IAPR International Workshops SSPR and SPR*, pages 83–93, 2002.

[Malouf03] R. Malouf. Markov models for language-independent named entity recognition. In *proceedings of the 6th conference on Natural language learning, vol. 20*, pages 1–4, 2003.

[Mao03] S. Mao, A. Rosenfeld, and T. Kanungo. Document structure analysis algorithms : a literature survey. In *Proceedings of SPIE Electronic Imaging*, pages 197–207, 2003.

[Mao04] S. Mao and G. R. Thoma. Bayesian learning of 2d document layout models for automated preservation metadata extraction. In *Proceedings of the 4th IASTED International Conference on Visualization, Imaging and Image Processing*, pages 329–334, 2004.

[Medvet11] E. Medvet, A. Bartoli, and G. Davanzo. A probabilistic approach to printed document understanding. *International Journal on Document Analysis and Recognition*, 14(4) :335–347, 2011.

[Messmer94] B. T. Messmer and H. Bunke. Efficient error-tolerant subgraph isomorphism detection. *Shape, Structure and Pattern Recognition*, 231-240, 1994.

[Messmer98] B. T. Messmer and Horst Bunke. A new algorithm for error-tolerant subgraph isomorphism detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5) :493–504, 1998.

[Messmer99] B. T. Messmer and H. Bunke. A decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32(12) :1979–1998, 1999.

[Miller00] D. Miller, S. Boisen, R. Schwartz, R. Stone, and R. Weischedel. Named entity extraction from noisy input : speech and ocr. In *Proceedings of the 6th Applied Natural Language Processing Conference*, pages 316–324, 2000.

[Monge96] A. Monge and C. Elkan. The field-matching problem : algorithm and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[Monge97] A. Monge and C. P. Elkan. An effcient domain-independent algorithm for detecting approximately duplicate database records. In *the proceedings of the SIGMOD workshop on Data Mining and Knowledge Discovery*, 1997.

[Moreau08] E. Moreau, F. Yvon, and O. Cappé. Appariement d'entités nommées coréférentes : combinaisons de mesures de similarité par apprentissage supervisé. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles*, pages 488–497, 2008.

[Nagy92] G. Nagy, S. Seth, and M. Viswanathan. A prototype document image analysis system for technical journals. *IEEE Computer*, 25(7) :10–22, 1992.

[Niklas10] K. Niklas. *Unsupervised Post-Correction of OCR Errors*. PhD thesis, Leibniz-Universitat Hannover Germany, 2010.

[Osesina12] J. R. Talburt, M. Bell, I. Osesina, and J. Talburt. A data-intensive approach to named entity recognition combining contextual and intrinsic indicators. *International Journal of Business Intelligence Research*, 3(1) :55–71, 2012.

[Packer10] T. L. Packer, J. F. Lutes, A. P. Stewart, D. W. Embley, E. K. Ringger, K. D. Seppi, and L. S. Jensen. Extracting person names from diverse and noisy ocr text. In *Proceedings of the Fourth Workshop on Analytics for Noisy Unstructured Text Data*, pages 19–26, 2010.

[Pouliquen06] B. Pouliquen, R. Steinberger, C. Ignat, I. Temnikova, A. Widiger, W. Zaghouani, and J. Žižka. Multilingual person name recognition and transliteration. *CORELA - Cognition, Representation, Langage*, 3(2) :115–123, 2006.

[Prudent05] Y. Prudent and A. Ennaji. A new learning algorithm for incremental self-organizing maps. In *Proceedings of the European Eymposium on Artificial Neural networks*, pages 7–12, 2005.

[Quinlan93] J. R. Quinlan. *C4.5 : Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

[Riesen07] Kaspar Riesen, Stefan Fankhauser, and Horst Bunke. Speeding up graph edit distance computation with a bipartite heuristic. In *Mining and Learning with Graphs*, 2007.

[Rodriquez12] K. J. Rodriquez, M. Bryant, T. Blanke, and M. Luszczynska. Comparison of named entity recognition tools for raw ocr text. In *Proceedings of The 11th Conference on Natural Language Processing*, pages 410–414, 2012.

