HAL
open science

Apports des techniques d'apprentissage semi-supervisées dans l'établissement de liens entre artefacts de conception

Emma Effa Bella

To cite this version:

Emma Effa Bella. Apports des techniques d'apprentissage semi-supervisées dans l'établissement de liens entre artefacts de conception. Apprentissage [cs.LG]. Sorbonne Université, 2019. Français. NNT : 2019SORUS093 . tel-03409774

HAL Id: tel-03409774

https://theses.hal.science/tel-03409774v1

Submitted on 30 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE DE DOCTORAT
DE SORBONNE UNIVERSITÉ

Spécialité : Informatique

École doctorale n°130: EDITE

réalisée

Laboratoire de Paris 6 / équipe MoVe

sous la direction de Marie-Pierre Gervais & Reda Bendraou

présentée par

Emma Lucie Effa Bella

pour obtenir le grade de :

DOCTEUR DE SORBONNE UNIVERSITÉ

Sujet de la thèse :

Apports des techniques d'apprentissage semi-supervisées dans l'établissement de liens entre artefacts de conception

soutenue le 28 octobre 2019

devant le jury composé de :

Mme	Mireille Blay-Fornarino	Rapporteur
Mme	Marianne Huchard	Rapporteur
M.	Fabrice Kordon	Examinateur
M.	Stephen Creff	Examinateur
M.	Laurent Wouters	Examinateur
Mme	Marie-Pierre Gervais	Directeur de thèse
M.	Reda Bendraou	Co-Directeur de thèse

REMERCIEMENTS

Après avoir remercié Mr Laurent WOUTERS, Docteur et Mr Ali KOUDRI, Docteur qui m'ont offert la possibilité d'effectuer cette thèse dans l'Institut de Recherche Technologique SystemX, et pour toutes nos discussions et leurs conseils qui m'ont accompagnés tout au long de cette thèse, j'adresse mes remerciements les plus chaleureux :

* à mes directeurs de thèse Mme Marie-Pierre GERVAIS et Mr. Reda BENDRAOU, Professeurs à l'Université de Nanterre, pour m'avoir donné les moyens et l'assistance nécessaire à la réalisation de nos travaux et au bon déroulement de cette thèse ; Qu'ils soient aussi remerciés pour leur gentillesse, leur disponibilité permanente, leurs conseils et pour les nombreux encouragements qu'ils m'ont prodigués ;

* aux rapporteurs, Mme Mireille BLAY-FORNARINO, Professeur à l'Université de Nice et Pr. Marianne HUCHARD, Professeur à l'Université de Montpellier, pour l'honneur qu'ils m'ont fait en acceptant d'être rapporteurs de cette thèse ;

* aux examinateurs, Mr Fabrice KORDON, Professeur à Sorbonne Université pour l'honneur qu'il me fait d'être dans mon jury de thèse et Mr Stephen CREFF, Docteur et Mr Laurent WOUTERS, Docteur pour avoir accepté de participer à mon jury de thèse et pour leur participation scientifique ainsi que le temps qu'ils ont consacrés à ma recherche.

* aux membres des équipes du projet ISC pour leurs conseils et leur encadrement : Mr. Stephen CREFF, Mme Anouk DUBOIS, Mlle Hanane FADIAW, Mr. Jérôme LENOIR et Mr. Sébastien MADELENAT, Mr. Mohamed HASSINE et Mr Leandro BATISTA ;

* au personnel d'IRT SystemX en particulier au personnel administratif et technique ;

* à mes collègues doctorants et post doctorants de l'équipe MoVe : Xhevahire TËRNAVA, Alex CHUESHEV, Anas SHATNAWI, Jabier MARTINEZ, Sara HOUHOU ;

* aux professeurs et ingénieurs de recherche de l'équipe Move ;

* au personnel administratif et technique de l'école doctorale en particulier à Samiha TABA ;

* à tous les membres de ma famille qui m'ont soutenue durant ces 3 dernières années.

* Je remercie finalement toute personne qui de près ou de loin a participé à la réalisation de mes travaux.

Emma Lucie Effa Bella **28 octobre 2019**

Sujet : Apports des techniques d'apprentissage semi-supervisées dans l'établissement de liens entre artefacts de conception

Résumé : Dans un environnement collaboratif de développement de systèmes complexes, plusieurs entreprises doivent échanger un nombre important de modèles hétérogènes et d'exigences. Durant les phases du cycle de vie du système, ces artefacts, reliés les uns aux autres et issus de différents outils de modélisations, évoluent constamment. Dans un tel environnement hétérogène et volatil, il est crucial de gérer l'impact des différents changements se produisant dans les différents espaces de conception. La traçabilité telle que définie par l'International Council on Systems Engineering (INCOSE) répond à ce besoin.
Toutefois, établir des liens entre des exigences et des modèles en ingénierie des systèmes complexes suppose de faire face à une volumétrie importante des artefacts. Par exemple, pour une spécification d'un véhicule autonome comprenant 3 000 exigences et 400 éléments de modèles, il faudrait en théorie vérifier de l'ordre d'un million de liens potentiels. Bien que plusieurs approches aient été proposées pour l'identification des liens de traçabilité, le processus de validation des liens est toujours chronophage et générateur d'erreurs. Ceci est principalement dû à la prépondérance d'opérations manuelles lors de ce processus.
Dans cette thèse, nous proposons une approche semi supervisée qui permet d'apprendre via un modèle probabiliste à reconnaître des liens de traçabilité valides ou non valides à partir de mesures et de scores de similarité. Cette approche fournit ainsi une mesure quantitative de confiance sur chaque lien candidat. Cette dernière permet potentiellement à l'expert en phase de validation d'optimiser son effort de vérification des liens tout en maîtrisant les risques d'erreur.
Nous avons évalué notre approche à la fois sur des jeux de données de référence utilisés dans la littérature de la traçabilité et sur des cas d'étude industriels. Nous avons montré la pertinence de notre approche pour l'identification des liens de traçabilité par rapport à des méthodes de traçabilité de l'état de l'art. Nous obtenons en effet une réduction des faux positifs d'environ 80% par rapport aux méthodes de l'état de l'art dans les cas industriels. Dans le même temps, nous conservons un nombre de liens valides (vrai positif), allant jusqu'à 75%.
Le prototype mis en place, appelé Aggregation Trace Links Support (ATLaS), est testé par les partenaires de l'Institut de Recherche SystemX dans le cadre du projet Éco-mobilité par Véhicules Autonomes (EVA).

(1700 car max, espaces inclus)

Mots clés : traçabilité, hétérogénéité, exigences, modèles, systèmes complexes, machine learning, traitement automatique du langage

Subject : Benefits of semi-supervised learning techniques in traceability links recovering between design artifacts

Abstract: During the development of complex systems, several enterprises exchange a large number of heterogeneous models and requirements. During the phases of the system's life cycle, these artifacts, linked to each other and derived from different modelling tools, are constantly evolving. In such a heterogeneous and volatile environment, it is necessary to manage the impact of the different changes occurring in the different design spaces. Traceability as defined by the International Council on Systems Engineering (INCOSE) meets this need.
However, establishing links between requirements and models in complex systems engineering requires dealing with a large volume of artifacts. For example, a specification of an autonomous vehicle with 3,000 requirements and 400 model elements, it would theoretically be necessary to check about one million of potential links. Although several approaches have been proposed for identifying traceability links, the validation process is always time-consuming and error-prone. This is mainly due to the predominance of manual operations during this process.
In this thesis, we propose a semi-supervised approach that learns through a probabilistic model to recognize links or no links from similarities measures and scores. This approach provides a quantitative confidence measure on each candidate link. This measure allows the expert in the validation phase to optimize his verification effort while reducing the risks of error.
We evaluated our approach on benchmarks in the traceability and on industrial case studies. The results show that our approach have better results than state-of-the-art traceability methods. We obtain a reduction of no links (false positive) of about 80% compared to state-of-the-art methods in industrial cases, while, keeping a number of links (true positive), up to 75% , at the same time. The prototype implemented, called Aggregation Trace Links Support (ATLaS), is being tested by the SystemX Research Institute's partners as part of the Eco-mobility by Autonomous Vehicles (EVA) project.

(1700 chars max, spaces included)

Keywords : traceability, heterogeneity, requirements, models, complex systems, machine learning, automatic language processing

Table des matières

I Introduction **1**

1 Contexte et motivations **3**
1.1 L'ingénierie des systèmes aujourd'hui . . . 3
1.2 Vers une meilleure ingénierie système . . . 4
1.3 Motivations . . . 5
1.4 Limites des approches existantes . . . 7
1.5 Contributions . . . 9
1.6 Organisation du mémoire . . . 10

II État de l'art et des pratiques **13**

2 Traçabilité en Ingénierie Systèmes et Logicielle **17**
2.1 Introduction . . . 17
2.2 Historique de la traçabilité . . . 17
2.2.1 Traçabilité des exigences . . . 17
2.2.2 Traçabilité des modèles . . . 18
2.3 Quelques définitions . . . 19
2.3.1 Définition de la traçabilité . . . 19
2.3.2 Modèle de rédaction d'exigences . . . 20
2.3.3 Métamodèles de traçabilité . . . 22
2.3.4 Taxonomie des types de liens de traçabilité . . . 22
2.4 Identification des types de liens . . . 23
2.4.1 Le lien de satisfaction : les techniques de Holbrook et al. . . . 23
2.4.2 Le lien de recouvrement : la technique de Zisman et al. . . . 25
2.4.3 Le lien de raffinement : la technique de Goknil et al. . . . 28
2.5 Conclusion . . . 30

3 Élicitation et maintenance des liens de traçabilité **33**
3.1 Introduction . . . 33
3.2 Élicitation des liens de traçabilité . . . 33
3.3 Maintenance des liens de traçabilité . . . 34
3.4 Validation des liens de traçabilité . . . 36
3.5 Approches d'élicitation et de maintenance des liens de traçabilité . . . 38

3.5.1 Les approches de Recherche d'Information . . . 38
3.5.2 Les approches basées sur l'apprentissage automatique . . . 39
3.6 Synthèse . . . 41
3.7 Problématiques de la thèse . . . 43
a) Réduction du coût de l'élicitation des liens de traçabilité . . . 43
b) Identification des types de liens . . . 44
3.8 Conclusion . . . 44

4 Bases méthodologiques **47**
4.1 Introduction . . . 47
4.2 Techniques du Traitement Automatique des Langues . . . 47
4.2.1 Bases du Traitement Automatique des Langues . . . 47
a) Segmentation en phrases . . . 47
b) Séparation des termes . . . 48
c) Retrait des mots vides . . . 48
d) Racinisation . . . 48
e) Lemmatisation . . . 48
f) Thesaurus et dictionnaires . . . 49
g) Etiquetage morpho-syntaxique (part-of-speech tagging) . . . 49
h) Séparation en syntagmes (Text chunking) . . . 49
i) Mesures de similarité et de dissemblance . . . 49
4.2.2 Techniques récentes du traitement automatique des langues . . . 50
a) Word embeddings . . . 50
b) Word2Vec . . . 50
c) Global Vectors (GLoVe) . . . 53
d) Similarité entre les plongements de mots . . . 54
e) Synthèse . . . 55
4.2.3 Sentence embeddings . . . 55
a) Smooth Inverse Frequency (SIF) . . . 55
b) Doc2Vec . . . 56
4.3 Techniques traditionnelles de Recherche d'Information en traçabilité . . . 59
4.3.1 Term Frequency - Inverse Document Frequency (TF-IDF) . . . 59
4.3.2 Vector Space Model (VSM) . . . 60
4.3.3 Latent Semantic Indexing (LSI) . . . 61
4.4 Techniques d'apprentissage automatique . . . 63
4.4.1 Cluster Hypothesis . . . 63
4.4.2 Latent Dirichlet Allocation (LDA) . . . 63
4.4.3 LabelSpreading . . . 65
4.5 Métriques d'évaluation des approches de traçabilité . . . 66
4.5.1 Rappel . . . 67
4.5.2 Précision . . . 67
4.5.3 F-mesure . . . 67
4.6 Conclusion . . . 68

III Contributions **69**

5 ATLaS, un framework d'identification des liens de traçabilité **73**
5.1 Introduction . . . 73
5.2 Présentation générale de l'approche . . . 73
5.3 Méthodologie de recherche . . . 75
5.3.1 Étape 1 : Collecte d'informations sur des paires d'artefacts . . . 75
5.3.2 Étape 2 : Élaboration d'un jeu d'exemples . . . 76
5.3.3 Étape 3 : Classification des liens et mesure de confiance . . . 77
5.4 Architecture du framework ATLaS . . . 77
5.4.1 Module1 : Pré-traitement des artefacts . . . 79
a) Segmentation en phrases et retrait des mots vides . . . 79
b) Séparation des termes . . . 79
c) Séparation en syntagmes . . . 79
5.4.2 Module 2 : Calcul des mesures et des scores de similarités . . . 79
a) Calcul des mesures de similarité . . . 80
b) Calcul des scores de similarité . . . 80
5.4.3 Module 3 : Calcul de la mesure de confiance . . . 82
a) Construction de la base d'exemples . . . 82
b) Classification des liens . . . 82
5.4.4 Module 4 : Identification des types de liens . . . 83
a) Liens de satisfaction . . . 83
b) Liens de recouvrement . . . 84
c) Liens de raffinement . . . 85
5.5 Intégration d'ATLaS dans l'espace collaboratif du projet EVA . . . 86
5.5.1 Projet Éco-mobilité par Véhicules Autonomes (EVA) . . . 87
5.5.2 Implémentation du framework ATLaS . . . 88
5.6 Conclusion . . . 89

6 Évaluation **91**
6.1 Introduction . . . 91
6.2 Cas d'étude . . . 91
6.2.1 Cas d'étude industriels ARC-IT . . . 91
6.2.2 Cas d'étude académiques . . . 92
a) Icebreaker . . . 92
b) HIPAA . . . 93
c) EasyClinic . . . 93
d) CM1-NASA . . . 93
6.3 Expérimentation de l'étape 1 : Collecte d'informations sur des paires d'artefacts 93
6.4 Expérimentations de l'étape 2 : Élaboration d'un jeu d'exemples . . . 95
6.5 Expérimentation de l'étape 3 : Classification des liens et mesure de confiance 96
6.5.1 Vérification de la Cluster Hypothesis . . . 96
6.5.2 Pouvoir discriminant du modèle . . . 99
6.5.3 Courbe Rappel - Précision . . . 102
6.5.4 Courbe de F-mesure . . . 105
6.6 Discussion . . . 111

6.7 Conclusion 112

IV Conclusion **113**

7 Conclusion et perspectives **115**
7.1 Contributions 115
7.2 Perspectives : vers un système de recommandation de liens de traçabilité . . . 115
7.3 Liste des publications liées à la thèse 116

A Algorithme de construction d'un sous-ensemble de paires d'artefacts **117**

B Quelques liens de traçabilité **119**

C Liste des abréviations **125**

Bibliographie **127**

Liste des tableaux

3.1 Évaluation des approches d'élicitation et de maintenance des liens de traçabilité 42
3.2 Récapitulatif des problématiques 45

4.1 Récapitulatif des avantages et des inconvénients des techniques du traitement automatique des langues 58
4.2 Récapitulatif des avantages et des inconvénients des techniques de Recherche d'Information 63
4.3 Récapitulatif des avantages et des inconvénients des techniques d'apprentissage automatique 66

6.1 Description des jeux de données industriels. 92
6.2 Description des jeux de données académiques 93
6.3 Exactitude des vrais et des faux liens dans les six cas d'études 95

Table des figures

1.1 Coûts engagés vs coûts effectifs 4
1.2 Gestion du changement et Processus collaboratif 6
1.3 Gestion de la traçabilité : vue méthodologique 8
1.4 Résumé des contributions 9

2.1 Traçabilité et phases de spécifications des exigences [1] 20
2.2 Modèle de rédaction d'exigences avec conditions [2] 21
2.3 lien de satisfaction entre deux exigences et un élément de modèle 25
2.4 lien de recouvrement *exigences - exigences* 27
2.5 lien de recouvrement *éléments de modèles - éléments de modèle* 27
2.6 lien de raffinement *exigences - exigences* 29
2.7 lien de raffinement *exigences - exigences* 30
2.8 lien de raffinement *éléments de modèle – éléments de modèle* 31

3.1 Processus d'élicitation des liens [3] 34
3.2 Processus de maintenance des liens [3] 35

4.1 Illustration de la technique CBOW avec le terme « *System* » de l'exigence *1245* 52
4.2 Illustration de la technique Skip-Gram avec le terme « *Terminate* » du but *113* 53
4.3 Paires de termes masculins et féminins et trios de termes comparatifs superlatifs pré-entraînés avec la technique de représentation des termes *Glove* [4] . 54
4.4 Illustration schématique du Word Move Distance avec l'exigence *1245* et le but *113* 54
4.5 TF-IDF des termes de l'exigence *1245* et du but *113* 60
4.6 Résumé du fonctionnement de la technique VSM 61
4.7 Résumé du fonctionnement de la technique LSI 62
4.8 Illustration du calcul de LSI avec l'exigence *1245* et le but *113* 62
4.9 Résumé du fonctionnement de la technique LDA 64
4.10 Illustration du calcul de LDA avec l'exigence *1245* et le but *113* 65
4.11 Illustration de l'algorithme LabelSpreading 66
4.12 Métriques de traçabilité [5] 67

5.1 Présentation de l'heuristique 77
5.2 Architecture d'ATLaS (Aggregation Trace Links Support) 78
5.3 Fonctionnement d'ATLaS avec la combinaison *VSM-LSI-LDA-S1-S2-S3* . . . 80
5.4 fonctionnement d'ATLaS dans l'espace collaboratif 87

6.1 Corrélation entre les mesures et les scores de similarité : cas d'étude académiques 94
6.2 Corrélation entre les mesures et les scores de similarités : cas d'étude industriels 94
6.3 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas Icebreaker 96
6.4 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas HIPAA 97
6.5 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas Easyclinic 97
6.6 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas CM1-NASA . . . 97
6.7 Vérification de la Cluster Hypothesis pour la classe des faux liens : cas d'étude académiques . . . 98
6.8 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas ARC-IT1 98
6.9 Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas ARC-IT2 98
6.10 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas Icebreaker . . . 99
6.11 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas HIPAA . . . 100
6.12 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas Easyclinic . . . 100
6.13 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas CM1-NASA . . . 101
6.14 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas ARC-IT1 . . . 101
6.15 Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas ARC-IT2 . . . 102
6.16 Courbe rappel-précision : cas Icebreaker . . . 103
6.17 Courbe rappel-précision : cas HIPAA . . . 103
6.18 Courbe rappel-précision : cas Easyclinic . . . 104
6.19 Courbe rappel-précision : cas CM1-NASA . . . 104
6.20 Courbe rappel-précision : cas ARC-IT1 . . . 104
6.21 Courbe rappel-précision : cas ARC-IT2 . . . 105
6.22 Courbe F-mesure : cas Icebreaker . . . 105
6.23 Courbe F-mesure : cas HIPAA . . . 106
6.24 Courbe F-mesure : cas Easyclinic . . . 106
6.25 Courbe F-mesure : cas CM1-NASA . . . 107
6.26 Courbe F-mesure : cas ARC-IT1 . . . 107
6.27 Courbe F-mesure : cas ARC-IT2 . . . 107
6.28 Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas Icebreaker . . . 108
6.29 Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas HIPAA . . . 109
6.30 Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas Easyclinic . . . 109
6.31 Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas CM1-NASA . . . 110
6.32 Récapitulatif des vrais liens trouvés et identifiés : cas ARC-IT1 . . . 110
6.33 Récapitulatif des vrais liens trouvés et identifiés : cas ARC-IT2 . . . 111

Première partie

Introduction

1 – Contexte et motivations

1.1 L'ingénierie des systèmes aujourd'hui

La complexité des systèmes est une réalité très pragmatique, que les parties prenantes d'un projet doivent affronter et surmonter tout au long du cycle de vie d'un système industriel. Cette complexité se retrouve de fait dans la totalité des grandes phases d'ingénierie d'un système, depuis l'analyse des besoins jusqu'à sa validation finale en passant par sa phase d'intégration. Ces différentes phases d'ingénierie ont par ailleurs évolué progressivement dans l'industrie au fil des ans pour passer d'un état des lieux très informel à une situation beaucoup plus mature qui nécessite désormais des environnements de conception outillés [6]. Nous pouvons notamment citer :

— L'ingénierie des exigences qui nécessite de pouvoir assurer la traçabilité et gérer en configuration de manière non ambiguë toutes les exigences d'un système pendant toute la durée de sa conception et en interaction avec l'ensemble de ses parties prenantes (régulateurs, utilisateurs, clients, responsables projets, etc.),

— La conception système qui exige de savoir réconcilier en permanence les approches de nombreuses disciplines scientifiques et techniques extrêmement hétérogènes (contrôle, électronique, informatique, matériaux, mécanique, signal, thermique, etc.) tout en respectant le plus souvent des logiques très fortes de sécurité et de sûreté,

— L'ingénierie des processus qui requiert la collaboration entre les différentes parties prenantes (experts, fournisseurs, clients, etc.) durant tout le cycle de vie du système et la prise en compte des contraintes organisationnelles du projet permettant de construire le système (délais, budget, réglementation, etc.). De manière plus générale, l'ingénierie des processus s'intéresse à l'utilisation optimale des ressources employées dans la construction du système dans un cadre contraint et réglementé, et ce quel que soit leur nature (humaine, logicielle, matérielle, etc.).

L'étude d'un système se fait généralement à travers le prisme de différents outils qui offrent autant de points de vue nécessaires aux parties prenantes du processus, de sa conception à sa validation opérationnelle. Bien que les experts raisonnent de manière conceptuelle sur un même ensemble d'objets, la prise de décision se fait dans des espaces techniques bien séparés qui ne facilitent pas la mise en relation entre paramètres de points de vue, ni ne permettent de détecter d'éventuels conflits entre ces derniers.

Aujourd'hui, le faible niveau de maîtrise de la cohérence globale des cycles d'ingénierie crée de manière récurrente de grandes difficultés dans l'intégration d'une part et dans la conduite du changement d'autre part. Par conséquent, un des plus grands défis de l'ingénierie système est de réussir à définir et à mettre en œuvre un processus de développement agile

qui minimise les coûts et les délais quand plusieurs organisations sont impliquées.

1.2 Vers une meilleure ingénierie système

Avec l'évolution des technologies du numérique se pose le problème de la gestion des risques dans un environnement de plus en plus ouvert et incertain. En particulier, je m'intéresse dans cette thèse aux risques liés à une mauvaise circulation / synchronisation des informations entre les partenaires d'un projet. Aujourd'hui, une manière de gérer les risques efficacement consiste, selon la communauté de l'ingénierie des modèles (IdM), à utiliser des modèles en lieu et place de documents au plus tôt dans les phases de conception. Cette démarche est largement argumentée par la littérature et par les expérimentations [7, 8, 9]. Je mentionnerai en particulier l'étude parue dans [10] et illustrée par la figure 1.1.

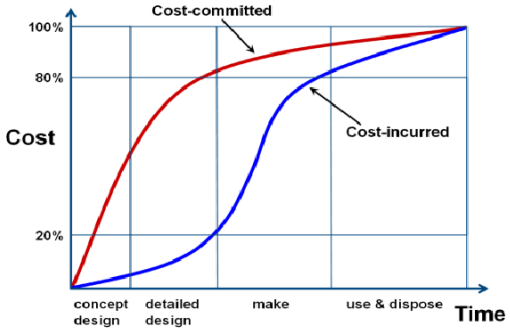

Fig. 1.1 *Coûts engagés vs coûts effectifs*

Cette figure montre le graphe des coûts généralement engagés dans le développement d'un système. On constate en effet sur cette courbe que 80% des coûts sont engagés durant les phases de conception préliminaires et de conception détaillée ; dans le même temps, les coûts effectifs de ces phases représentent à peine 20% des coûts effectifs totaux ; cette courbe traduit le fait que les efforts consentis dans les phases d'ingénierie système ne sont toujours pas à la hauteur de ses enjeux. Ainsi, l'utilisation de modèles dans les phases de conception permettrait de réduire les erreurs de spécifications quand plusieurs partenaires sont impliqués, et éviter ainsi un effet boule de neige coûteux à bien des égards. Dans la pratique, les approches proposées par la communauté de l'ingénierie des modèles, centrées sur l'utilisation de modèles et de transformations de modèles apportent effectivement un plus comparées aux approches centrées sur les documents, mais elles ne résolvent pas le problème de la **gestion du changement** ni celui du partage d'une **vision commune**.

Pour aller plus loin, une large étude menée par le Standish Group [11] sur un nombre important de projets IT de diverses tailles vient renforcer cette nécessité de mettre plus d'effort en phase amont du cycle de développement. Cette étude indique en particulier que les raisons principales d'échecs des projets sont :

1. La mauvaise vision / compréhension / maîtrise du besoin,
2. Le manque d'implication des différentes parties prenantes, en particulier du client, très tôt dans le cycle de développement,

3. La grande difficulté pour les équipes impliquées dans le cycle de vie du produit de collaborer (silos entre métiers),
4. La grande difficulté des acteurs du développement à faire les bons choix du fait de l'absence de métriques pertinentes.

Tous ces points sont liés à l'intégration cohérente des points de vue d'experts et à la capacité de les réconcilier lorsque changement il y a. En effet, les parties prenantes se sentent davantage impliquées si elles ont une vision claire du problème, si elles savent mesurer l'impact global des choix pris localement, dans leur domaine et au sein de leur organisation ; il faut également que ces parties aient accès aux informations pertinentes (en temps réel) qui peuvent impacter leurs propres choix. Cela nécessite effectivement de "dé-siloter" d'une certaine manière les espaces métiers et techniques.

1.3 Motivations

Les problèmes présentés dans la section précédente sont d'autant plus compliqués que ces dernières années ont vu l'émergence de l'entreprise étendue pour répondre à la nécessité de satisfaire ou anticiper rapidement un besoin du marché. L'entreprise étendue est une solution organisationnelle pouvant se manifester au travers d'une collaboration pérenne ou opportuniste lorsqu'une entreprise n'est en mesure d'apporter une solution seule. Cela permet surtout pour une entreprise d'assurer sa survie dans un environnement de plus en plus ouvert et compétitif. Dans ce paradigme émergent, chaque entreprise amène son savoir-faire et ses actifs afin de créer de la valeur ajoutée, et ainsi occuper ou renforcer une place stratégique sur le marché.

Comme l'illustre la figure 1.2, c'est dans un contexte d'entreprise étendue et dans une démarche Model-Based System Engineering (MBSE) inter-disciplinaires que notre thèse s'inscrit.

Les entreprises impliquées dans l'ingénierie collaborative doivent avant tout être en mesure de comprendre les changements impactant le cycle de vie du système. Ensuite, il s'agit d'accompagner et de maîtriser ce changement, ce qui ne peut se faire sans le partage d'une vision commune et un minimum de collaboration.

La conduite du changement nécessite notamment de mettre en place des outils permettant d'interfacer les différents processus internes / externes des entreprises impliquées et de mettre en relation les données d'ingénierie, au-delà des problèmes posés par la propriété intellectuelle. L'acceptation collective d'un contrat pour répondre à un besoin va nécessairement remettre en cause les référentiels internes de chaque entreprise ainsi qu'une grande partie de ses acquis. La manière dont ces derniers seront mis à contribution et questionnés va conditionner la qualité de la réponse apportée collectivement. L'objectif est en définitive d'aboutir à la construction d'un système qui réponde au mieux au besoin et à la stratégie des entreprises impliquées, tout en respectant les contraintes internes (coût, délai ou performance) ou externes (réglementation, certification ou environnement). À travers notre thèse, nous nous proposons de fournir des moyens permettant de contribuer au partage d'une vision commune et à son maintien à travers le changement.

Dans les faits, une bonne partie de l'information permettant de bien comprendre les tenants et aboutissants d'un projet, et permettant de bien justifier les choix de conception /

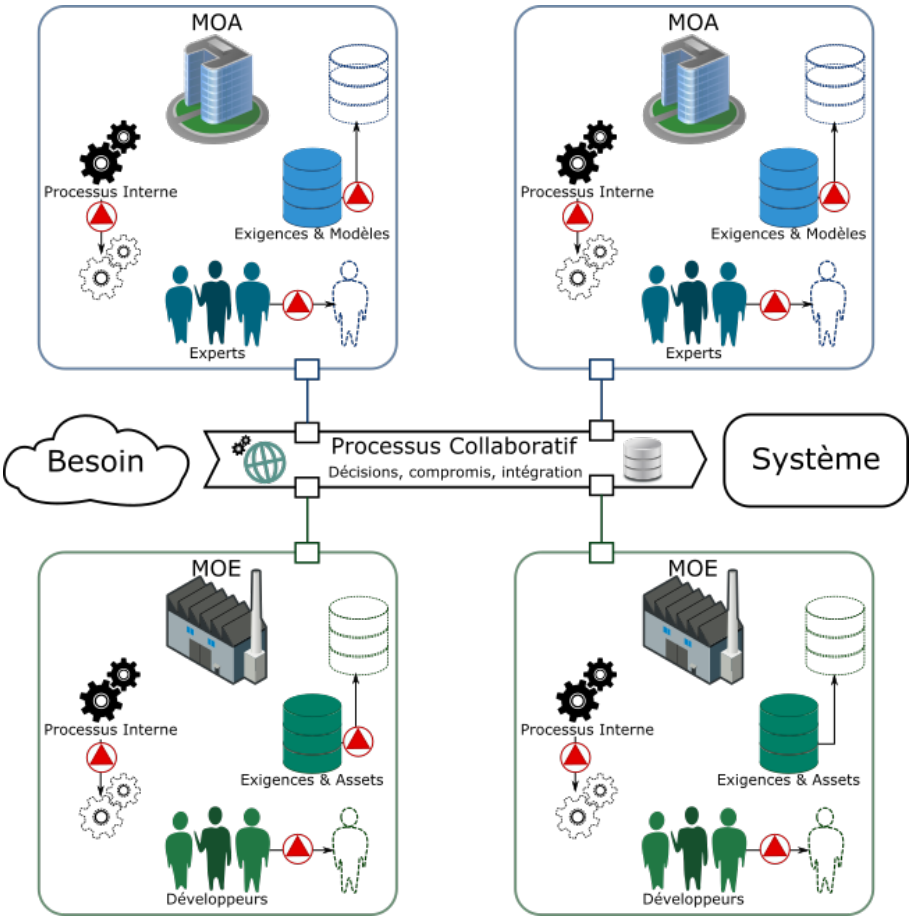

Fig. 1.2 *Gestion du changement et Processus collaboratif*

développement, est généralement implicite. Cette information non partagée, implicite, peut nuire à la compréhension globale des spécifications et à une bonne prise de décision commune.

Afin d'améliorer cette situation et contribuer à une meilleure compréhension du problème et des solutions proposées, nous proposons d'expliciter les liens de traçabilité qui n'ont pas été exprimés entre artefacts d'ingénierie. La syntaxe et la sémantique associées à ces liens devront permettre de raisonner globalement sur un ensemble d'artefacts d'ingénierie de manière à déterminer la cohérence et dans une certaine mesure la complétude des spécifications et des développements.

Par nos travaux, nous souhaitons contribuer à la construction et à la mise à jour d'une cartographie de l'information nécessaire à une meilleure compréhension globale du besoin, des choix réalisés collectivement pour le satisfaire, et en suivre les évolutions au-delà des silos métiers / techniques / technologiques. Étant donné la complexité du sujet, nous prenons le parti de simplifier le problème en ne considérant dans nos expérimentations que les liens entre exigences et des modèles. La raison de ce choix est que la compréhension des exigences, de leurs relations et de leurs déclinaisons (modèles) constitue à notre sens un facteur déterminant dans la conduite du changement [12]. En effet, comprendre et expliciter les exigences, leur nature et leurs relations ne peut qu'améliorer la vision globale d'un projet et consolider

ainsi la prise de décision collective. Par exemple, comprendre comment une exigence métier est reliée à une exigence réglementaire ou comment elle est dérivée en exigences techniques permettrait de faire de meilleurs choix et surtout de bien les justifier.

1.4 Limites des approches existantes

Dans un contexte d'entreprise étendue, la conception de systèmes complexes passe par la collaboration de différentes parties prenantes. Ces dernières produisent et utilisent une variété d'artefacts. Par exemple, la spécification des systèmes d'une voiture de 2004 comprend environ 20 000 pages [13]. Au cours du projet, cette volumétrie d'artefacts va être soumise à des changements fréquents. Les artefacts étant corrélés entre eux, la modification de l'un d'entre eux conduit à modifier ceux avec lesquels il est en relation.

Dans de telles situations, la ***traçabilité*** joue un rôle important. Elle aide à comprendre quels artefacts sont reliés les uns aux autres et permet de savoir quelles caractéristiques ont déjà été spécifiées, mises en œuvre ou testées.

Bien qu'il existe un vaste corpus de connaissances sur la traçabilité [14, 15, 16, 17, 18], il est courant de constater que, dans la pratique, la traçabilité n'est soit pas du tout établie [19], soit seulement établie lorsque les normes et les contrats l'exigent [20].

Ainsi, la mise en œuvre de la traçabilité dans les industries est encore confrontée à de nombreux problèmes [18]. Les sections suivantes décrivent brièvement quelques-uns de ces problèmes.

Problème 1 : Hétérogénéité des artefacts et des outils. Dans la conception d'un système, différents artefacts et outils sont utilisés. Cette hétérogénéité est due tout d'abord au fait que les modèles sont spécifiques aux disciplines qui les produisent. Par exemple, un modèle peut décrire l'architecture du système tandis qu'un autre décrit les pannes et les défaillances du système. Elle est également liée au fait que ces modèles sont exprimés avec différents langages de modélisation. De plus, les artefacts spécifiés dans ces différents outils peuvent contenir des informations redondantes. Ce qui peut entraîner des incohérences lorsque le système évolue et que seuls certains artefacts sont mis à jour. La plupart des outils de traçabilité ne prennent pas en charge les liens vers des artefacts situés à l'extérieur des outils ou ne prennent en charge que les liens vers des outils spécifiques.

Problème 2 : Interprétation sémantique des liens. Les liens de traçabilité peuvent être de différents types suivant leur fonction et les artefacts qu'ils relient. Les taxonomies de liens de traçabilité sont généralement définies dans un modèle d'information de traçabilité. Il peut, par exemple, prendre la forme d'un métamodèle, d'un schéma de base de données ou d'une ontologie. Cependant, cette classification des liens de traçabilité dépend du domaine d'étude, de l'entreprise ou même du projet [17]. Elle n'est donc pas standardisée. De plus, les taxonomies de liens de traçabilité sont suggérées sur la base de leurs propriétés structurelles et sémantiques, elles n'ont pas de niveau d'abstractions communs, et il existe très peu de définitions formelles des différents types de liens [21]. La sémantique d'un lien est donc soit fusionnée avec l'outillage quand il existe ou fonction de l'interprétation de chaque partenaire [22].

Problème 3 : Efficacité et confiance dans les outils de traçabilité. Il est important de s'assurer que les outils de traçabilité soient robustes, adaptés aux contextes

industriels et supportent le passage à l'échelle [23, 24]. Dans le domaine de l'ingénierie système, où le nombre d'exigences peut être supérieure à 2000, aucune solution concrète n'a encore été suggérée à cet égard [18, 25]. Ainsi, l'utilisation de ces outils de traçabilité diminue la motivation des industries qui considèrent que leur entretien est très coûteux au regard des bénéfices engendrés [26].

Problème 4 : Charge de travail allouée à la traçabilité. L'élicitation de liens de traçabilité prend beaucoup de temps lorsqu'elle est effectuée manuellement. En effet, le coût de cette activité devient rapidement rédhibitoire dès lors qu'il y a une forte volumétrie d'artefacts. De plus, les liens de traçabilité deviennent obsolètes lorsque les artefacts qu'ils relient évoluent. Ces derniers doivent donc être mis à jour (maintenus). La création et la maintenance manuelle des liens de traçabilité sont des tâches chronophages et génératrices d'erreurs.

Par souci de concision, ces problèmes seront souvent référencés dans la suite du document par :

— ***problème 1 : Hétérogénéité*** pour le *Problème 1 : Hétérogénéité des artefacts et des outils*,
— ***problème 2 : Interprétation*** pour le *Problème 2 : Interprétation sémantique des liens*,
— ***problème 3 : Efficacité et confiance*** pour le *Problème 3 : Efficacité et confiance dans les outils de traçabilité*
— et ***problème 4 : Charge de travail*** pour le *Problème 4 : Charge de travail alloué à la traçabilité.*

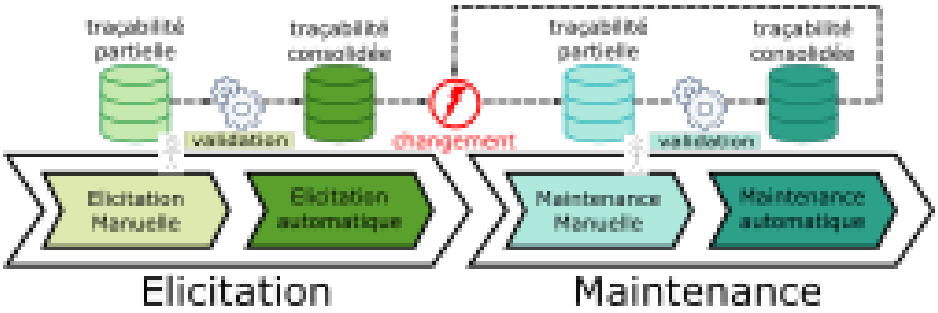

Fig. 1.3 *Gestion de la traçabilité : vue méthodologique*

La figure 1.3 synthétise le ***problème 4 : Charge de travail*** auquel nous souhaitons principalement apporter une réponse. L'élicitation manuelle des liens de traçabilité aboutit à une base d'exigences partielle, formalisée (modèles) ou non (exigences textuelles). L'élicitation automatique des liens entre exigences permet de compléter et consolider cette base afin d'apporter une meilleure vision et servir de meilleurs choix. Dans cette figure, chaque base représente l'agrégation des différentes bases des parties impliquées. Lorsque la base est consolidée, celle-ci est amenée à changer au gré des évolutions en termes de besoins, de technologies ou encore de législation. Ainsi la modification d'une exigence quelconque nécessite pour conserver une cohérence globale de revoir les liens de traçabilité de toutes les exigences qui lui sont associées, directement ou indirectement. Ces deux activités d'élicitation et de maintenance sont aujourd'hui essentiellement humaines, chronophages et sujettes à oubli ou erreur. De fait, la base d'exigences se retrouve dans un état partiel (liens manquants,

caduques, ou faux) ; ceci nécessite alors de disposer d'un outil de création / maintenance automatique qui permettra de corriger / compléter la base.

De plus, comme l'illustre la figure 1.3, une tâche commune à l'élicitation et à la maintenance est la validation des liens. C'est une tâche importante pour ces activités car elle assure l'intégrité des liens de traçabilité créés ou maintenus. De ce fait, elle nécessite des interventions humaines. Ce qui fait d'elle la tâche la plus coûteuse de la traçabilité. Notamment, le coût de la traçabilité dans le domaine réglementaire peut facilement dépasser 1 million de dollars US [27]. Soulignons, tout de même, que ce coût non négligeable doit être mis en balance avec des coûts bien plus élevés liés à l'élicitation ou la maintenance de liens permettant de démontrer la conformité et la sûreté de fonctionnement de systèmes. Ainsi, une étude approfondie de cette tâche pour réduire les efforts humains investis est donc crucial. Par conséquent, nos travaux de thèse sont principalement centrés sur le ***problème 4 : Charge de travail***.

1.5 Contributions

Au regard du ***problème 4 : Charge de travail***, nous proposons de fournir une démarche outillée d'aide à l'élicitation des liens de traçabilité. Nous proposons un framework de traçabilité basé sur la combinaison de méthodes de recherche d'information et d'apprentissage automatique. Pour cela, nous faisons l'hypothèse qu'en analysant les liens générés par ces dernières, nous pouvons déduire des connaissances destinées à une méthode d'apprentissage automatique. Ces connaissances sont alors utilisées pour réduire le nombre de liens de traçabilité non valides et par conséquent le temps à investir pour la validation des liens. Notre démarche est résumée par la figure 1.4.

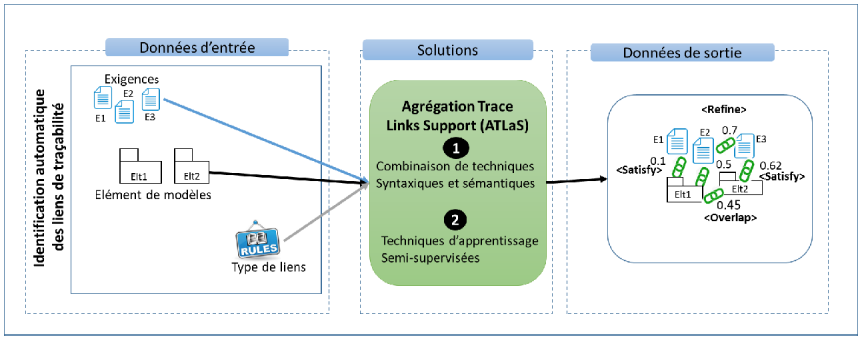

Fig. 1.4 *Résumé des contributions*

Le framework ATLaS (Aggregation Trace Links Support) génère automatiquement des liens de traçabilité typés. Il prend en entrée des paires d'artefacts (exigences ou éléments de modèles) et des règles formelles de définitions des types de liens. Des mesures de similarités syntaxiques et sémantiques entre paires d'artefacts sont utilisées pour construire le vecteur de descripteurs de chaque paire d'artefacts. En sortie, il fournit le type du lien et une mesure de confiance qui est utilisée pour filtrer les « *liens non valides* ». L'exemple de la figure 1.4 montre que l'exigence **E3** et l'élément de modèle **Elt2** sont liés par un lien **<Satisfy>** ayant une mesure de confiance de **0.48**.