[Rohini00] S. Rohini, N. Cheng, and L. Wei. A hybrid approach for named entity and sub-type tagging. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 247–254, 2000.

[Sagot14] B. Sagot and K. Gabor. Détection et correction automatique d'entités nommées dans des corpus ocrisés. In *21ème conférence sur le Traitement Automatique des Langues Naturelles*, pages 437–442, 2014.

[Sanfeliu02] A. Sanfeliu, F. Serratosa, and R. Alquezar. Synthesis of function-described graphs and clustering of attributed graphs. *International Journal of Pattern Recognition and Artificial-Intelligence*, 16(6) :621–655, 2002.

[Sanfeliu83] A. Sanfeliu and K. S. Fu. A distance measure between attributed relational graphs for pattern recognition. *IEEE Transactions on System, Man and Cybernetics*, 13(3) :353–362, 1983.

[Sanroma12] G. Sanromà, R. Alquezar, and F. Serratosa. A new graph matching method for point-set correspondence using the em algorithm and softassign. *Computer Vision and Image Understanding*, 116(2) :292–304, 2012.

[Santosh13a] K. C. Santosh and A. Belaïd. Client-driven content extraction associated with table. In *Machine Vision and Applications*, pages 277–280, 2013.

[Santosh13b] K. C. Santosh and A. Belaïd. Document information extraction and its evaluation based on client's relevance. In *Proceedings of International Conference on Document Analysis and Recognition*, pages 35–39, 2013.

[Santosh13c] K. C. Santosh and A. Belaïd. Pattern-based approach to table extraction. In *Proceedings of the IAPR Iberian Conference on Pattern Recognition and Image Analysis*, pages 766–773. Volume 7887 of Lecture Notes in Computer Science, 2013.

[Santosh15] K. C. Santosh. g-dice : graph mining-based document information content exploitation. *International Journal on Document Analysis and Recognition*, 18(4) :337–355, 2015.

[Schenker03] Schenker Adam, Last Mark, Bunke Horst, and Kandel Abraham. *Graph Based Representations in Pattern Recognition*, chapter Comparison of Distance Measures for Graph-Based Clustering of Documents, pages 202–213. Springer Berlin Heidelberg, 2003.

[Sebastian04] T. B. Sebastian, P. N. Klein, and B. B. Kimia. Recognition of shapes by editing their shock graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5) :550–571, 2004.

[Seong93] D. S. Seong, H. S. Kim, and K. H. Park. Incremental clustering of attributed graphs. *IEEE Trans. Systems, Man, and Cybernetics*, 23(5) :1399–1411, 1993.

[Serratosa00] F. Serratosa, R. Alquezar, and A. Sanfeliu. Efficient algorithms for matching attributed graphs and function described graphs. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 867–872, 2000.

[Shapiro82] L. G. Shapiro and R. M. Haralick. Organization of relational models for scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(6) :595–602, 1982.

[Shearer01] K. Shearer, H. Bunke, and S. Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34(5) :1075–1091, 2001.

[Sole-Ribalta11] A. Sole-Ribalta and F. Serratosa. Models and algorithms for computing the common labelling of a set of attributed graphs. *Computer Vision and Image Understanding*, 115(7) :929–945, 2011.

[Srinivas10] M. Srinivas and C. Krishna Mohan. Efficient clustering approach using incremental and hierarchical clustering methods. In *Proceedings of International Joint Conference on Neural Networks*, pages 1–7, 2010.

[Staelin07] C. Staelin, M. Elad, D. Greig, O. Shmueli, and M. Vans. Biblio : Automatic meta-data extraction. *International Journal on Document Analysis and Recognition*, 10(2) :113–126, 2007.

[Tahmasebi13] N. Tahmasebi, K. Niklas, G. Zenz, and T. Risse. On the applicability of word sense discrimination on 201 years of modern english. *International Journal on Digital Libraries*, 13(3-4) :135–153, 2013.

[Talburt00] J. R. Talburt and M. Bell. A Bayesian approach to the identification of postal address lines utilizing word frequencies derived from expert coded corpora. In *Proceedings of the third International Symposium on Soft computing for history, Soft computing, multimedia and image processing*, pages 728–732, 2000.