Notre démarche s'appuie sur une technique d'apprentissage semi-supervisée. Notons que

ces techniques n'ont pas encore été utilisées dans la communauté de la traçabilité. Elles nécessitent très peu de données validées pour faire de l'apprentissage. Ainsi ces techniques permettent d'aborder le ***problème 3 : Efficacité et confiance*** en tenant compte de la volumétrie des artefacts et de la rareté des données validées. Nous proposons donc, différentes stratégies pour améliorer la robustesse et la performance des outils de traçabilité. Dans ce document, nous faisons également l'évaluation de notre approche sur des cas d'étude industriels. Cette évaluation a permis d'identifier les différences entre les jeux de données de référence utilisés dans la littérature et les jeux de données industriels. Nous tirons de cette évaluation des caractéristiques d'un système de recommandation de liens de traçabilité.

L'attribution des types de liens est notre contribution par rapport au ***problème 2 : Interprétation***. Toutefois, ce dernier touche différents aspects de l'entreprise qui ne sont pas pris en compte dans nos travaux. Pour ce faire, nous faisons les hypothèses suivantes :

— les différentes parties prenantes se sont accordées sur une taxonomie de liens de traçabilité,
— les différentes parties prenantes partagent la même interprétation des types de liens.

Le framework ATLaS est intégré à un espace collaboratif. Ce dernier est une plateforme d'intégration applicative qui permet à différents outils de partager et d'échanger leurs données durant du cycle de développement des systèmes. Au travers de cet espace collaboratif, le ***problème 1 : Hétérogénéité*** est abordé indirectement.

En résumé, la thèse défendue dans ce document est que les méthodes semi-supervisées sont une piste porteuse pour l'élicitation et la maintenance des liens de traçabilité en industrie. Cette proposition repose sur les contributions suivantes :

— définition d'une mesure de confiance pour faciliter la validation des liens,
— combinaison de techniques syntaxiques et sémantiques afin de capturer plus de sémantique,
— usage de techniques semi-supervisées pour améliorer les performances des outils de traçabilité,
— prise en compte de l'hétérogénéité des artefacts,
— identification des différences entre les jeux de données de référence utilisés dans la littérature et les jeux de données industriels,
— caractérisation d'un système de recommandation de liens de traçabilité.

1.6 Organisation du mémoire

La suite de cette thèse est organisée en six chapitres. Le Chapitre 2 introduit les principes généraux de la traçabilité en ingénierie des exigences et modèles. Il présente la définition de la traçabilité retenue dans ce document et un tour d'horizon des types de liens, corroborée par un exemple fil conducteur du véhicule autonome fourni par nos partenaires industriels. Le Chapitre 3 propose une revue de la littérature des approches d'élicitation et de maintenance des liens de traçabilité selon les problèmes précédemment cités. Le Chapitre 4 introduit l'environnement scientifique sur lequel s'appuie cette thèse. Plus précisément, il introduit des notions générales du traitement automatique des langues et des techniques semi-supervisées. Il présente aussi des techniques récentes du traitement du langage. La contribution de la thèse est présentée dans le chapitre 5. Elle consiste en des approches permettant de créer des liens de traçabilité typés avec une mesure de confiance à partir d'un ensemble d'exigences et

d'éléments de modèles. Ce chapitre décrit notre méthodologie et un prototype implémentant les contributions de cette thèse. Le prototype implémenté est l'outil support *ATLaS* basé sur des techniques semi-supervisées qui assiste l'analyste lors de la validation des liens de traçabilité. Le Chapitre 6 détaille le cas d'application, les expérimentations et les évaluations effectués pour la validation de notre approche. Ces expérimentations et évaluations sont illustrées dans ce même chapitre avec différents cas d'études. Enfin, nous concluons et présentons les perspectives de travail dans le Chapitre 7.

Deuxième partie

État de l'art et des pratiques

Avant-propos

Le chapitre précédent introduit le contexte industriel et scientifique dans lequel se situent nos travaux de thèse. S'axant sur l'ingénierie systèmes, il soulève des problématiques de la traçabilité qui se posent aujourd'hui, dans un contexte d'entreprises étendues. Parmi ces derniers rappelons :

— **Problème 1 : Hétérogénéité des artefacts et des outils** : la plupart des outils de traçabilité ne prennent pas en charge les liens vers des artefacts situés à l'extérieur des outils ou ne prennent en charge que les liens vers des outils spécifiques ;

— **Problème 2 : Interprétation sémantique des liens** : La sémantique d'un lien est principalement fonction de l'interprétation de chaque partenaire ou est fusionnée avec l'outillage ;

— **Problème 3 : Efficacité et confiance dans les outils de traçabilité** : les outils de traçabilité ne prennent pas en compte la volumétrie des artefacts et leur fréquence de modifications ;

— **Problème 4 : Charge de travail allouée à la traçabilité** : la création et la maintenance manuelle des liens de traçabilité sont chronophages et sujettes à oubli ou erreur.

Cette partie de la thèse intitulée « État de l'art et des pratiques » examine les approches actuelles selon ces quatre problèmes. Elle est divisée en trois chapitres, les deux premiers portant sur la traçabilité et le dernier sur les approches du traitement automatique des langues, ainsi que les techniques d'apprentissage automatique de cette communauté.

— Le chapitre 2 introduit les définitions de la traçabilité en ingénierie des exigences et modèles et la taxonomie des liens de traçabilité.

— Le chapitre 3 propose un tour d'horizon des différentes approches actuelles de traçabilité et précise notre problématique de thèse.

— Le chapitre 4 présente les techniques du traitement automatique des langues, les techniques de Recherche d'Information(RI) de la communauté de la traçabilité, ainsi que les techniques d'apprentissage automatique de cette communauté.

2 – Traçabilité en Ingénierie Systèmes et Logicielle

2.1 Introduction

Ce chapitre commence par un historique de l'état de l'art et des pratiques de la traçabilité des exigences et de celle des modèles. Il présente un état de l'art sur la traçabilité, et ce faisant, pose les définitions retenues dans le cadre de cette thèse. Il présente également un axe important de mon étude à savoir l'identification automatique des types de liens existant entre les exigences et les modèles (*les liens exigences - exigences, les liens éléments de modèle - éléments de modèle et les liens exigences - éléments de modèle*). Cette présentation est illustrée à partir d'exemples portant sur la conception du système « véhicule autonome ». Ces exemples sont tirés du cas d'usage industriel du projet Éco-mobilité par Véhicules Autonomes (EVA) de l'Institut de Recherche Technologique SystemX (cf. Annexe B).

2.2 Historique de la traçabilité

La plus grande partie de la recherche sur la traçabilité [1] a été effectuée au cours des deux dernières décennies par la communauté de l'ingénierie des exigences. Au cours des dernières années, elle a pris de l'importance, et les questions de traçabilité font maintenant l'objet de recherches dans de nombreux autres domaines, notamment en ingénierie systèmes et logicielle.

Les sous-sections suivantes présentent brièvement l'historique de l'état de l'art et des pratiques de la traçabilité des exigences et des modèles.

2.2.1 Traçabilité des exigences

La traçabilité existe depuis près de 40 ans maintenant, le premier outil de traçabilité des exigences a été présenté en 1978 [28]. Elle permet de créer des flux de connexions entre des artefacts (exigences textuelles et éléments de modèles) tout le long du cycle de vie d'un projet. Elle concerne principalement *les liens exigences - exigences* et *les liens exigences - éléments de modèle*. En pratique, la traçabilité des exigences est généralement gérée dans des outils de gestion d'exigences. L'outil le plus courant est DOORS [29]. La plupart des outils de gestion des exigences fonctionnent plus ou moins de la même manière. Ces outils organisent les exigences et les autres artefacts sous forme de listes arborescentes. Chaque artefact se voit

1. Une définition précise de la traçabilité est fournie en section 2.3.1

attribuer un identifiant unique et peut être annoté à l'aide d'un ensemble d'attributs pouvant stocker des métadonnées (priorité, statut ou risque). De plus, ces outils offrent des capacités de création manuelle, de visualisation de liens ou d'analyse d'impacts. Les outils les plus avancés proposent généralement des interfaces d'intégration avec d'autres outils d'ingénierie. Ils offrent également la possibilité de trier et de filtrer les liens de traçabilité et de générer des rapports.

Certains outils prennent en charge la rédaction des exigences textuelles. Le processus de rédaction des exigences implique la participation de nombreuses parties prenantes ayant des domaines d'expertises variés. Il est alors essentiel de disposer d'un document d'exigences bien rédigé. Bien que les modèles de rédactions d'exigences soient largement utilisées dans les industries aujourd'hui, ces modèles ne permettent pas de pleinement spécifier les systèmes en langage naturel. C'est ainsi que les recherches dans la traçabilité des exigences se sont orientées vers d'autres communautés pour chercher des solutions (métamodèle de traçabilité, ontologies). Une autre problématique dans la traçabilité des exigences et dans la communauté de la traçabilité en générale est le coût de la validation des liens. Cette notion sera abordée plus en détail dans la section 3.4 du chapitre 3. De plus, les quatre problèmes que nous avons mentionnés dans le chapitre 1 ne sont pas couverts par les outils actuels.

2.2.2 Traçabilité des modèles

En pratique, la traçabilité des modèles est beaucoup moins répandue que la traçabilité des exigences. Elle concerne principalement *les liens exigences - éléments de modèle* et *les liens éléments de modèle - éléments de modèle*. Malgré l'existence de nombreux prototypes de recherche [30, 31], très peu d'outils industriels intègrent la capacité de générer des liens de traçabilité à partir de la transformation de modèles. Une application industrielle notable de cette technique est la synchronisation bidirectionnelle des modèles et du code dans les outils UML round-trip engineering (tels que IBM Rational Software Modeler). Winkler et al. ainsi que Galvao et al. [1, 32] ont présenté des enquêtes sur les approches de traçabilité des modèles en les comparant à l'aide de différents critères, notamment le support des outils. Certaines approches fournissent une prise en charge native des outils, mais dans la plupart des cas (langages de transformation ATL [31] ou Kermeta [33]), la traçabilité n'est pas prise en charge de manière native et n'est possible que par extension des outils existants.

Plus récemment, les recherches se sont concentrées sur la définition et l'utilisation de métamodèles de traçabilité pour expliciter les liens conceptuels entre les paires d'artefacts. C'est ainsi que de nombreux métamodèles ou ontologies [34, 35, 36] ont été proposés dans la littérature. Cependant, ces derniers sont souvent construits de manière ad-hoc et sont le plus souvent chronophages à personnaliser à des contextes spécifiques. Un avantage de la traçabilité des modèles par rapport à celle des exigences est la représentation des artefacts en graphes afin de faciliter l'élicitation des liens.

De plus, une hypothèse de travail récurrente de ces techniques est que tous les artefacts sont ou devraient découler des exigences. Ainsi, les autres artefacts sont mis sous forme textuelle avant d'être comparés avec les exigences. La représentation de modèle en graphe orienté qui a été utilisée plusieurs fois dans la littérature comme un formalisme commun pour présenter tous les modèles issus de langages de modélisation hétérogènes [37, 38] pourrait être exploitée afin de tenir compte des métadonnées des artefacts autres que les exigences. Ceci pourrait augmenter les performances des approches enracinées dans la traçabilité des

exigences.

Des solutions pour palier le « *problème 1 : Hétérogénéité* » ont été proposées. Citons notamment, l'initiative Open Services for Lifecycle Collaboration (OSLC) [39] qui a défini un ensemble de spécifications afin de faciliter l'intégration de différents logiciels de conception à l'aide d'un réseau sémantique ou d'une ontologie. Toutefois, le « *problème 2 : Interprétation* » et le « *problème 4 : Charge de travail* » que nous avons mentionnés dans le chapitre 1 n'ont pas encore trouvé de solutions.

2.3 Quelques définitions

2.3.1 Définition de la traçabilité

La traçabilité a son origine dans l'ingénierie des exigences [1]. Elle a ensuite été adoptée par d'autres communautés notamment l'Ingénierie Dirigée par les Modèles (IDM) et l'ingénierie systèmes. Il existe plusieurs définitions de la traçabilité dans la littérature. Par exemple, le glossaire standard ISO/IEC/IEEE 24765 [40] de l'ingénierie des systèmes et du logiciel fournit la définition suivante de la traçabilité :

> « the degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor -successor or master-subordinate relationship to one another ».

Comme autre définition de la traçabilité, citons celle du glossaire standard IEEE de la technologie de l'information [41]. Dans ce dernier, la traçabilité est définie comme :

> « the identification and documentation of derivation paths (upward) and allocation or flowdown paths (downward) of work products in the work product hierarchy ».

Néanmoins, en ingénierie des exigences, la définition de Gotel et Finkelstein [42] a été largement utilisée. Ces auteurs ont défini la traçabilité comme suit :

> « the ability to describe and follow the life of a requirement, in both forward and backward direction, to its subsequent deployment and use, and through periods of ongoing refinement and iteration in any of these phases ».

Dans la communauté IDM, la traçabilité est généralement limitée à la transformation de modèle. Aizenbud-Reshef et al. [43] ont ainsi modifié la définition précédente comme suit :

> « any relationship that exists between artifacts involved in the software engineering life cycle ».

Ces définitions fournissent des bases solides pour comprendre et interpréter la traçabilité. Elles permettent de définir, de décrire, de capturer et de suivre les liens de traçabilité tout le long du cycle de vie du système. Cependant, elles ne font pas clairement de distinction entre deux cas de figures de l'élicitation des exigences : avant et après la création du document de spécification. L'identification des liens de traçabilité commence dès le début du projet. Durant les phases préliminaires du projet, le besoin exprimé par le client est réécrit sous forme d'exigences dans un document de spécification. Ce référentiel d'exigences est ensuite transformé tout le long du projet en différentes déclinaisons. La figure 2.1 résume ce processus.

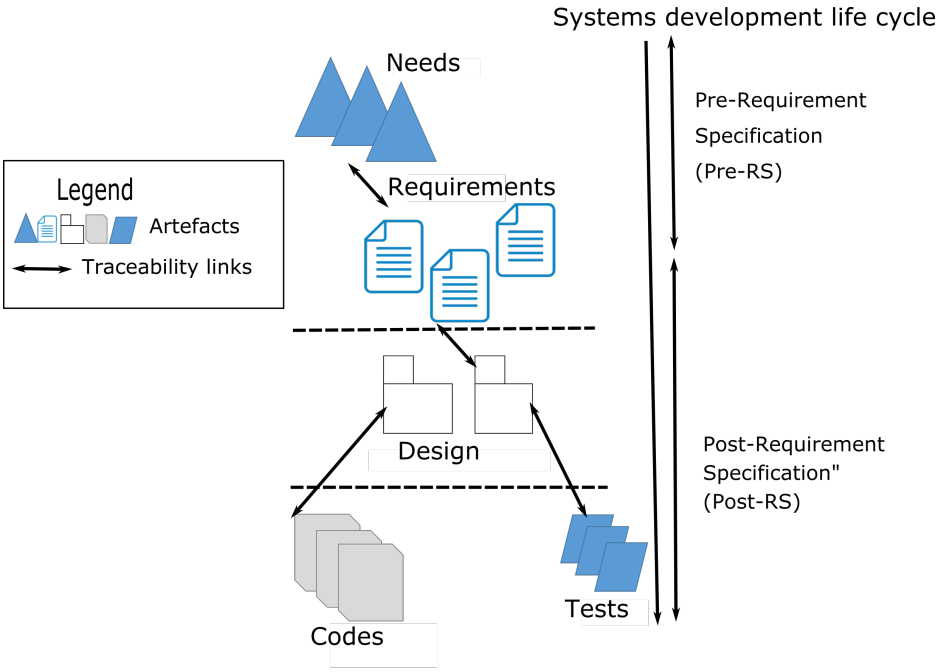

Fig. 2.1 *Traçabilité et phases de spécifications des exigences [1]*

— Le premier cas de figure est la traçabilité « Pré-Requirement Specification - Pré-RS ». Ce premier cas porte sur le cycle de vie des exigences avant leur inclusion dans le document de spécification (traitement d'informations informelles ou conflictuelles).
— Le second cas de figure est la traçabilité « Post-Requirement Specification - Post-RS ». Ce deuxième cas concerne la mise en œuvre progressive des exigences spécifiées à travers leurs différentes déclinaisons (modèles, codes).

Nos travaux se situent dans ce second cas de figure. Contrairement aux autres définitions, celle d'Edwards et Howell [44] porte sur la traçabilité Post-RS en ignorant les processus antérieurs [44]. Pour ces derniers la traçabilité est :

Définition 2.3.1. Edwards et Howell « a technique used to provide a relationship between the requirements, the design and the final implementation of the system ».

Cette définition est donc celle qui est retenue dans ce document. La traçabilité est donc définie comme une technique qui permet de créer des liens entre des exigences, des modèles, des tests et des codes.

La sous-section suivante présente les règles d'écriture d'une exigence qui facilite leur compréhension et donc leur traçabilité.

2.3.2 Modèle de rédaction d'exigences

Une étape importante de la traçabilité Post-RS est la gestion du référentiel d'exigences. Ce dernier est un ensemble vivant d'exigences qui évoluent dans le temps.

Selon le référentiel CMMI (Capability Maturity Model Integration) [45], une exigence du système est une condition ou une capacité que doit posséder un système ou un composant

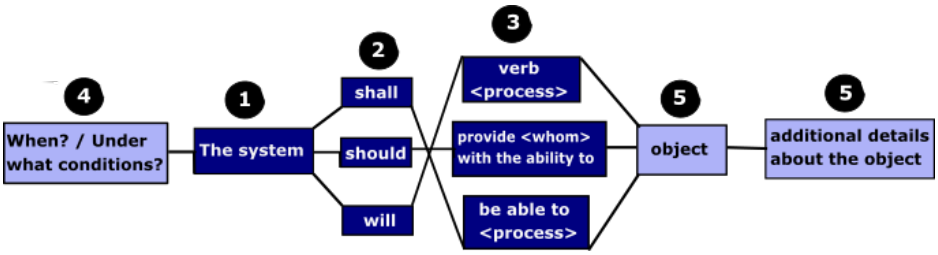

Fig. 2.2 *Modèle de rédaction d'exigences avec conditions [2]*

du système pour remplir un contrat, se conformer à une norme, une spécification ou tout autre document imposé formellement.

Ainsi, les exigences doivent être claires et précises pour être bien comprises par l'ensemble des parties prenantes. Afin de faciliter leur compréhension, Pohl préconise que l'écriture des exigences doit suivre un canevas (Requirements Templates) [2]. Il définit dans ce canevas qu'une exigence est constituée de cinq éléments dont trois obligatoires et deux facultatifs. La figure 2.2 illustre ces cinq éléments, qui sont :

1. Le premier élément obligatoire de l'exigence est le sujet, par exemple « The system ». Notons qu'une exigence doit obligatoirement être constituée d'une phrase principale.
2. Le deuxième élément obligatoire est le verbe modal, par exemple « shall », « will », « should » ...
3. Le troisième élément obligatoire est la sous phrase décrivant la capacité, l'activité ou le comportement du système ou du composant du système.
4. Le quatrième élément, qui est facultatif, est une condition. Une exigence peut aussi inclure des phrases supplémentaires qui décrivent sous quelles conditions les fonctionnalités souhaitées doivent être exécutées. Ces dernières sont alors exprimées avec des conjonctions temporelles (during, after, before, as soon as, when, ...), des conjonctions logiques (if), des opérations logiques (or, and, xor) ou avec certaines expressions telles que «same» ou «such as».
5. Le cinquième élément, qui est également facultatif, est constitué d'un supplément d'information sur le système ou sous-système. Des informations complémentaires peuvent également être ajoutées sous forme libre pour fournir plus de détails. Ces informations fournissent alors des caractéristiques particulières de l'exigence comme sa complétude, sa faisabilité, ses standards de conformité, son niveau d'ambiguïté [46].

La construction automatique de la cartographie des artefacts nécessite la capture de leur syntaxe et de leur sémantique. Bien que les modèles de rédaction d'exigences définissent la structure syntaxique de base des exigences, ils ne permettent pas de capturer les concepts et les nuances de sens dues à la différence du vocabulaire entre les parties prenantes. Un moyen très répandu pour le faire est la construction de glossaires appropriés ou de métamodèle de traçabilité. Les glossaires permettent de référencer certains concepts (termes) des domaines d'étude du système et leur définition validée par les parties prenantes, les méta-modèles y explicitent des informations relationnelles.

2.3.3 Métamodèles de traçabilité

Contrairement aux glossaires, les métamodèles de traçabilité permettent de capturer les concepts métiers et les liens entre ces concepts. Ils constituent ainsi l'expression abstraite de la traçabilité d'un projet [47]. Ils contribuent à établir une compréhension commune de la sémantique des artefacts et des liens de traçabilité [48] entre les différentes parties prenantes.

Plusieurs métamodèles [49, 50, 51], ont été proposés dans la littérature dans différents domaines. Par exemple, Taromirad et al. [50] ont proposé un métamodèle pour la sûreté de fonctionnement. Celui-ci capture non seulement des concepts métiers de la sûreté de fonctionnement comme les dangers préliminaires, ou l'arbre de défaillance mais également des concepts métiers de l'ingénierie logicielle tels que les exigences systèmes et logicielles ou encore le code. Il permet également de spécifier les relations entre ces concepts. Par exemple, il définit la relation « dérive de (derived) » entre les exigences systèmes et logicielles.

Plus généralement, les métamodèles de traçabilité permettent de définir les types d'artefacts et les liens de traçabilité associés à ces types. Il existe de ce point de vue deux types de métamodèles : ceux qui définissent des liens de traçabilité non typés et d'autres qui définissent les liens de traçabilité typés [1]. Le type du lien de traçabilité est une métadonnée qui peut être stockée comme un attribut du lien. La définition de quelques-uns de ces types de liens est présentée dans la section suivante.

2.3.4 Taxonomie des types de liens de traçabilité

L'automatisation de la traçabilité nécessite une connaissance des types de relations pouvant exister entre les artefacts. Différents auteurs, comme Espinoza et al. [52] ou Spanoudakis et al. [16], ont étudié ces relations et ont suggéré des taxonomies des types de liens de traçabilité. Comme mentionné dans l'introduction du chapitre 2, section 2.1, nous nous intéressons aux types de liens existant entre les exigences et les modèles, plus précisément aux types de liens *exigences - exigences, éléments de modèle - éléments de modèle* et *exigences - éléments de modèle*. Les types de liens présentés ci-dessous sont les plus récurrents dans les taxonomies précédemment évoquées.

Il est important de noter que dans la littérature, les types de liens et leur sémantique sont très variables suivant les auteurs [22]. Pour plus de détails sur les taxonomies des types de liens, Lehnert et al. [53] ont fait une revue systématique sur 10 ans (1991 et 2011) et les ont évaluées et classées suivant un ensemble de critères tels que la granularité, le passage à l'échelle, l'outil de support etc.

Les types de liens et leur définition, que nous présenterons dans la suite, sont ceux des auteurs qui ont les travaux les plus aboutis concernant l'identification automatique des liens. Ces types de liens et leur définition sont :

Le lien de satisfaction (satisfaction link). Il permet de relier des exigences avec des éléments de modèles. Il indique que les éléments de modèle permettent de réaliser des propriétés définies dans les exigences [1] ;

Le lien de recouvrement (overlap link). Il permet de relier des exigences entre elles et des modèles entre eux. Il est utilisé pour associer des artefacts qui décrivent le même aspect du système [1] ;

Le lien de raffinement (refine link). Il relie des exigences entre elles et des modèles entre eux. Les éléments de modèles ou les exigences peuvent apporter plus de précisions ou des informations supplémentaires aux éléments de modèles ou aux exigences

qu'ils raffinent [1].

Dans cette section, la définition de la traçabilité et les types de liens retenus ont été présentés. Dans la section suivante, quelques méthodes pour identifier automatiquement les types des liens précédemment cités sont présentées.

2.4 Identification des types de liens

Dans les projets de petite taille, il est possible d'associer manuellement les types de chaque lien. Les analystes se basent sur le document de spécification et les règles métiers établies par les parties prenantes pour caractériser le type de chaque lien. Cependant, de nombreux projets d'ingénierie sont très volumineux et comportent quelques milliers d'exigences et des centaines d'éléments de modèle. Malheureusement, la volumétrie de ces projets rend difficile et fastidieuse la détermination manuelle des types de liens. L'utilisation de techniques automatisées pour réduire les efforts des analystes et le temps nécessaire pour effectuer cette tâche en conjonction avec l'activité humaine devient dans ce cas cruciale.

Ainsi, l'identification automatique des types de liens est l'un des problèmes (*problème 2 : Interprétation*) abordés dans notre thèse. Elle tient une place importante dans la validation des liens de traçabilité. Cette notion sera abordée plus en détail dans le chapitre 3. Dans la littérature, plusieurs auteurs ont proposé des techniques pour automatiser l'identification des types de liens. Nous présentons ici celles que nous avons retenues pour l'identification de chaque type de liens car elles sont en adéquation avec nos travaux.

2.4.1 Le lien de satisfaction : les techniques de Holbrook et al.

Holbrook et al. [21] ont proposé quatre techniques pour identifier automatiquement le lien de satisfaction. Toutes ces techniques font usage d'un thésaurus de domaine contenant une liste de synonymes construite par analyse d'un pourcentage donné des artefacts (dans leurs expérimentations, les auteurs ont utilisé 25%). Un bref aperçu de ces techniques est présenté ci-après.

Méthode naïve. Elle est basée sur une simple idée de suivi du pourcentage de termes communs entre les exigences et les éléments de modèles. Le pourcentage total de paires de termes ou de syntagmes liées constitue la mesure de similarité pondérée entre l'exigence et l'élément de modèle.

Règles linguistiques. Elles ont pour objectif d'identifier les termes communs entre les exigences et les éléments de modèles. Pour les définir, les analystes inspectent manuellement un sous-ensemble d'artefacts et exploitent le modèle de rédaction des exigences et les conventions de nommage des éléments de modèles.

Term Frequency - Inverse Document Frequency (TF-IDF). Elle utilise la méthode algébrique de recherche d'information Vector Space Model (VSM) [2] [54] avec la méthode de pondération TF-IDF [3]. Elle a pour objectif d'identifier les similarités entre les termes ou les syntagmes en se basant sur leur importance dans les documents.

Règles linguistiques et TF-IDF. Elle combine la technique des règles linguistiques avec le TF-IDF afin d'améliorer les performances de cette dernière.

2. Une présentation détaillée de VSM est fournie en section 4.3.2 du chapitre 4
3. Une présentation détaillée de TF-IDF est fournie en section 4.3.1 du chapitre 4

Toutes ces techniques, liées au cas applicatif des auteurs, utilisent la définition formelle du lien de satisfaction proposée par Holbrook et al.[21]. Cette définition est fonction des termes contenus dans les artefacts. Elle se présente comme suit :

> Given a set of requirements with each requirement,
> R, divided into unique $terms(R(t_{r1}, t_{r2}, \dots))$ or $phrases(R(p_{r1}, p_{r2}, \dots))$
> and a set of design elements with each design element,
> D, divided into unique $terms(D(t_{d1}, t_{d2}, \dots))$ or $phrases(D(p_{d1}, p_{d2}, \dots))$,
> a satisfaction mapping is a set of $pairsofterms(t_{rn}, t_{dm})$ where t_{rn} is a term in a set of requirements and t_{dm} is a term in the set of design elements where t_{rn} is directly correlated to t_{dm}

Une version simplifiée de cette définition est :

> Étant donné une exigence R, divisée en termes uniques
> $R = \{term_{R_1}, term_{R_2}, \dots, term_{R_{|R|}}\}$,
> et un élément de modèle Elt, divisé en termes uniques
> $Elt = \{term_{Elt_1}, term_{Elt_2}, \dots, term_{Elt_{|Elt|}}\}$,
> un lien de satisfaction ($\xrightarrow[Satisfaction]{}$) existe entre R et Elt s'il respecte la condition :
> $Elt \xrightarrow[Satisfaction]{} R \; si \; \exists \{term_{R_i}, term_{Elt_j}\} \mid term_{R_i} \in R, \; term_{Elt_j} \in Elt \wedge term_{Elt_j} \xrightarrow[Correlé]{} term_{R_i}$
> Avec $i \in \{1, \dots |R|\}$ et $j \in \{1, \dots, |Elt|\}$, et
> $\xrightarrow[Correlé]{} \triangleq synonyme| \; hyponyme| \; hiérarchie \; conceptuelle$

Un terme peut principalement être un nom ou un verbe. Deux termes sont considérés comme étant directement corrélés s'il y a une synonymie, une hyponymie entre eux ou encore si ces termes appartiennent à une même hiérarchie conceptuelle. Par exemple, dans la hiérarchie conceptuelle « security », les termes « login » et « authentification » sont directement corrélés.

Elle concerne aussi les syntagmes.

> Étant donné une exigence R, divisée en syntagmes nominaux ou verbaux uniques $R = \{chunk_{R_1}, chunk_{R_2}, \dots, chunk_{R_{|R|}}\}$,
> et un élément de modèle Elt, divisé en syntagmes nominaux ou verbaux uniques $Elt = \{chunk_{Elt_1}, chunk_{Elt_2}, \dots, chunk_{Elt_{|Elt|}}\}$,
> un lien de satisfaction ($\xrightarrow[Satisfaction]{}$) existe entre R et Elt s'il respecte la condition :
> $Elt \xrightarrow[Satisfaction]{} R \; si \; \exists \{chunk_{R_i}, chunk_{Elt_j}\} \mid chunk_{R_i} \in R, \; chunk_{Elt_j} \in Elt \wedge chunk_{Elt_j} \xrightarrow[Correlé]{} chunk_{R_i}$
> Avec $i \in \{1, \dots |R|\}$ et $j \in \{1, \dots, |Elt|\}$, et
> $\xrightarrow[Correlé]{} \triangleq synonyme| \; hyponyme| \; hiérarchie \; conceptuelle$

Par exemple, la figure 2.3 présente un lien de satisfaction[4] entre deux exigences et un élément de modèle.

Dans cette figure 2.3, les éléments de modèle *UserId* et *password* permettent de satisfaire partiellement l'exigence *160003* car ils vont permettre à l'utilisateur de s'authentifier. Les termes « *UserId* et *password* » sont directement corrélés au terme « authenticate ». De même

4. le lien de satisfaction indique que les éléments de modèle permettent de réaliser des propriétés définies par les exigences

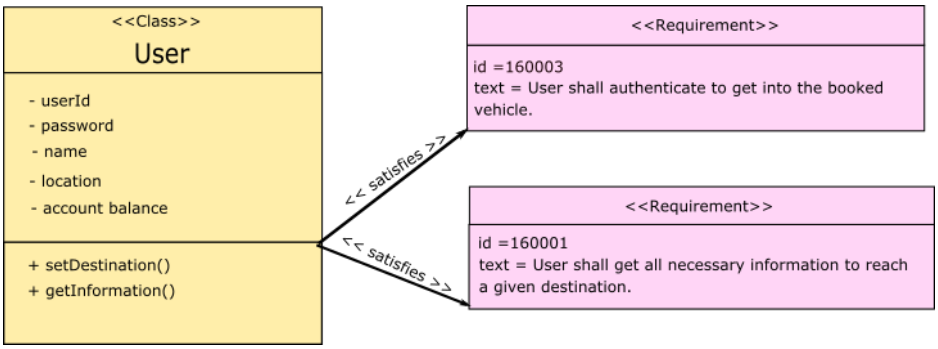

Fig. 2.3 *lien de satisfaction entre deux exigences et un élément de modèle*
Les exigences opérationnelles 160003 et 160001 ont été associées à la classe « User » qui permet de les satisfaire à l'aide des attributs (userdId et password) et des fonctions (setDestination() et getInformation()).

l'élément de modèle *getInformation()* permet de satisfaire partiellement l'exigence *160001* car cette fonction est utilisée pour fournir des informations à l'utilisateur. Les syntagmes verbaux « *getInformation()* » et « *get necessary information* » sont identiques et donc directement corrélées. Il faut noter que ces exigences sont partiellement satisfaites car les éléments de modèles *UserId* et *password*, *getInformation()* ne définissent pas comment les exigences sont satisfaites.

2.4.2 Le lien de recouvrement : la technique de Zisman et al.

Zisman et al. [55, 56, 57] ont proposé une technique pour identifier les liens de recouvrement. Elle est basée sur des règles linguistiques. Les relations syntaxiques requises par ces règles sont définies à l'aide d'une séquence de termes ayant des rôles grammaticaux spécifiques dans les textes. De même que dans les techniques de Holbrook et al.[21], les termes sont des noms et des verbes. Les artefacts et même les règles sont exprimés en XML. Un exemple de règle d'identification de lien de recouvrement peut alors être :

Exists
SEQUENCE(<x1/NN1>, <x2/NN1>, <x3/Dquotes>*, <x4/VM>,<x5/VBI>, <x6/VVN>) in requirements.xml/ UMLmodels.XML ;
<x7/CLASS>, <x8/OPERATION> in UMLmodels.XML
such that
OPERATION_OF(<x8>,<x7>)
and MEMBER_OF(<x6>,OP_SYNONYMS(OP_STEROTYPE(<x8>))) and
CONTAINS(NAME(<x7>), <x1>) or CONTAINS(NAME(<x8>), <x2>))
Generate Relation
OVERLAPS (SEQUENCE (<x1/NN1>, <x2/NN1>, <x3/Dquotes>*, <x4/VM>, <x5/VBI>, <x6/VVN>), <x8>).

Cette règle spécifie qu'un lien de recouvrement est créé s'il existe une séquence de noms (<NN1>), suivie de zéro ou plusieurs occurrences de guillemets (<Dquotes>) et une séquence de syntagmes verbales (<VM>, <VBI>, <VVN>) dans une exigence et une opération d'une classe dans un modèle telle que le troisième verbe de la locution verbale (<VVN>) est un

synonyme du stéréotype de l'opération, le nom de la classe est identique ou est une variante morphologique du premier nom de la séquence (<x1/NN1>) et le nom de l'opération est identique ou est une variante morphologique du deuxième nom de la séquence <x2/NN1>. Comme exemple de variante morphologique, citons « authentification », « authentifying », « authentified » qui sont des variantes morphologiques de « authentify ». Cette technique utilise un dictionnaire de synonymes construits par les différentes parties prenantes.

Les auteurs ont également proposé une définition du lien de recouvrement. Ils le définissent comme suit :

> R1 overlaps with an element R2, if R1 and R2 refer to common features of a system or its domain.

Cette définition n'étant pas précise, elle est présentée dans ce document telle que les auteurs l'ont définie dans leur expérimentation et en utilisant le formalisme proposé par Holbrook et al. [21] pour le lien de satisfaction. Pour les exigences, elle peut donc être transcrite comme suit :

> Étant donné deux exigence R1 et R2, divisées en termes uniques
> $R1 = \{term_{R1_1}, term_{R1_2}, \ldots, term_{R1_{|R1|}}\}$,
> et $R2 = \{term_{R2_1}, term_{R2_2}, \ldots, term_{R2_{|R2|}}\}$,
> un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre R1 et R2 s'il respecte la condition :
> $R2 \xrightarrow[Recouvrement]{} R1 \ si \ \exists \ \{term_{R1_i}, term_{R2_j}\} \mid term_{R1_i} \in R1, \ term_{R2_j} \in R2 \ \wedge \ term_{R2_j} \xrightarrow[identique]{} term_{R1_i} \vee term_{R2_j} \xrightarrow[sémantiquement\ proche]{} term_{R1_i}$
> Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$, et
> $\xrightarrow[sémantiquement\ proche]{} \triangleq synonyme| \ hyponyme| \ antonymie| \ troponymie$

Deux termes sont considérés comme sémantiquement proches s'il y a une synonymie, une hyponymie, une antonymie, ou troponymie entre eux, ou encore si un des termes est une variante morphologique de l'autre.

Cette définition au niveau des syntagmes donne :

> Étant donné deux exigences R1 et R2, divisées en syntagmes nominaux ou verbaux uniques
> $R1 = \{chunk_{R1_1}, chunk_{R1_2}, \ldots, chunk_{R1_{|R1|}}\}$,
> et $R2 = \{chunk_{R2_1}, chunk_{R2_2}, \ldots, chunk_{R2_{|R2|}}\}$,
> un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre R1 et R2 s'il respecte la condition :
> $R2 \xrightarrow[Recouvrement]{} R1 \ si \ \exists \ \{chunk_{R1_i}, chunk_{R2_j}\} \mid chunk_{R1_i} \in R1, \ chunk_{R2_j} \in R2 \ \wedge \ chunk_{R2_j} \xrightarrow[identique]{} chunk_{R1_i} \vee chunk_{R2_j} \xrightarrow[sémantiquement\ proche]{} chunk_{R1_i}$
> Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$, et
> $\xrightarrow[sémantiquement\ proche]{} \triangleq synonyme| \ hyponyme| \ antonymie| \ troponymie$

Par exemple, les figures 2.4 et 2.5 présentent respectivement des exemples de liens de recouvrement *exigences - exigences* et *éléments de modèle - éléments de modèle*.

Dans la figure 2.4, les exigences *Req1* et *Req2* sont en recouvrement car elles ont en commun le syntagme nominale « reduction of costs of repairing ». L'application de cette définition sur les éléments de modèle concerne principalement leur noms. En effet, les éléments de modèle sont généralement nommés suivant les éléments du système qu'ils représentent.

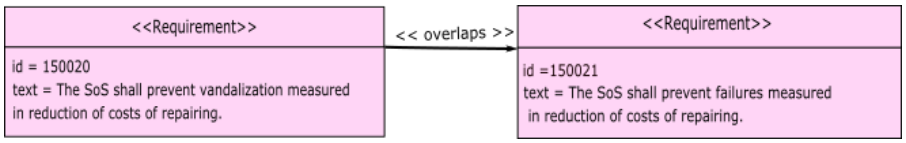

Fig. 2.4 *lien de recouvrement exigences - exigences*
Ces deux exigences concernent la prévention des coûts liée à la dégradation du matériel.

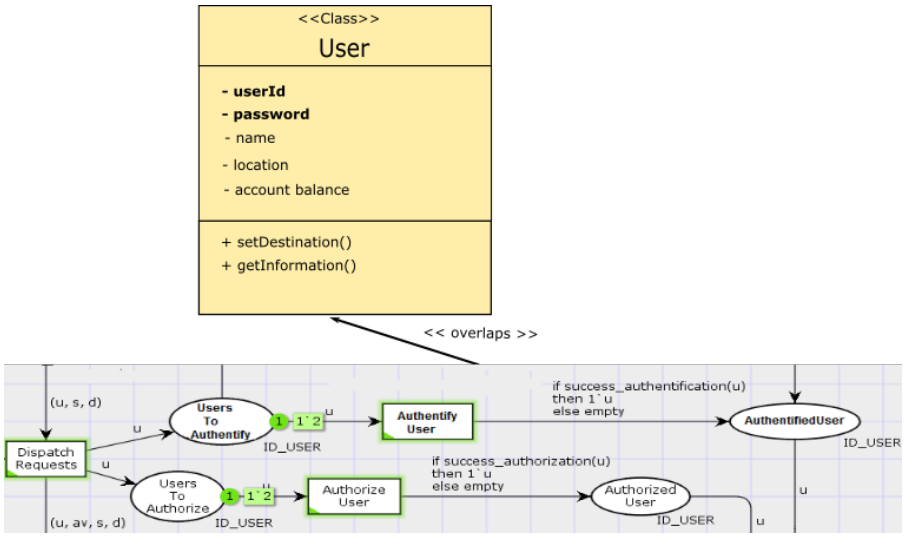

Fig. 2.5 *lien de recouvrement éléments de modèle - éléments de modèle*
Les termes en gras (userId, password, Users To Authentify, Authentify User, AuthentifiedUser) font référence à une fonction commune du système : l'authentification.

Les noms d'éléments de modèle peuvent ne contenir qu'un seul terme : un nom ou un verbe. Par exemple, un élément de modèle peut porter le nom « User » ou « Dispatch ». Les noms peuvent également être composés de plusieurs termes. Dans ce cas, les termes sont séparés soit par les majuscules, soit par un tiret, ou soit par un espace. Ces conventions de nommage sont généralement définies dans les projets et constituent les bonnes pratiques. Ainsi, dans la figure 2.5, il y a un lien de recouvrement entre les éléments de modèle « userId », « password » du diagramme de classe et les éléments de modèle « Users To Authentify », « Authentify User », « AuthentifiedUser » du diagramme d'état-transistion.

Pour les éléments de modèle, la définition formelle du lien de recouvrement s'écrit comme suit :

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en termes uniques
$Elt1 = \{term_{Elt1_1}, term_{Elt1_2}, \ldots, term_{Elt1_{|Elt1|}}\}$,
et $Elt2 = \{term_{Elt2_1}, term_{Elt2_2}, \ldots, term_{Elt2_{|Elt2|}}\}$,
un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :

$Elt2 \xrightarrow[Recouvrement]{} Elt1 \; si \; \exists \{term_{Elt1_i}, term_{Elt2_j}\} \mid term_{Elt1_i} \in Elt1, \; term_{Elt2_j} \in Elt2 \; \wedge \; (term_{Elt2_j} \xrightarrow[identique]{} term_{Elt1_i} \vee term_{Elt2_j} \xrightarrow[sémantiquement\ proche]{} term_{Elt1_i})$

Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$, et

$\overrightarrow{\textit{sémantiquement proche}} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

Cette définition au niveau des syntagmes donne :

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en syntagmes nominaux ou verbaux uniques
$Elt1 = \{chunk_{Elt1_1}, chunk_{Elt1_2}, \ldots, chunk_{Elt1_{|Elt1|}}\}$,
et $Elt2 = \{chunk_{Elt2_1}, chunk_{Elt2_2}, \ldots, chunk_{Elt2_{|Elt2|}}\}$,
un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :
$Elt2 \xrightarrow[Recouvrement]{} Elt1\ si\ \exists\ \{chunk_{Elt1_i}, chunk_{Elt2_j}\}\ |\ chunk_{Elt1_i} \in Elt1,\ chunk_{Elt2_j} \in Elt2 \wedge (chunk_{Elt2_j} \xrightarrow[identique]{} chunk_{Elt1_i} \vee chunk_{Elt2_j} \xrightarrow[\textit{sémantiquement proche}]{} chunk_{Elt1_i})$
Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$, et
$\overrightarrow{\textit{sémantiquement proche}} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

2.4.3 Le lien de raffinement : la technique de Goknil et al.

Goknil et al. [58] ont proposé un métamodèle de traçabilité et un ensemble de règles pour définir les types de liens. Les auteurs ont également proposé une définition formelle du lien de raffinement sur la base de prédicats. Ils représentent les exigences par un tuple, $R \equiv < P, S >$ où P est un prédicat (propriété du système) et S est un sous-ensemble du système sur lequel la propriété P porte. Cette définition se présente comme suit :

Étant donné $R1 \equiv < P1, S1 >$ et $R2 \equiv < P2, S2 >$ deux exigences distinctes.
Supposons $\exists m \in \mathbb{N}, P2 = q_1 \wedge q_2 \wedge \ldots \wedge q_{m-1} \wedge q_m$. En posant $q^1 i \Rightarrow q_i, i \in 1...m$ (c'est à dire que des propriétés P1 de R1 vont impliquer des propriétés P2 de R2) l'exigence R1 raffine l'exigence R2 si et seulement si en remplaçant tous les qi dans $P2$ on a :

— $P1 = p_1 \wedge p_2 \wedge \ldots p_n \wedge q^1 1 \wedge q^1 2 \wedge \ldots \wedge q^1 m$
— $\exists s \in S2, s \notin S1$

Une formule simplifiée de cette définition dans un contexte linguistique pourrait être :

Étant donné deux exigences R1 et R2, divisées en syntagmes nominaux ou verbaux uniques
$R1 = \{chunk_{R1_1}, chunk_{R1_2}, \ldots, chunk_{R1_{|R1|}}\}$,
et $R2 = \{chunk_{R2_1}, chunk_{R2_2}, \ldots, chunk_{R2_{|R2|}}\}$,
un lien de raffinement ($\xrightarrow[Raffinement]{}$) existe entre R1 et R2 s'il respecte la condition :
$R2 \xrightarrow[Raffinement]{} R1\ si\ \exists\ \{chunk_{R1_i}, chunk_{R2_j}\}\ |\ chunk_{R1_i} \in R1,\ chunk_{R2_j} \in R2\ \wedge\ chunk_{R2_j} \xrightarrow[identique]{} chunk_{R1_i}\ \wedge\ chunk_{R2_j in\{syntagmes\ avec\ condition\ temporelle\}}$
Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$

Par exemple, la Figure 2.6 présente un exemple de lien de raffinement *exigences - exigences*. L'exigence *600007* précise sous quelle condition l'exigence *650003* doit être réalisée. La phrase « *The System of System must be able to detect an obstacle* » est partagée par les deux exigences et la phrase « *when the obstacle is at 14m on a dry road (19m on a wet road) of an autonomous vehicle* » contient une condition temporelle introduite par « when » et le supplément d'information.

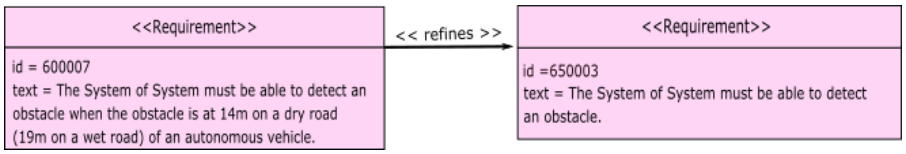

Fig. 2.6 *lien de raffinement exigences - exigences*
L'exigence 600007 précise sous quelles conditions l'exigence 650003 doit être réalisée.

Toutefois, cette définition est trop contraignante à cause de la condition d'identité entre les phrases des exigences ($chunk_{R1_i}$ et $chunk_{R2_j}$). Effectivement, cette dernière ne permet de capturer qu'un très petit groupe de liens de raffinement car dans la pratique très peu de paires d'artefacts reliés par une relation de raffinement sont rédigées de cette manière (respectent cette configuration). Une extension de cette définition pourrait être de tenir compte de la proximité sémantique entre les paires d'artefacts. De ce constat, nous retenons dans nos travaux qu'une condition nécessaire mais non suffisante pour créer de telles relations est que la paire d'artefacts ait, non seulement une relation de recouvrement, mais également qu'un des artefacts contient des informations complémentaires à l'autre. Ces dernières peuvent être introduites à l'aide de conditions temporelles ou logiques. Dans ce document, seul ce type d'information sera considéré pour l'identification des liens de raffinement.

La formule simplifiée de la définition retenue pour les exigences est donc :

Étant donné deux exigences R1 et R2, divisées en syntagmes nominaux ou verbaux uniques
$R1 = \{chunk_{R1_1}, chunk_{R1_2}, \ldots, chunk_{R1_{|R1|}}\}$,
et $R2 = \{chunk_{R2_1}, chunk_{R2_2}, \ldots, chunk_{R2_{|R2|}}\}$,
un lien de raffinement ($\xrightarrow[Raffinement]{}$) existe entre R1 et R2 s'il respecte la condition :
$R2 \xrightarrow[Raffinement]{} R1 \; si \; \exists \; \{chunk_{R1_i}, chunk_{R2_j}\} \mid chunk_{R1_i} \in R1, \; chunk_{R2_j} \in R2 \wedge chunk_{R2_j} \xrightarrow[Recouvrement]{} chunk_{R1_i} \wedge chunk_{R2_j in\{syntagmes \; avec \; condition \; temporelle\}}$
Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$

Cette définition pour les modèles est :

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en syntagmes nominaux ou verbaux uniques
$Elt1 = \{chunk_{Elt1_1}, chunk_{Elt1_2}, \ldots, chunk_{Elt1_{|R|}}\}$,
et $Elt2 = \{chunk_{Elt2_1}, chunk_{Elt2_2}, \ldots, chunk_{Elt2_{|Elt|}}\}$,
un lien de raffinement ($\xrightarrow[Raffinement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :
$Elt2 \xrightarrow[Raffinement]{} Elt1 \; si \; \exists \; \{chunk_{Elt1_i}, chunk_{Elt2_j}\} \mid chunk_{Elt1_i} \in Elt1, \; chunk_{Elt2_j} \in Elt2 \wedge chunk_{Elt2_j} \xrightarrow[Recouvrement]{} chunk_{Elt1_i} \wedge chunk_{Elt2_j in\{syntagmes \; avec \; condition \; temporelle\}}$
Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$

Par exemple, la figure 2.7 et la figure 2.8 présentent des exemples de lien de raffinement *exigences - exigences* et de lien de raffinement *éléments de modèle - éléments de modèle* respectivement.

Dans la figure 2.7, les exigences *600016* et *650001* sont en recouvrement car elles partagent le même concept ou terme « pedestrian ». De plus, l'exigence *650001* complète

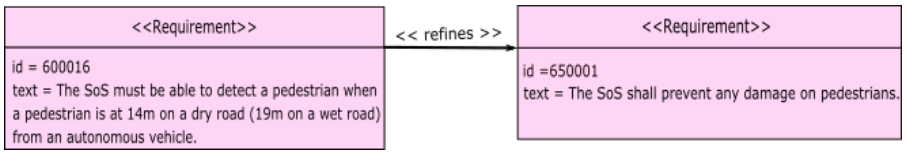

Fig. 2.7 *lien de raffinement exigences - exigences*
L'exigence 600016 précise sous quelles conditions l'exigence 650001 doit être réalisée.

l'exigence *650001* en introduisant la condition temporelle qui précise quand celle-ci doit être respectée. Dans la figure 2.8, le diagramme de flux et le diagramme de capacité opérationnelle sont en recouvrement car ils partagent le terme « Destination » et le diagramme de flux apporte une précision sur l'ensemble des actions qui permettent de réaliser la capacité « Reach destination ».

2.5 Conclusion

Dans ce chapitre, nous avons abordé les concepts de base liés à la traçabilité. Tout d'abord, nous avons mis l'accent sur les définitions de la littérature que nous avons adoptées dans nos travaux. Par la suite nous avons présenté les taxonomies des types de liens et leurs définitions formelles. Puis, nous avons brièvement présenté la traçabilité des exigences et celle des modèles.

Le chapitre suivant présente l'état de l'art sur les deux thématiques que nous abordons plus spécifiquement dans ce travail de thèse, à savoir : l'élicitation et la maintenance des liens de traçabilité.

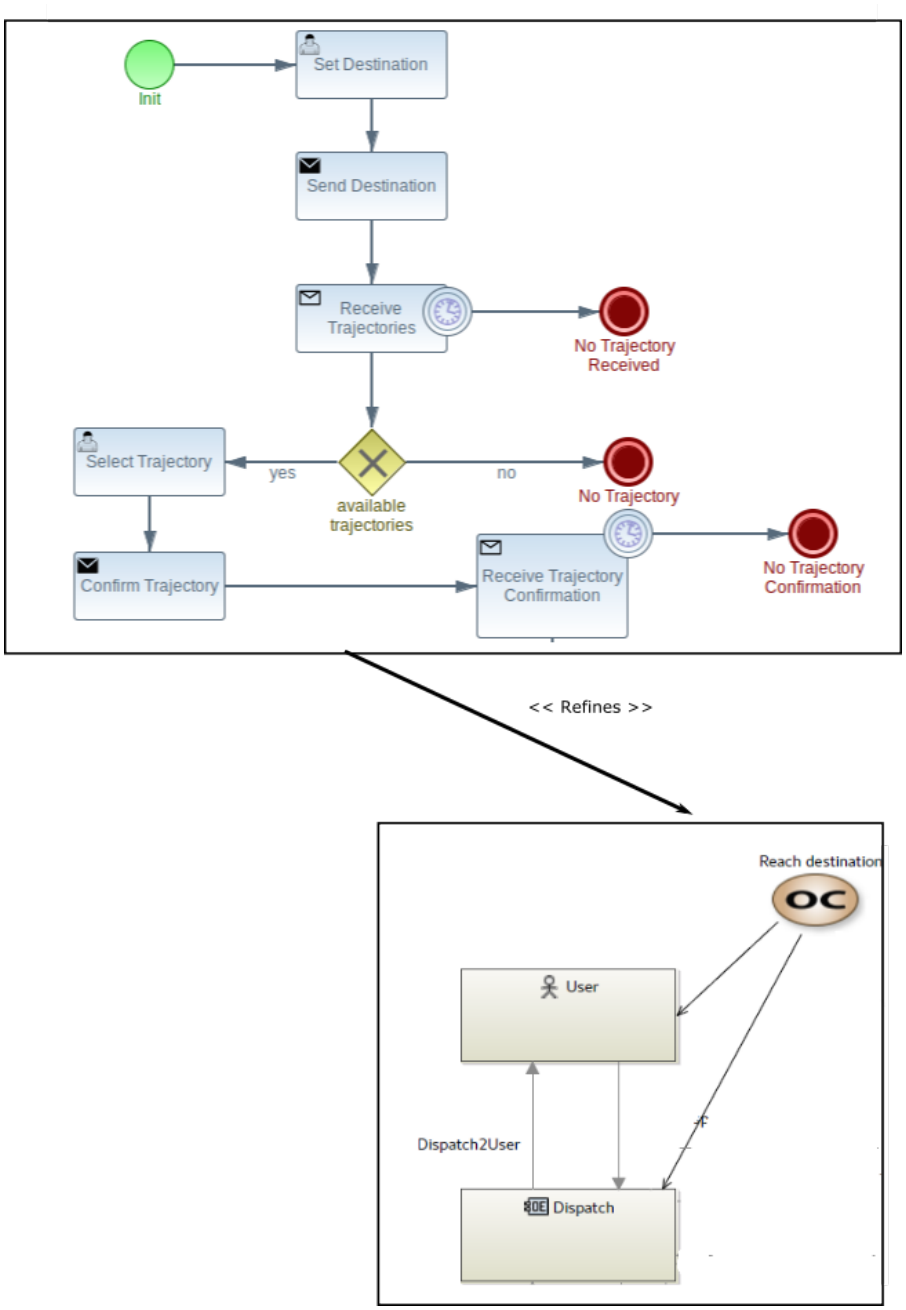

Fig. 2.8 *lien de raffinement éléments de modèle – éléments de modèle*
le diagramme de flux apporte une précision sur l'ensemble des actions qui permettent de réaliser la capacité « Reach destination ».

3 – Élicitation et maintenance des liens de traçabilité

3.1 Introduction

La traçabilité permet d'assurer que le système produit répond aux besoins du client et est en accord avec la stratégie des parties prenantes. Elle englobe les différents aspects qui cadrent le cycle de vie du système : opérationnel, technique, technologique, légal, normatif ou environnemental. Elle permet ainsi de mieux comprendre et appréhender les liens entre exigences, leurs mises en œuvre concrètes (éléments de modèles), leurs vérifications et les différents choix qui en découlent.

Ce chapitre porte en particulier sur les deux piliers de la traçabilité que sont l'élicitation et la maintenance des liens entre artefacts. Il est structuré en cinq sections. Nous présentons dans les deux premières sections un bref aperçu de l'élicitation et de la maintenance des liens de traçabilité. Dans la troisième section, nous abordons la validation des liens, qui est une sous activité commune à ces deux activités. Nous passons en revue dans la quatrième section quelques approches d'élicitation et de maintenance des liens de traçabilité selon les quatre problèmes que nous avons mentionnés dans le chapitre 1. Nous concluons ce chapitre par une synthèse qui soulève des limites des approches existantes.

3.2 Élicitation des liens de traçabilité

Elle désigne le processus englobant la planification, la création, la représentation, le stockage et la validation des liens de traçabilité. Ce processus est illustré dans la figure 3.1.

La planification consiste à déterminer les stratégies de traçabilité (les artefacts à tracer, l'usage des liens). Celles-ci sont établies à chaque début de projet. De ce fait, bien que des artefacts soient préexistants sur un projet, les liens peuvent ne pas encore être créés. L'idée derrière la création des liens de traçabilité est de générer manuellement ou automatiquement des liens entre les artefacts sources et cibles. Elle a pour objectif de fournir une aide aux experts pour formaliser les liens de traçabilité qui n'ont pu pour une raison ou une autre être explicités. Durant la validation des liens, chaque partie prenante vérifie l'intégrité des liens créés. La validation est étudiée en détail dans la section 3.4. La représentation et le stockage des liens dépendent généralement des outils.

Il existe aujourd'hui un nombre important d'outils d'élicitation de liens de traçabilité. Ceux-ci offrent des capacités de création manuelle ou de visualisation des liens. Les outils les plus avancés proposent généralement des interfaces permettant l'intégration avec d'autres

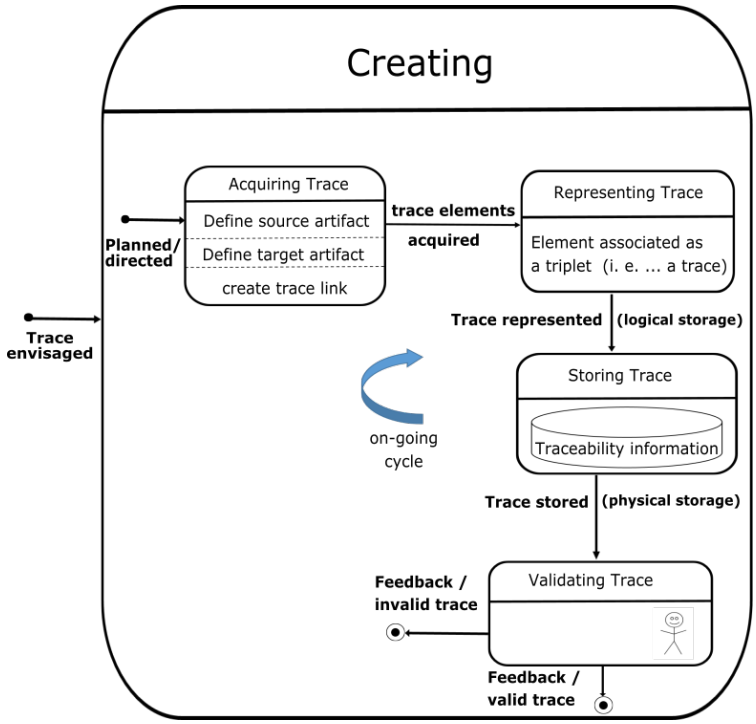

Fig. 3.1 *Processus d'élicitation des liens [3]*
Le processus d'élicitation des liens concerne la création, la représentation, le stockage et la validation des liens.

outils d'ingénierie. Cependant, aucun des quatre problèmes que nous avons mentionnés dans le chapitre 1 n'est couvert par les outils actuels.

Dans la littérature, plusieurs approches permettent d'identifier semi-automatiquement ou automatiquement les liens de traçabilité, chacune avec des degrés différents d'efficacité et de performance. Cependant leur faible performance explique leur faible adoption par les entreprises qui considèrent que ces approches sont très coûteuses au regard des bénéfices engendrés [59]. De plus, à ce coût s'ajoute la mise à jour des liens lorsque les artefacts évoluent. L'activité de traçabilité qui permet de prendre en compte l'évolution des artefacts est la maintenance des liens de traçabilité.

3.3 Maintenance des liens de traçabilité

Selon la stratégie de traçabilité, le processus de maintenance peut être réalisé en continu ou à la demande. La maintenance continue des liens de traçabilité consiste à mettre à jour les liens de traçabilité à chaque modification d'artefacts [60] tandis que la maintenance à la demande, comme son nom l'indique, consiste à mettre à jour les liens de traçabilité à la demande des parties prenantes.

La maintenance des liens intervient à la suite de l'élicitation des liens comme illustré par la figure 3.2.

Elle permet de préserver l'investissement réalisé lors de l'élicitation des liens de traçabilité en mettant à jour les liens lorsque les artefacts liés évoluent. Ces mises à jour peuvent

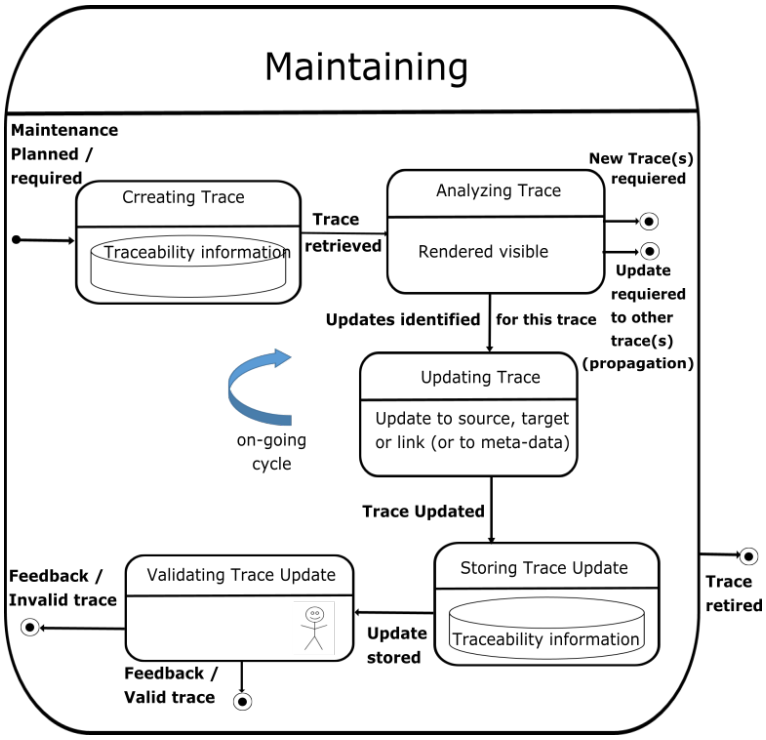

Fig. 3.2 *Processus de maintenance des liens [3]*
Le processus de maintenance des liens concerne l'analyse, la mise à jour/la création de nouveaux liens, le stockage et la validation des liens.

nécessiter la propagation des modifications et/ou la création de nouveaux liens. Les liens maintenus doivent ensuite être validés puis enregistrés.

La maintenance constitue l'un des plus grands défis de la traçabilité [61]. Cette tâche est la plus complexe à gérer car il est difficile de juger les conséquences possibles sur les artefacts existants lorsqu'un artefact est ajouté ou subit un changement.

La plupart des outils commerciaux de gestion des exigences traitent ce problème en marquant simplement les liens nécessitant potentiellement une mise à jour comme suspects ou en notifiant l'utilisateur des modifications faites sur les artefacts et les liens potentiellement impactés.

Dans la littérature, contrairement à l'élicitation des liens, peu de travaux ont été consacrés à la maintenance des liens [62]. Cela est dû en grande partie à l'absence de jeux de données reflétant l'évolution des liens de traçabilité entre plusieurs types d'artefacts [61]. Dans le cas de figure où plusieurs artefacts évoluent indépendamment les uns des autres et sont issus d'espaces technologiques hétérogènes, le coût associé à la maintenance des liens de traçabilité peut très vite devenir rédhibitoire, surtout lorsque les modifications des artefacts sont fréquentes. A ce coût s'ajoute celui de la validation des liens après la mise à jour des liens. Cette notion est présentée en détail dans la section suivante. Pour conclure, le « *problème 1 : Hétérogénéité* » et le « *problème 4 : Charge de travail* » ne sont pas couverts par les outils actuels.

3.4 Validation des liens de traçabilité

La validation des liens est une sous-activité commune à l'élicitation et à la maintenance des liens. Elle occupe une place essentielle [62] dans ces deux activités car elle permet d'assurer la viabilité et la crédibilité des liens créés ou maintenus. Durant la validation des liens, quelle que soit la façon dont les liens de traçabilité sont initialement créés ou maintenus, chaque lien est examiné et son type est défini. Ce dernier est transcrit par les propriétés des liens de traçabilité, lesquelles ont été étudiées de façon détaillée dans la sous-section 2.3.4 Taxonomie des liens de traçabilité du chapitre 2.

La validation des liens est la principale tâche qui assure l'intégrité des liens de traçabilité créés ou maintenus. Elle demande une intervention humaine, qui ne peut pas être automatisée [63]. Elle est ainsi la tâche la plus coûteuse de l'élicitation et de la maintenance des liens, du fait de la quantité de travail manuel demandé et de l'expertise nécessaire pour réaliser cette tâche.

Des solutions ont donc été proposées pour faciliter et aider l'humain dans la validation de liens de traçabilité. Dans la littérature, il existe trois principales techniques [61] pour réaliser cette sous-activité :

La rétroaction des analystes. Elle consiste à recueillir les réactions des analystes après vérification des liens en augmentant ou diminuant la pondération des termes utilisées pour calculer les scores de similarité selon qu'un terme apparaisse dans un lien rejeté ou accepté [64, 65]. Des études ont été menées en vue d'évaluer cette technique. Par exemple, des *eyetrackers* ont été utilisés pour explorer comment les analystes vérifient les liens de traçabilité. Ces études ont montré que, même si cette technique réduit l'effort de validation, elle influe négativement la qualité des liens de traçabilité obtenus [66].

L'analyse sémantique. Elle consiste à exploiter la sémantique et le contexte de chaque lien de traçabilité. La sémantique d'un lien peut être analysée pour comprendre sa logique. De même, la relation entre des paires d'artefact de même type ou de types différents peut être analysée afin de tirer des conclusions sur liens existants ou manquants. Par exemple, Ghabi et al. [67] ont constaté que des éléments de code (méthodes, classes, fichiers) qui font appel à d'autres éléments de code partagent généralement les mêmes exigences.

Les mesures de confiance. Elle consiste à définir des métriques qui servent d'indicateurs de qualité des liens créés. Compte tenu du nombre potentiellement élevé de liens de traçabilité dans un projet, et du coût de l'effort de validation, cette technique permet d'attribuer des scores de confiance à chaque lien. Elle permet ainsi de communiquer un degré de confiance des liens aux analystes.

Chacune de ces techniques offre un soutien potentiel pour améliorer et comprendre l'intégrité des liens de traçabilité. Toutefois, un défi aujourd'hui est de développer des outils qui intègrent plus de connaissances du système d'étude afin de soutenir non seulement la validation des liens mais également l'élicitation et la maintenance des liens.

Le moment où la validation des liens est faite durant le processus d'élicitation est le principal facteur de différenciation des approches d'élicitation et de maintenance des liens. Ces approches peuvent ainsi être divisées en trois catégories :

Catégorie 1 : Validation avant la création des liens. Dans cette catégorie, un ensemble de règles définies prescrivent la création de liens entre les artefacts. Elle re-

groupe :

Les approches orientées évènements permettent la création de liens entre artefacts suivant le pattern *Publish–Subscribe* [68]. Ces approches consistent à analyser les actions des utilisateurs afin d'en déduire les opérations de création, de maintenance ou de suppression à effectuer sur les artefacts associés. Quelques auteurs ont utilisé ces approches pour éliciter et maintenir les liens de traçabilité. Notamment Cleland-Huang et al. [69, 70] ont proposé une approche orientée événements pour éliciter les liens et une autre pour les maintenir. Leur approche pour la maintenance des liens de traçabilité est appelée « Event-based traceability (EBT)».

Les approches à base de règles permettent de gérer le cycle de vie des liens de traçabilité à l'aide de règles de similarité (sur la base d'un vocabulaire métier). Des auteurs ont utilisé ces approches pour éliciter et maintenir les liens de traçabilité. Par exemple, Zisman et al. [55, 57, 56] ont utilisé l'étiquetage morpho-syntaxique [1]combiné avec des règles pour dériver des liens de traçabilité. De même, Goknil et al. [71] ont proposé une approche à base de règles pour maintenir les liens de traçabilité et pour analyser l'impact des changements entre exigences.

Les approches basées sur des ontologies utilisent des bases de connaissance afin de connecter sémantiquement les artefacts. De nombreuses ontologies ont été proposées dans la littérature. Par exemple, Martins et al. [72] ont utilisé les ontologies pour créer des liens entre des exigences. De même, Guo et al. [73, 36] ont mis au point un système expert de traçabilité (DoCIT) utilisant à la fois une ontologie et des méthodes heuristiques pour raisonner sur les concepts du domaine afin de fournir des liens de traçabilité.

Les approches basées sur des métamodèles de traçabilité s'appuient sur des méta-modèles afin de créer et maintenir des liens de traçabilité. Ces méta-modèles capturent l'ensemble des relations entre artefacts et définissent la sémantique des types de liens de traçabilité [74]. De nombreux métamodèles de traçabilité ont été proposés dans la littérature. Par exemple, Taromirad et al. [50] ont proposé une approche pour construire un cadre de traçabilité multi-domaines. De même, Ramesh et al. [75] ont proposé quatre modèles de référence avec une taxonomie de liens de traçabilité permettant de capturer les besoins de traçabilité de différentes parties prenantes.

Catégorie 2 : Validation pendant la création des liens. Dans cette catégorie, les liens de traçabilité sont créés et validés en même temps par les parties prenantes. Elle concerne les **approches basées sur les valeurs**. Ces approches permettent la création de liens par l'attribution d'un niveau de priorité sur les artefacts devant être reliés. Plusieurs auteurs ont utilisé ces approches pour éliciter et maintenir les liens de traçabilité. Par exemple, Zemont et al. [76] ont proposé une approche qui donne un cadre pour évaluer les priorités des exigences et des liens de traçabilité dans une entreprise. De même, Egyed et al. [77] ont introduit une approche basée sur les valeurs (value-based approaches) pour équilibrer les coûts et les bénéfices liés à l'élicitation et au maintien des liens de traçabilité.

Catégorie 3 : Validation après la création des liens. Dans cette catégorie, les liens de traçabilité sont générés automatiquement et cette liste de liens est revue par l'ana-

1. Une présentation détaillée de l'étiquetage morpho-syntaxique est fournie en section 4.2.1 du chapitre 4

lyste. Ce dernier peut alors accepter ou rejeter le lien. Cette catégorie regroupe :
— Les approches de Recherche d'Information (RI),
— Les approches basées sur l'apprentissage automatique.
Ces approches ayant montré des résultats encourageants pour la traçabilité d'artefacts textuels comme illustré par de nombreuses publications récentes [78, 79, 80], nous les étudions en détail dans la section suivante.

Notons que ces approches utilisent des techniques de traitement automatique des langues que nous présenterons dans le chapitre 4.

3.5 Approches d'élicitation et de maintenance des liens de traçabilité

Afin de faire face aux différents défis liés à l'élicitation et à la maintenance de nombreuses approches ont été proposées dans la littérature. Les approches d'apprentissage automatique et de Recherche d'Information sont brièvement décrites dans les sous-sections suivantes.

3.5.1 Les approches de Recherche d'Information

Elles s'appuient sur des mesures quantitatives de similarité entre artefacts. Une paire d'artefacts va être considérée liée lorsque la mesure de similarité est supérieure à un certain seuil. Ces approches basées sur l'analyse du langage naturel ont été largement utilisées pour l'élicitation des liens de traçabilité. Elles ont pour principal objectif d'identifier des documents partageant des similarités syntaxiques et sémantiques. Elles représentent les documents sous forme de vecteurs de dimensions prédéfinies afin de faciliter leur mise en correspondance. Il existe plusieurs techniques de recherche d'information, comme Vector Space Model (VSM) [2] [54] ou Latent Semantic Indexing (LSI) [3] [81].

Ces dernières ont été utilisées pour identifier et maintenir les liens de traçabilité. Notamment, Kleffmann et al. [82] proposent une approche combinant la technique LSI avec la distance de Levenshtein [83] pour l'identification dynamique de liens de traçabilité. Ces liens sont créés entre des diagrammes informels (dessinés à la main) sur des écrans muraux et des exigences textuelles. La distance de Levenshtein permet de mesurer la différence entre deux chaînes de caractères en calculant le nombre minimum d'opérations (nombre d'insertions, suppressions ou substitutions de caractères) requis pour passer d'une chaîne à une autre [84]. Les auteurs observent les activités des utilisateurs et définissent des liens de traçabilité sur la base de leurs observations. Ainsi, un lien est créé entre une exigence et un modèle lorsque l'on observe que ces derniers sont ouverts de manière consécutive à plusieurs reprises. Les liens sont également identifiés à l'aide d'une analyse linguistique (avec LSI). La distance de Levenshtein est utilisée pour calculer les similitudes entre les termes contenus dans les artefacts. Lorsque la mesure de similarité de Levenshtein obtenue est supérieure au premier seuil défini, la technique LSI est appliquée. Le lien de traçabilité est créé si la mesure de similarité de LSI est supérieure au deuxième seuil défini.

Récemment, les approches de RI [85], [86] ont été utilisées pour extraire les patterns de changements entre les artefacts. Quelques-unes de ces approches sont présentées dans

2. Une présentation détaillée de VSM est fournie en section 4.3.2 du chapitre 4
3. Une présentation détaillée de LSI est fournie en section 4.3.3 du chapitre 4

la suite. Elles proposent des mécanismes pour la détection et la propagation d'impact de changements entre artefacts. Après modification, elles permettent d'identifier les expressions nominales ou verbales impactées dans les artefacts liés sur la base du seuil de similarité et suggèrent les modifications à apporter.

Par exemple, Chetan et al. [85] ont proposé une approche pour réaliser une propagation d'impact de changements entre exigences. Plus précisément, pour deux exigences liées, lorsque l'une des exigences est modifiée, cette approche indique quels syntagmes nominaux ou verbaux sont impactés dans l'autre exigence et suggère les modifications à apporter. Elle repose sur l'étiquetage morpho-syntaxique (part-of-speech tagging - POS) et les mesures de similarité à la fois lexicales et sémantiques. L'étiquetage morpho-syntaxique est utilisé pour identifier les termes supprimés ou ajoutés dans une exigence après un changement en tenant compte de la position et des attributs grammaticaux des termes. Les mesures de similarité (Path-based measure) sont utilisées pour calculer le degré de similarité entre les syntagmes modifiées et les syntagmes des autres exigences. Lorsque ces mesures sont supérieures au seuil fixé, les exigences concernées sont considérées comme impactées.

Pour plus de détails sur les approches basées sur la RI, Borg et al. [87] ont fait une revue systématique de quelques-unes d'entre elles sur 10 ans (2003-2013) et les ont évaluées suivant un ensemble de critères.

Le principal inconvénient de ces approches est qu'elles capturent de manière insuffisante la sémantique contenue dans les artefacts (polysémie, non prise en compte du contexte, etc.) [88], [14] (cf. *problème 2 : Interprétation*). Elles ont l'avantage de reporter la validation des liens après leur création permettant ainsi de réduire le temps d'élicitation. Cependant, leur imprécision, due à cette capture approximative de la sémantique, rend la validation des liens d'autant plus chronophage (cf. *problème 3 : Efficacité et confiance* et *problème 4 : Charge de travail*).

3.5.2 Les approches basées sur l'apprentissage automatique

Les techniques d'apprentissage automatique ont été utilisées avec succès pour améliorer les performances des techniques de RI. Elles construisent un classificateur de liens de traçabilité à partir d'exemples [89]. Ce dernier prend en entrée des descripteurs de paires d'artefacts et indique en sortie s'il existe ou non un lien. Dans la littérature, ces techniques peuvent être classées en trois groupes : les techniques d'apprentissage non supervisées, supervisées et semi-supervisées.

Les techniques d'apprentissage non supervisées permettent de trouver des informations sémantiques entre des termes à partir de textes non annotées ou sans utiliser des jeux d'entrainement. Les techniques d'apprentissage non supervisées les plus utilisées dans la communauté de la traçabilité sont la technique Latent Dirichlet Allocation (LDA) [90, 91, 92] et les algorithmes de clustering [93, 94]. LDA [4] est un modèle génératif probabiliste qui permet de regrouper des documents par thèmes [95] et les algorithmes de clustering permettent de diviser un ensemble d'objets en groupes d'objets similaires [96].

Quelques auteurs, comme Panichella et al. [92], ont proposé une approche basée sur LDA pour identifier des liens de traçabilité entre les exigences et les codes sources.

4. Une présentation détaillée de LDA est fournie en section 4.4.2 du chapitre 4

LDA étant une technique paramétrique, elle nécessite lors de son implémentation la définition d'un certain nombre de paramètres (le nombre de thème, le nombre d'itérations, etc). Son efficacité repose donc fortement sur ces paramètres. Les auteurs ont par conséquent adapté, configuré et combiné les Algorithmes Génétiques (AG) avec LDA pour déterminer ses configurations optimales. L'objectif de leur approche est d'identifier efficacement les configurations de LDA qui produisent de meilleures performances. L'approche utilise une version simplifiée des algorithmes génétiques basée sur la sélection d'individus ayant les meilleures configurations de LDA. Ces dernières sont alors conservées pour la création de nouvelles configurations. Leurs résultats ont montré que leur approche est capable d'identifier des configurations robustes de LDA.

Les techniques d'apprentissage supervisées font de l'apprentissage d'informations sémantiques à partir de textes annotés, contrairement aux techniques d'apprentissage non supervisées. Elles utilisent des jeux d'entrainement où les termes sont annotés suivant leur contexte d'utilisation.

De nombreux auteurs ont utilisé ces techniques pour éliciter les liens de traçabilité. Par exemple, Guo et al. [78] ont proposé une approche qui combine les réseaux de neurones récurrents (Recurrent Neural Network - RNN) [97]) avec le *Word embedding* [98] pour identifier des liens de traçabilité entre les exigences et les modèles. Les réseaux de neurones récurrents permettent d'identifier le sens d'un mot dans une phrase donnée en se basant sur les sens des mots qui le précèdent. Ils sont alors considérés comme ayant une mémoire. Ce traitement d'information est possible grâce à des architectures de réseau de neurones particulières telles que « Long Short Term Memory - LSTM » [99] et « Gated Recurrent Unit - GRU » [100]. Le *Word embedding* est utilisé pour transformer les textes en entrées numériques en fonction de leur sens. Ces données permettent ensuite d'alimenter les réseaux de neurones. Les auteurs ont évalué différentes architectures de réseaux de neurones (LSTM, GRU). Leurs résultats ont montré que l'architecture de réseau de neurones bidirectional Gated Recurrent Unit - BI-GRU [100]) a surpassé de manière significative non seulement les autres architectures évaluées mais également les techniques classiques de RI à savoir VSM et LSI.

Les techniques d'apprentissage semi-supervisées utilisent à la fois les données annotées et non annotées. Elles sont donc des techniques intermédiaires entre les techniques supervisées qui n'utilisent que des données annotées et les techniques non supervisées qui n'utilisent que des données non annotées. Le principal intérêt de ces techniques est qu'elles nécessitent très peu de données annotées [101]. De ce fait, elles sont généralement utilisées lorsque les jeux de données deviennent volumineux rendant l'annotation manuelle des données fastidieuses [102]. De nombreux chercheurs ont découvert que les données non étiquetées, lorsqu'elles sont utilisées conjointement avec une petite quantité de données étiquetées, peuvent améliorer considérablement la précision d'apprentissage [101]. Nous n'avons pas trouvé de travaux qui appliquaient ces techniques à la traçabilité lors de notre revue bibliographique. Selon nous, il s'agit là d'une lacune importante.

Sachant que l'annotation ou l'étiquetage des données nécessite souvent un travail manuel très coûteux, alors que les données non étiquetées sont beaucoup plus faciles à

obtenir, l'apprentissage semi-supervisé s'est révélé très utile dans de nombreux problèmes du monde réel [102]. Une application typique en traçabilité pourrait être la validation de liens, dans laquelle les liens validés manuellement, du fait de leur coût, représentent toujours une très petite partie des liens générés par les techniques de recherche d'information, et le nombre de liens non validés reste important.

En conclusion, notons que ces récentes expérimentations montrent que ces approches obtiennent de meilleures performances [78] que les autres approches. Leur inconvénient majeur est qu'elles nécessitent un volume important d'exemples. Toutefois, il existe des techniques d'apprentissage automatique qui requièrent peu ou pas d'exemples mais elles sont moins performantes et précises [89] (cf. *problème 3 : Efficacité et confiance* et *problème 4 : Charge de travail*).

3.6 Synthèse

Malgré le travail de recherche considérable consacré à la création et la maintenance de liens, ces tâches demeurent aujourd'hui essentiellement humaines, fastidieuses et sujettes aux erreurs. Les différentes approches de gestion du cycle de vie des liens de traçabilité que nous avons parcourues présentent des faiblesses récurrentes, en particulier sur les questions de pertinence et de performance. Le tableau 3.1 résume les forces et faiblesses des différentes approches présentées.

Les approches de recherche d'information ont montré des résultats encourageants dans la réduction des coûts de création / maintenance des liens. Cependant, elles capturent approximativement la sémantique et de ce fait la validation des liens de traçabilité générés / mis à jour est chronophage.

Les approches d'apprentissage automatique corrigent ces défauts en limitant les erreurs humaines (qualité augmentée) en allégeant le processus de vérification manuelle [78] ; mais elles restent encore coûteuses.

Approches	Techniques	Forces	Faiblesses
Recherche d'Information	VSM	— Création rapide des liens	— **Capture approximative de la sémantique** (*problème 2 : Interprétation*) — **Passage à l'échelle** (*problème 3 : Efficacité et confiance*) — **Validation chronophage des liens créés** (*problème 4 : Charge de travail*)
	LSI		
	LSI + distance de Levenshtein (Kleffman et al.)	— Détection des changements — Création rapide des liens — bonnes performances	— **Validation chronophage des liens créés** (*problème 4 : Charge de travail*)
	Étiquetage morpho - syntaxique + Path-based measures (Chetan et al.)	— Détection des changements — Propagation d'impact ciblée	— **Validation chronophage des liens créés** (*problème 4 : Charge de travail*)
Apprentissage non supervisé	LDA	— Création rapide des liens	— Méthode paramétrique — **Capture approximative de la sémantique** (*problème 2 : Interprétation*) — **Passage à l'échelle** (*problème 3 : Efficacité et confiance*) — **Validation chronophage des liens créés** (*problème 4 : Charge de travail*)
	LDA + Algorithmes Génétiques (Panichella et al.)	— Création rapide des liens — Bonnes performances	— **Validation chronophage des liens créés** (*problème 4 : Charge de travail*) — temps de calcul
Apprentissage supervisé	RNN + Word embedding (Guo et al.)	— Meilleures performances	— **Utilisation d'un volume significatif d'exemples de liens de traçabilité** (*problème 3 : Efficacité et confiance* et *problème 4 : Charge de travail*)
Apprentissage semi-supervisé			

Table 3.1 *Évaluation des approches d'élicitation et de maintenance des liens de traçabilité*
Les textes en gras présentent les problèmes abordés dans notre thèse.

3.7 Problématiques de la thèse

En définitive, le défi en traçabilité est l'automatisation complète des activités d'élicitation et de maintenance des liens de traçabilité [61]. Ce défi a pour principal objectif la réduction des coûts liés à la capture, l'exploitation et la maintenance des liens de traçabilité. L'analyse des approches présentées dans la section précédente montre que ce défi nécessite encore beaucoup de travaux de recherche [59, 61].

De plus, l'analyse de l'état de l'art a permis de ressortir les différentes lacunes des méthodes actuelles par rapport aux problématiques de la traçabilité en entreprise étendue. L'objectif global de cette thèse étant de soutenir la traçabilité dans un contexte MBSE inter-disciplinaires, elle se doit d'aborder les quatre problèmes mentionnés dans le chapitre 1.

En outre, l'apprentissage semi-supervisé s'est révélé très utile dans de nombreux problèmes du monde réel [102]. Il pourrait donc être appliqué dans notre contexte industriel. Notre proposition consiste alors à étudier l'intégration des techniques de RI et d'apprentissage semi-supervisées enrichies avec celles du traitement automatique des langues, afin d'en mesurer les performances dans l'élicitation des liens de traçabilité. Ainsi, à travers nos travaux, nous répondons aux deux questions de recherche suivantes :

— Question 1 : Quelles méthodes de traçabilité permettent d'obtenir de bonnes performances sur des jeux de données industriels ?
— Question 2 : Quel est l'apport des méthodes de plongements lexicaux basées sur des modèles neuronaux à la capture de la sémantique des liens de traçabilité ?