[Talburt07a] N. Wu, J. R. Talburt, C. Heien, N. Pippenger, C. Chiang, E. Pierce, E. Gulley, and J. Moore. A method for entity identification in open source documents with partially redacted attributes. *Journal of Computing Sciences in Colleges*, 22(5) :138–144, 2007.

[Talburt07b] J. R. Talburt, N. Wu, E. M. Pierce, and R. R. Hashemi. Entity identification using indexed entity catalogs. In *Proceedings of the International Conference on Information and Knowledge Engineering*, pages 338–342, 2007.

[Talburt07c] J. R. Talburt, N. Wu, E. M. Pierce, C. C. Chiang, C. Heien, E. Gulley, and J. Moore. Entity identification in documents expressing shared relationships. In *Proceedings of the 11th WSEAS International Conference on SYSTEMS*, pages 224–229, 2007.

[Tan03] C. L. Tan and Q. H. Liu. Extraction of newspaper headlines from microfilm for automatic indexing. *International Journal of Document Analysais and Recognition*, 6(3) :201–210, 2003.

[Tanaseichuk15] O. Tanaseichuk, A. Hadj Khodabakshi, D. Petrov, J. Che, T. Jiang, B. Zhou, A. Santrosyan, and Y. Zhou. An efficient hierarchical clustering algorithm for large datasets. *Austin Journal of Proteomics, Bioinformatics and Genomics*, 2(1), 2015.

[Tao08] D. Tao, X. Tang, and X. Li. Which components are important for interactive image searching ? *IEEE Trans Circuits Syst Video Technol*, 18(1) :1–11, 2008.

[Tejada02] S. Tejada, C. A. Knoblock, and S. Minton. Learning domain-independent string transformation weights for high accuracy object identification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 350–359, 2002.

[Tsujimoto92] S. Tsujimoto and H. Asada. Major components of a complete text reading system. In *Proceedings of IEEE, vol. 80, no. 7*, pages 1133–1149, 1992.

[Vijaya04] P. A. Vijaya, M. Narasimha Murty, and D. K. Subramanian. Leaders-subleaders : An efficient hierarchical clustering algorithm for large datasets. *Pattern Recognition Letters*, 25(4) :505–513, 2004.

[Wacholder97] N. Wacholder, Y. Ravin, and M. Choi. Disambiguation of proper names in text. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 202–208, 1997.

[Wallis01] W. D. Wallis, P. Shoubridge, M. Kraetz, and D. Ray. Graph distances using graph union. *Pattern Recognition Letters*, 22(6-7) :701–704, 2001.

[Wang01] S. Wang, Y. Cao, and S. Cai. Using citing information to understand the logical structure of document images. *International Journal of Document Analysais and Recognition*, 4(1) :27–34, 2001.

[Wick07] M. Wick, M. Ross, and E. Learned-Miller. Context-sensitive error correction : Using topic models to improve ocr. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, pages 1168–1172, 2007.

[Wilson97] R. C. Wilson and E. R. Hancock. Structural matching by discrete relaxation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6) :634–648, 1997.

[Winkler99] W. E. Winkler. The state of record linkage and current research problems. Technical report, Statistics of Income Division, Internal Revenue Service Publication R99/04, 1999.

[Witten05] I. H. Witten and E. Frank. *Data Mining : Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann Publishers Inc., 2005.

[Wong90] A. K. C. Wong, M. You, and S. C. Chan. An algorithm for graph optimal monomorphism. *IEEE Transactions on System, Man and Cybernetics*, 20(3) :628–638, 1990.

[Xu01] L. Xu and I. King. A pca approach for fast retrieval of structural patterns in attributed graphs. *IEEE Transactions on System, Man and Cybernetics*, 31(5) :812–817, 2001.

[Yadav15] A. Yadav. Incremental k-means clustering algorithms : A review. *International Journal of Latest Trends in Engineering and Technology*, pages 136–140, 2015.

[Zou10] J. Zou, D. Le, and G. R. Thoma. Locating and parsing bibliographic references in html medical articles. *International Journal on Document Analysis and Recognition*, 13(2) :107–119, 2010.