Ces questions peuvent être regroupées en deux grandes problématiques. La Question 1 concerne le *problème 3 : Efficacité et confiance* et le *problème 4 : Charge de travail* et plus particulièrement la problématique « ***Réduction du coût de l'élicitation des liens de traçabilité*** ». La Question 2 concerne le *problème 2 : Interprétation* et est représentée par la problématique « ***Identification des types de liens*** ». Le « *problème 1 : Hétérogénéité* » est abordé indirectement dans nos travaux au travers de l'espace collaboratif fourni avec le cas d'étude des partenaires de l'IRT SystemX.

Les sous-sections suivantes détaillent ces problématiques auxquelles nous souhaitons apporter une réponse à travers notre thèse.

a) Réduction du coût de l'élicitation des liens de traçabilité

L'utilisation du langage naturel pour la description des exigences a motivé l'usage des méthodes de traitement automatique des langues pour supporter l'élicitation des liens de traçabilité [103]. Ces méthodes sont généralement associées aux techniques de RI. Ainsi, la validation des liens générés par ces méthodes se fait généralement après la création des liens. Ces derniers sont séparés en deux ensembles : les liens approuvés forment le groupe des « liens (links) » référencés dans ce document par « *vrais liens* » et les liens rejetés forment le groupe de « non-liens (no links) » référencés dans ce document par « *faux liens* ».

Du fait de la volumétrie des artefacts et de leurs changements fréquents, ce processus de validation devient chronophage [104]. Aussi, la minimisation des «*faux liens*» lors de la génération permettrait de réduire les efforts liés à ce travail de validation. Dans ce cadre, les approches d'apprentissage supervisées ont récemment permis de réduire considérablement le nombre de «*faux liens*» générés [78]. Cependant, ces techniques nécessitent de larges volumes

d'exemple de liens de traçabilité pour être réellement efficaces.

Dans la pratique, la constitution d'une base d'exemples demande beaucoup de temps. Pour contourner cette limitation du nombre d'exemples nécessaires à la création de liens, de nombreuses stratégies ont été appliquées. Notamment les techniques qui consistent à étudier et à imiter étroitement les décisions des analystes lors de la validation des liens [105, 73]. Toutefois, très peu de travaux ont été faits dans ce sens. Aujourd'hui, les stratégies qui semblent les plus pertinentes sont celles qui combinent différentes techniques [69, 106] ou celles qui sont capable de modifier leur propre comportement dans le but d'optimiser leurs performances [92, 107]. Ces stratégies visent à tirer profit des avantages des techniques existantes tout en compensant leurs faiblesses. Bien que ces stratégies soient prometteuses, l'élicitation de liens qui maximise la création de «*vrais liens*» tout en minimisant la création de «*faux liens*» reste un objectif très difficile à atteindre. En effet, les expérimentations montrent qu'il est aujourd'hui plus probable d'éliciter un «*faux lien*» qu'un «*vrai lien* » [108].

b) Identification des types de liens

Les types de liens de traçabilité peuvent être utilisés pour faciliter le processus de vérification de la conformité d'un système à ses exigences, pour faire l'analyse d'impacts, pour comprendre l'évolution des artefacts ou pour justifier la logique qui sous-tend certains aspects de conception et de mise en œuvre du système. Malgré son importance, le soutien à l'identification des types de liens dans les environnements et les outils actuels ne sont pas toujours satisfaisants. L'inconvénient majeur de ces supports est l'incapacité à identifier et à maintenir automatiquement des relations de traçabilité impliquant des artefacts qui sont exprimés en langage naturel et créés indépendamment par des outils non interopérables et qui évoluent de manière autonome. En conséquence, les types de liens doivent être établis et maintenus manuellement lors de la validation, ce qui est fastidieux et complexe.

Dans la littérature quelques techniques ont été proposées pour pallier cette problématique. Toutefois, une limite commune à l'ensemble des techniques d'identification des types de liens présentées dans le chapitre 2 est la création manuelle de dictionnaires de synonymes. Cette tâche pouvant être fastidieuse, des auteurs comme Divya et al. [109] ont proposé des techniques pour construire automatiquement des dictionnaires de termes sémantiquement proches à partir de documents donnés. Citons également les travaux de Kof et al. [110], de De Nicola et al. [111], et de Gacitua et al. [112] qui ont apporté une contribution significative dans le domaine d'extraction d'ontologies à partir d'exigences. Récemment la technique *Word embedding* a été utilisée avec succès dans diverses tâches de traitement du langage naturel [98, 113]. Elle pourrait donc également être mise à profit pour réaliser cette tâche afin d'obtenir de meilleures performances.

Le tableau 3.2 donne un résumé des problèmes de la traçabilité en entreprise étendue traités, des questions de recherche et des problématiques de la thèse.

3.8 Conclusion

Dans ce chapitre, nous avons examiné la littérature en réalisant une étude bibliographique des travaux relatifs à l'élicitation et la maintenance des liens de traçabilité. De façon générale, les approches étudiées ont des lacunes sur deux axes : avant et après l'élicitation

<table>
<tr><td>Problèmes de la traçabilité en entreprise étendue</td><td>Questions de recherche</td><td>Problématique de thèse</td></tr>
<tr><td>Problème 1 : Hétérogénéité des artefacts et des outils</td><td></td><td></td></tr>
<tr><td>Problème 2 : Interprétation sémantique des liens</td><td>Question 2 : Quel est l'apport des méthodes de plongements lexicaux basées sur des modèles neuronaux à la capture de la sémantique des liens de traçabilité ?</td><td>Identification des types de liens</td></tr>
<tr><td>Problème 3 : Efficacité et confiance dans les outils de traçabilité</td><td rowspan="2">Question 1 : Quelles méthodes de traçabilité permettent d'obtenir de bonnes performances sur des jeux de données industriels ?</td><td rowspan="2">Réduction du coût de l'élicitation des liens de traçabilité</td></tr>
<tr><td>Problème 4 : Charge de travail alloué à la traçabilité</td></tr>
</table>

Table 3.2 *Récapitulatif des problématiques*

des liens de traçabilité. Concernant le premier axe, les approches nécessitent beaucoup de temps pour la construction de bases de connaissance du projet (ontologies, métamodèles). Pour le deuxième axe, les approches reportent la validation des liens après la création, celle-ci restant néanmoins encore chronophage (*problème 3 : Efficacité et confiance* et *problème 4 : Charge de travail*).

Pour la maintenance des liens, la transformation de modèle, les approches à base de règles et orientées événements restent les plus utilisées. La principale lacune de ces approches est qu'elles ne prennent pas en charge la maintenance de liens de traçabilité et d'artefacts issus de cadre de modélisation hétérogène (*problème 1 : Hétérogénéité*).

Au regard de ces lacunes et des quatre problèmes, mentionnés dans le chapitre 1, qui se posent dans la traçabilité en entreprise étendue, des questions scientifiques ont été identifiées. Leur domaine a été restreint aux approches de Recherche d'Information et d'apprentissage automatique des langues. Ces questions sont :

— Question 1 : Quelles méthodes de traçabilité permettent d'obtenir de bonnes performances sur des jeux de données industriels ? (*problème 3 : Efficacité et confiance* et *problème 4 : Charge de travail*)
— Question 2 : Quel est l'apport des méthodes de plongements lexicaux basées sur modèles neuronaux à la capture de la sémantique des liens de traçabilité ? (*problème 2 : Interprétation*)

Elles sont regroupées dans deux problématiques, à savoir :

— Réduction du coût de l'élicitation des liens de traçabilité
— Identification des types de liens

Afin de répondre à ces problématiques, nous allons étudier l'environnement scientifique des communautés de recherche d'information, de traitement automatique des langues, et de l'apprentissage semi-supervisé dans le cadre de la traçabilité.

Ainsi, le chapitre suivant introduit quelques notions du traitement automatique des langues en traçabilité. Il présente également des techniques de Recherche d'Information et des techniques semi-supervisées sur lesquelles s'appuient nos travaux de recherche.

4 – Bases méthodologiques

4.1 Introduction

Ce chapitre présente les différentes techniques et hypothèses sur lesquelles s'appuient nos travaux. Nous allons tout d'abord présenter des techniques du traitement automatique des langues (TAL), qui ont surpassé les performances des techniques traditionnelles de la traçabilité. Nous allons ensuite présenter en détail les techniques de Recherche d'Information qui ont été les plus utilisées dans la communauté de la traçabilité. Nous illustrerons ces techniques avec les énoncés d'artefacts tirés du cas d'usage de nos partenaires industriels ou des cas d'usage publics utilisés pour l'évaluation de notre approche. Puis nous montrerons l'apport des méthodes semi-supervisées dans notre contexte industriel et en faisant une brève présentation de la méthode non supervisée « LDA » et de la méthode semi-supervisée « LabelSpreading ». Nous finirons en présentant les métriques d'évaluations des techniques les plus utilisées dans la communauté de la traçabilité.

4.2 Techniques du Traitement Automatique des Langues

Les techniques du Traitement Automatique des Langues sont généralement utilisées pour transformer les données textuelles afin d'en faciliter l'analyse.

Les sous-sections présentent les techniques de bases et les techniques récentes du traitement automatique des langues.

4.2.1 Bases du Traitement Automatique des Langues

Le traitement automatique des langues fait référence à la compréhension, l'analyse et la manipulation informatisées, et la génération du langage naturel [114]. L'ensemble des techniques utilisées dans le chapitre 5 sont présentées dans la suite. Dans cette section, les techniques seront illustrées à l'aide de l'énoncé de l'exigence *650001*, tirée du cas d'usage de nos partenaires industriels. Rappelons l'énoncé de cette exigence « *R650001 : The system of systems shall prevent any damage on pedestrian* ». Cet énoncé est également disponible dans l'annexe B.

a) Segmentation en phrases

Elle consiste à séparer les phrases d'un texte en se basant sur le principe qu'à l'écrit, une phrase commence obligatoirement par une majuscule et se termine par un signe de

ponctuation à savoir le point, le double point, le point-virgule ou encore les trois points de suspension.

b) Séparation des termes

Elle consiste à séparer les termes d'une phrase en se basant sur le principe que dans une langue, les termes sont séparés par un séparateur de mots. Pour la langue anglaise, qui est la langue dans laquelle nos données d'études sont écrites, le séparateur de mot est le blanc typographique.

Par exemple, la séparation des termes de l'exigence *R650001* donne [{*The*}, {*system*}, {of}, {*systems*}, {*shall*}, {*prevent*}, {*any*}, {*damage*}, {*on*}, {*pedestrian*}]

c) Retrait des mots vides

Un mot vide (en anglais stop word) est un mot non significatif figurant dans un texte [115]. En traçabilité, les mots vides sont des termes qui apparaissent si fréquemment dans les artefacts qu'ils ne sont plus pertinents pour l'identification des liens. Un mot est considéré ***vide*** lorsque sa fréquence est plus ou moins la même dans chacun des textes d'un corpus. De ce fait, ce mot n'est plus discriminant car il ne permet pas de distinguer les textes les uns par rapport aux autres. Les mots vides sont principalement des mots caractéristiques d'une langue comme les prépositions, les articles, les pronoms. Des listes pré-établies de mots vides sont disponible dans la plupart des algorithmes de traitement des langues.

d) Racinisation

La *racinisation* consiste à réduire les termes à leur racine ou radical. Elle permet ainsi d'obtenir une forme tronquée des termes, commune à toutes les variantes morphologiques d'un terme. Par exemple, la racinisation de « *fishing* », « *fished* », « *fish* » donne « *fish* ». Les termes ayant le même radical sont considérés comme similaires. L'efficacité de cette technique est limitée [116]. En effet, elle augmente le nombre de faux positifs lorsque la réduction des termes est excessive ou moindre (*overstemming* et *understemming*) [117]. Par exemple, la racinisation de « *university* » et « *universe* » donne « *univers* », bien que ces deux termes n'aient pas des sens similaires.

e) Lemmatisation

Elle consiste à regrouper les termes d'une même famille dans un texte, afin de réduire ces termes à leur forme canonique ou lemme. Par exemple, elle va réduire un verbe à sa forme infinitive et un nom à sa forme masculin singulier « *systems* » en « *system* ». Cependant, elle n'agrège que les variantes morphologiques flexionnelles. Par exemple, « *chevaux* » va être réduit à « *cheval* », ce qui n'est pas le cas de « *chevalerie* ». Cette technique permet d'améliorer les performances des modèles d'extraction d'information [78]. Ne tenant pas compte du sens des termes qu'elle regroupe, elle manque souvent des regroupements de termes tels que « *good* » et « *better* ». Ce genre de regroupement est généralement réalisé à l'aide de dictionnaires ou de thesaurus.

f) Thesaurus et dictionnaires

Les thésaurus et les dictionnaires sont des ressources lexico-sémantiques qui fournissent les connexions sémantiques entre les termes qu'elles contiennent. Ils proposent une représentation où la signification des termes dépend de relations hiérarchiques (hyperonymie, hyponymie, etc). Il existe plusieurs ressources lexico-sémantiques informatisées : *WordNet, BabelNet, Wiktionary* ou encore les encyclopédies universelles (*wikipédia*). Elles peuvent être associées aux techniques de recherche d'information afin d'améliorer leurs performances.

g) Etiquetage morpho-syntaxique (part-of-speech tagging)

Elle consiste à analyser la syntaxe des textes et à associer les termes du texte avec leurs attributs grammaticaux. Cette technique permet d'expliquer comment un terme est utilisé dans une phrase. Il y a huit principales étiquettes morpho-syntaxiques : les noms, les pronoms, les adjectifs, les verbes, les adverbes, les prépositions, les conjonctions et les interjections. Par exemple, l'étiquetage morpho-syntaxique de l'exigence *R650001* donne [{ *Determiner*, *The*}, { *Noun*, *system*}, { Preposition, of}, { *Noun*, *systems*}, { *Verb*, *shall*}, { *Verb*, *prevent*}, { *Determiner*, *any*}, { *Noun*, *damage*}, { *Preposition*, *on*}, { *Noun*, *pedestrian*}]

Les résultats d'une analyse lexicale fournie par l'étiquetage morpho-syntaxique peuvent être utilisés pour identifier des taxonomies de termes clés dans des documents d'exigences textuelles. L'étiquetage morpho-syntaxique est généralement associé à d'autres techniques du traitement des langues.

h) Séparation en syntagmes (Text chunking)

Un syntagme est un groupe d'un ou plusieurs termes qui forment un constituant syntaxique ou sémantique d'une phrase [118]. Il existe cinq types de syntagmes : verbal, nominal, adverbial, adjectival, prépositionnel.

La séparation en syntagmes consiste donc à séparer une phrase en syntagmes en se basant sur l'étiquetage morpho-syntaxique. En traçabilité des exigences, après avoir exécuté la séparation en syntagmes, seuls les syntagmes verbaux et nominaux sont généralement considérés, parce que ce sont les principaux éléments qui sont porteurs de sens dans les artefacts (exigences et éléments de modèles). La séparation en syntagmes de l'exigence *R650001* donne [{ *Noun Phrase*, *The system of systems*}, { *Verbal Phrase*, *shall prevent*}, { *Noun Phrase*, *any damage on pedestrian*}]

i) Mesures de similarité et de dissemblance

Les mesures de similarité et de dissemblance jouent un rôle très important dans certaines tâches du traitement automatique des langues comme la recherche d'information, l'analyse des sentiments [119]. Elles sont utilisées pour définir le degré de similarité ou de dissemblance (distance) entre des termes. Elles peuvent être syntaxiques ou sémantiques.

Les mesures syntaxiques calculent les scores de similarité en fonction du contenu des chaînes de caractères des segments de texte, parfois combinés avec les fréquences. Elles sont généralement les mieux adaptées pour faire correspondre les variations du même terme ou de la même phrase et pour traiter des termes mal orthographiés ou ne se trouvant pas dans des dictionnaires [120].

Les mesures sémantiques sont calculées à partir des corrélations capturées dans des ressources lexico-sémantiques. Elles sont les plus appropriées pour faire correspondre des termes qui sont syntaxiquement différents mais qui ont un sens proche. Elles fournissent en outre une base précise pour faire face aux variations morphologiques des termes. Les mesures de similarité sémantique sont généralement appliquées dans des contextes locaux c'est-à-dire qu'elles donnent une valeur de proximité entre deux termes [121].

Nous utilisons toutes ces techniques et y ferons référence dans les chapitres suivants.

4.2.2 Techniques récentes du traitement automatique des langues

Les techniques classiques de recherche d'information présentées dans la section 4.3 considèrent les termes et les groupes de termes comme des entités non porteuses de sens, plus précisément elles considèrent que les termes sont indépendants les uns des autres. De ce fait, ces techniques représentent les documents par des vecteurs de fréquences de vocabulaire (nombre d'occurrences de termes) disponible dans les documents. Le *word* et *sentence embeddings* viennent repenser ce modèle de représentation en projetant les termes dans un espace qui les rapprochent suivant leur sens en matière de distances statistiques. Ces récentes techniques du traitement des langues sont détaillées dans les sous-sections suivantes. Elles seront également illustrées avec les énoncés de l'exigence *1245* et du but *113*.

a) Word embeddings

Word Embeddings [98], traduit en français par « plongement de mots », fait référence à des techniques qui consistent à représenter des termes en des vecteurs réels de d-dimension en incorporant la syntaxe et les relations sémantiques entre les termes, de sorte que les termes similaires ont des représentations vectorielles similaires. Elles sont également capables de coder les relations syntaxiques et sémantiques entre les termes comme des relations linéaires [78].

Cette technique est basée sur l'hypothèse de distribution de Harris [122]. Cette hypothèse stipule que les termes, qui apparaissent dans un même contexte, peuvent être caractérisés par les termes qui les entourent. Un contexte peut être un seul terme ou un groupe de termes. Des termes associés aux mêmes termes ont ainsi tendance à avoir des significations similaires. Par conséquent, si le même terme apparaît dans deux contextes différents, il aura deux représentations vectorielles distinctes.

Les techniques de Word Embeddings sont généralement utilisées pour alimenter les algorithmes d'apprentissage automatique (*machine learning*) tel que les réseaux de neurones afin de réaliser différentes tâches de traitement automatique des langues. Les techniques de Word Embeddings représentent les termes en les projetant dans un espace qui les rapproche suivant leur sens en matière de distance statistiques. Cette représentation des termes dans un espace de dimension est appelée Word2vec [123].

b) Word2Vec

Word2vec est un groupe de modèles utilisé pour produire des représentations vectorielles de termes. Ces modèles sont composés de réseaux de neurones artificiels entraînés pour reconstruire le contexte linguistique des termes. Ils ont été développés par une équipe de

recherche chez Google sous la direction de Tomas Mikolov [123]. Ces derniers ont démontré que les relations sémantiques sont souvent préservées dans les opérations vectorielles sur des vecteurs de termes, par exemple, *vec(King) - vec(Man) + vec(Woman)* est proche de *vec(Queen)*. En effet, avec un corpus assez grand et suffisamment d'usages et de contextes, *Word2vec* peut faire des suppositions très précises sur la signification d'un terme à partir des différentes occurrences du terme dans le corpus d'apprentissage.

L'entrainement des plongements de mots (Word Embeddings) consiste à trouver des représentations vectorielles de termes de sorte que les termes ayant une signification similaire soient associés à une même représentation vectorielle [124]. Il existe de nombreuses représentations vectorielles des termes les plus courants disponibles publiquement. Cette liste de termes avec leurs représentations vectorielles est appelée modèle pré-entrainé. Ces derniers sont des ensembles de termes associés à leurs vecteurs numériques, de sorte que ces vecteurs capturent les différentes significations du terme présentes dans le corpus d'apprentissage. Ils se sont avérées être très utiles dans différentes tâches de traitement du langage [125, 126, 127]. Parmi ces modèles pré-entraînés, celui de *Google word2vec* est populaire pour sa simplicité et son efficacité. Le modèle pré-entrainé *Google word2vec* a été entrainé sur environ 100 milliards de termes de *Google News*. Son entrainement a été réalisé avec la technique *Continuous Bag of Words (CBOW)* avec un vocabulaire d'environ 3 millions d'entités. Il peut être téléchargé en ligne sur le site internet *https ://code.google.com/archive/p/word2vec/*.

Word2vec est un groupe de modèles qui a pour objectif la représentation vectorielle de termes à l'aide de technique d'entrainement. Il existe deux principales techniques d'entraînement : la technique *CBOW* et la technique *Skip-Gram*. Ces techniques sont détaillées dans les deux sous-sections suivantes.

b).1 Continuous Bag of Words (CBOW) La technique d'entraînement CBOW a pour objectif d'obtenir la représentation appropriée d'un terme, en se basant sur une fenêtre de termes adjacents (fenêtre de contexte), afin de retrouver le prochain terme du même contexte. Un contexte peut être un seul terme ou un groupe de termes positionnés avant/après dans un terme dans une phrase et une fenêtre de contexte est caractérisée par un nombre de termes du contexte.

CBOW entraîne un réseau de neurones pour prédire un terme en fonction de son contexte. Par exemple, la figure 4.1 montre le calcul de la représentation vectorielle du terme « System » de l'exigence *1245* avec une fenêtre de contexte égale à 1. L'entrée du réseau de neurones prend une fenêtre de contexte autour du terme. Ce réseau de neurones contient quatre couches : la couche d'entrée (*input layer*), la couche cachée d'entrée (*input hidden layer*), la couche cachée de sortie (*output hidden layer*) et la couche de sortie (*output layer*). La couche d'entrée et la couche de sortie ont la même taille, voir figure 4.1 size[1 x V]. Il y a deux ensembles de pondération (tableaux Input-hidden layer et output-hidden layer dans la figure 4.1). Le premier ensemble est entre la couche d'entrée et la couche cachée d'entrée et le second est entre la couche cachée de sortie et la couche de sortie. Le réseau neuronal prend également en entrée un hyper-paramètre arbitraire N qui permet de définir le nombre de dimensions pour représenter un terme et le nombre de neurones dans les couches cachées, dans notre exemple N = 2.

Dans la couche d'entrée, les termes sont représentés dans un espace qui les positionne en fonction des termes adjacents (représentation word2vec). Les valeurs de cette couche sont multipliées par l'ensemble de pondération de la couche cachée d'entrée. Dans la figure 4.1, il

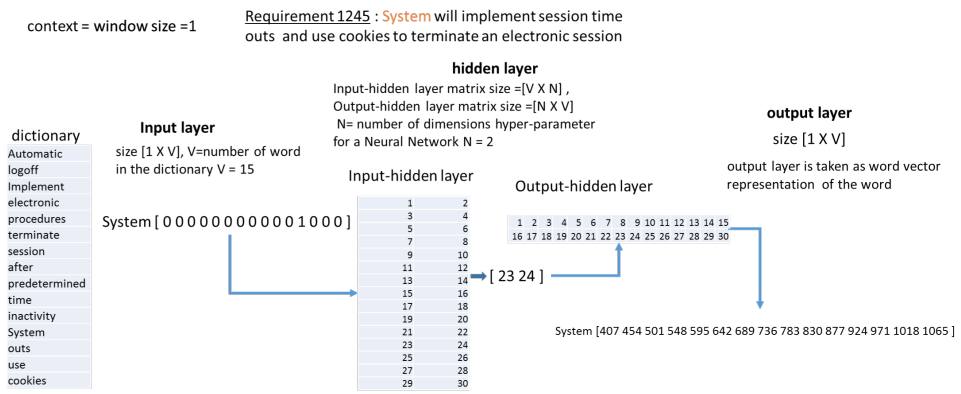

Fig. 4.1 *Illustration de la technique CBOW avec le terme « System » de l'exigence 1245 La couche d'entrée est multipliée par l'ensemble de pondération de la couche cachée d'entrée ; le résultat est multiplié avec l'ensemble de pondération de la couche cachée de sortie ; le vecteur obtenu correspond à la représentation vectorielle du terme « System ».*

s'agit de la ligne correspondante dans la matrice cachée d'entrée donnant le vecteur [23,24]. Ce résultat est multiplié par l'ensemble de pondération de la couche cachée de sortie et la couche de sortie est calculée. Ce résultat donne la représentation vectorielle du terme considéré.

b).2 Skip-Gram La technique d'entraînement Skip-Gram a pour objectif d'obtenir le vecteur de représentation du terme qui permet de déduire son contexte.

Skip-Gram entraîne un réseau de neurones pour prédire, sur une fenêtre de contexte donnée, les termes pouvant apparaître autour d'un terme défini. Par exemple, la figure 4.2 illustre le calcul de la représentation vectorielle du terme « Terminate » du but *113* avec une fenêtre de contexte égale à 1, et en tenant compte de deux contextes. Le premier contexte est le terme « Terminate » dans l'exigence *1245* et le deuxième contexte le terme « Terminate » dans l'exigence *59104*. Rappelons l'énoncé de l'exigence « *59104 : Electronic session must terminate after a pre determined time of inactivity* ».

Comme la technique d'entrainement CBOW, Skip-Gram prend en entrée du réseau de neurones, une fenêtre de contexte autour du terme, deux ensembles de pondération et un hyper-paramètre arbitraire N. Le réseau de neurones contient également quatre couches : la couche d'entrée (*input layer*), la couche cachée d'entrée (*input hidden layer*), la couche cachée de sortie (*output hidden layer*) et la couche de sortie (*output layer*).

La couche d'entrée correspond à la représentation des termes en vecteur word2vec. Cette représentation est également calculée pour les termes de contexte. Ces derniers sont des termes qui se trouvent avant ou après le terme d'entrée en fonction de la fenêtre de contexte. Dans l'exemple, les termes cibles sont « *cookies* » et « *session* ». La représentation prise dans l'exemple est celle de la position des termes dans le dictionnaire V, voir figure 4.2. Les valeurs de la couche d'entrée sont multipliées par l'ensemble de pondération de la couche cachée d'entrée. Le vecteur obtenu est multiplié par l'ensemble de pondération de la couche cachée de sortie. Ce calcul donne une nouvelle représentation vectorielle du terme d'entrée. L'erreur est calculée en soustrayant la représentation vectorielle des termes de contexte avec la nouvelle représentation vectorielle du terme d'entrée. Par conséquent, pour n termes de

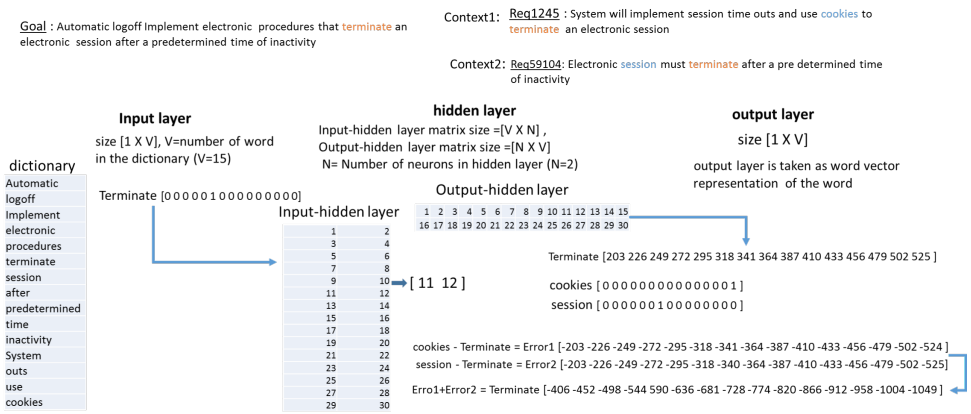

Fig. 4.2 *Illustration de la technique Skip-Gram avec le terme « Terminate » du but 113 La couche d'entrée est multipliée par l'ensemble de pondération de la couche cachée d'entrée ; le résultat est multiplié avec l'ensemble de pondération de la couche cachée de sortie ; un nouveau vecteur de représentation du terme « Terminate » est obtenu ; les vecteur de représentation des termes de contexte sont obtenu grâce à leur position dans le vocabulaire V ; les vecteurs d'erreur sont calculés par soustraction des vecteurs des termes de contextes avec le nouveau vecteur du terme « Terminate » ; la somme des vecteurs d'erreur permet d'obtenir le vecteur d'erreur final, qui correspond à la représentation vectoriel du terme « Terminate ».*

contexte, il y aura n vecteurs d'erreur. La somme par éléments est prise sur tous les vecteurs d'erreur pour obtenir un vecteur d'erreur final. Ce vecteur d'erreur final correspond à la représentation vectorielle du terme d'entrée.

c) Global Vectors (GLoVe)

Global Vectors for word representation en abrégé *GLoVe* [4] est une autre méthode non-supervisée de dérivation des vecteurs de représentation de termes. Elle a été développée comme un projet open-source à Stanford [4]. Le modèle pré-entrainé de *GLoVe* peut être téléchargé en ligne sur le site internet *https ://nlp.stanford.edu/projects/glove/*.

GLoVe permet de calculer la fréquence à laquelle les termes coexistent les uns avec les autres dans un corpus donné. Pour le faire, les termes sont projetés dans un espace significatif où la distance entre les termes est liée à leur similarité sémantique [128]. L'apprentissage est effectué sur des statistiques globales agrégées de co-occurrence des termes d'un corpus. Les représentations résultantes présentent des sous-structures linéaires intéressantes de l'espace vectoriel du terme. Comme le modèle *Google word2vec*, le modèle *GLoVe* est très efficace pour les tâches d'analogie et de similarité de termes. Par exemple, la figure 4.3 présente le lien sémantique entre plusieurs termes. Une régularité géométrique peut être observée entre les différents termes. La même régularité est observé entre les termes « *short* » et « *slow* » et leur comparatifs « *shorter* », « *slower* » et superlatifs « *shortest* », « *slowest* ». De même, les termes masculins « *brother* », « *nephew* », « *uncle* » et leur féminin « *sister* », « *niece* », « *aunt* », présente une régularité géométrique identique.

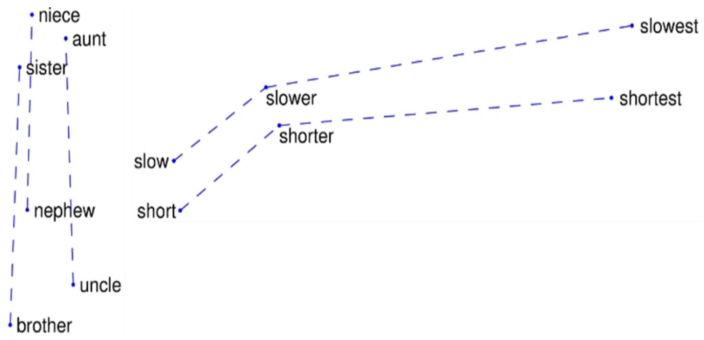

Fig. 4.3 *Paires de termes masculins et féminins et trios de termes comparatifs superlatifs pré-entraînés avec la technique de représentation des termes* Glove *[4]*

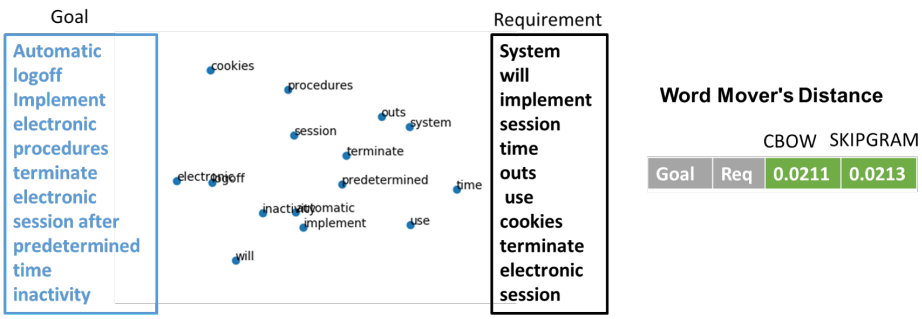

Fig. 4.4 *Illustration schématique du Word Move Distance avecl'exigence 1245 et le but 113. Tous les termes des deux artefacts sont projetés dans espace vectoriel ; la distance entre l'exigence 1245 et le but 113 est la distance cumulative minimale que tous les mots du but 113 doivent parcourir pour correspondre exactement aux mots de l'exigence 1245 ; cette distance calculé avec CBOW et Skip-Gram est proche de 0, elle est donc représentée avec une couleur de variante verte.*

d) Similarité entre les plongements de mots

Les plongements de mots sont des représentations numériques des similitudes contextuelles entre les mots. Ils peuvent donc être manipulés pour effectuer différentes tâches du traitement automatique des langues comme le calcul du degré de similitude entre deux termes. Ce degré est exprimé sous forme de distance, plus la distance entre deux termes est proche de zéro, plus ces termes sont considérés comme proches.

La similarité entre deux documents avec les plongements de mots peut être définie par différentes métriques parmi lesquelles, la distance Word Mover's Distance (WMD) [113]. Cette distance représente les documents en tant que nuage de points pondérés des représentations vectorielles des termes. La distance entre deux documents A et B est la distance cumulative minimale que les termes du document A doivent parcourir pour correspondre exactement au nuage de points du document B. La figure 4.4 montre une illustration schématique de cette métrique. Elle n'a pas d'hyper-paramètres et est simple à mettre en œuvre.

e) Synthèse

Contrairement aux techniques traditionnelles de recherche d'information comme LSI ou VSM, qui sont basées sur la fréquence des mots et traitent chaque terme comme un symbole unique, les techniques de Word Embeddings ne considèrent pas les termes comme étant indépendants les uns des autres et permettent de gérer la polysémie. Le modèle Skip-gram peut en effet capturer plusieurs sémantiques pour un seul terme, il aura ainsi plusieurs représentations vectorielles du même terme suivant les contextes et les usages. Bien que, Skip-Gram associé au sous-échantillonnage négatif surpasse généralement toutes les autres méthodes [129, 130], il ne fonctionne bien qu'avec de petites quantités de données. Dans le même temps, CBOW est plus rapide, et a de meilleures représentations pour les termes les plus fréquents. GLoVe est également un modèle de représentation vectorielle de termes implémenté dans différentes librairies populaires (Gensim, Spacy) [4]. Il a de bonnes performances et converge plus rapidement que les modèles Word2Vec.

Les techniques Word embedding peuvent être considérées comme l'une des principales raisons du succès des récents modèles d'apprentissage automatique pour les tâches de traitement de la langue. Elles ont ainsi été utilisées avec succès dans la communauté de la traçabilité [78, 79, 80].

4.2.3 Sentence embeddings

Le succès des techniques de word embeddings a motivé l'implémentation de techniques pour capturer la sémantique de textes plus longs, tels que les phrases et les paragraphes [131, 132, 133]. Comme les techniques de word embeddings qui représentent les termes comme des vecteurs, les techniques de *sentence embeddings* capturent le sens d'un syntagme ou d'une phrase dans un seul vecteur. Ces techniques peuvent ainsi être utilisées pour améliorer la précision de l'identification liens de traçabilité.

Plusieurs techniques d'apprentissage de représentation des phrases ont été proposées dans la littérature. Dans les sous-sections suivantes une sélection de ces techniques est brièvement introduite.

a) Smooth Inverse Frequency (SIF)

Parmi les différentes techniques de sentence embeddings, on peut trouver la méthode d'Arora [131], appelées *Smooth Inverse Frequency (SIF)*. SIF est une méthode non supervisée qui utilise les plongements de mots calculés à l'aide de GLoVe sur des corpus non annotés comme Wikipedia. Elle représente ensuite la phrase par une moyenne pondérée des vecteurs de représentation des termes puis modifie ces vecteurs en utilisant une décomposition en valeurs singulières (Singular Value Decomposition - SVD) ou une analyse en composantes principales (Principal Component Analysis - PCA).

Prendre la moyenne des vecteurs de représentation des termes d'une phrase tend à donner trop de poids à des termes qui ne sont pas pertinents d'un point de vue sémantique comme « *but* », « *just* ». Pour résoudre ce problème, la méthode SIF utilise deux techniques : la pondération et la suppression des composants communs.

— La pondération : comme le TF-IDF, SIF prend la moyenne pondérée des termes incorporés dans la phrase. Chaque plongement de mot est pondéré par $a/(a + p(w))$,

où a est un paramètre qui est réglé à $0,001$ et $p(w)$ est la fréquence estimée du mot dans le corpus d'apprentissage [131].

— La suppression des composants communs : SIF calcule le composant principal des vecteurs de représentation résultant pour un ensemble de phrases. Elle soustrait ensuite de ces phrases leurs projections sur leur première composante principale. Elle élimine ainsi les variations liées à la fréquence et à la syntaxe des termes qui sont moins pertinentes sur le plan sémantique.

Cette méthode permet d'obtenir des performances nettement supérieures à la moyenne non pondérée pour une variété de tâches de similarité textuelle, et sur la plupart de ces tâches a de meilleures performances que certaines méthodes supervisées. Son principal inconvénient est qu'elle ne tient pas compte de l'ordre des termes dans la phrase. Effectivement, les différences dans l'ordre des termes vont souvent de pair avec les différences de signification.

b) Doc2Vec

Doc2Vec [134] est un algorithme non supervisé qui permet de générer des vecteurs de représentation pour des textes longs comme les phrases, les paragraphes ou les documents. Cet algorithme est une adaptation de *Word2Vec* qui permet de générer des vecteurs de représentation pour les termes. Son objectif est de trouver les représentations vectorielles des différents documents d'un corpus donné.

Word2Vec est basé sur l'hypothèse que la représentation vectorielle d'un terme devrait permettre de prédire les termes voisins, tandis que l'hypothèse sous-jacente de *Doc2Vec* est que la représentation vectorielle du document devrait permettre de prédire les termes dans ce dernier. Dans l'algorithme *Doc2Vec*, les techniques d'entrainement exploitent l'idée que la prédiction des termes voisins d'un terme donné repose fortement sur le document. Par exemple, même avec une forte occurrence de l'expression « *primary forest* » dans un corpus, si le thème du document concerne l'« *apprentissage automatique* », le terme « *random* » aura une plus forte probabilité d'apparaître avant le terme « *forest* » au lieu du terme « *primary* » puisque l'expression « *random forest* » est plus plausible pour le thème « *apprentissage automatique*. Inversement, si le thème du document porte sur la « *nature*, le terme « *primary* » aura une plus forte probabilité d'apparaitre avant le terme « *forest* ».

Le principal objectif de *Doc2Vec* est d'associer des documents arbitraires à des thèmes. Cet algorithme apprend à corréler les thèmes et les termes plutôt que des termes avec d'autres termes. Il prend en entrée un ensemble des phrases du document, représenté sous forme d'objets. Chacun de ces objets représente une phrase et se compose de deux listes : une liste de mots et une liste de thèmes. L'algorithme parcourt deux fois l'ensemble des phrases : une première fois pour construire le vocabulaire, et une deuxième fois pour créer le modèle du document, en apprenant la représentation vectorielle de chaque terme et de chaque thème dans le jeu de données.

Les techniques d'entrainement de *Doc2Vec* sont *Distributed Bag of Words (DBOW)* et *Distributed Memory (DM)* :

— *Distributed Bag of Words (DBOW)* est l'équivalent du modèle Skip-Gram de *Word2Vec*. Les vecteurs de représentation des documents sont obtenus en entrainant un réseau de neurones à prédire une distribution de probabilités de termes dans un paragraphe à partir d'un terme du document choisi aléatoirement.

— *Distributed Memory (DM)* est l'équivalent du modèle *CBOW* de *Word2Vec*. Il agit

comme une mémoire en retenant des termes voisins d'un terme donné ou du thème du document. Les vecteurs de représentation des termes représentent le concept des termes tandis que le vecteur de représentation du document a pour but de représenter le concept du document.

L'algorithme Doc2Vec est implémenté dans différentes librairies populaires comme gensim [1]. Du fait de ses bonnes performance, il est utilisé pour transformer les textes en vecteurs dans les techniques traditionnelle de recherche d'information comme LDA et LSI. La principale limite de cet algorithme est qu'il repose sur la qualité des données (sensibilité à la rareté de termes mal orthographiés et aux termes n'existant pas dans le corpus d'apprentissage).

Le tableau 4.1 récapitule les avantages et les inconvénients des approches du traitement automatique des langues présentées dans cette section.

1. https ://radimrehurek.com/gensim/models/doc2vec.html

Techniques du traitement automatique des langues	Avantages	Inconvénients
Segmentation	— rapide — simple à mettre en œuvre	— dépendance aux langues — sensibilité à la rareté de termes mal orthographiés ou inexistants dans le dictionnaire
Séparation des termes		
Retrait des mots vides		
Séparation en syntagmes		
Racinisation	— réduction d'erreurs due à la non concordance de termes	— non prise en compte des synonymes et des homonymes
Lemmatisation		
thesaurus et dictionnaires	— description des relations sémantiques entre termes	— mise à jour coûteuse — combinaison de dictionnaires difficile à cause des divergences structurelles et de contenu
Etiquetage morpho-syntaxique	— aide à la construction du graphe de co-occurrence	— dépendance aux langues — sensibilité à la rareté de termes mal orthographiés ou inexistants dans le dictionnaire
Mesure de similarité syntaxique	— traitement de termes mal orthographiés ou inexistants dans des dictionnaires	— moins bonnes performances que les mesures sémantiques
Mesure de similarité sémantique	— aide aux correspondances de termes syntaxiquement différents mais ayant un sens proche	— application dans des contextes locaux (fournissent des valeurs de proximité entre deux termes)
Word2vec	— représentation vectorielle de termes — suppositions précises sur la signification d'un terme suivant ses différentes occurrences dans des documents	— demande un volume important de données annotées — temps de calcul
Glove	— efficacité pour des tâches d'analogie et de similarité de termes	— dépendance aux langues — performances liées à la qualité des données
WMD	— prise en compte de la dépendance des termes — gestion de la polysémie, de la synonymie	— temps de calcul — configuration des hyper-paramètres
SIF	— prise en compte de la dépendance des termes — gestion de la polysémie, de la synonymie	— performances liées à la qualité des données — temps de calcul
Doc2Vec		

Table 4.1 *Récapitulatif des avantages et des inconvénients des techniques du traitement automatique des langues*

4.3 Techniques traditionnelles de Recherche d'Information en traçabilité

De nombreuses techniques de recherche d'information ont été utilisées dans la communauté de la traçabilité. Nous présentons les plus utilisées dans les sous-sections suivantes.

Nous illustrerons chaque technique avec l'énoncé de l'exigence *1245* et du but *113*, du cas d'étude public HIPAA (Healthcare Insurance Portability and Accountability Act) disponible sur le site internet de COEST [2] (Center of Excellence for Software & Systems Traceability). Ce cas d'étude académique est présenté dans la section 6.2.2 du chapitre 6. Rappelons l'énoncé de l'exigence « *1245 : System will implement session timeouts and use cookies to terminate an electronic session* » et l'énoncé du but « *113 : Automatic logoff Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity* ». Ces énoncés sont également disponibles dans l'annexe B.

4.3.1 Term Frequency - Inverse Document Frequency (TF-IDF)

Créé dans les années 80 par Salton [135], le Cosinus de Salton (Term Frequency - Inverse Document Frequency - TF-IDF) est l'ancêtre des algorithmes de fouille de texte. La fréquence ***TF*** est le nombre de fois qu'un terme donné apparaît dans un document et la fréquence ***IDF*** d'un terme est le logarithme du rapport entre le nombre total de documents d'une collection et le nombre de documents contenant ce terme.

Cette technique évalue la rareté d'un terme dans un corpus de documents ayant un même champ lexical. Elle permet donc de classer les documents selon leur pertinence par rapport à un terme donné ou un groupe de termes. Cette dernière est une méthode de pondération qui est généralement associée à d'autres techniques de recherche d'information. Elle est également utilisée pour l'identification des types de liens. Elle se calcule comme suit :

$$TF(t,d) = \frac{\text{Nombre de fois que le terme t apparaît dans le document d}}{\text{Nombre de termes dans le document d}}$$

$$IDF(t) = log(\frac{\text{Nombre de documents}}{\text{Nombre de documents contenant le terme t}})$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Par exemple, le calcul du TF-IDF des termes contenus dans les énoncés de l'exigence *1245* et du but *113* est illustré dans la figure 4.5. Lorsqu'un terme apparaît dans tous les documents, il est probable que ce terme ne soit pas pertinent pour un document particulier. C'est le cas des termes « Terminate », « Electronic » et « Session » qui ont les plus faibles fréquences dans les deux artefacts. Cependant, lorsqu'un terme apparaît dans un sous-ensemble de documents, il est probable que ce dernier ait une certaine pertinence pour les documents dans lesquels il est présent. C'est le cas du terme « Automatics » dans l'exigence *1245* et du terme « System » dans le but *113*.

Le principal avantage de cette technique est qu'elle permet d'établir la description des documents dans un modèle vectoriel. Sa principale limite est qu'elle ne prend pas en compte le sens des combinaisons de termes. Par exemple, les termes de l'expression « *give up* » ont un sens lorsqu'ils sont pris ensemble et des sens différents lorsqu'ils sont considérés séparément.

2. Center of Excellence for Software & Systems Traceability, http ://www.coest.org

	TF-IDF	
Terms	Goal	Requirement
Automatic	0,10841917	0
logoff	0,10841917	0
Implement	0,08333333	0,1
electronic	0,16666667	0,1
procedures	0,10841917	0
terminate	0,08333333	0,1
session	0,08333333	0,2
after	0,10841917	0
predetermined	0,10841917	0
time	0,08333333	0,1
inactivity	0,10841917	0
System	0	0,130103
outs	0	0,130103
use	0	0,130103
cookies	0	0,130103

Fig. 4.5 *TF-IDF des termes de l'exigence 1245 et du but 113*
Les termes communs aux deux énoncés ont les plus faibles valeurs de TF-IDF tandis que les autres termes ont des valeurs de TF-IDF plus ou moins élevées.

4.3.2 Vector Space Model (VSM)

Le modèle vectoriel, de l'anglais Vector Space Model (VSM) [136] est une méthode algébrique, utilisée en recherche d'information pour la recherche documentaire, la classification ou le filtrage de données. L'ensemble de représentation des documents est un vocabulaire comprenant des termes d'indexation, notamment, les termes les plus significatifs d'un corpus donné.

L'importance des termes est calculée avec la fréquence des termes dans le document ou avec la méthode de pondération des termes TF-IDF. Chaque document est ainsi représenté par un vecteur, dont la dimension correspond à la taille du vocabulaire (voir figure 4.5). Cette représentation vectorielle des documents permet de projeter les documents textuels dans un espace vectoriel et de définir une notion de proximité entre documents (voir espace vectoriel dans la figure 4.6). La similarité entre deux documents est définie par la valeur du produit scalaire des vecteurs représentant les documents. Lorsque le produit scalaire de deux documents est nul, cela indique qu'ils sont strictement orthogonaux. Dans l'espace du langage naturel, cela traduit l'absence de termes communs entre ces documents. Les documents sont considérés liés lorsque le produit scalaire excède un seuil prédéfini. La figure 4.5 résume le fonctionnement de cette technique avec l'exigence *1245* et le but *113*. Les énoncés des artefacts sont transformés en vecteurs puis leur mesure de similarité est définie par le produit scalaire des deux vecteurs. Leur mesure de similarité est inférieure à 0.5.

Son principal avantage est qu'elle est relativement simple à implémenter. Ceci pourrait

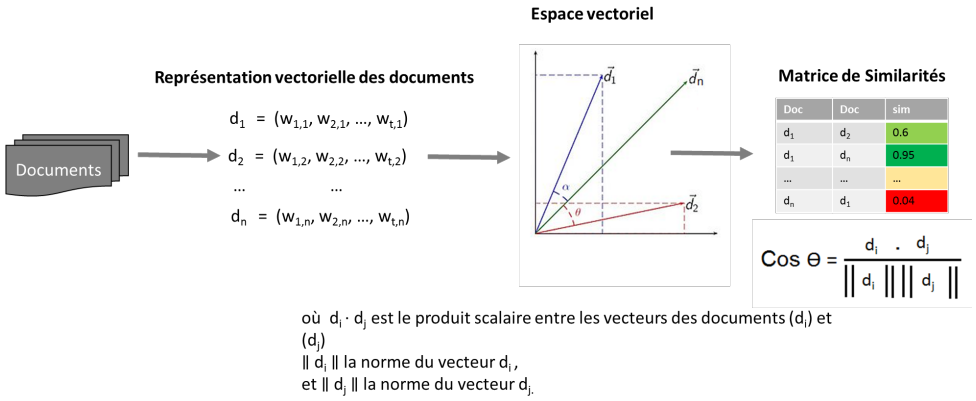

Fig. 4.6 *Résumé du fonctionnement de la technique VSM*
Les documents sont considérés liés lorsque le produit scalaire excède le seuil de 0.5. Les mesures de similarité les plus élevées sont représentés avec des variantes de la couleur verte tandis que les mesures de similarité les plus basses sont représentées avec des variantes de la couleur rouge.

expliquer pourquoi elle fait partie des techniques les plus utilisées dans la communauté de la traçabilité [93, 94, 137, 106]. Cependant, sa principale limite est qu'elle ne tient pas compte de la dépendance des termes [14] et de ce fait ne prend pas en compte la synonymie, la morphologie des termes et la polysémie. De plus, son efficacité comme celle des autres techniques de recherche d'information repose fortement sur les seuils de similarité définis, sur la méthode de pondération et sur la qualité du vocabulaire.

4.3.3 Latent Semantic Indexing (LSI)

L'indexation sémantique latente, de l'anglais Latent Semantic Indexing (LSI) [81], consiste à analyser des documents afin d'identifier les concepts contenus dans ces documents. Cette technique a été développée pour combler des lacunes de VSM. En effet, VSM ne prend pas en compte la synonymie. De plus, les matrices *termes-documents* (voir figure 4.7) dont les lignes correspondent aux termes et les colonnes correspondantes aux documents sont susceptibles d'avoir plusieurs dizaines de milliers de lignes et de colonnes pour un corpus de taille moyen. Le calcul de ces matrices pourrait donc demander de grandes capacités de calcul. LSI va donc être utilisé pour réduire le nombre de lignes tout en préservant la structure de similarité entre les colonnes. Cette réduction de matrice de rang plus faible, noté K, donne une approximation de la matrice originale.

La matrice termes-documents caractérise l'occurrence de chaque terme dans chacun des documents (voir étape 1 dans la figure 4.7). Cette occurrence est généralement normalisée avec la technique de pondération TF-IDF. La réduction de dimension permet à LSI d'identifier les concepts partagés par les différents documents. Elle est réalisée à l'aide de la décomposition en valeurs singulières (voir étape 2 dans la figure 4.7). Cette opération consiste à diviser une matrice en deux sous-matrices orthogonales et une sous-matrice diagonale. Les produits de ces sous-matrices donnent les corrélations entre les termes et entre les documents. Ensuite LSI transforme la matrice *termes-documents* en une « relation » entre les termes et les « concepts », et une relation entre ces concepts et les documents. Les documents sont alors représentés sous forme de vecteurs dans un espace de concepts et

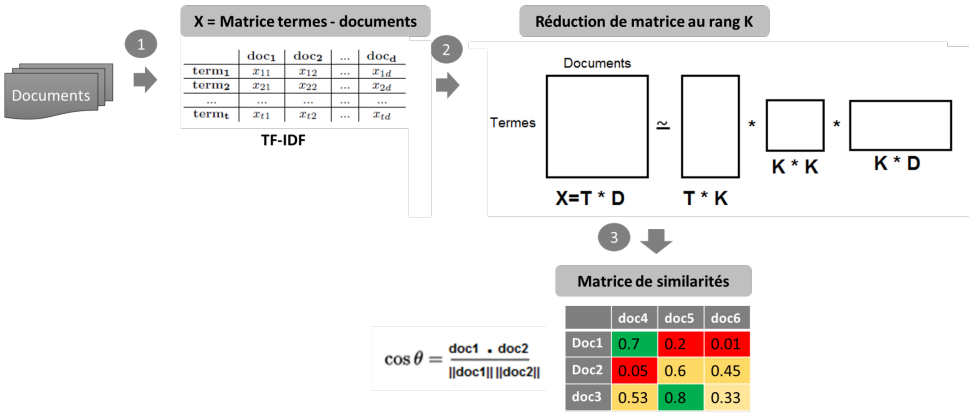

Fig. 4.7 *Résumé du fonctionnement de la technique LSI*
La matrice termes-documents *est décomposée en trois sous-matrices. Le produit de ces matrices donne les similarités entre les documents.*

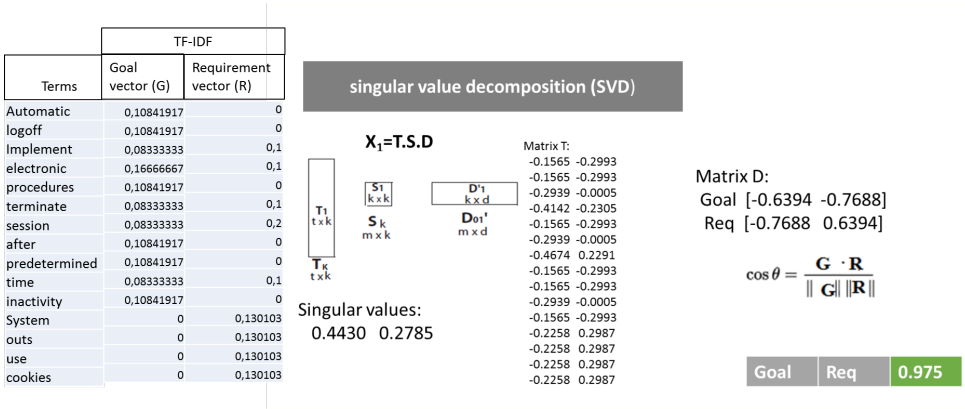

Fig. 4.8 *Illustration du calcul de LSI avec l'exigence 1245 et le but 113*
Les énoncés des artefacts sont transformés en matrice termes-documents *; Cette matrice est réduite au rang $K = 2$; la mesure de similarité entre les artefacts est supérieure à 0.5*

comparés par produits scalaires comme dans le cas de VSM (voir étape 3 dans la figure 4.7). La figure 4.7 résume le fonctionnement de LSI et le calcul de cette technique est illustré dans la figure 4.8 avec l'exigence *1245* et le but *113*.

Le principal avantage de LSI est qu'elle permet d'évaluer le niveau de corrélation entre les termes d'un document. Elle a donc été utilisée par de nombreux auteurs dans la littérature [82, 138, 139, 140] pour identifier les liens de traçabilité. Cependant, bien que la réduction de dimension de la matrice *termes-documents* permette la combinaison de certaines dimensions, ces dernières peuvent ne pas être pertinentes. De plus, chaque occurrence de termes est considérée comme ayant la même signification car le terme est représenté par un seul point dans l'espace. Ainsi, comme VSM, LSI ne permet pas de gérer la polysémie. En outre, comme indiqué dans la section4.6, ces résultats sont généralement difficiles à interpréter.

Le tableau 4.2 récapitule les avantages et les inconvénients des approches de Recherche d'Information présentées dans cette section.

Techniques de Recherche d'Information	Avantages	Inconvénients
TFIDF	— simple à mettre en œuvre — capacité de description de documents dans un espace vectoriel	— non prise en compte du sens des termes composés
VSM	— simple à mettre en œuvre	— non prise en compte de la dépendance des termes — non prise en compte de la synonymie, la polysémie et la morphologie des termes
LSI	— évaluation des niveaux de corrélation entre termes d'un document	— non prise en compte de la dépendance des termes — non prise en compte de la polysémie

Table 4.2 *Récapitulatif des avantages et des inconvénients des techniques de Recherche d'Information*

4.4 Techniques d'apprentissage automatique

Comme introduit dans la section 3.5.2 du chapitre 3, les techniques d'apprentissage automatique ont été largement utilisées pour améliorer les performances des techniques traditionnelles de recherche d'information [50]. Elles sont utilisées pour décrire des caractéristiques sur des paires de liens, pour définir des stratégies de classification des liens ou pour créer des modèles prédictifs. Les techniques d'apprentissage automatique sont généralement classées en trois groupes : les techniques non supervisées, les techniques supervisées et les techniques semi supervisées. Ces techniques reposent sur une hypothèse, dite « *cluster hypothesis* ».

4.4.1 Cluster Hypothesis

L'hypothèse de *cluster hypothesis*, dans le contexte d'apprentissage automatique, indique que les points proches sont susceptibles d'avoir la même étiquette ou que les points du même groupe sont susceptibles d'avoir la même étiquette. Cette hypothèse peut également indiquer, dans le contexte de Recherche d'Information, que les documents étroitement associés tendent à être pertinents pour les mêmes requêtes [115]. Elle peut ainsi être interprétée dans le contexte de la traçabilité par le fait que les « *vrais liens* » tendent à être plus similaires entre eux que les « *faux liens* » [94].

Cette hypothèse a été vérifiée à plusieurs reprises dans la communauté de l'apprentissage automatique [141, 142], de la Recherche d'Information [143, 144], et de la traçabilité [94, 145].

4.4.2 Latent Dirichlet Allocation (LDA)

Une des techniques d'apprentissage non supervisées très utilisée dans la communauté de la traçabilité est la technique Latent Dirichlet Allocation (LDA) [90, 91, 92]. L'Allocation de

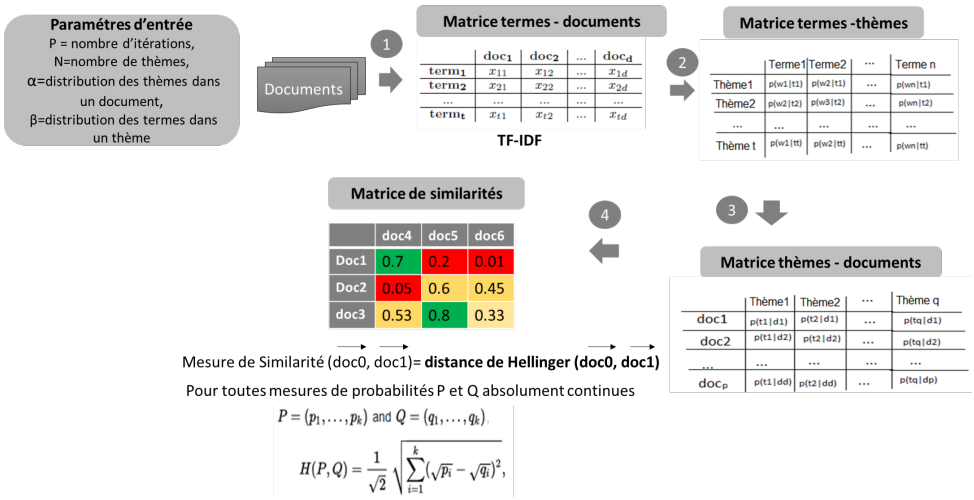

Fig. 4.9 *Résumé du fonctionnement de la technique LDA*
Les documents sont représentés en une matrice termes-documents *normalisée avec la technique de pondération TF-IDF. Cette matrice est transformée en matrice* termes-thèmes *puis en matrice* thèmes-documents *; la similarité entre les documents est calculée avec la distance de Hellinger.*

Dirichlet latente, de l'anglais Latent Dirichlet Allocation (LDA) [95], est un modèle génératif probabiliste qui permet de regrouper des documents par thèmes. La méthode LDA pose comme hypothèse que chaque document est un mélange de thèmes et que chaque terme est attribuable suivant une probabilité à un de ces thèmes. Les documents sont représentés en matrices *termes-documents* normalisées avec la technique de pondération TF-IDF (voir étape 1 dans la figure 4.9). Cette matrice est transformée en matrice *termes-thèmes* (voir étape 2 dans la figure 4.9) puis en matrice *thèmes-documents* (voir étape 3 de la figure 4.9).

LDA prend en entrée quatre paramètres : le nombre d'itérations, le nombre de thèmes, un coefficient de distribution de thèmes dans un document et un coefficient de distribution de termes dans un thème (voir figure 4.9). Cette technique attribue initialement un thème aléatoire à chaque terme de l'ensemble des documents. Puis LDA améliore le modèle généré aléatoirement en attribuant à chaque terme dans chaque document, la plus forte probabilité qu'il appartienne à un thème en faisant l'hypothèse que tous les thèmes sont correctement attribués aux autres termes, sauf pour le terme courant. Cette opération est répétée en fonction du nombre d'itérations définies. Ce dernier est généralement assez grand pour permettre aux allocations de se stabiliser. Au final, les documents sont représentés par un vecteur contenant la proportion de chaque thème qui les constitue. La similarité entre deux documents est définie par la distance de Hellinger [146] entre les vecteurs les représentant (voir étape 4 de la figure 4.9). Les documents sont considérés liés lorsque la mesure de similarité excède un seuil prédéfini. La figure 4.9 résume le fonctionnement de cette technique et son calcul est illustré dans la figure 4.10 avec l'exigence *1245* et le but *113*.

LDA a été utilisée par différents auteurs [90, 91, 92] dans la littérature. Son principal avantage est qu'elle permet de modéliser les relations entre les thèmes. Cependant son principal inconvénient est qu'elle est paramétrique et qu'il n'y a pas de mesure objective pour définir le meilleur choix des paramètres pour un jeu de données défini. De plus, comme les autres techniques traditionnelles de recherche d'information, elle considère que les termes

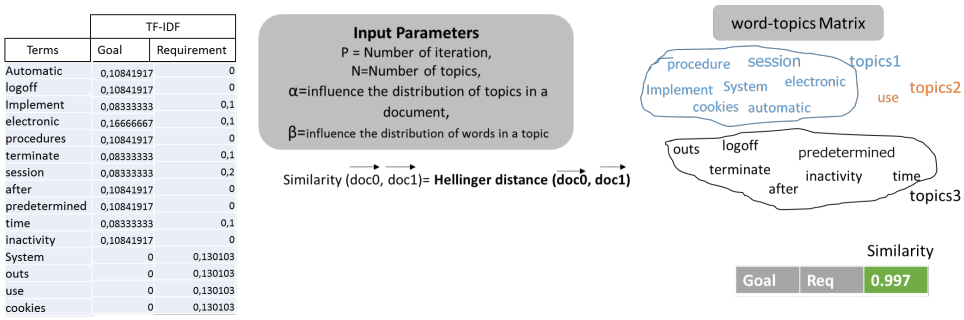

Fig. 4.10 *Illustration du calcul de LDA avec l'exigence 1245 et le but 113*
Les énoncés des artefacts sont transformés en matrice termes-documents *; les termes des énoncés sont classés en 3 thèmes ; la mesure de similarité entre les artefacts est supérieure à 0.5*

sont indépendants les uns des autres.

4.4.3 LabelSpreading

L'algorithme *label propagation* [147], et sa variante *labelSpreading*, sont des techniques d'apprentissage semi-supervisées qui sont utilisées pour la détection de structure dans de grands réseaux complexes. L'implémentation de *labelSpreading* est disponible en ligne dans scikit-learn [3].

LabelSpreading prend en données d'entrée, un petit sous-ensemble de liens annotés à l'aide d'une valeur d'étiquette prédéfinie et un grand ensemble de liens non annotés. La valeur d'étiquette indique la communauté à laquelle les liens appartiennent. La valeur des étiquettes est modifiée en fonction de celles de ses liens voisins. Par conséquent, l'efficacité de cet algorithme pour l'identification des liens de traçabilité repose, non seulement sur le *cluster hypothesis*, mais également sur son fonctionnement qui permet à chaque lien de diffuser progressivement sa valeur d'étiquette à ses voisins jusqu'à ce qu'un état global stable soit atteint [147]. Par exemple, la figure 4.11 montre l'algorithme *LabelSpreading* apprenant une structure interne complexe. Dans le graphe de gauche, le cercle extérieur contient un lien représenté par un point bleu foncé annoté par *outer label* ; le cercle intérieur contient quant à lui un lien représenté par un point bleu clair annoté par *inner label* ; le reste des liens sont non annotés et portent l'étiquette -1. Les points annotés sont bien positionnés dans leur structure respective, ce qui permet à l'algorithme de propager correctement les bonnes valeurs des étiquettes autour des cercles (voir résultat dans le graphe de droite).

Par rapport à d'autres algorithmes, *LabelSpreading* présente des avantages en matière de durée de fonctionnement et de quantité d'informations nécessaires pour apprendre une structure de réseau. De plus, parmi les différentes techniques semi-supervisées, l'algorithme *LabelSpreading* est le plus robuste au bruit dans la classification du texte, c'est-à-dire qu'il est capable de modifier les valeurs des étiquettes des données annotées lors de l'apprentissage [147]. Son principal inconvénient est qu'il produit un ensemble de solutions et non une solution unique.

Le tableau 4.3 récapitule les avantages et les inconvénients des approches d'apprentissage automatique présentées dans cette section.

3. https ://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelSpreading.html

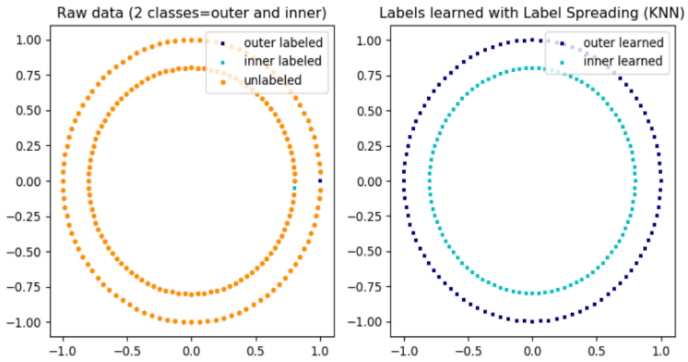

Fig. 4.11 *Illustration de l'algorithme LabelSpreading*
la structure des observations non annotées est cohérente avec la structure des deux classes outer label et inner label, de sorte que l'étiquette de chaque classe peut être propagée par apprentissage à l'ensemble des cercles.

Techniques d'apprentissage automatique	Avantages	Inconvénients
LDA	— Rapide — modélisation des relations entre les thèmes de documents	— manque de mesure objective pour la définition de la meilleure configuration des paramètres — non prise en compte de la dépendance des termes
LabelSpreading	— demande très peu de données d'exemples pour l'apprentissage — possibilité de modifier les valeurs des étiquettes des données d'exemples (robuste)	— proposition d'un ensemble de solutions et non d'une solution unique

Table 4.3 *Récapitulatif des avantages et des inconvénients des techniques d'apprentissage automatique*

4.5 Métriques d'évaluation des approches de traçabilité

L'évaluation des techniques d'identification de la traçabilité repose principalement sur trois métriques : le *rappel*, la *précision* et la *F-mesure*. L'évaluation des techniques de traçabilité se fait à l'aide de seuils dont la valeur est comprise entre 0 et 1 avec une incrémentation par pas de 0.1. Les seuils les plus bas sont compris entre 0 et 0.3 et les seuils de 0.7 à 1 constituent les seuils les plus élevés.

Une technique d'identification des liens de traçabilité atteint son optimum à un seuil spécifique. Le seuil optimal est défini dans ce document comme le seuil où le meilleur compromis entre le rappel et la précision est atteint. En d'autres termes, c'est le seuil où la technique récupère le plus de *vrais liens* tout en récupérant très peu de *faux liens*, c'est-à-dire que le

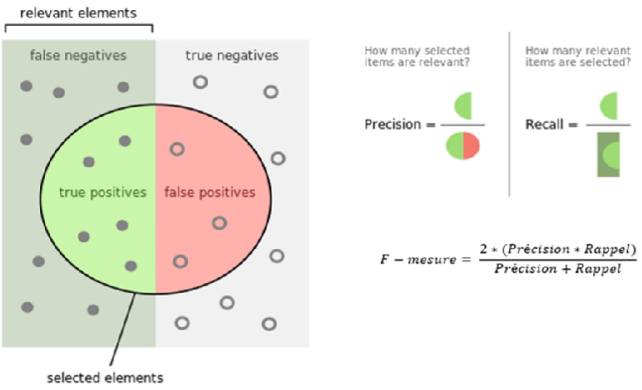

Fig. 4.12 *Métriques de traçabilité [5]*

nombre des *vrais liens* trouvés se rapproche le plus possible du nombre total de vrais liens et le nombre de *faux liens* trouvés tend vers 0.

Les différentes métriques de traçabilité sont définies dans les sous-sections suivantes. La figure 4.12 illustre ces métriques.

4.5.1 Rappel

Le rappel, de l'anglais *Recall*, est la proportion de vrais liens identifiés. Cette métrique permet de définir le nombre de vrais liens retrouvés (True positive) au regard du nombre de vrais liens réellement existant (true positive + False negative). Ainsi, elle permet d'évaluer si les techniques de traçabilité trouvent la totalité des vrais liens. Elle se calcule comme suit :

$$Rappel = \frac{\text{True positive}}{\text{True positive} + \text{False negative}}.$$

4.5.2 Précision

La précision est la proportion de vrais liens (true positive) parmi les liens identifiés (True positive + False positive). Elle mesure l'exactitude de la liste de liens générés par les techniques de traçabilité, c'est-à-dire qu'elle évalue si ces techniques ne font aucune erreur. Elle se calcule comme suit :

$$Précision = \frac{\text{True positive}}{\text{True positive} + \text{False positive}}$$

4.5.3 F-mesure

La F-mesure est la moyenne harmonique de la précision et du rappel. Obtenir une haute précision et un rappel élevé est un exercice d'équilibre car lorsque la précision augmente, le rappel a tendance à diminuer et vice versa. La F-mesure représente cet équilibre. Elle peut donc être utilisée pour évaluer des approches de traçabilité. Elle se calcule comme suit :

$$F - mesure = \frac{2 * (Rappel * Précision)}{Rappel + Précision}$$

Nous utilisons intensivement ces mesures et y ferons référence dans la partie 3 de ce document.

4.6 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art des méthodes de Recherche d'information. Nous avons tout d'abord présenté les techniques fondamentales du traitement des langues. Elles sont utilisées dans la plupart des traitements de texte et permettent de formater les données textuelles. Ensuite les techniques traditionnelles et récentes de recherche d'information ont été présentées. Malgré leurs lacunes, elles fournissent des informations sous forme numérique qui peuvent alimenter des méthodes d'apprentissage automatique. Ceci est aussi vrai pour les techniques récentes de recherche d'information. Dans différentes communautés et notamment dans la communauté de la traçabilité, les techniques récentes de recherche d'information combinées avec les techniques supervisées d'apprentissage automatique ont surpassé de manière significative les performances des approches traditionnelles.

Les principes généraux des méthodes semi-supervisées et l'hypothèse *cluster hypothesis* sur laquelle la plupart de ces techniques se basent ont également été présentés. Le principal avantage des méthodes semi-supervisées est qu'elles utilisent très peu de liens pour l'apprentissage. De plus, l'hypothèse *cluster hypothesis* a déja été vérifiée plusieurs fois dans la communauté de la traçabilité. Il y a donc un intérêt à utiliser de telles techniques pour l'identification des liens de traçabilité.

Des tableaux récapitulant les avantages et les inconvénients de l'ensemble des techniques de traitement automatique, l'ensemble des techniques de Recherche d'Information et l'ensemble des techniques de l'apprentissage automatique étudiées dans ce chapitre ont été fournis.

La partie suivante est consacrée à notre contribution, qui consiste en une approche semi-supervisée combinant différentes techniques de recherche d'information.

Troisième partie

Contributions

Avant-propos

Dans la partie précédente, la définition de la traçabilité retenue a été présentée. Les techniques de traitement du langage sur lesquelles se basent nos travaux ont également été présentées. Une étude des approches de l'état de l'art a été faite selon les quatre problèmes de la traçabilité en entreprise étendue identifiés. Les problématiques de thèse à savoir la *Réduction du coût de l'élicitation des liens de traçabilité* et l'*Identification des types de liens* ont été détaillées. Ces problématiques seront abordées sous le prisme des approches de Recherche d'Information, d'apprentissage automatique et de traitement automatique des langues au travers des deux questions de recherche suivantes :

— Question 1 : Quelles méthodes de traçabilité permettent d'obtenir de bonnes performances sur des jeux de données industriels ?
— Question 2 : Quel est l'apport des méthodes de plongements lexicaux basées sur des modèles neuronaux à la capture de la sémantique des liens de traçabilité ?

Cette partie de thèse intitulée « Contributions » présente les méthodes proposées pour répondre aux problématiques soulevées et détaille les expérimentations et évaluations réalisées. Elle est divisée en deux chapitres, le premier portant sur les méthodes proposées, le framework *ATLaS* implémenté et le cas d'application et le deuxième sur les cas d'étude utilisés et les évaluations réalisées.

— Le chapitre 5 décrit les approches proposées, les quatre modules du framework *ATLaS* et détaille le cas d'application.
— Le chapitre 6 présente les cas d'étude utilisés et les expérimentations et les évaluations effectuées.

5 – ATLaS, un framework d'identification des liens de traçabilité

5.1 Introduction

Dans le contexte de notre travail, nous avons étudié plusieurs approches d'élicitation des liens de traçabilité. Ces dernières introduites dans le chapitre 3 proposent des approches présentant divers inconvénients. On peut citer entre autres : le coût de l'élicitation des liens de traçabilité et de l'identification des types de liens. Pour remédier à cela nous proposons de construire un framework d'identification des liens de traçabilité, appelé « *Aggregation Trace Links Support* » en abrégé *(ATLaS)*, basé sur la combinaison de techniques de Recherche d'Information, de techniques du Traitement Automatique des Langues et de techniques semi-supervisées. Pour cela, nous avons proposé une heuristique permettant, dans un premier temps, de construire une base d'exemples de « *vrais liens* » et de « *faux liens* », puis, dans un deuxième temps, d'alimenter les techniques semi-supervisées à l'aide de cette base d'exemples. En d'autres termes, notre framework *ATLaS* permet d'affecter une mesure de confiance aux liens de traçabilité potentiels et fournir ainsi à l'analyste en phase de validation une mesure quantitative de confiance sur chaque lien candidat afin de l'aider à optimiser son effort de validation des liens de traçabilité.

La suite de ce chapitre est donc organisée comme suit : la section 5.2 donne une présentation générale de l'approche proposée. La section 5.3 présente la méthodologie de recherche suivie durant notre étude. La section 5.4 présente l'architecture logicielle du framework *ATLaS*. Enfin, la section 5.5 présente l'espace collaboratif, les plates-formes de traitement du langage et de l'apprentissage automatique utilisées dans ATLaS.

5.2 Présentation générale de l'approche

En ingénierie système, l'hétérogénéité des artefacts est due à la diversité des métiers qui interviennent dans la conception de systèmes complexes. Par exemple, un modèle décrit un aspect du système et est généralement créé ou dérivé pour un objectif particulier. Les modèles des différents aspects du système sont toutefois rarement manipulés indépendamment les uns des autres. La collaboration entre les parties prenantes repose donc sur un ensemble de processus dans lesquels ils seront amenés à communiquer et à échanger au travers d'outils. À cause de la forte volumétrie des artefacts produits, la synchronisation des informations échangées peut passer par l'identification de similarités entre les artefacts.

La plupart des techniques de recherche d'information utilisées pour l'identification de

liens de traçabilité s'appuient sur des mesures quantitatives de similarité entre des artefacts. Ces mesures peuvent reposer sur des similarités morphologiques entre les termes apparaissant dans les artefacts (LSI, VSM), la proximité sémantiques des termes (word embedding), ou sur la proximité thématique des artefacts (LDA). Ainsi, chaque mesure de similarité fournit une caractérisation ou une description de la paire d'artefacts considérée selon un critère donné. Un vecteur contenant des mesures de similarité peut alors être construit pour toute paire d'artefacts potentiellement liée. Par conséquent, plus les mesures de similarité considérées vont être diverses et complémentaires, plus le vecteur contiendra de l'information nécessaire pour décider qu'un lien existe effectivement entre les artefacts considérés.

Nous nous proposons donc de construire à partir d'une base d'exemples de « *vrais liens* » et de « *faux liens* » de traçabilité, un modèle permettant d'estimer à partir d'un vecteur de descripteurs d'une paire d'artefacts, la probabilité que cette paire d'artefacts soit effectivement liée. Cette approche réalise en quelque sorte une agrégation de plusieurs méthodes complémentaires de traçabilité. Elle se ramène à résoudre un problème de classification binaire, les deux classes à discriminer étant la classe des « *vrais liens* » et celle des « *faux liens* ».

De façon formelle : soient $m_1, m_2, \ldots, m_p$ des mesures de similarité entre paires d'artefacts et p le nombre de mesures de similarité. Pour deux artefacts x et y, $m_j(x,y)_{j=1\ldots p} \in \mathbb{R}$ est un nombre qui indique le degré de similarité entre x et y, pour j allant de $\{1, \ldots, p\}$. Notons V la fonction qui à une paire d'artefacts (x, y) associe son *vecteur de descripteurs* :

$$V(x,y) = (m_1(x,y), m_2(x,y), \ldots, m_p(x,y))$$

.

En supposant que nous disposons d'un ensemble de paires d'artefacts $\{(x_i, y_i)_{i=1,\ldots,n}\}$ et d'un *vecteur d'annotation* $a = (a_1, ..., a_n)$ tel que :

$$\begin{cases} a_i = 0 & \text{si } x_i \text{ et } y_i \text{ ne sont pas liés (« faux » liens),} \\ a_i = 1 & \text{s'il existe un lien de traçabilité entre } x_i \text{ et } y_i \text{ (« vrais » liens),} \\ a_i = -1 & \text{s'il n'y a aucune information connue sur la relation entre } x_i \text{ et } y_i \text{ (lien non annoté).} \end{cases}$$

L'apprentissage consiste alors à calculer une *fonction de prédiction* P de sorte à minimiser le coût de la forme :

$$\sum_{i=1}^{n} L(P(V(x_i, y_i)), a_i)$$

où $L(r, s)$ est une fonction à valeur réelle positive minimale pour $r = s$ [148].

Une fois la fonction de prédiction P calculée, pour un nouveau couple d'artefacts (t, z), le réel $P(V(t, z))$ fournira la probabilité qu'il existe un lien de traçabilité entre t et z. Cette probabilité est notre mesure de confiance.

D'un point de vue métier Ingénierie Systèmes, la mesure de confiance calculée par notre approche fournit à l'analyste en phase de validation un critère de priorisation. Notre approche peut être directement utilisée pour détecter des liens de traçabilité en considérant comme vrais des liens pour lesquels la mesure de confiance fournie par *ATLaS* est supérieure à un seuil fixé par l'utilisateur.

5.3 Méthodologie de recherche

La méthodologie de recherche mise en œuvre pour élaborer notre proposition est expérimentale et est constituée de trois étapes.

La première étape consiste à collecter sur chaque paire d'artefacts les informations caractérisant au maximum leur similarité ou différence. Les techniques étudiées en partie 1 (RI, apprentissage automatique et TAL) produisant des informations de type différent, ceci nous amène à les utiliser conjointement afin que la collecte soit maximale.

À partir de ces informations, la phase suivante va consister à décider si une paire d'artefacts est liée ou non. Pour ce faire, on utilise une technique de classification qui produit pour chaque paire d'artefacts une probabilité, appelée mesure de confiance. Selon cette mesure, la décision de l'existence d'un lien (vrai lien) ou pas (faux lien) (cf. section 3.7) est alors possible. Toutefois, une telle technique se fonde sur un jeu d'exemples.

Ainsi la deuxième étape consiste à élaborer ce jeu d'exemples.

La troisième étape consiste à calculer la mesure de confiance pour décider de l'existence d'un lien ou pas.

Ces trois étapes sont détaillées ci-après.

5.3.1 Étape 1 : Collecte d'informations sur des paires d'artefacts

L'hypothèse de cette étape est qu'une collecte utilisant conjointement des techniques multi-domaines maximise le type d'informations obtenues.

En effet, l'étude de techniques utilisées en traçabilité menée dans l'état de l'art (cf. section 3.5), nous permet d'identifier que certaines d'entre elles sont fondées sur le même type de calcul. Par exemple, VSM, LSI, et LDA sont trois techniques qui calculent le nombre d'occurrences des termes. Toutefois, les informations produites ne sont pas forcément les mêmes d'une technique à l'autre. Ainsi, utiliser ces trois techniques plutôt qu'une seule d'entre elles permet d'obtenir un volume d'informations plus élevé, car résultant de l'agrégation des informations produites par chacune d'elles.

Pour autant, ces techniques capturent le même type d'informations, i.e., mesurent le nombre de termes communs entre artefacts, sans capturer leur sens (la sémantique).

A contrario, des techniques telles que Word Embedding et Sentence Embedding permettent de capturer la sémantique via la construction de dictionnaires contextuels (cf. section 4.2.2).

Afin d'enrichir l'ensemble de mesures agrégat des techniques VSM, LSI et LDA en apportant les aspects sémantiques qu'elles n'intègrent pas, nous construisons de tels dictionnaires et les utilisons pour définir trois scores de similarité S1, S2 et S3 [1].

Ainsi l'utilisation conjointe de plusieurs techniques issues de différents domaines, que nous dénommons par la suite VSM-LSI-LDA-S1-S2-S3, maximise l'information nécessaire à la détermination des liens.

En résumé, cette étape consiste à calculer les corrélations croisées des mesures produites par chaque technique VSM, LSI et LDA et des scores de similarité. Son expérimentation, décrite en section 6.3 du chapitre 6, permet d'étayer l'idée que des techniques multi-domaines apportent une complémentarité des informations qu'elles produisent.

1. La construction des dictionnaires contextuels et le calcul des scores de similarité sont décrits en détail dans la section 5.4.2

5.3.2 Étape 2 : Élaboration d'un jeu d'exemples

La classification de l'étape 3 nécessitant un jeu d'exemples, cette étape consiste à construire un ensemble réduit de liens annotés vrais ou faux.

Ainsi nous commençons par extraire de l'ensemble de l'étape 1 les mesures que chaque technique VSM, LSI et LDA a produit isolément (Figure 5.1). Puis nous appliquons sur ces trois ensembles de mesures une heuristique fondée sur l'hypothèse que deux artefacts ayant un nombre important de termes communs ont de fortes chances d'être liés. À l'inverse, deux artefacts ayant peu de termes communs ont moins de chances d'être liés. Ainsi nous définissons l'heuristique comme suit :

— chaque ensemble de mesures VSM, LSI et LDA est trié par ordre de similarité ;
— dans chacun des trois ensembles, seules les mesures de similarité les plus élevées et les plus basses sont sélectionnées. Le pourcentage de mesures sélectionnées est un paramètre de l'heuristique (par exemple, il est de 10% sur la Figure 5.1) ;
— l'intersection des mesures de similarité les plus élevées de chaque ensemble est déclarée constituer l'ensemble des « vrais » liens ;
— l'union des mesures de similarité les plus basses de chaque ensemble est déclarée constituer l'ensemble des « faux » liens.

Le jeu d'exemples est ainsi constitué des paires d'artefacts annotées comme étant liées (« vrai » lien) ou pas (« faux » lien). Les paires d'artefacts dont les mesures de similarité sont absentes de l'intersection ou de l'union ne sont pas dans le jeu d'exemples.

La figure 5.1 illustre cette heuristique.

En faisant une intersection des mesures de similarité les plus élevées et une union des mesures de similarité les plus basses, les paires d'artefacts sont sélectionnées selon un critère plus strict, afin d'en renforcer la fiabilité. Ce choix repose de manière plus fondamentale sur la « *Cluster Hypothesis* » (cf. section 4.4.1). Cette hypothèse a été exploitée de façon prolifique par différents auteurs s'appuyant également sur les techniques VSM, LSI, LDA [94, 144, 145]. En effet, les paires d'artefacts considérées comme liées (vrais liens) sont concentrées dans la région des fortes valeurs de similarité pour les trois techniques, tandis que les paires d'artefacts considérées comme non liées (faux liens) sont potentiellement plus dispersées. Cette interprétation de la « *Cluster Hypothesis* » dans la communauté de la traçabilité (cf. section 4.4.1) est une restriction de l'interprétation de cette dernière dans la communauté de l'apprentissage automatique, car elle ne met l'accent que sur l'homogénéité locale de la classe des paires d'artefacts liées (vrais liens).

L'expérimentation de cette étape 2, décrite en section 6.4, a consisté à évaluer cette heuristique sur des cas d'étude pour lesquels les liens valides sont connus, i.e., l'étiquetage en vrais liens et faux liens est disponible. Ces cas d'étude sont disponibles sur le site internet COEST [2] (Center of Excellence for Software & Systems Traceability) et le site internet d'ARC-IT [3](Architecture Reference for Cooperative and Intelligent Transportation).

Les résultats obtenus permettent de valider l'heuristique pour les faux liens. Par contre, l'évaluation de l'heuristique pour les vrais liens n'est pas probante. Néanmoins, l'objectif ici est de construire le jeu d'exemples utilisé par la classification. Ainsi, même si le nombre de vrais liens est peu élevé, identifier au maximum les faux liens est donc fondamental dans la

2. Center of Excellence for Software & Systems Traceability, http ://www.coest.org
3. Architecture Reference for Cooperative and Intelligent Transportation https ://local.iteris.com/arc-it/index.html

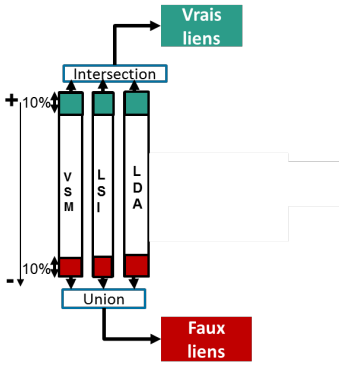

Fig. 5.1 *Présentation de l'heuristique*

détermination du résultat final de la classification.

5.3.3 Étape 3 : Classification des liens et mesure de confiance

Le jeu d'exemples de l'étape précédente constitue une faible fraction de l'ensemble des liens possibles (de l'ordre de 20%) dans nos expérimentations numériques. Aussi nous optons pour une approche de classification semi-supervisée qui nécessite un nombre réduit de données annotées pour faire l'apprentissage.

Nous nous fondons sur la « *cluster hypothesis* » telle qu' interprétée dans la communauté de l'apprentissage automatique (cf. section 4.4.1). Dans cette communauté, cette hypothèse repose sur l'homogénéité locale des classes (vrais liens, faux liens) dans un espace de descripteurs. Ainsi, dans l'espace des mesures et des scores de similarité introduits à la première étape, une paire d'artefacts étiquetée comme « vrai lien » ou comme « faux lien » a probablement pour plus proches voisins des paires d'artefacts ayant la même étiquette.

Il existe plusieurs algorithmes d'apprentissage semi-supervisé reposant sur la « *cluster hypothesis* » [102]. Considérant que les étiquettes de référence obtenues à la deuxième étape sont possiblement entachées d'erreurs, nous avons retenu l'algorithme d'apprentissage semi-supervisé dénommé *labelSpreading*, en raison de sa capacité à modifier et donc corriger les étiquettes initialement fournies.

Les expérimentations de cette étape, décrite en section 6.5, ont consisté d'une part en la vérification de la « *cluster hypothesis* » pour les deux classes (vrais liens et faux liens) et d'autre part en l'évaluation du taux de bonne classification du modèle construit lors de l'apprentissage. Cette dernière s'appuie sur des métriques fondées sur le nombre de liens trouvés, i.e., le rappel, la précision et la F-mesure.

5.4 Architecture du framework ATLaS

Pour mettre en œuvre cette méthodologie, nous avons défini l'architecture logicielle d'un framework que nous appelons ATLaS. Elle est composée de plusieurs modules. Ceux-ci réalisent les calculs effectués dans les différentes étapes de la méthodologie. Le module 2 "*Calcul des mesures et des scores de similarités*" réalise les calculs effectués dans l'étape 1 et le module 3 "*Calcul de la mesure de confiance*" réalise les calculs effectués dans les étapes 2 et

3.

ATLaS prend en entrée des exigences décrites en langage naturel et des modèles exprimés sous forme textuelle (artefacts). Il génère en sortie une liste de liens de traçabilité avec leurs mesures de confiance et leur type (Figure 5.2).

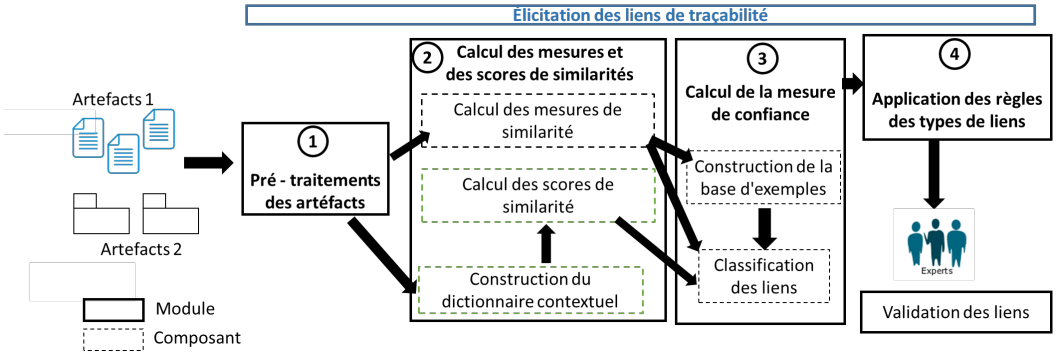

Fig. 5.2 *Architecture d'ATLaS (Aggregation Trace Links Support)*

1. Dans le module 1 "*Pré-traitement des artefacts*", les artefacts sont transformés en sacs de mots (Bags of Words - BoW) et syntagmes.
2. Dans le module 2 "*Calcul des mesures et des scores de similarités*", les sacs précédemment produits sont utilisés pour calculer les mesures et les scores de similarités entre les artefacts. Ces mesures et ces scores constituent le *vecteur de descripteurs* de chaque paire d'artefacts. L'ensemble de ces vecteurs forme la *matrice de descripteurs*. Ainsi, ce module est composé de trois composants : le composant "*Calcul des mesures de similarité*", le composant "*Calcul des scores de similarité*", et le composant "*Construction du dictionnaire contextuel*".
3. Le module 3 "*Calcul de la mesure de confiance*", intègre l'heuristique définie pour construire un petit jeu d'entrainement. Celui-ci associé à la *matrice de descripteurs* permet de calculer une *fonction de prédiction* (ou un *modèle prédictif*) qui fournira une mesure de confiance déterminant la probabilité qu'une paire d'artefacts soit liée (« vrai ») ou non (« faux »). Ce module est composé de deux composants : le composant "*Construction de la base d'exemples*" et le composant "*Classification des liens*".
4. Dans le module 4 "*Identification des type de liens*", les définitions formelles des types de liens sont appliquées sur la liste des liens considérés comme vrais.

Notre approche sera illustrée avec le même exemple paire d'artefacts *Exigence 1245 - But 113* utilisé dans le chapitre 4. Cet exemple est tiré du cas d'étude HIPAA qui est l'un des jeux de données académiques utilisés pour l'évaluation d'*ATLaS* dans le chapitre 6 Évaluation. Il est composé de l'exigence *1245* dont la description est « *System will implement session timeouts and use cookies to terminate an electronic session* » et du but *113* « *Automatic logoff Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity* ».

Les 4 modules du framework *ATLaS* sont détaillés dans les sections suivantes.

5.4.1 Module1 : Pré-traitement des artefacts

L'objectif de ce module est de convertir les artefacts exprimés en langage naturel afin de faciliter le calcul des techniques de Recherche d'Information et de Traitement Automatique des Langues. Pour ce faire, il convient d'extraire le texte des artefacts et d'effectuer quelques opérations de base du traitement du langage pour les diviser en sacs de mots et syntagmes. Ces opérations sont indispensables pour construire des corpus de texte utilisables.

Les sous-sections suivantes et la figure 5.3 détaillent ces opérations.

a) Segmentation en phrases et retrait des mots vides

Les artefacts sont segmentés en phrases. Ensuite, les « *mots vides* (stopword) » [115], qui sont des termes qui se répètent fréquemment dans un document de telle sorte qu'ils ne sont plus pertinents pour l'identification des liens comme les articles "*a*" ou "*the*", sont retirés. Le résultat de ces opérations est envoyé aux opérations de séparation des termes et de séparation en syntagmes.

b) Séparation des termes

Les "*token*" ou termes sont obtenus par découpage des phrases. Les ponctuations et les chiffres sont ensuite supprimés.

Notons que cette opération se fait différemment pour les modèles et les exigences. Dans le cas des modèles, les termes composés sont divisés suivant la règle "*camel case*". Cette règle consiste à écrire un ensemble de termes en les liant sans espace ni ponctuation, et en mettant en capitale la première lettre de chaque terme. Par exemple, "GetForecastHistory" est divisé en "Get", "Forecast" et "History".

c) Séparation en syntagmes

Le "*Text chunking*", ou la séparation en syntagmes, est une technique qui consiste à décomposer les phrases en segments indépendants [149]. Elle est utilisée pour détecter les syntagmes nominaux et verbaux dans les phrases. Seuls ces types de syntagmes sont considérés car ils constituent les principaux éléments qui donnent le sens aux phrases. Par exemple, l'application de cette opération sur l'exigence "*1245*" avec la description "*System will implement session timeouts and use cookies to terminate an electronic session.*" donne les syntagmes nominaux et verbaux "*System*, "*will implement*", "*session timeouts*", "*use cookies to terminate*" et "*electronic session*".

Au terme du pré-traitement, chaque artefact est transformé en sac de mots et en sac de syntagmes. Ceux-ci sont envoyés dans le module 2 au composant *Construction du dictionnaire contextuel*. Dans le même temps, les sacs de mots sont également envoyés au composant *Calcul des mesures et des scores de similarités* (Figure 5.3).

5.4.2 Module 2 : Calcul des mesures et des scores de similarités

Ce module prend en entrée des sacs de mots, des syntagmes nominaux et verbaux puis il calcule les mesures et des scores de similarités.

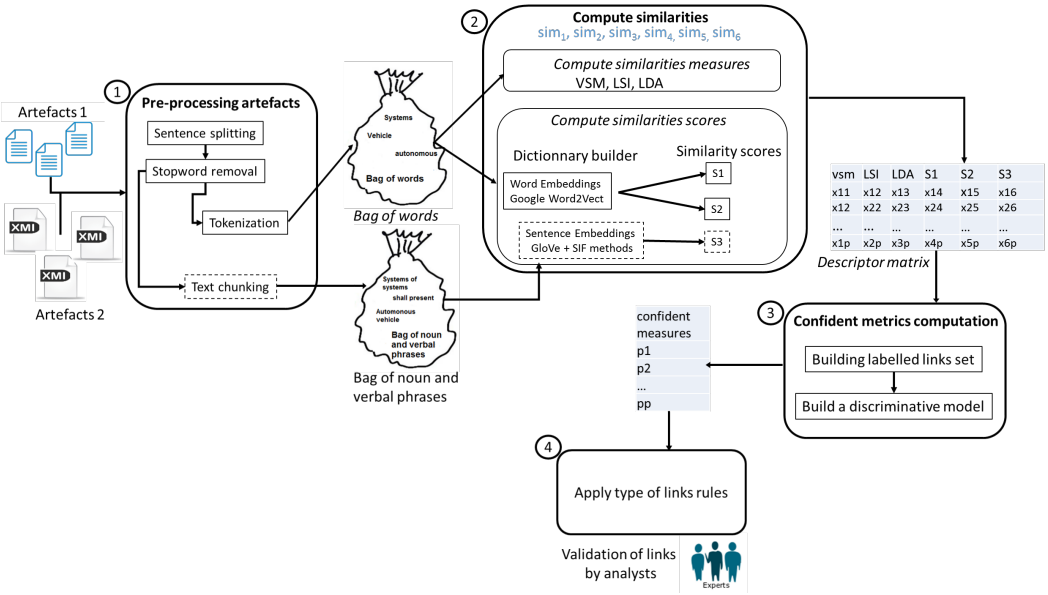

Fig. 5.3 *Fonctionnement d'ATLaS avec la combinaison* VSM-LSI-LDA-S1-S2-S3

Un vecteur contenant les mesures de similarité est construit pour chaque paire d'artefacts. Ces vecteurs représentent les lignes de la *matrice de descripteurs*. Cette dernière est la sortie ce module.

a) Calcul des mesures de similarité

Pour chaque paire d'artefacts, les mesures de similarités des techniques *LSI*, *LDA* et *VSM* sont calculées. Ces mesures seront utilisées pour construire la base d'exemples. Elles sont également positionnées sur la matrice des descripteurs.

b) Calcul des scores de similarité

Des dictionnaires contextuels construits à partir des techniques de *Word* et *sentence Embedding* ont été utilisés pour définir trois scores de similarité.

La construction des dictionnaires contextuels et le calcul des scores de similarité sont décrits en détail dans les sous-sections suivantes.

b).1 Construction du dictionnaire contextuel Cette activité permet de déterminer les représentations vectorielles des termes et des syntagmes contenus dans les artefacts en fonction de leur signification contextuelle. Les sacs de mots et de syntagmes sont les données d'entrée de ce composant. Le résultat de cette activité est une liste de paires de termes et de syntagmes synonymes.

Apprendre le plongement de mots ou *Word Embeddings* consiste à rechercher des représentations vectorielles de termes ayant une signification similaire à un terme donné et à associer ces représentations à ce dernier [124]. De nombreuses listes de termes avec leurs représentations vectorielles, appelées modèles pré-entraînés, sont disponibles publiquement. Parmi ces modèles pré-entraînés, nous avons utilisé *Google Word2Vec* model [98], qui est

populaire pour sa simplicité et son efficacité (cf. section 4.2.2). Il a été formé sur environ 100 milliards de termes de *Google News* et sa taille de vocabulaire est de 3 millions d'entités.

Outre *Google Word2Vec* model, nous avons également utilisé le modèle pré-entraîné *GloVe* [4] (cf. section 4.2.2). Comme le modèle Google *Word2Vec*, le modèle *GloVe* est très efficace pour formuler des tâches d'analogie et de similarité. Ce modèle est intégré dans la méthode *Smooth Inverse Frequency (SIF)* d'Arora et al. [131] qui permet de calculer le plongement des syntagmes nominaux et verbaux (cf. section 4.2.3).

Le modèle pré-entraîné *Google Word2Vec* et la méthode *SIF* sont utilisés pour déterminer les dictionnaires contextuels des termes et des syntagmes nominaux et verbaux afin d'identifier les paires de termes et de syntagmes synonymes.

b).2 Présentation des scores de similarité Avec cette liste de paires de termes et de syntagmes synonymes, nous avons défini trois scores de similarité.

— Le premier score de similarité S_1 est la mesure de similarité de la méthode "*Naive Satisfaction Method* [150, 151], l'un des scores les plus populaires dans la communauté de traçabilité. Comme la plupart des scores de similarité utilisés dans l'analyse des textes et la traçabilité, S_1 met l'accent sur les termes communs partagés par des paires d'artefacts.

— Le deuxième score de similarité S_2 est similaire au premier (S_1) mais se positionne au niveau d'un syntagme. Il calcule le rapport des syntagmes nominaux et verbaux communs entre les artefacts, y compris les synonymes.

— Enfin, le troisième score, S_3, est également calculé au niveau des termes comme S_1, mais il se calcule après avoir filtré certains termes considérés comme non pertinents.

En effet, dans une collection de textes, certains termes peuvent respecter une distribution uniforme. Ils peuvent alors être considérés comme des *mots vides* pour cette collection bien qu'ils n'en soient pas généralement. Par exemple, dans le domaine de l'ingénierie systèmes, des termes comme "*System*" ou "*shall*" sont récurrents dans les exigences et donc, leur distribution uniforme fait d'eux des termes non pertinents en ce qui concerne la traçabilité des exigences. Ces termes peuvent donc donner lieu à de nombreux *faux liens* s'ils sont pris en compte. Nous supposons alors que ces termes sont plus fréquents que ceux qui sont réellement utiles pour identifier les *vrais liens*.

A partir de cette hypothèse, la fréquence de chaque terme peut alors être calculée dans le corpus et les termes avec une fréquence supérieure à un seuil choisi sont filtrés. Un nouveau score de similarité S_3 peut ainsi être défini comme le rapport entre les termes sémantiquement importants (c.-à-d. les termes qui n'ont pas été filtrés) entre deux artefacts.

Ces scores de similarité sont calculés à l'aide des équations suivantes :

$$S_1 = \frac{N_{common}}{N_{TotalWords}}.$$

$$S_2 = \frac{N_{commonNounAndVerbalPhrases}}{N_{TotalPhrases}}.$$

$$S_3 = \frac{N_{commonImportantWords}}{N_{TotalWords}}.$$

Avec

— N_{common} le nombre de termes communs, y compris les synonymes,

— $N_{TotalWords}$ le nombre total de termes des deux artefacts,
— $N_{commonNounAndVerbalPhrases}$ le nombre de syntagmes nominaux et verbaux identiques et leurs synonymes entre deux artefacts,
— $N_{TotalPhrases}$ le nombre total syntagmes nominaux et verbaux des deux artefacts,
— et $N_{commonImportantWords}$ le nombre de termes importants communs entre deux artefacts, y compris les synonymes.

Ainsi, tous les scores de similarité (S_1, S_2, S_3) décrits ci-dessus et les mesures de similarité calculées avec LSI, LDA et VSM fournissent un ensemble d'informations pour chaque paire d'artefacts. Le vecteur de descripteur est donc composé de 6 composantes pour la combinaison *VSM-LSI-LDA-S1-S2-S3* conformément à la méthodologie présentée à la section 5.2.

Chaque paire d'artefacts est décrite par un *vecteur de descripteurs*. Ainsi, pour la combinaison *VSM-LSI-LDA-S1-S2-S3*, une paire d'artefacts (x, y) avec (m_{S_1}, m_{S_2}, m_{S_3}, m_{LSI}, m_{LDA}, m_{VSM}), les 3 scores de similarité et les 3 mesures de similarité fournies par les techniques VSM, LSI, LDA, le vecteur descripteurs associé est égal à :

$$(m_{S_1}(\text{x, y}),\ m_{S_2}(\text{x, y}),\ m_{S_3}(\text{x, y}),\ m_{LSI}(\text{x, y}),\ m_{LDA}(\text{x, y}),\ m_{VSM}(\text{x, y})).$$

Dans notre exemple, le vecteur de descripteurs de la paire d'artefacts *Exigence 1245* et le *But 113* avec la combinaison *VSM-LSI-LDA-S1-S2-S3* est $((0.23)_{S_1}$, $(0.18)_{S_2}$, $(0.2)_{S_3}$, $(0.96)_{LSI}$, $(0.99)_{LDA}$, $(0.21)_{VSM})$.

5.4.3 Module 3 : Calcul de la mesure de confiance

Nous utilisons la matrice de descripteurs comme entrée du module 3. L'idée sous-jacente ici étant que chaque vecteur de cette matrice puisse contenir suffisamment d'informations pour décider si un lien existe ou non entre deux artefacts donnés. Ce module est composé de deux composants : le composant "*Construction de la base d'exemples*" et "*Classification des liens*".

Les sous-sections suivantes décrivent ces composants.

a) Construction de la base d'exemples

Nous avons mis en place une heuristique pour annoter certaines paires d'artefacts comme étant liées ou non. Cette heuristique est décrite dans la section 5.3.2 et est implantée par ce composant. Il prend en entrée les mesures de similarité fournies par les techniques VSM, LSI, et LDA. Il fournit en sortie la base des paires d'artefacts liées (« vrais » liens) ou non (« faux » liens).

Par exemple, supposons qu'une paire d'artefacts *Exigence A - But B* a les mesures de similarité (0.29), (0.02) et (0.46) fournies respectivement par les techniques VSM, LSI, LDA. Cette paire d'artefacts sera donc considérée comme n'étant pas liée à cause de la faible valeur de la mesure de similarité fournie par LSI. Elle sera donc intégrée à la liste des paires d'artefacts considérées comme non liées (« faux » liens).

b) Classification des liens

Ce composant a pour objectif de construire un modèle prédictif. Celui-ci permet de trouver les structures communautaires (vrais liens, faux liens) afin de les regrouper en clusters

de vrais et de faux liens. Il prend en entrée la matrice de descripteurs issue de l'étape 1 et la base d'exemples issue de l'étape 2. En sortie, le modèle prédictif fournit une probabilité qui définit le degré d'appartenance à la communauté de vrais liens. Cette probabilité constitue alors la mesure de confiance sur le lien de traçabilité entre deux artefacts. Les liens ayant une mesure de confiance supérieure ou égale à un seuil fixé par l'utilisateur sont considérés comme vrais liens, tout le reste étant marqué comme faux liens.

Reprenons la paire d'artefacts composée de l'exigence *1245* et du But *113*, la probabilité que cette paire d'artefacts soit liée est de 55%, soit une mesure de confiance égal à 0.55.

5.4.4 Module 4 : Identification des types de liens

Ce module prend en entrée les définitions des types de liens et la liste des vrais liens associés à leur mesure de confiance générées par le module 3. L'ensemble des règles formelles des types de liens est appliqué sur chaque lien afin de leur attribuer leur type. Il est important de noter que les règles d'identification des types de liens reposent fortement sur la nature des artefacts.

Rappelons que les types de liens qui nous intéressent dans nos travaux, sont les types de liens existant entre les exigences et les modèles, plus précisément les types de liens *exigence-exigence, modèle-modèle* et *exigence-modèle*. Ces liens sont les liens de satisfaction, les liens de recouvrement et les liens de raffinement. Ils ont été présentés en détail dans la section 2.4 du chapitre 2.

Les sous-sections suivantes présentent comment les règles d'identification des liens sont appliquées pour chaque type de liens.

a) Liens de satisfaction

Ce lien est plus étudié que les autres dans la littérature. De plus, différents auteurs s'accordent sur sa définition et sur ses règles d'identification, ce qui n'est pas le cas pour tous les autres types de liens.

Les définitions et les règles d'identification des types de liens ont été détaillées dans la section 2.4 du chapitre 2. Ces règles d'identification des types de liens se basent sur la nature des artefacts et sur les termes ou syntagmes contenus dans ceux-ci. Rappelons les règles d'identification du lien de satisfaction avec les termes et les syntagmes respectivement :

Étant donné une exigence R, divisée en termes uniques
$R = \{term_{R_1}, term_{R_2}, \ldots, term_{R_{|R|}}\}$,
et un élément de modèle Elt, divisé en termes uniques
$Elt = \{term_{Elt_1}, term_{Elt_2}, \ldots, term_{Elt_{|Elt|}}\}$,
un lien de satisfaction ($\xrightarrow[Satisfaction]{}$) existe entre R et Elt s'il respecte la condition :
$Elt \xrightarrow[Satisfaction]{} R \; si \; \exists \{term_{R_i}, term_{Elt_j}\} \mid term_{R_i} \in R, \; term_{Elt_j} \in Elt \wedge term_{Elt_j} \xrightarrow[Correlé]{} term_{R_i}$
Avec $i \in \{1, \ldots |R|\}$ et $j \in \{1, \ldots, |Elt|\}$, et
$\xrightarrow[Correlé]{} \triangleq synonyme | \; hyponyme | \; hierarchie \; conceptuelle$

Étant donné une exigence R, divisée en syntagmes nominaux ou verbaux uniques $R = \{chunk_{R_1}, chunk_{R_2}, \ldots, chunk_{R_{|R|}}\}$,
et un élément de modèle Elt, divisé en syntagmes nominales ou verbales uniques $Elt = \{chunk_{Elt_1}, chunk_{Elt_2}, \ldots, chunk_{Elt_{|Elt|}}\}$,

un lien de satisfaction ($\xrightarrow[Satisfaction]{}$) existe entre R et Elt s'il respecte la condition :

$Elt \xrightarrow[Satisfaction]{} R$ si $\exists$ $\{chunk_{R_i}, chunk_{Elt_j}\}$ | $chunk_{R_i} \in R$, $chunk_{Elt_j} \in Elt$ $\wedge$ $chunk_{Elt_j} \xrightarrow[Correlé]{} chunk_{R_i}$

Avec $i \in \{1, \ldots |R|\}$ et $j \in \{1, \ldots, |Elt|\}$, et

$\xrightarrow{Correlé} \triangleq synonyme|\ hyponyme|\ hierarchie\ conceptuelle$

Afin de conserver la sémantique capturée et le filtre des *faux liens* réalisé, cette règle dans le framework *ATLaS* doit être exprimée à partir des mesures et des scores de similarité définis dans ce dernier. Notons qu'elles se basent sur la similarité entre les termes et les syntagmes des artefacts.

Le type de lien n'est attribué qu'aux liens considérés comme vrais, c'est-à-dire les liens ayant une mesure de confiance supérieure ou égale au seuil fixé par l'utilisateur. Pour la combinaison *VSM-LSI-LDA-S1-S2-S3*, où les termes et les syntagmes sont considérés, les deux règles précédentes sont réécrites de la façon suivante :

Soit **R** une exigence et **Elt** un élément de modèle

Soit S_1**,** S_2**,** S_3 les scores de similarité entre R et Elt

Soit **seuil** le seuil fixé par l'utilisateur

Si $MesureDeConfiance(R, Elt) >=$ **seuil** et $S_1(R, Elt) > 0$ ou $S_2(R, Elt) > 0$ ou $S_3(R, Elt) > 0 \rightarrow$ Elt satisfait R.

b) Liens de recouvrement

Ce lien, comme le lien de raffinement, est très peu étudié dans la littérature. Rappelons les règles d'identification de ce type de lien pour les liens *exigence-exigence* et *modèle-modèle* respectivement :

Étant donné deux exigence R1 et R2, divisées en termes uniques

$R1 = \{term_{R1_1}, term_{R1_2}, \ldots, term_{R1_{|R|}}\}$,

et $R2 = \{term_{R2_1}, term_{R2_2}, \ldots, term_{R2_{|R|}}\}$,

un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre R1 et R2 s'il respecte la condition :

$R2 \xrightarrow[Recouvrement]{} R1$ si $\exists$ $\{term_{R1_i}, term_{R2_j}\}$ | $term_{R1_i} \in R1$, $term_{R2_j} \in R2$ $\wedge$ $term_{R2_j} \xrightarrow[identique]{} term_{R1_i} \vee term_{R2_j} \xrightarrow[sémantiquement\ proche]{} term_{R1_i}$

Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$, et

$\xrightarrow{sémantiquement\ proche} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

Étant donné deux exigences R1 et R2, divisées en syntagmes nominaux ou verbaux uniques

$R1 = \{chunk_{R1_1}, chunk_{R1_2}, \ldots, chunk_{R1_{|R|}}\}$,

et $R2 = \{chunk_{R2_1}, chunk_{R2_2}, \ldots, chunk_{R2_{|R|}}\}$,

un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre R1 et R2 s'il respecte la condition :

$R2 \xrightarrow[Recouvrement]{} R1$ si $\exists$ $\{chunk_{R1_i}, chunk_{R2_j}\}$ | $chunk_{R1_i} \in R1$, $chunk_{R2_j} \in R2$ $\wedge$ $chunk_{R2_j} \xrightarrow[identique]{} chunk_{R1_i} \vee chunk_{R2_j} \xrightarrow[sémantiquement\ proche]{} chunk_{R1_i}$

Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$, et

$\xrightarrow{sémantiquement\ proche} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en termes uniques
$Elt1 = \{term_{Elt1_1}, term_{Elt1_2}, \ldots, term_{Elt1_{|Elt1|}}\}$,
et $Elt2 = \{term_{Elt2_1}, term_{Elt2_2}, \ldots, term_{Elt2_{|Elt2|}}\}$,
un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :
$Elt2 \xrightarrow[Recouvrement]{} Elt1$ *si* $\exists \{term_{Elt1_i}, term_{Elt2_j}\} \mid term_{Elt1_i} \in Elt1,\ term_{Elt2_j} \in Elt2 \ \wedge\ term_{Elt2_j} \xrightarrow[identique]{} term_{Elt1_i} \vee term_{Elt2_j} \xrightarrow[sémantiquement\ proche]{} term_{Elt1_i}$
Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$, et
$\xrightarrow[sémantiquement\ proche]{} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en syntagmes nominaux ou verbaux uniques
$Elt1 = \{chunk_{Elt1_1}, chunk_{Elt1_2}, \ldots, chunk_{Elt1_{|Elt1|}}\}$,
et $Elt2 = \{chunk_{Elt2_1}, chunk_{Elt2_2}, \ldots, chunk_{Elt2_{|Elt2|}}\}$,
un lien de recouvrement ($\xrightarrow[Recouvrement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :
$Elt2 \xrightarrow[Recouvrement]{} Elt1$ *si* $\exists \{chunk_{Elt1_i}, chunk_{Elt2_j}\} \mid chunk_{Elt1_i} \in Elt1,\ chunk_{Elt2_j} \in Elt2 \wedge chunk_{Elt2_j} \xrightarrow[identique]{} chunk_{Elt1_i} \vee chunk_{Elt2_j} \xrightarrow[sémantiquement\ proche]{} chunk_{Elt1_i}$
Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$, et
$\xrightarrow[sémantiquement\ proche]{} \triangleq synonyme|\ hyponyme|\ antonymie|\ troponymie$

Comme pour le lien de satisfaction, ces règles sont exprimées en fonction des mesures et des scores de similarité définis dans le framework *ATLaS*. Ce qui donne respectivement pour les liens *exigence-exigence* et *modèle-modèle* :

Soit **R1** et **R2** deux exigences
Soit S_1, S_2, S_3 les scores de similarité entre R1 et R2
Soit **seuil** le seuil fixé par l'utilisateur
Si $MesureDeConfiance(R1, R2) >=$ **seuil** et $S_1(R1, R2) > 0$ ou $S_2(R1, R2) > 0$ ou $S_3(R1, R2) > 0 \rightarrow$ R2 est en recouvrement avec R1 ;

Soit **Elt1** et **Elt2** deux éléments de modèle
Soit S_1, S_2, S_3 les scores de similarité entre Elt1 et Elt2
Soit **seuil** le seuil fixé par l'utilisateur
Si $MesureDeConfiance(Elt1, Elt2) >=$ **seuil** et $S_1(Elt1, Elt2) > 0$ ou $S_2(Elt1, Elt2) > 0$ ou $S_3(Elt1, Elt2) > 0 \rightarrow$ Elt2 est en recouvrement avec Elt1.

Dans cette configuration, il peut arriver qu'un lien considéré comme vrai ne respecte pas l'ensemble des conditions définies, ce lien est alors marqué par le type « *Unknow* ». Il peut alors être soit retiré de la liste des vrais liens par l'utilisateur, soit ce dernier peut lui attribuer un autre type lors de la validation. Cette remarque est également vraie pour le lien de raffinement.

c) Liens de raffinement

Ce lien est complexe à définir et à identifier dans les communautés de la traçabilité et des modèles. Toutefois, les règles d'identification de ce lien retenues dans cette thèse sont respectivement pour les liens *exigence-exigence* et *modèle-modèle* :

Étant donné deux exigences R1 et R2, divisées en syntagmes nominaux ou verbaux uniques
$R1 = \{chunk_{R1_1}, chunk_{R1_2}, \ldots, chunk_{R1_{|R1|}}\}$,
et $R2 = \{chunk_{R2_1}, chunk_{R2_2}, \ldots, chunk_{R2_{|R2|}}\}$,
un lien de raffinement ($\xrightarrow[Raffinement]{}$) existe entre R1 et R2 s'il respecte la condition :
$R2 \xrightarrow[Raffinement]{} R1$ *si* $\exists\ \{chunk_{R1_i}, chunk_{R2_j}\} \mid chunk_{R1_i} \in R1,\ chunk_{R2_j} \in R2 \wedge chunk_{R2_j} \xrightarrow[Recouvrement]{} chunk_{R1_i} \wedge chunk_{R2_j in\{syntagmes\ avec\ condition\ temporelle\}}$
Avec $i \in \{1, \ldots |R1|\}$ et $j \in \{1, \ldots, |R2|\}$

Étant donné deux éléments de modèle Elt1 et Elt2, divisés en syntagmes nominaux ou verbaux uniques
$Elt1 = \{chunk_{Elt1_1}, chunk_{Elt1_2}, \ldots, chunk_{Elt1_{|R|}}\}$,
et $Elt2 = \{chunk_{Elt2_1}, chunk_{Elt2_2}, \ldots, chunk_{Elt2_{|Elt|}}\}$,
un lien de raffinement ($\xrightarrow[Raffinement]{}$) existe entre Elt1 et Elt2 s'il respecte la condition :
$Elt2 \xrightarrow[Raffinement]{} Elt1$ *si* $\exists\ \{chunk_{Elt1_i}, chunk_{Elt2_j}\} \mid chunk_{Elt1_i} \in Elt1,\ chunk_{Elt2_j} \in Elt2 \wedge chunk_{Elt2_j} \xrightarrow[Recouvrement]{} chunk_{Elt1_i} \wedge chunk_{Elt2_j in\{syntagmes\ avec\ condition\ temporelle\}}$
Avec $i \in \{1, \ldots |Elt1|\}$ et $j \in \{1, \ldots, |Elt2|\}$

Ce lien est également exprimé à l'aide des mesures et des scores de similarité. Les règles appliquées sur les liens *exigence-exigence* et *modèle-modèle* sont respectivement :

Soit **R1** et **R2** deux exigences
Soit S_1**,** S_2**,** S_3 les scores de similarité entre R1 et R2
Soit **seuil** le seuil fixé par l'utilisateur
Si $MesureDeConfiance(R1, R2) >=$ **seuil** et $(S_1(R1, R2) > 0$ ou $S_2(R1, R2) > 0$ ou $S_3(R1, R2) > 0)$ et $\exists$**ph** $\in$ **R2** / **ph** $\in$ liste{ *'during', 'after', 'before', 'soon', 'when', 'if', ...*} $\rightarrow$ R2 raffine R1 ;

Soit **Elt1** et **Elt2** deux éléments de modèle
Soit S_1**,** S_2**,** S_3 les scores de similarité entre Elt1 et Elt2
Soit **seuil** le seuil fixé par l'utilisateur
Si $MesureDeConfiance(Elt1, Elt2) >=$ **seuil** et $(S_1(Elt1, Elt2) > 0$ ou $S_2(Elt1, Elt2) > 0$ ou $S_3(Elt1, Elt2) > 0)$ et $\exists$**ph** $\in$ **Elt2** / **ph** $\in$ liste{ *'during', 'after', 'before', 'soon', 'when', 'if', ...*} $\rightarrow$ Elt2 raffine Elt1.

Le framework *ATLaS* génère ainsi des liens de traçabilité candidats vers l'espace collaboratif pour validation ; chaque lien étant couplé à une mesure de confiance et à son type.

5.5 Intégration d'ATLaS dans l'espace collaboratif du projet EVA

Dans cette section, nous présentons le prototype que nous avons développé ainsi que l'espace collaboratif au sein duquel il est intégré et les principaux outils associés. Cet espace a été développé dans le cadre du projet Éco-mobilité par Véhicules Autonomes (EVA) de l'IRT SystemX.

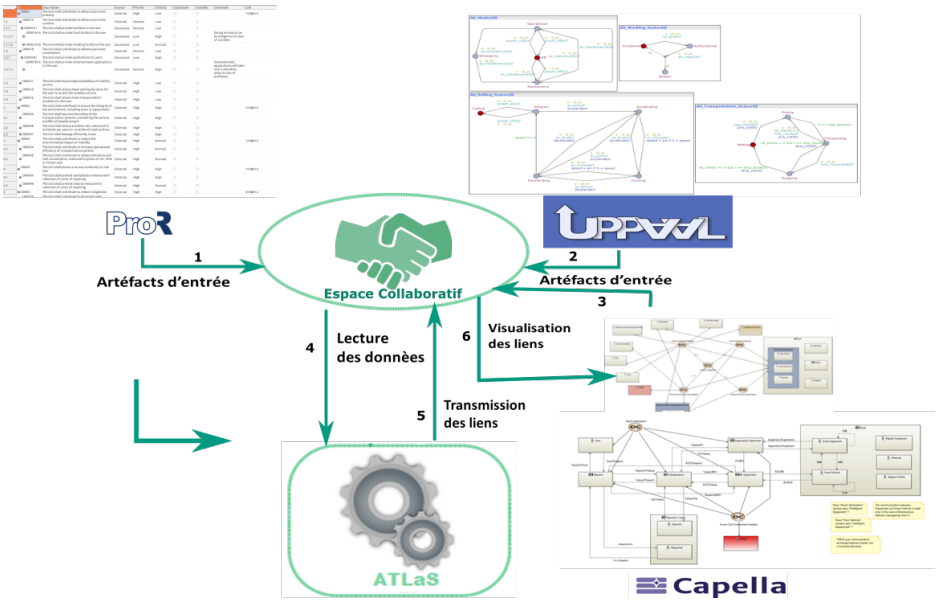

Fig. 5.4 *fonctionnement d'ATLaS dans l'espace collaboratif*

5.5.1 Projet Éco-mobilité par Véhicules Autonomes (EVA)

Ce projet vise à développer un service complet de transports électriques intelligents sans conducteur à la demande, et à en équiper l'infrastructure du plateau de Saclay de manière pérenne.

Des travaux d'analyse et de modélisation du système de systèmes ont été menés dans le but d'établir une spécification formelle des différents systèmes : le véhicule, la caméra, le satellite et le GPS. Les différents partenaires du projet ont donc produit plusieurs artefacts (exigences, modèles, etc ...) dans différents espaces de modélisation (Capella [4], CPN [5], UPPAAL [6], PRoR [7] , etc) et à l'aide de langages de modélisation hétérogènes (Capella, réseau de Petri [152], etc).

Dans le cadre de ce projet, un espace collaboratif [153, 154] a été développé. C'est une plateforme d'intégration applicative qui permet à différents outils de partager et d'échanger leurs données au cours du cycle de développement des systèmes. Cette plateforme est associée aux différents outils à travers des connecteurs et permet l'intégration des données dans un triplestore. Un triplestore est une base de données spécialement conçue pour le stockage et la récupération de données RDF (Resource Description Framework) exprimées sous forme de triplets. Un triplet RDF est une association : sujet, prédicat, objet.

— Le sujet représente la ressource à décrire ;
— Le prédicat représente un type de propriété applicable à cette ressource ;
— L'objet représente une donnée ou une autre ressource : c'est la valeur de la propriété.

ATLaS a été intégré à cet espace. Afin d'évaluer ATLaS, une expérimentation avec les outils PRoR, UPPAAL et Capella a été réalisée. La figure 5.4 illustre le fonctionnement

4. https ://www.polarsys.org/capella/
5. http ://cpntools.org/
6. http ://www.uppaal.org/
7. https ://www.eclipse.org/rmf/pror/

d'ATLaS dans l'espace collaboratif et les paragraphes suivants présentent le rôle de ces différents outils.

ProR est un outil open source qui permet la gestion des exigences. L'*id*, la *description*, les liens entre exigences, et le niveau de priorité de l'exigence sont les informations qui caractérisent chaque exigence. Notons que les liens entre exigences ne sont pas souvent renseignés.

UPPAAL est un environnement d'outil intégré pour la modélisation, la validation et la vérification de systèmes temps réel modélisés en réseaux d'automates. Il permet de représenter les modes et les états des fonctions systèmes et leurs différentes interactions. Les propriétés des modes et états sont définies par l'utilisateur. Celles-ci sont : l'*id*, le *nom de l'état*, la *transition*, le *quid des gardes* et les *événements* définissant le déroulement de l'automate.

Capella est un outil Open Source d'ingénierie systèmes basé sur les modèles (Model-Based Systems Engineering-MBSE). Il permet de représenter les fonctions systèmes. De même que les modes et états, les propriétés des fonctions systèmes sont définies par l'utilisateur. Celles-ci sont : l'*id*, le *nom*, le *type* et la *description*. Cet outil permet également de visualiser les exigences, les modèles et les liens générés par ATLaS.

ATLaS est un outil d'aide à l'identification de liens de traçabilité. Les exigences de PRoR et les modèles de UPPAAL ou de Capella sont envoyés à ATLaS. Après traitement, les liens générés sont également renvoyés sous forme de triplets dans l'espace collaboratif.

5.5.2 Implémentation du framework ATLaS

Le framework *ATLaS* a été implémenté avec différents outils et librairies existants :
- VSM a été implémenté via l'outil Tracelab [155] ;
- LSI et LDA ont été implémentées avec la librairie Gensim [8] ;
- les dictionnaires de termes ont été construits à l'aide du modèle pré-entraîné *Google Word2Vec* de plongement de mots (Word Embeddings) disponible publiquement ;
- les dictionnaires de syntagmes ont été construits à l'aide du modèle pré-entraîné *GloVe* de plongement de mots (Word Embeddings) et la méthode SIF [131] disponibles publiquement ;
- la technique semi-supervisée *LabelSpreading* est basée sur l'implémentation de scikit-learn [9].

Au vue de la taille des données des cas d'étude industriels, qui nécessite une bonne puissance de calcul ainsi qu'une quantité importante de mémoire, nous avons utilisé un super calculateur de 176 cœurs de CPU physiques et 2 To de RAM. Pour les cas d'étude académiques, nous avons utilisé un ordinateur intel(R) core(TM) i5-5300U CPU de 8Go de RAM.

L'implémentation actuelle la méthode *LabelSpreading* dans scikit-learn n'est pas conçue pour des calculs sur de gros volumes de données. Ainsi, faute de trouver une implémentation de la méthode *LabelSpreading* à plus grande échelle, nous avons choisi de faire l'apprentissage à partir d'un sous-ensemble des paires d'artefacts puis d'extrapoler cet apprentissage sur

8. https ://radimrehurek.com/gensim/index.html
9. https ://scikit-learn.org/stable/modules/generated/sklearn.semi supervised.LabelSpreading.html

l'ensemble des paires d'artefacts. Nous avons ainsi défini un algorithme pour construire ce sous-ensemble pour l'apprentissage. Cet algorithme est détaillé dans l'annexe A.

5.6 Conclusion

Dans ce chapitre, nous avons décrit l'approche proposée pour la réduction du coût de l'élicitation des liens de traçabilité et pour l'identification des types de liens. Celle-ci est constituée de trois étapes :

— Étape 1 : collecte d'informations sur des paires d'artefacts ;
— Étape 2 : élaboration d'un jeu d'exemples ;
— Étape 3 : classification des liens et mesure de confiance.

Contrairement aux approches étudiées dans le Chapitre 3, notre démarche présente en premier lieu plusieurs avantages :

— l'usage combiné de liens validés et non validés à travers les techniques semi-supervisées ;
— la capture de plus de sémantique grâce à l'exploitation des atouts des techniques récentes de traitements du langage ;
— et la combinaison de différentes techniques, qui permet de tirer les avantages de ces dernières tout en minimisant leurs limites.

En deuxième lieu, afin notamment de minimiser le travail de l'expert lors de la validation des liens, nous avons fourni une mesure de confiance calculée à partir des techniques de Recherche d'Information, des techniques d'apprentissage et des techniques du Traitement Automatique des Langues. En dernier lieu, nous avons présenté les définitions des types de liens appliquées aux liens considérés comme vrais.

L'évaluation de notre approche a été réalisée sur différents cas d'étude académiques et industriels disponibles publiquement. Ceci nous a permis de souligner le fait que les problématiques qui se posent dans les cas d'études académiques sont plus accentuées dans les cas industriels. Ce point est spécifiquement traité dans le chapitre suivant.

6 – Évaluation

6.1 Introduction

Ce chapitre traite des expérimentations réalisées dans notre méthodologie de recherche. Cette méthodologie est composée de trois étapes :

— Étape 1 : collecte d'informations sur des paires d'artefacts ;
— Étape 2 : élaboration d'un jeu d'exemples ;
— Étape 3 : classification des liens et mesure de confiance.

Comme support à l'approche proposée dans cette thèse, nous avons développé l'outil ATLaS (Aggregation Trace Links Support). L'objectif est de prouver la faisabilité des concepts et de valider la démarche introduite dans le chapitre précédent en utilisant des cas d'étude académiques et industriels. ATLaS est un framework fournissant aux utilisateurs deux fonctionnalités majeures. Sa principale fonctionnalité a pour objectif de faciliter la validation des liens identifiés en réduisant le nombre de *faux liens* et en associant une mesure de confiance à chaque lien.

La suite de ce chapitre est organisée comme suit : la section 6.2 présente les cas d'études académiques et industriels. La section 6.3 présente les expérimentations réalisées dans l'étape 1. Ensuite la section 6.4 présentent les expérimentations réalisées dans l'étape 2. Puis la section 6.5 présentent les expérimentations réalisées dans l'étape 3. Enfin une synthèse des évaluations est proposée dans la section 6.6. Cette section est suivie d'une conclusion.

6.2 Cas d'étude

Afin d'évaluer la pertinence et la faisabilité de la solution proposée, nous avons utilisé les cas d'étude académiques disponibles sur le site internet du COEST [1] (Center of Excellence for Software & Systems Traceability), et les cas d'étude industriels disponibles sur le site internet d'ARC-IT [2].

6.2.1 Cas d'étude industriels ARC-IT

Cette sous-section présente les cas d'étude d'ARC-IT (Architecture Reference for Cooperative and Intelligent Transportation) en version 8.2 tels que décrits sur le site internet [3].

ARC-IT est une architecture de référence fournie par le Département des Transports des États-Unis. Il fournit un cadre commun pour la conception de systèmes de transport

1. Center of Excellence for Software & Systems Traceability, http ://www.coest.org
2. https ://local.iteris.com/arc-it/index.html
3. https ://local.iteris.com/arc-it/index.html

intelligents (intelligent transportation systems - ITS). Il comprend un ensemble d'artefacts d'ingénierie inter-connectés, organisés en quatre vues axées sur différents points de vue : organisationnel (entreprise), fonctionnel, physique et communicatif. Cette architecture est composée de plusieurs artefacts allant des besoins des parties prenantes aux modèles de conception physiques.

Les paragraphes suivants décrivent les cas d'étude évalués dans le cadre de nos travaux, avec l'approche proposée.

Arc-IT1. Le premier cas d'étude ARC-IT, noté ici ***Arc-IT1***, concerne les besoins définis par les parties prenantes et les exigences fonctionnelles. Il comprend 483 besoins, 2395 exigences et 3260 liens de traçabilité validés.

Arc-IT2. Le deuxième cas d'étude ARC-IT, noté ici ***Arc-IT2***, concerne les exigences fonctionnelles et les fonctions systèmes. Il comprend 2395 exigences, 364 fonctions systèmes et 2395 liens de traçabilité validés.

Le tableau 6.1 présente le nombre total d'exigences, d'éléments de modèles, la taille du corpus et du vocabulaire, et le nombre de liens validés pour chaque cas d'étude industriels.

	ARC-IT1	ARC-IT2
Nombre d'exigences	2878	2395
Nombre d'éléments de modèles	0	364
Taille du Corpus	66667	76643
Taille du vocabulaire	2379	2331
Nombre de liens validés	3260	2395

Table 6.1 *Description des jeux de données industriels.*

6.2.2 Cas d'étude académiques

Les expériences de validation de notre approche ont également été menées sur des cas d'étude académiques. Ces cas d'étude sont : Icebreaker, HIPAA (Healthcare Insurance Portability and Accountability Act), EasyClinic, et CM1- NASA (instrument spatial de la NASA). Les paragraphes suivants les présentent brièvement.

a) Icebreaker

Le système *Icebreaker* gère les services de dégivrage pour empêcher la formation de glace sur les routes. Il reçoit des données d'une série de stations météorologiques et de capteurs routiers dans un district donné. Il utilise ces informations pour prévoir les conditions de gel et planifier la dispersion du sel et autres matériaux de dégivrage. Il permet également de maintenir les cartes du district, de gérer l'inventaire des matériaux de dégivrage. Ce dernier est aussi utilisé pour entretenir, expédier et suivre les camions en temps réel. Ce système construit par un groupe d'étudiants, comprend 202 exigences fonctionnelles, 73 classes UML et 452 liens de traçabilité validés.

b) HIPAA

Tous les logiciels liés aux soins de santé aux États-Unis doivent être conformes à la loi *HIPAA*. Cette loi exige que les entités couvertes utilisent des sauvegardes administratives et techniques pour protéger les informations médicales des patients, y compris l'utilisation et la divulgation des informations médicales de ceux-ci. Le jeu de données *HIPPA* fournit la traçabilité entre 10 sauvegardes techniques *HIPAA* et des exigences de 10 systèmes Informatiques de Santé. Il comprend donc 10 sauvegardes techniques, 1881 exigences reparties dans ces 10 systèmes et 243 liens de traçabilité validés.

c) EasyClinic

L'ensemble de données EasyClinic a également été construit par un petit groupe d'étudiants. Il contient plusieurs artefacts en anglais et en italien, notamment des cas d'utilisation, des diagrammes d'interaction, des cas de test et un diagramme de classes. Les cas de test n'ayant pas été pris en compte lors des expérimentations, ce système comprend 113 éléments de modèles et 953 liens de traçabilité validés.

d) CM1-NASA

L'ensemble de données CM1-NASA consiste en une configuration complète décrite par des exigences de haut niveau et un document de conception décrit par des exigences de bas niveau d'un instrument spatial de la NASA (National Aeronautics and Space Administration). Le texte des documents a été modifié par la NASA avant la diffusion publique afin de cacher l'identité de l'instrument. L'ensemble de données CM1-NASA a 21 exigences de haut niveau, 52 exigences bas niveau et 44 liens de traçabilité validés.

Le tableau 6.2 présente le nombre total d'exigences, d'éléments de modèles, la taille du corpus et du vocabulaire, et le nombre de liens validés pour chaque cas d'étude académiques.

	Icebreaker	HIPAA	EasyClinic	CM1-NASA
Nombre d'exigences	201	1881	0	63
Nombre d'éléments de modèles	73	10	113	0
Taille du Corpus	4282	44062	18286	7934
Taille du vocabulaire	439	2718	618	853
Nombre de liens validés	452	243	953	44

Table 6.2 *Description des jeux de données académiques*

6.3 Expérimentation de l'étape 1 : Collecte d'informations sur des paires d'artefacts

Dans cette section, nous procédons à l'analyse des corrélations croisées entre les mesures de similarité VSM, LSI, LDA et les scores de similarité S_1, S_2, S_3 (cf. section 5.4.2). Pour chaque jeu de données, nous calculons les mesures et les scores de similarité de tous les liens possibles. À partir de ces valeurs, nous calculons les coefficients de corrélation de

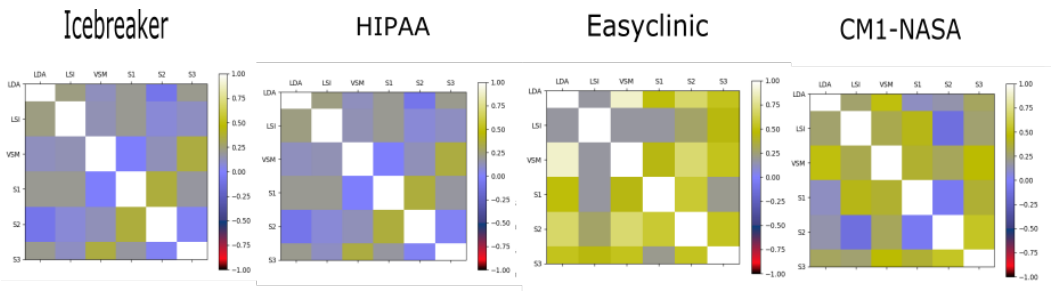

Fig. 6.1 *Corrélation entre les mesures et les scores de similarité : cas d'étude académiques*
Les mesures de similarité fournies par les techniques VSM, LSI et LDA sont fortement corrélées entre elles. De même les scores de similarité S_1, S_2 et S_3 sont fortement corrélés entre eux. Toutefois les informations capturées par les mesures de similarité fournies par les techniques VSM, LSI et LDA et les scores de similarité S_1, S_2 et S_3 sont différentes.

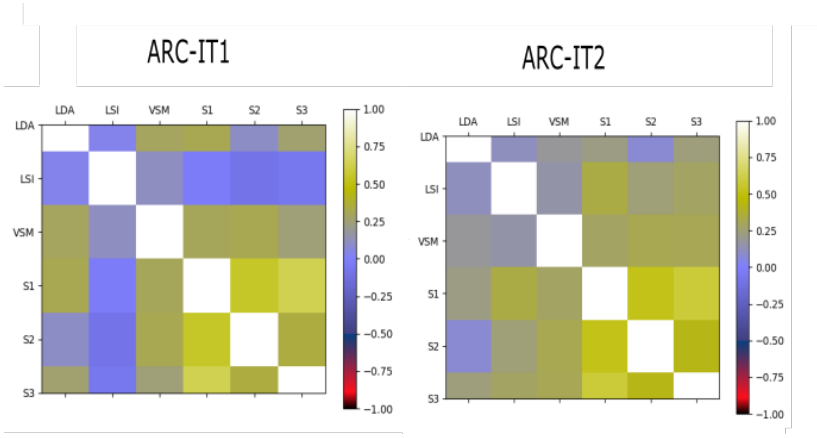

Fig. 6.2 *Corrélation entre les mesures et les scores de similarités : cas d'étude industriels*
Les mesures de similarité fournies par les techniques VSM, LSI et LDA sont fortement corrélées entre elles. De même les scores de similarité S_1, S_2, S_3 sont fortement corrélés entre eux. Toutefois les informations capturées par les mesures de similarité fournies par les techniques VSM, LSI et LDA et les scores de similarité S_1, S_2 et S_3 sont différentes.

Pearson [156] entre les mesures et les scores de similarité sur chaque jeu de donnée. Cela nous indique le degré de similitude des informations extraites des données par ces mesures et ces scores de similarités, et par là même, leur complémentarité éventuelle.

Les matrices de corrélations croisées ainsi obtenues sur chaque jeu de données sont présentées sur la figure 6.1 pour les cas d'étude académiques et la figure 6.2 pour les cas d'étude industriels.

Nous observons que dans la plupart des cas, les mesures de similarités VSM, LSI, LDA sont fortement corrélées entre elles. Par exemple, pour le cas *ARC-IT1* et pour le cas *ARC-IT2*, les mesures de similarité VSM, LSI, LDA ont une dépendance moyenne entre 0.25 et 0.75 (Figure 6.2).

De même, les scores de similarité S_1, S_2, S_3 sont fortement corrélés entre eux. Par exemple, pour les cas *HIPAA* et *Icebreaker*, les scores S_1, S_2 et S_3 ont une dépendance moyenne comprise entre 0.25 et 0.75 (Figure 6.1).

Toutefois, les mesures de similarité sont faiblement corrélées aux scores de similarité, variant de 0 à 0.25, pour la plupart des cas d'étude académiques (Figure 6.1) et pour les cas d'étude industriels (Figure 6.2).

Ceci corrobore l'hypothèse de complémentarité entre les deux familles de techniques utilisées.

6.4 Expérimentations de l'étape 2 : Élaboration d'un jeu d'exemples

Dans cette section, nous évaluons la précision de l'heuristique permettant de constituer un jeu d'exemples en vue de l'apprentissage d'un modèle de classification des liens de traçabilité. Ce jeu d'exemples est constitué de paires d'artefacts étiquetées comme étant liées (« vrais liens ») ou non (« faux liens ») par ladite heuristique. Nous nous appuyons sur les cas d'étude académiques et industriels présentés dans la section 6.2. Rappelons que pour chaque cas d'étude, les liens valides sont connus.

Cas d'étude	Exactitude des vrais liens	Exactitude des faux liens
Icebreaker	30.8 %	99.3 %
HIPAA	6.4%	98.9%
EasyClinic	57%	99.9%
CM1-NASA	25 %	99.3 %
ARC-IT1	1.7%	99.8%
ARC-IT2	4.8%	99.7%

Table 6.3 *Exactitude des vrais et des faux liens dans les six cas d'études*

Le tableau 6.3 montre le pourcentage d'exactitude de notre heuristique pour les différents cas d'étude. Le pourcentage d'exactitude des *faux liens* est supérieur à 98% pour tous les cas d'étude tandis que le pourcentage d'exactitude des *vrais liens* est inférieur à 50% pour les cas d'étude académiques et inférieur à 5% pour les cas d'étude industriels. On observe une différence de précision importante entre l'identification des liens valides et celle des liens non valides. Cette différence est plus accentuée dans les cas d'études industriels. Ce phénomène est lié à ce que nous appellerons le *fléau de la cardinalité*. Nous désignons par là le fait que lorsque le nombre d'artefacts augmente, le nombre de liens valides devient négligeable devant le nombre de liens non valides. De ce fait, la probabilité de se tromper dans la détection des liens valides augmente. C'est ce qui est observé dans les cas d'étude industriels pour lesquels il y a beaucoup plus d'artefacts que dans les cas d'étude académiques. Par exemple, le cas d'étude *ARC-IT2* contient 2395 exigences et 364 éléments de modèle ; le nombre de liens possibles (paires d'artefacts) est donc égal à 871 780 (2395 x 364) pour 2395 *vrais liens*, soit 0.27% de liens valides. En comparaison, le cas d'étude *Icebreaker* contient 201 exigences et 73 éléments de modèles ; le nombre de liens possibles est donc égal à 14 673 (201 x 73) pour 452 liens *vrais liens*, soit 3.1% de liens valides.

6.5 Expérimentation de l'étape 3 : Classification des liens et mesure de confiance

Dans cette section, nous évaluons les performances globales de classification.

6.5.1 Vérification de la Cluster Hypothesis

Comme précédemment mentionné, la technique de classification semi-supervisée retenue repose sur l'hypothèse que l'on a une homogénéité locale des classes dans l'espace de descripteurs.

Afin de vérifier la « *Cluster Hypothesis* », sur chaque cas d'étude et pour chaque lien, nous calculons le nombre de liens de la même classe que le lien considéré parmi ses 100 plus proches voisins.

Les résultats obtenus sont présentés sous forme d'histogramme sur les figures 6.3 à 6.9. L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de lien ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche du cas d'étude *Icebreaker* indique qu'il y a 250 vrais liens ayant dans leur plus proches voisins entre 0 et 10 vrais liens.

Il en ressort que cette hypothèse est largement vérifiée pour les faux liens, comme illustré par la Figure 6.7 pour les cas d'étude académiques. Cette tendance est aussi observée dans les cas d'étude industriels. Toutefois, sur l'ensemble des cas d'étude, le voisinage des vrais liens est dominé par des faux liens, ce de façon plus marquée dans les cas d'étude industriels. C'est une conséquence du fléau de la cardinalité. Il est à noter que l'ajout des scores de similarité S_1, S_2 et S_3 augmente systématiquement le nombre de vrais liens au voisinage d'un vrai lien. Autrement dit, l'utilisation de la combinaison des mesures de similarité VSM, LSI et LDA et des scores de similarités S_1, S_2 et S_3 (*VSM-LSI-LDA-S1-S2-S3*) améliore la discrimination des vrais liens et des faux liens. Toutefois, la disposition majoritaire de faux liens au voisinage des vrais liens dans l'espace des descripteurs aura pour conséquence inévitable de produire des faux-positifs. Cet aspect sera examiné de façon plus détaillée dans la sous-section 6.5.2.

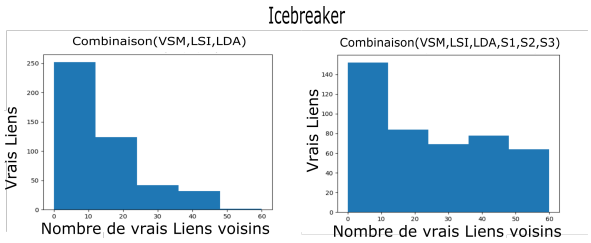

Fig. 6.3 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas Icebreaker L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 250 vrais liens ayant dans leur plus proches voisins entre 0 et 10 vrais liens.*

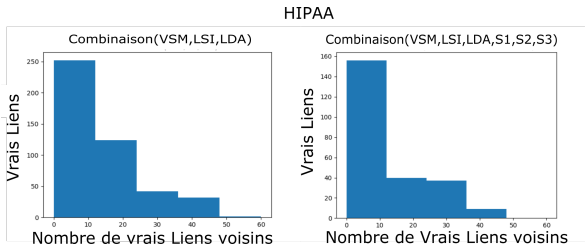

Fig. 6.4 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas HIPAA*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 250 vrais liens ayant dans leur plus proches voisins entre 0 et 10 vrais liens.

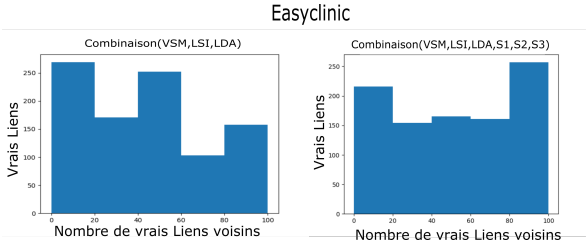

Fig. 6.5 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas Easyclinic*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 250 vrais liens ayant dans leur plus proches voisins entre 0 et 20 vrais liens.

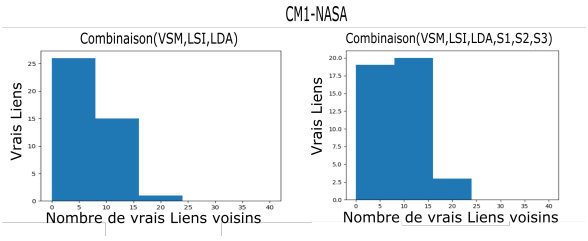

Fig. 6.6 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas CM1-NASA*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 25 vrais liens ayant dans leur plus proches voisins entre 0 et 5 vrais liens.

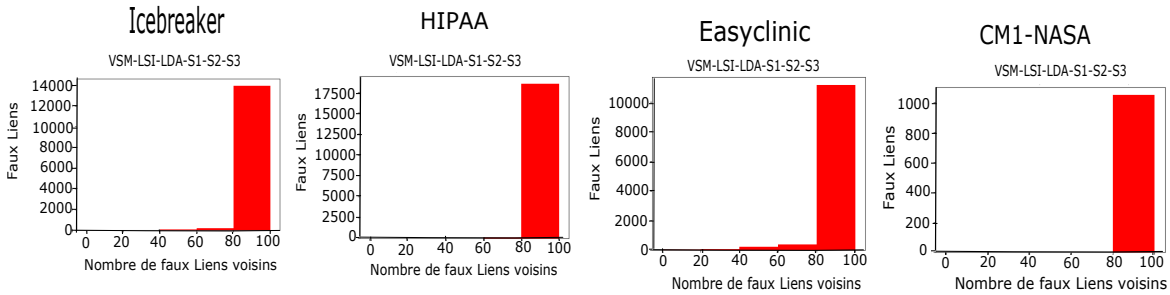

Fig. 6.7 *Vérification de la Cluster Hypothesis pour la classe des faux liens : cas d'étude académiques*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme du cas Icebreaker *indique qu'il y a 14 000 faux liens ayant dans leur plus proches voisins entre 80 et 100 faux liens.*

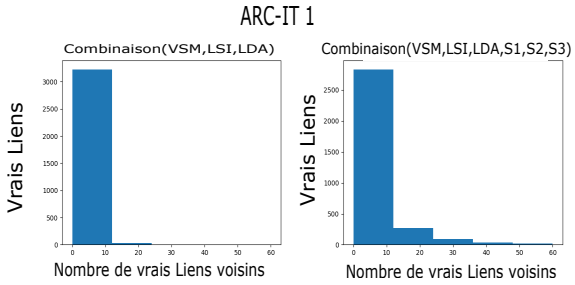

Fig. 6.8 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas ARC-IT1*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 3000 vrais liens ayant dans leur plus proches voisins entre 0 et 10 vrais liens.

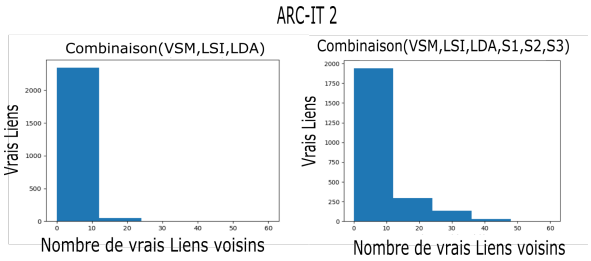

Fig. 6.9 *Vérification de la Cluster Hypothesis pour la classe des vrais liens : cas ARC-IT2*
L'axe des abscisses représente le nombre de liens parmi les 100 plus proches voisins d'un lien donné qui sont de la même classe que ce dernier ; l'axe des ordonnées représente le nombre de liens ayant parmi leurs 100 plus proches voisins x liens de la même classe, x étant donné par l'axe des abscisses. Par exemple, l'histogramme à gauche indique qu'il y a 2000 vrais liens ayant dans leur plus proches voisins entre 0 et 10 vrais liens.

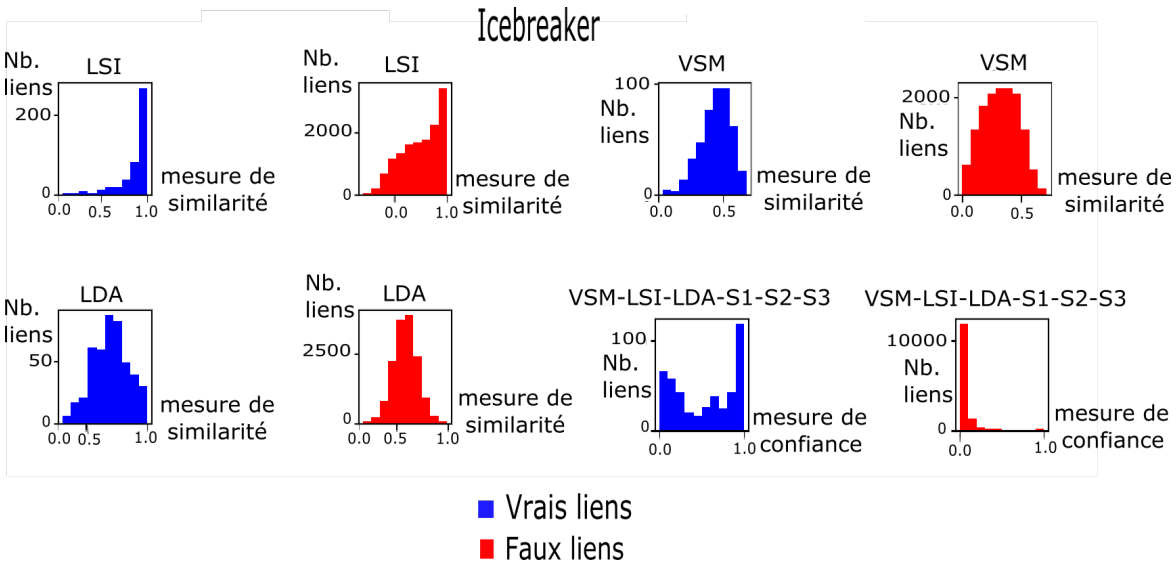

Fig. 6.10 *Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas Icebreaker*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

6.5.2 Pouvoir discriminant du modèle

Dans cette section, nous analysons la capacité de notre modèle prédictif à discriminer les *vrais* et les *faux liens*. Pour rappel, le modèle prédictif consiste en une fonction qui prend en entrée un vecteur de descripteurs d'une paire d'artefacts et renvoie en sortie la probabilité qu'il existe un lien de traçabilité (vrai lien) entre ces deux artefacts. Cette probabilité est notre mesure de confiance. Pour chaque cas d'étude, nous analysons la répartition de cette mesure de confiance entre les *vrais liens* et les *faux liens*. A titre comparatif, nous avons également analysé la répartition des mesures de similarité fournies par les techniques VSM, LSI et LDA.

Les résultats obtenus sont présentés sous forme d'histogramme. Les figures 6.10 à 6.15 illustrent ces résultats pour chaque cas d'étude. Dans chaque figure, les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

Lorsque les techniques VSM, LSI, LDA sont utilisées, un nombre important de faux-liens se voit attribuer des mesures de similarité élevées. La combinaison *VSM-LSI-LDA-S1-S2-S3* par contre permet de baisser significativement les mesures de confiance affectées aux faux liens, et dans certains cas, d'augmenter les mesures de confiance affectées aux vrais liens. Ainsi, le modèle prédictif a bien un meilleur pouvoir discriminant lorsque *VSM-LSI-LDA-S1-S2-S3* est utilisée.

Dans la sous-section suivante, nous allons quantifier précisément la qualité de la classification selon les combinaisons utilisées. Cette analyse reposera sur le rappel, la précision et la F-mesure qui sont les métriques les plus utilisées pour évaluer les techniques en traçabilité (cf. section 4.12)

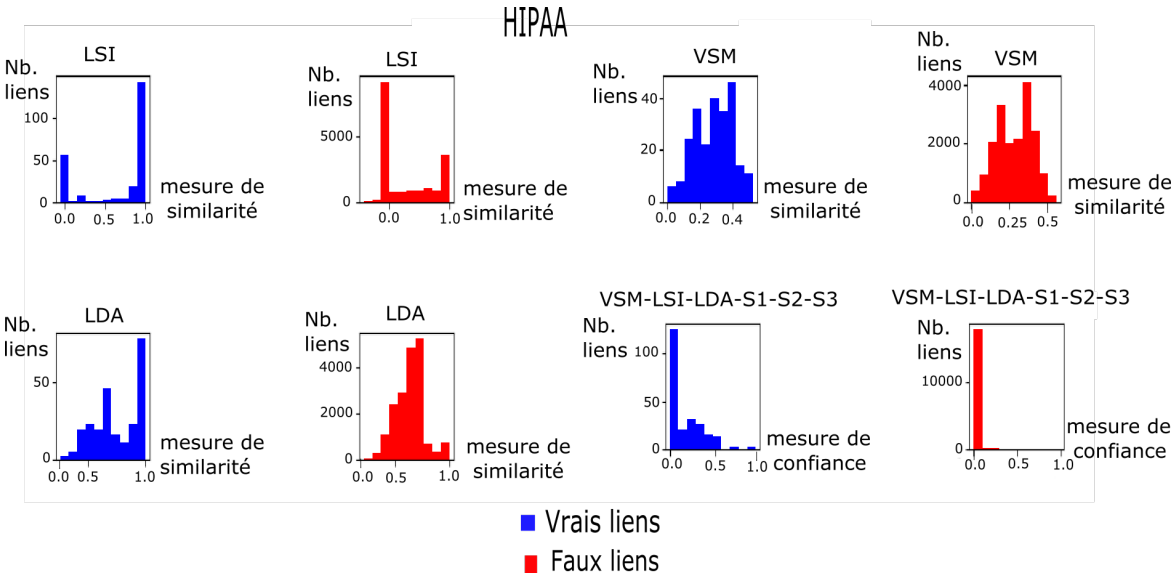

Fig. 6.11 *Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas HIPAA*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

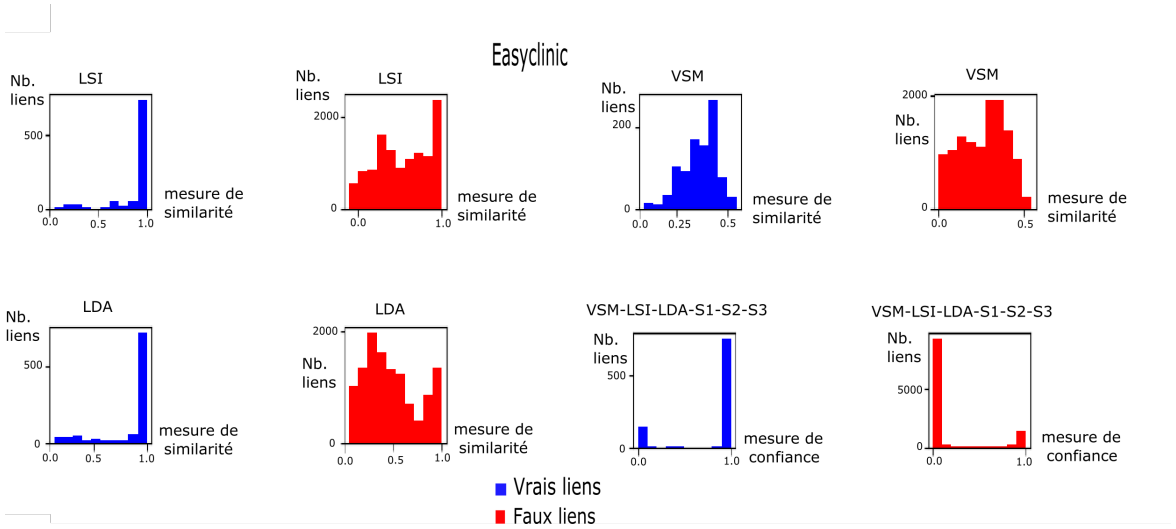

Fig. 6.12 *Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas Easyclinic*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

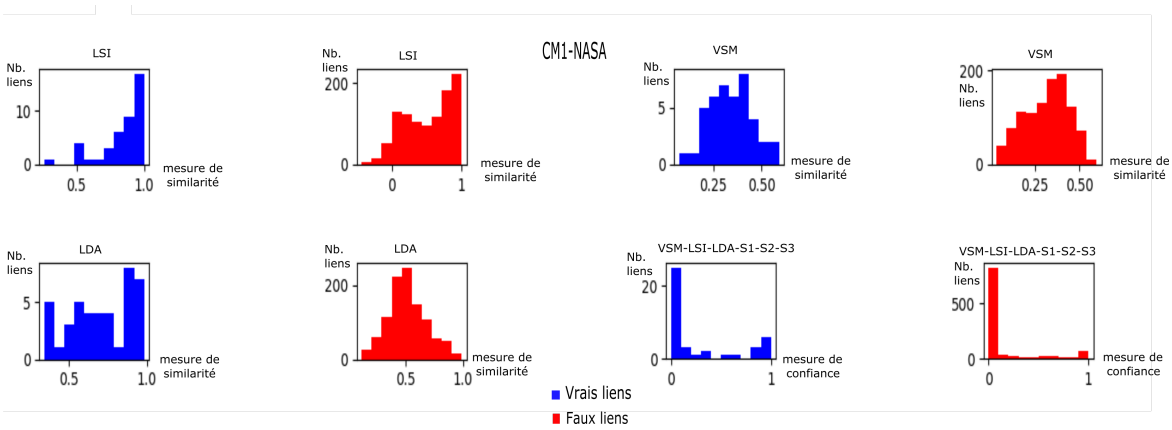

Fig. 6.13 *Répartition des mesures de confiance et de similarité des vrais et faux liens : cas CM1-NASA*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

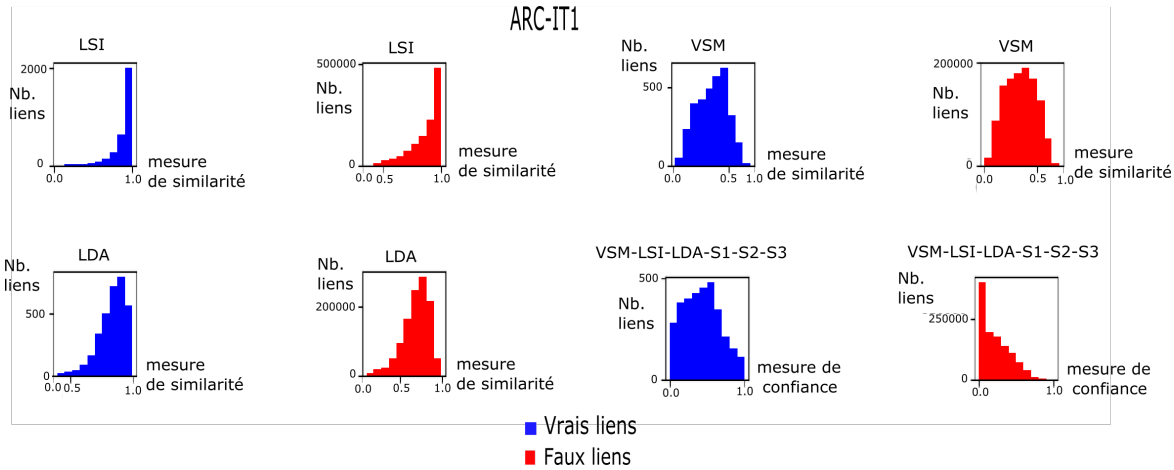

Fig. 6.14 *Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas ARC-IT1*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

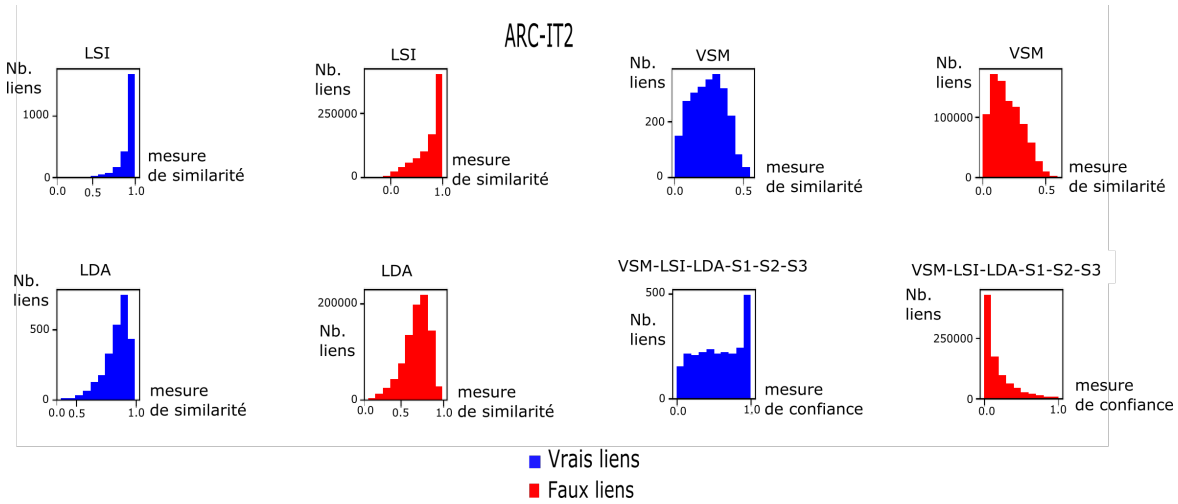

Fig. 6.15 *Répartition des mesures de confiance et de similarité des vrais et des faux liens : cas ARC-IT2*
Les vrais liens sont représentés en bleu et les faux liens en rouge. L'axe des abscisses représente les mesures de confiance ou les mesures de similarité et l'axe des ordonnées représente le nombre de liens.

6.5.3 Courbe Rappel - Précision

Dans cette section, *VSM-LSI-LDA-S1-S2-S3* et les techniques VSM, LSI et LDA sont évaluées via leurs rappels et précisions pour différents seuils sur les six cas d'étude. Le *rappel* est la proportion de *vrais liens* trouvés par rapport au nombre total de *vrais liens*. La *précision* est la proportion de *vrais liens* trouvés parmi l'ensemble des liens étiquetés « vrai ». Rappelons qu'une paire d'artefacts est considérée liée (« vrai lien ») lorsque la mesure de similarité est supérieure ou égale à un seuil donné.

La courbe *Rappel - Précision* illustre les scores de rappels et de précisions à différentes mesures de confiance ou mesures de similarité. Elle permet d'avoir un aperçu des points forts de chaque technique. De manière générale, plus une courbe est globalement éloignée de l'origine, plus la technique correspondante a de meilleures performances.

Les Figures 6.16 à 6.27 présentent les courbes de *Rappel - Précision* de *VSM-LSI-LDA-S1-S2-S3* et des techniques LSI, LDA, et VSM à différents seuils sur les six cas d'étude. Ces figurent montrent que *VSM-LSI-LDA-S1-S2-S3* a globalement de meilleures performances que les techniques LSI, LDA, et VSM.

Le cas *CM1-NASA* est le seul cas où les techniques LSI et LDA, notamment LSI ont de meilleurs résultats que *VSM-LSI-LDA-S1-S2-S3*. D'une part, la forte spécialisation du vocabulaire fait que les scores de similarité S_1, S_2 et S_3 ne sont pas fiables et limitent la précision de l'identification des liens valides. D'autre part la petite taille de ce jeu de données met l'hypothèse d'homogénéité locale des classes complètement en défaut. Dans le cas d'étude *HIPAA*, bien que les performances de *VSM-LSI-LDA-S1-S2-S3* soient meilleures que LSI, LDA ou VSM, leur courbe reste proche de l'origine et les résultats obtenus ne sont pas très satisfaisants. Ceci est dû au faible nombre de *vrais liens* en proportion du nombre total de liens possibles (paires d'artefacts). Pour les cas d'étude *Easyclinic* et *Icebreaker*, la courbe *VSM-LSI-LDA-S1-S2-S3* est globalement éloignée de l'origine. Dans ces deux cas,

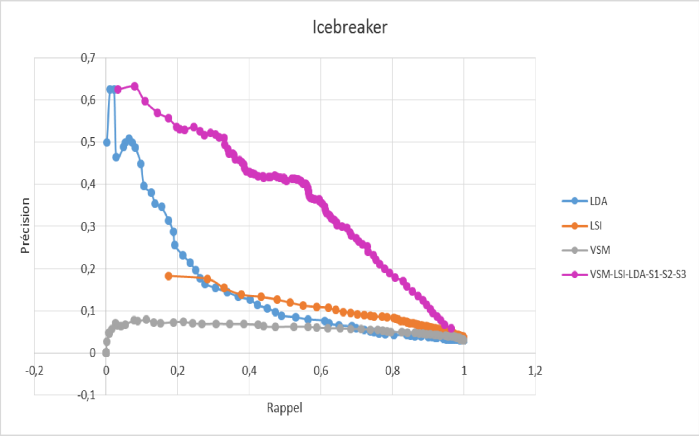

Fig. 6.16 *Courbe rappel-précision : cas Icebreaker*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

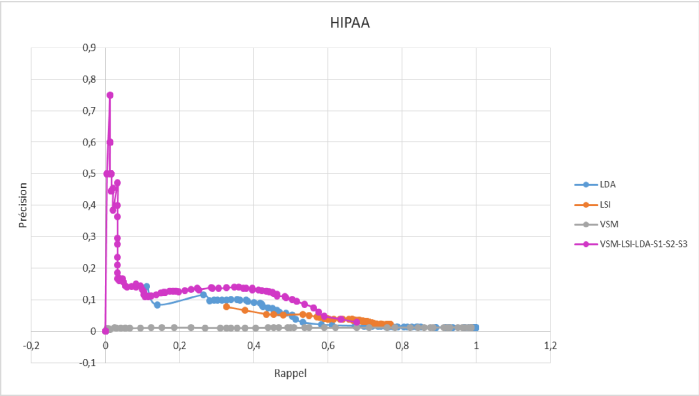

Fig. 6.17 *Courbe rappel-précision : cas HIPAA*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

notre approche a d'excellentes performances, en particulier sur le cas d'étude *Easyclinic*.

Dans les cas d'étude industriels, *VSM-LSI-LDA-S1-S2-S3* a de meilleures performances que celles de LSI, LDA et VSM. Toutefois, les courbes de *VSM-LSI-LDA-S1-S2-S3* restent très proches de l'origine pour les mêmes raisons que dans le cas d'étude académique *HIPAA*.

-

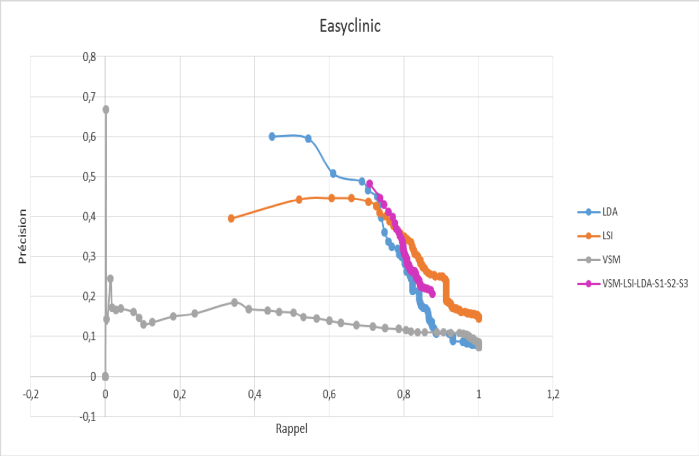

Fig. 6.18 *Courbe rappel-précision : cas Easyclinic*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances entre les seuils* 0.7 *et* 0.8

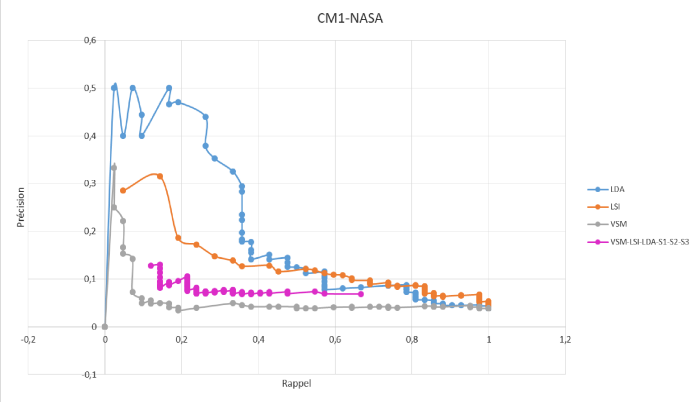

Fig. 6.19 *Courbe rappel-précision : cas CM1-NASA*
LDA et LSI ont les meilleures performances.

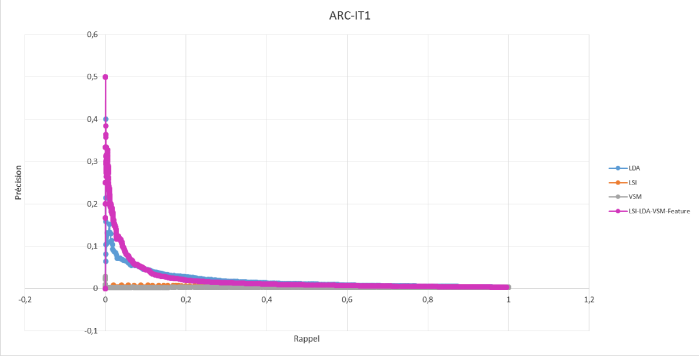

Fig. 6.20 *Courbe rappel-précision : cas ARC-IT1*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

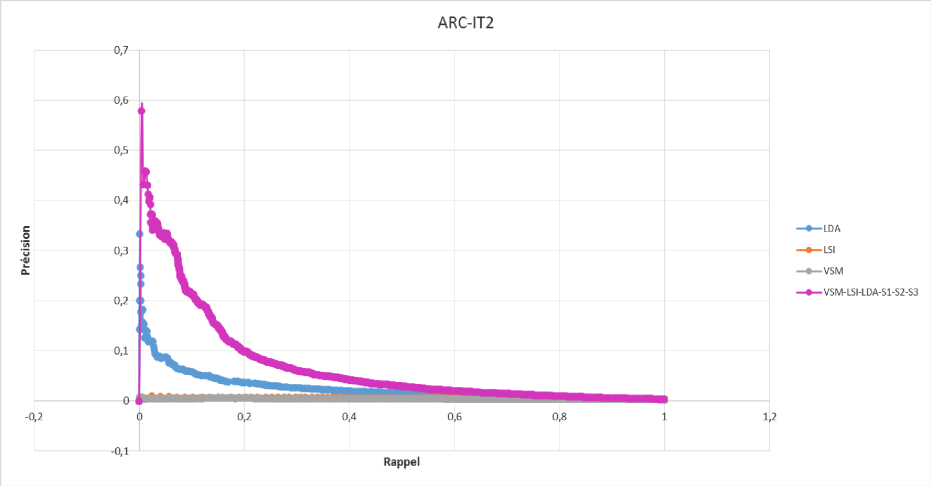

Fig. 6.21 *Courbe rappel-précision : cas ARC-IT2*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

6.5.4 Courbe de F-mesure

Dans cette sous-section, les performances de notre approche et celles des techniques VSM, LSI, et LDA sont évaluées. La F-mesure est la moyenne harmonique du rappel et de la précision. Elle permet de montrer les compromis entre la précision et le rappel, elle peut ainsi être utilisée pour fournir un aperçu de la performance d'une technique.

VSM-LSI-LDA-S1-S2-S3 et les techniques VSM, LSI et LDA sont donc évaluées via leurs F-mesures à différents seuils sur les six cas d'étude. Les figures 6.22 à 6.27 présentent ces résultats. Notons que chaque technique atteint son optimum à un seuil spécifique. L'optimum est le meilleur compromis entre le rappel et la précision.

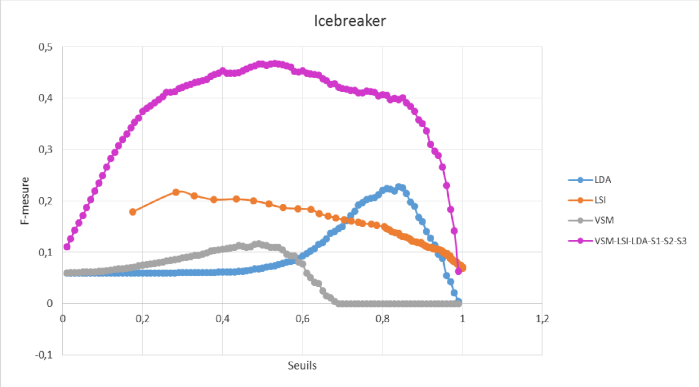

Fig. 6.22 *Courbe F-mesure : cas Icebreaker*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

Ces figures montrent dans les cas académiques que *VSM-LSI-LDA-S1-S2-S3* donnent globalement de meilleurs résultats que les techniques VSM, LSI et LDA prises séparément. En effet, *VSM-LSI-LDA-S1-S2-S3* est meilleure que ces techniques pour tous les seuils. Ce constat est valable dans tous les cas d'étude académiques à l'exception du cas d'étude *CM1-*

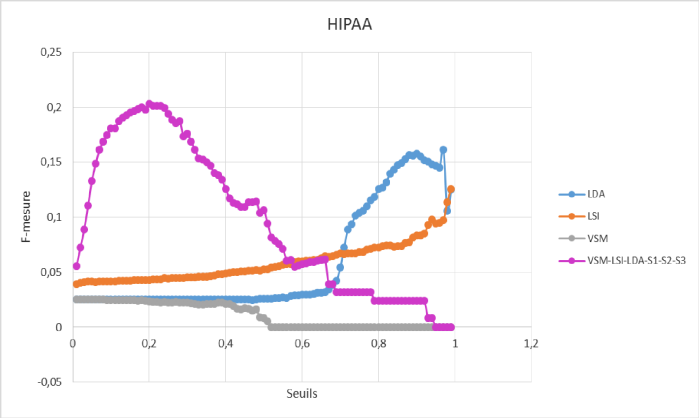

Fig. 6.23 *Courbe F-mesure : cas HIPAA*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances entre les seuils* 0.1 *et* 0.7 *et LSI et LDA ont les meilleures performances aux seuils* 0.7 *à* 1

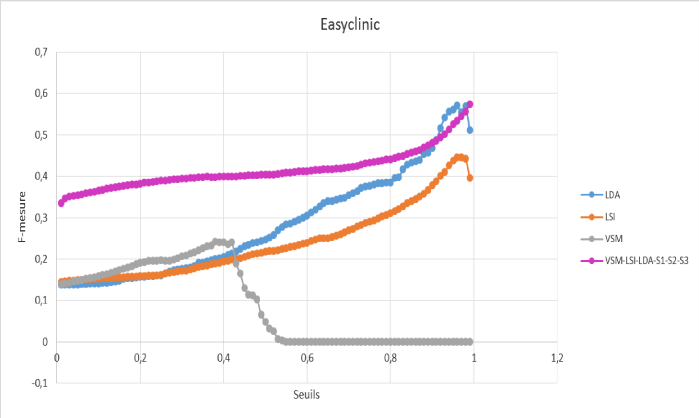

Fig. 6.24 *Courbe F-mesure : cas Easyclinic*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

NASA. Ceci s'explique par le fait que dans ce cas d'étude, les artefacts ont été délibérément modifiés de telle sorte que même des experts du domaine reconnaissent difficilement le système dont il est question [65]. De ce fait, le vocabulaire technique utilisé dans ce cas d'étude est essentiellement ad-hoc.

Dans les cas d'étude industriels, les performances de *VSM-LSI-LDA-S1-S2-S3* sont également meilleures que celles de VSM, LSI et LDA prises séparément, en comparaison aux cas d'étude académiques.

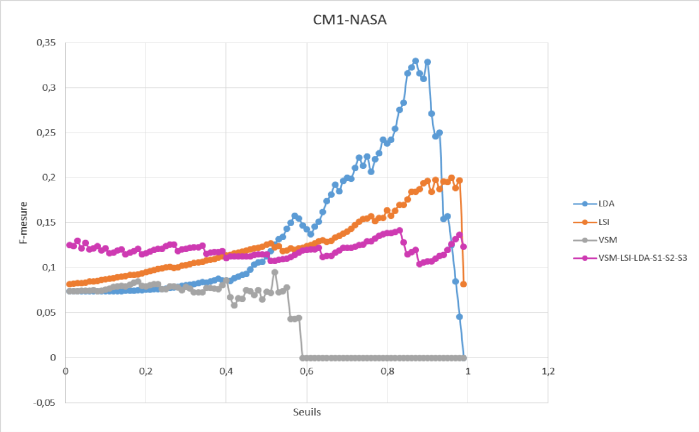

Fig. 6.25 *Courbe F-mesure : cas CM1-NASA*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances aux seuils* 0 *à* 0.3. *LDA et LSI ont les meilleures performances aux seuils* 0.4 *à* 1

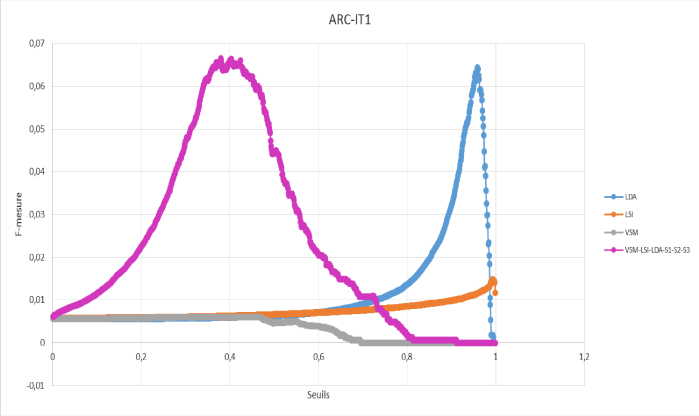

Fig. 6.26 *Courbe F-mesure : cas ARC-IT1*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances aux seuils* 0.1-0.8.

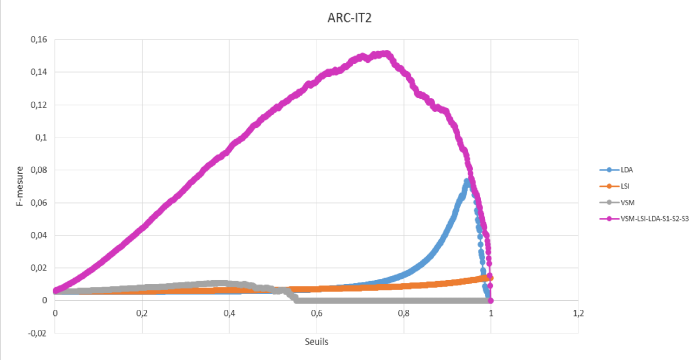

Fig. 6.27 *Courbe F-mesure : cas ARC-IT2*
VSM-LSI-LDA-S1-S2-S3 *a les meilleures performances sur l'ensemble des seuils.*

Les tableaux des figures 6.28 à 6.33 présentent :

— en colonne *Nb liens identifiés* : le nombre de « vrais » liens trouvés par la méthode ;

— en colonne *Nb liens corrects* : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas.

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
Icebreaker	Nb_exigences = 201	0,1	**2654**	**387**	12084	451	14673	452	13654	448
		0,2	1309	330	11127	447	14669	452	11348	439
	Nb_elt_modèles = 73	0,3	914	288	10153	442	14589	452	8453	396
		0,4	736	269	9103	438	13978	443	**5383**	**309**
	Vrais liens = 452	**0,5**	629	252	8043	430	11552	416	2267	156
		0,6	541	225	6973	412	**6889**	**339**	455	35
	Nb_liens_candidats = 14673	0,7	440	187	5809	393	2502	222	13	0
		0,8	345	162	**4388**	**364**	575	113	0	0
		0,9	226	119	2616	281	98	44	0	0

Fig. 6.28 *Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas Icebreaker*

Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaires et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
HIPAA	Nb_exigences = 1881	0,1	**1051**	**117**	8622	185	18810	242	17610	223
		0,2	703	96	8100	180	18810	242	13551	164
	Nb_elt_modèles = 10	0,3	441	60	7550	176	18767	241	9227	106
		0,4	282	33	6868	174	18309	235	3243	37
	Vrais liens = 243	**0,5**	134	20	6335	172	15687	206	**241**	**2**
		0,6	38	8	5440	170	12206	183	0	0
	Nb_liens_candidats = 18810	0,7	9	4	4686	165	**4529**	**129**	0	0
		0,8	5	3	4186	161	1496	109	0	0
		0,9	4	3	**3173**	**142**	850	86	0	0

Fig. 6.29 *Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas HIPAA*
Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaires et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
Easyclinic	Nb_elt_modèles = 113	0,1	**3445**	**803**	11468	950	12451	953	11031	949
		0,2	3204	795	10724	926	10760	921	8701	921
	Nb_elt_modèles = 113	0,3	3059	791	9448	891	9111	888	6304	750
		0,4	2984	785	8019	870	7211	841	**2495**	**414**
	Vrais liens = 953	**0,5**	2903	779	7040	870	5728	827	162	27
		0,6	2802	775	6306	869	4316	803	0	0
	Nb_liens_candidats = 12769	0,7	2693	770	5093	815	**3462**	**784**	0	0
		0,8	2511	765	**4117**	**789**	3083	778	0	0
		0,9	2193	756	2943	736	2140	723	0	0

Fig. 6.30 *Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas Easyclinic*
Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaire et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
CM1-NASA	Nb exigences niv1 = 21	0,1	**238**	**17**	922	42	1092	42	1039	41
		0,2	199	14	843	42	1068	42	862	36
	Nb exigences niv2 = 52	0,3	173	13	754	41	999	42	646	27
		0,4	157	11	679	41	844	38	**284**	**14**
	Vrais liens = 44	**0,5**	149	11	610	41	562	34	67	4
		0,6	125	10	540	36	323	26	0	0
	Nb liens candidats = 1092	0,7	105	9	458	35	**168**	**21**	0	0
		0,8	88	9	**336**	**31**	84	15	0	0
		0,9	70	6	182	22	25	11	0	0

Fig. 6.31 *Récapitulatif des vrais liens trouvés et identifiés par toutes les méthodes : cas CM1-NASA*
Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaires et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
ARC-IT1	Nb_exigences = 2395	0,1	**753391**	**2980**	1118624	3257	1156782	3260	1119961	3182
		0,2	560278	2602	1092986	3244	1149367	3260	936605	2732
	Nb_besoins= 483	0,3	381663	2201	1059047	3229	1130229	3260	694551	2086
		0,4	239566	1772	1014350	3210	1097662	3259	431435	1304
	Vrais liens = 3260	**0,5**	130073	1314	954335	3173	1025677	3225	**181092**	**427**
		0,6	56437	832	874277	3094	882269	3142	26464	58
	Nb_liens_candidats = 1156785	0,7	18575	483	768828	2942	639389	2905	2	0
		0,8	4793	266	625359	2687	322668	2218	0	0
		0,9	766	114	**412384**	**2058**	**61185**	**1034**	0	0

Fig. 6.32 *Récapitulatif des vrais liens trouvés et identifiés : cas ARC-IT1*
Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaires et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

Ces tableaux montrent que le nombre de *faux-positifs* est drastiquement réduit par *VSM-*

			VSM-LSI-LDA-S1-S2-S3		LSI		LDA		VSM	
		seuil	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects	Nb liens identifiés	Nb liens corrects
ARC-IT2	Nb_exigences = 2395	0,1	158456	1822	849433	2395	871780	2395	650359	2076
		0,2	57406	1334	827382	2393	870901	2395	383953	1526
	Nb_elt_modèles = 364	0,3	27330	1025	799104	2391	860412	2395	181642	887
		0,4	14732	793	761998	2384	830960	2394	51244	278
	Vrais liens = 2395	0,5	8171	614	717263	2378	776692	2383	6834	29
		0,6	4577	468	662302	2344	666365	2327	0	0
	Nb_liens_candidats = 871780	0,7	2578	361	591030	2284	470422	2161	0	0
		0,8	1345	262	489809	2150	220193	1759	0	0
		0,9	660	180	330019	1766	35958	829	0	0

Fig. 6.33 *Récapitulatif des vrais liens trouvés et identifiés : cas ARC-IT2*
Ce tableau présente en colonne Nb liens identifiés : le nombre de « vrais » liens trouvés par la méthode ; en colonne Nb liens corrects : parmi l'ensemble des liens de la colonne adjacente, le nombre de liens qui figurent dans l'ensemble des liens valides fournis par l'étude de cas. La couleur verte représente les meilleures précisions, la couleur jaune les précisions intermédiaires et la couleur rouge les précisions les plus faibles. VSM-LSI-LDA-S1-S2-S3 *permet d'obtenir les meilleures précisions.*

LSI-LDA-S1-S2-S3 aux seuils les plus faibles ((0.1 à 0.3).

Ces résultats confirment la pertinence d'une combinaison multi-domaines de techniques utilisées en traçabilité.

6.6 Discussion

Il ressort de nos expérimentations que les performances de notre approche reposent sur plusieurs facteurs. Nous avons établi l'intérêt de l'usage de dictionnaires contextuels. Afin d'améliorer la capture de la sémantique, il serait souhaitable de construire ces dictionnaires à partir de larges corpus textuels métiers ; cela permettrait notamment d'améliorer les performances en cas de forte spécialisation du vocabulaire technique, i.e. lorsque que le sens métier des termes techniques est très éloigné de leur sens courant. La capture de la sémantique est néanmoins limitée par l'ambiguïté inhérente au langage naturel, en particulier, lorsque les artefacts textuels sont élaborés par de nombreux experts pouvant avoir des niveaux d'expression hétérogènes (locuteurs natifs et non natifs). La capture de la sémantique est également limitée par l'écart sémantique entre les artefacts. Nous entendons par là leur niveau de spécification. En effet, plus deux artefacts se situent à des niveaux de spécification du système éloignés, moins l'identification d'un éventuel lien sémantique est fiable. Cette difficulté peut toutefois être mitigée par l'amélioration du dictionnaire contextuel, comme précédemment suggéré.

6.7 Conclusion

Dans ce chapitre, nous avons présenté les cas d'étude académiques et industriels sur lesquels notre approche a été évaluée. Les résultats des différentes expérimentations ont également été présentés.

Ces expérimentation ont montré que notre approche a de meilleures performances que les techniques traditionnelles de traçabilité à savoir VSM, LSI, LDA. En particulier, notre approche permet de réduire drastiquement le nombre de *faux liens*.

Nous présentons dans le chapitre suivant quelques perspectives ouvertes par nos travaux.

Quatrième partie

Conclusion

7 – Conclusion et perspectives

Dans ce chapitre, nous faisons le bilan des contributions de cette thèse (cf. section 7.1), présentons les perspectives ouvertes par notre travail (cf. section 7.2) et donnons finalement la liste de nos publications et communications.

7.1 Contributions

Les contributions issues de nos travaux de thèse s'inscrivent dans le domaine de la traçabilité en entreprises étendues en abordant les problèmes suivants :

— *Problème 1* : Hétérogénéité des artefacts et des outils ;
— *Problème 2* : Interprétation sémantique des liens ;
— *Problème 3* : Efficacité et confiance dans les outils de traçabilité ;
— *Problème 4* : Charge de travail allouée à la traçabilité.

Nous avons en effet proposé une méthode semi-supervisée d'identification de liens de traçabilité. Cette méthode repose à la fois sur des méthodes classiques d'extraction d'information utilisées en traçabilité et des techniques récentes issues du traitement automatique des langues permettant une capture avancée de la sémantique des artefacts textuels.

Nous avons évalué cette méthode à la fois sur des cas d'études académiques et industriels. Elle améliore considérablement la précision de l'identification des liens de traçabilité en comparaison des méthodes d'extraction d'information couramment utilisées à cette fin, tout en réduisant drastiquement le nombre de faux positifs. L'identification des liens se base sur une mesure de confiance qui peut être exploitée par un expert lors de la validation des liens afin d'optimiser son effort de vérification de la validité des liens.

Notre méthode a été implémentée dans un framework dénommé *ATLaS*, lequel a été intégré à une plateforme collaborative qui permet à différents outils de partager et d'échanger leurs données au cours du cycle de développement des systèmes.

Nous avons en outre proposé une définition linguistique du lien de raffinement dans la perspective d'une extension d'*ATLaS* à la problématique du typage de liens de traçabilité.

7.2 Perspectives : vers un système de recommandation de liens de traçabilité

Bien que nous qualifiions notre approche de "semi-supervisée", elle ne requiert pas d'étiquetage préalable de paires d'artefacts fourni par un expert. Une heuristique d'étiquetage a en effet été élaborée à cette fin. Toutefois, plutôt que de faire intervenir l'expert uniquement

en bout de chaîne pour valider les liens candidats identifiés par le modèle construit, il pourrait être intéressant, à budget temps égal, de solliciter l'expert en amont lors de phase de construction du jeu d'exemples, en lui proposant sélectivement quelques paires d'artefacts à étiqueter, afin d'apprendre un meilleur modèle. La difficulté résiderait alors dans le choix des paires d'artefacts à présenter à l'expert, qui maximiserait le bénéfice de son intervention, à budget temps fixé. C'est un axe de recherche qui nous semble intéressant à explorer en exploitant de façon plus avancée la distribution spatiale des paires d'artefacts dans l'espace de descripteurs.

D'autre part, nous pouvons raisonnablement supposer que pour deux projets similaires en ingénierie système, soit du point de vue du domaine métier, soit du point de vue du système conçu, les artefacts et par conséquent les liens de traçabilité produits dans les deux cas partagent nécessairement des propriétés statistiques qui se transposent d'une certaine manière dans l'espace de descripteurs. Un modèle de classification de liens de traçabilité appris pour l'un des projets devrait pouvoir être exploité pour l'autre, moyennant un degré d'adaptation lié au degré de dissemblance entre les deux projets. Telle est l'ambition de l'apprentissage par transfert, branche prometteuse de l'apprentissage automatique qui, nous semble-t-il, est susceptible d'être d'un apport décisif pour la traçabilité dans le domaine industriel. A ce propos, le premier défi sur les cas d'étude industriels est la volumétrie des artefacts qui conduit comme nous l'avons vu au fléau de la cardinalité. Un axe intéressant de recherche serait alors d'élaborer ou d'apprendre automatiquement des règles métiers qui permettraient préalablement d'éliminer les liens candidats absurdes du point de vue de la conception du système à l'étude, de sorte à se ramener à un nombre de liens candidats simplement proportionnel au nombre d'artefacts.

7.3 Liste des publications liées à la thèse

Cette section présente les publications que nous avons réalisées pendant cette thèse :

— Semi-supervised Approach for Recovering Traceability Links in Complex Systems, Conference Proceedings, 2018 23rd International Conference on Engineering of Complex Computer Systems (ICECCS), Emma Effa Bella, Marie-Pierre Gervais, Reda Bendraou, Laurent Wouters and Ali Koudri.

— ATLaS : A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised Techniques, Conference Proceedings, 2019 23rd ieee international edoc conference - the enterprise computing conference, Emma Effa Bella, Stephen Creff, Marie-Pierre Gervais and Reda Bendraou.

Autres Publications :

— Collaborative systems engineering : Issues & challenges, 2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD), Laurent Wouters, Stephen Creff, Effa Bella Emma, and Ali Koudri.

— Towards Semantic-Aware Collaborations in Systems Engineering, 2017 24th Asia-Pacific Software Engineering Conference (APSEC), Laurent Wouters, Stephen Creff, Effa Bella Emma, and Ali Koudri.

A – Algorithme de construction d'un sous-ensemble de paires d'artefacts

L'implémentation actuelle de la méthode *LabelSpreading* dans scikit-learn [1] n'est pas conçue pour des calculs sur de gros volumes de données. Ainsi, faute de trouver une implémentation de la méthode *LabelSpreading* à plus grande échelle, nous avons choisi de faire l'apprentissage à partir d'un sous-ensemble des paires d'artefacts puis d'extrapoler cet apprentissage sur l'ensemble des paires d'artefacts.

Cet algorithme vise à s'assurer que le sous-ensemble des paires d'artefacts sélectionné pour l'apprentissage est représentatif de l'ensemble des paires d'artefacts. L'idée de base est la suivante : lorsqu'une paire d'artefacts est sélectionnée pour le sous-ensemble, il n'est pas nécessaire d'avoir des paires d'artefacts appartenant à son voisinage le plus proche dans le sous-ensemble car elles sont susceptibles d'appartenir à la même classe. Le voisinage est défini en termes de distance euclidienne entre les *vecteurs de descripteurs* des paires d'artefacts. Le sous-ensemble pour l'apprentissage est construit comme suit : le voisinage de chaque paire d'artefacts est calculé et représenté sous forme de graphe. Ensuite, ce graphe est divisé en plusieurs parties. Le nombre de parties constitue le facteur de sous-échantillonnage et est un paramètre de l'algorithme. Il est défini suivant le nombre de paires d'artefacts maximum que peut contenir le sous-ensemble. Les paires d'artefacts sont sélectionnées aléatoirement. Lorsqu'une paire d'artefacts est sélectionnée, ses voisins les plus proches sont écartés. Cette opération est répétée jusqu'à ce qu'il n'y ait plus de paires d'artefacts.

1. https ://scikit-learn.org/stable/modules/generated/sklearn.semi_supervised.LabelSpreading.html

B – Quelques liens de traçabilité

Cette annexe présente la liste des liens utilisés pour servir d'exemples dans le document. Elle présente également pour chaque cas d'étude quelques liens de traçabilité trouvés par la combinaison *VSM-LSI-LDA-S1-S2-S3*) et quelques liens manqués par toutes les techniques.

ICEBREAKER		
N.	Liste des liens trouvés par la combinaison *VSM-LSI-LDA-S1-S2-S3*	
1	9044 Truck Management	72 material Type truck Condition capacity get Capacity set Material Type get Material Type truck Plate NUmber set Truck Plate NUmber get Truck Plate NUmber set Truck ID set Capacity set Truck Condition get Truck ID Truck Database 317 truck ID get Truck Type set Truck Type truck Type get Truck Condition
2	9134 PurchaseOrders shall be automatically generated when de-icing material stocks fall under a predetermined limit.	21 get Time set Time time get Inventory Update set Observe material create PurchaseOrder date Inventory Maintenance 371 set Date get Date
3	9010 Weather forecasts data history shall be recorded.	104 get Forecast add Forecast set Forecast time set Time get Time date set Date get Date Historical Forecast Database 366 forecast
4	9082 A list of authorized users shall be maintained.	57 display Failed Authorization set Password modify User Info display Invalid Entry display Successfull Authorization password user ID set User ID get User ID Access Control GUI 349 get Password display Access Level
5	9171 The scheduler shall minimize the total mileage of the de-icing schedule.	53 analyze District Weather Data Generate De-Ice Schedule analyze Freezing Time Scheduler 308 get District Data set District Data district Data

ICEBREAKER		
N.	Liste des liens manqués par toutes les techniques	
1	9003 Road temperature readings shall be recorded.	27 set Length location set Location get Location road Section ID set Road Section ID Road Section 324 Get Current freezing status length Set freezing conditions get Length get Road Section ID
2	9016 Data received from the road sensors shall be updated regularly.	6 set WeatherStation ID WeatherStation ID validate Station handle Failed Station Data alert Failed Station handle Data Transmission WeatherStation ID time set Time get Time date set Date WeatherStation ID get WeatherStation ID get Update Time WeatherStation Input Proxy 335 get Date
3	9018 De-icing shall not be scheduled further than two days in advance.	53 analyze District Weather Data Generate De-Ice Schedule analyze Freezing Time Scheduler 308 get District Data set District Data district Data
4	9020 The scheduled de-icing shall be for a valid district.	32 district ID Map Database 361 remove Map add Map district Name get District Name set District ID get District ID map ID Assign sensors to road section get Map ID update Map map set Map get Map set Map ID set District Name
5	9020 The scheduled de-icing shall be for a valid district.	37 handle Road Closing handle Road Reopen district ID set District ID get District ID remove Map add Map zoom Network Map GUI 369 district ID display Map map set Map get Map get District ID

HIPAA		
N.	Liste des liens manqués par toutes les techniques	
1	432 Consumers may also want to decide who will view and communicate their personal health information	111 Access Control Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in 164 308 a 4
2	596 System will support transaction logs that will include the MID of the editor transaction type and transaction date	112 Audit Controls Implement hardware software and or procedural mechanisms that record and examine activity in information systems that contain or use electronic protected health information
3	33386 All data is required to be encrypted using SSL TLS	114 Encryption Implement a mechanism to encrypt electronic protected health information whenever deemed appropriate
4	230545 When passwords are used the system shall provide an administrative function that resets passwords	118 Person or entity authentication Implement procedures to verify that a person or entity seeking access to electronic protected health information is the one claimed .
5	33356 System defines a single set of credentials for each party connecting to the system	120 Unique user identification. Assign a unique name and/or number for identifying and tracking user identity.

EASYCLINIC		
N.	Liste des liens trouvés par la combinaison *VSM-LSI-LDA-S1-S2-S3*	
1	3038 Access section patient performed by code tax The service was launched following the express request by the actor The Operator logs on to form on the management section of the patient body of GUILoginPaziente insert the tax code that Patient has made a request and select the function The validation then passes control to the instance of GUILoginPazienteHandler which delegates to the instance of PazienteManager the task of managing the service this growth deals to validate the data entered by using The body of the patient outcome is the notified backward up to the instance of GUILoginPazienteHandler which requires the application of GUIPazienteHandler to enable access by the Section Patient applicant and that object shall display main form for instance GUIPaziente This feature is been described by the sequence diagram of Fig 3 13	7020 Validation patient He works to perform the functions necessary to authenticate an Patient Has an interest in validating the patient to access to the section relating to his information The Operator has been recognized by System See UcValOp The patient must provide the operator with the Hospital or Card its tax code The patient is not recognized system Success The patient is recognized by System Operator and access to its section The Operator active enforcement of a depending on a specific patient 1 View the mask for validating the Patient 2 input the code of Hospital Card 3 Confirm validation 4 Verify that the code Card s Hospital identifies a patient within S I O 5 Get information Patient 6 Transfer in the memory Patient data Validation through tax code 2 1 injects the tax code Patient 2 2 Reinserts running from step 3 Operator cancels function validation 3 1 ends the execution of use case with failure Login failed 4 1 Show appropriate message 4 2 Reinserts execution from step 2
2	3047 Show anagrafica The service was launched following the express request by the Patient The actor Patient access to the mask main for instance GUIPrincipale on the management of all services provided by the system to the request of patients and there select the display of their anagrafica Master Control then goes to the instance of GUIPrincipaleHandler which delegates to the instance of GUIAnagraficaHandler the task of managing the service at this Point control is passed to the instance of PazienteManager which deals with finding by the panel of the Patient Patient data applicant Results are sent backward up to the instance of that GUIAnagraficaHandler turn is responsible for their display on the screen on the management of registries of patients for instance GUIAnagrafica This feature is was described by collaboration diagram of Fig 3 9	7023 Show anagrafica Allows a patient of view their anagrafica data using the specific locations displaced of the outpatient He has an interest in access to your anagrafica information The patient has been recognized by system See UcAccTB The patient obtains display of their data anagrafica The patient activates the execution of service display their data anagrafica 1 Access to the database and retrieves information about anagrafica Patient 2 View information anagrafiche Patient Patient decides to print Information displayed 2 1 Start printing the list of Reservations 2 2 generates the report and sends it to Printer 2 1 The age of a patient is characterized by following data name surname sex code Tax date of birth place of birth province of birth place of address home province of residence address ZIP code Additional telephone and Notes

EASYCLINIC		
N.	Liste des liens manqués par toutes les techniques	
1	3014 Book visit It allows the operator to meet request for a booking service made by an ambulatory patient The Operator has an interest in the record renting an outpatient service requested by a patient The Operator has been recognized by System See UcValOp The patient is been recognized by the system See UcValPaz and has issued to The module R S A properly completed The data in the S I O not be modified Success The outpatient service required is properly booked The operator activates the execution of service booking service Outpatient 1 First visit The Operator has received a request First visit part of a patient Inserting anagrafica patient 2 View the mask for booking of service Outpatient 3 Select the type of visit that the patient wants make 4 Inserts motivation of the Request 5 Confirm data inserted 6 Verify that the sequence Reservations are valid See BrValSeq Includes 7 Select dates available 8 Confirm your reservation 9 Stores data Notify the operator 10 that the operation was concluded successfully Sequence reservations invalid 6 1 Show an appropriate message 6 2 Reinserts execution from step 2 Operator cancels the operation Reservations 5 1 ends the use case with failure 8 1 9 1 For a first visit has shown to this point that will be saved even the personal data In addition the patient always in the case of a Before visiting the system will view and print the card in hospital patient	7081 Test case reservation a First Date C31 from a visit 20 06 2003 new patient of the outpatient Version 0 02 000 Use Case Meets the request for a reservation UcPreVis service by Outpatient a patient High Priority Set up There are no reservations Description test Input type selected visit First visit The Oracle system enables the reservation of visit Classes cover valid CE1 CE5 Classes are not valid None
2	3031 Operator Login Scenario validation run The service was launched following the express request by the actor Operator The access to the screen on the management of the system for instance GUILogin enter login and password and check the function The validation then passes control to the instance of GUILoginHandler which delegates to the instance of OperatoreManager the task of managing the latter deals with validate the data entered by using the panel The successful operator is notified of a backward up the application of which requires GUILoginHandler to the instance of GUIPrincipaleHandler to enable access to Patient applicant and that object shall display the main form for instance GUIPrincipale This feature is was described by the sequence diagram of Fig 3 2	7051 Test case Operator Login Date C01 s login through 20 06 2003 correct and incorrect password torque login password in S I O Version 0 02 000 Use Case He performs the functions UcValOpe necessary to authenticate an Operator High Priority The couple set up login password Rocolo Virago is recorded in the S I O Description test Input Login Rocolo Password Virago The Oracle system allows access the operator Classes cover valid CE1 CE2 CE6 CE7 Classes are not valid None

CM1-NASA		
N.	Liste des liens manqués par toutes les techniques	
1	51221 The DPU CCM shall implement a mechanism whereby large memory loads and dumps can be accomplished incrementally	512144 Memory Upload and Download Handling Data can be upload to several types of locations including DRAM EEPROM hardware registers and EEPROM filesystem ial D MEM DAT UPLD command specify the target location If the destination is the EEPROM filesystem a block number is provided in lieu of a memory address which is used by the DPU FSW to formulate a filename of the form eefs1 DPU blk where is the block number In this case once the entirety of the uploaded data is received by the DPU FSW the uploaded data is then written to that file in the EEPROM filesystem If a file already exists with that name it is overwritten The EEPROM filesystem can be reinitialized using the command D MEM DISK INIT
2	51231 The DPU CCM shall record an error to the LAST BOOT IVEC location in EEPROM and discontinue strobing the Watchdog Timer should an unrecoverable software error occur An unrecoverable software error is defined as an error that causes a loss of commandability or ground communication	512111 Flight Software Initialization The Command and Control CSC is initialized by spawning the CCM Control Task ccmCtrlTask from the operating system startup task usrRoot After the task is spawned it calls a function ccmInit which creates the error event queue instantiates needed semaphores and installs various ISRs Finally it spawns the remaining tasks which comprise the DPU FSW When the CCM Control Task starts up it reads DPU configuration startup defaults from the SYSTEM CONFIG AREA in EEPROM If the SYSTEM CONFIG AREA checksum is bad hard coded defaults are used The CCM Control Task initializes the remaining CSCs by calling the applicable initialization function or if the CSC has an associated task by spawning the task using the VxWorks function taskSpawn When the CCM Control Task initializes a CSC it passes the startup defaults read from EEPROM as parameters to the task initialization function In addition to its task initialization activities ccmInit also initializes the command length verification table by calling ccmCmdLengthInit The command processor uses the created table to verify expected command lengths for newly received commands

ARC-IT1		
N.	Liste des liens trouvés par la combinaison *VSM-LSI-LDA-S1-S2-S3*	
1	R5524 ;The vehicle shall be capable of joining a platoon by determining the recommended platoon entry location and the timing for the vehicle to enter a platoon.	N511 ;The Connected Vehicle needs to be able to move into a platoon and inform the vehicle driver of how, when, and where to safely join a platoon.
2	R2164 ;The center shall receive citation records from roadside check facilities.	N787 ;Commercial Vehicle Administration needs to have a safety screening capability that acquires safety inspection data from commercial vehicles using electronic communications with fleets and with vehicles along the roadside.
3	R2540 ;The vehicle shall provide position warnings to the driver when an object gets close enough to the vehicle to become a hazard if no action is taken by the driver.	N819 ;Drivers need their vehicle to provide safety warnings to them when a safety compromising situation is detected by on board systems.
4	R6301 ;The center shall be able to receive driver activity logs from peer centers such as other commercial vehicle service providers.	N805 ;Commercial Vehicle Service Providers need to collect and manage driver log information in order provide the information to the appropriate regulators and to make Fleet and Freight Management aware that their vehicles are being operated safely.
5	R5552 ;The center shall send eco-driving recommendations to connected vehicles so that the vehicle or the driver can adjust their driving behavior to save fuel and reduce emissions.	N519 ;The vehicle driver needs to be able to receive customized real-time driving advice so that they can adjust their driving behavior to save fuel and reduce emissions.

ARC-IT 1		
N.	Liste des liens manqués par toutes les techniques	
1	R2042 ;The public interface for travelers shall present information to the traveler in a form suitable for travelers with physical disabilities.	N752 ;Transit Operations needs to be able to collect transit fares at transit stations using electronic payment methods in or der to support bus rapid transit or train systems.
2	R2043 ;The parking element shall process the financial requests and manage an interface to a Financial Institution.	N175 ;Regional Transportation Operations need to have an electronic payment functionality that operates across different modes or systems.
3	R2082 ;The center shall collect the log of passenger boardings and alightings from the paratransit vehicles.	N264 ;Transit Operations needs to provide and update manifests to properly manage demand response transit vehicles based upon traveler requests.
4	R2105 ;The vehicle shall prioritize safety and warning messages to supersede advisory and broadcast messages.	N674 ;Traveler Information needs to be able to inform as much of the traveling public as possible using any available means to increase mobility and safety through better information.
5	R2160 ;The center shall package data concerning commercial vehicle safety and credentials into profiles (detailed and historical data).	N728 ;Commercial Vehicle Administration needs to be able to inform the appropriate parties of issues dealing with the clearance of a commercial vehicle or its driver in order to maintain the smooth flow of goods through its roadways.

ARC-IT2		
N.	Liste des liens trouvés par la combinaison *VSM-LSI-LDA-S1-S2-S3*	
1	R5205 ;The center shall receive request from shippers for the available loads to be shipped from intermodal customer locations in the region.	F8 ;The 'Drayage Operations Optimization Service' provides a portal for shippers and receivers to post their loads in need of transport and provide an opportunity for commercial vehicles to find a load to haul on their trip back to/from an intermodal facility. It connects load matching services with container and chassis/equipment availability information and appointment/reservations services from the intermodal terminals to provide an optimized, regionally integrated view of intermodal container transport for drayage operators.
2	R2539 ;The vehicle shall present information to the driver in audible or visual forms without impairing the driver's ability to control the vehicle in a safe manner.	F405 ;'Vehicle Control Warning' monitors areas around the vehicle and provides warnings to a driver so the driver can take action to recover and maintain safe control of the vehicle. It includes lateral warning systems that warn of lane departures and obstacles or vehicles to the sides of the vehicle and longitudinal warning systems that monitor areas in the vehicle path and provide warnings when headways are insufficient or obstacles are detected in front of or behind the vehicle. It includes on-board sensors, including radars and imaging systems, and the driver information system that provides the visual, audible, and/or haptic warnings to the driver.

ARC-IT 2		
N.	Liste des liens manqués par toutes les techniques	
1	R147 ;The center shall correlate electric vehicle needs to charging station capacities.	F100 ;'TIC Travel Services Information' disseminates information about traveler services such as lodging, restaurants, and service stations. Tailored traveler service information is provided on request that meets the constraints and preferences specified by the traveler. This application also supports reservations and advanced payment for traveler services including parking and loading zone use.
2	R1244 ;The transit vehicle shall receive acknowledgments of the emergency request from the center and output this acknowledgment to the transit vehicle operator or to the travelers.	F336 ;'Transit Vehicle Security' provides security and safety functions on-board the transit vehicle. It includes surveillance and sensor systems that monitor the on-board environment, silent alarms that can be activated by transit user or vehicle operator, operator authentication, and a remote vehicle disable function. The surveillance equipment includes video (e.g. CCTV cameras), audio systems and/or event recorder systems. The sensor equipment includes threat sensors (e.g. chemical agent, toxic industrial chemical, biological, explosives, and radiological sensors) and object detection sensors (e.g. metal detectors).
3	R1560 ;The center shall aggregate collected environmental probe data and disseminate the aggregated environmental probe data to other centers.	F32 ;'TIC Road Weather Advisories and Warnings' provides road weather advisories to drivers and other travelers.
4	R1759 ;The center shall support online route guidance for specialty vehicles, such as commercial vehicles.	F96 ;'TIC Trip Planning' provides pre-trip and en-route trip planning services for travelers. It receives origin, destination, constraints, and preferences and returns trip plan(s) that meet the supplied criteria. Trip plans may be based on current traffic and road conditions, transit schedule information, and other real-time traveler information. Candidate trip plans are multimodal and may include vehicle, transit, and alternate mode segments (e.g., rail, ferry, bicycle routes, and walkways) based on traveler preferences. It also confirms the trip plan for the traveler and supports reservations and advanced payment for portions of the trip. The trip plan includes specific routing information and instructions for each segment of the trip and may also include information and reservations for additional services (e.g., parking) along the route.
5	R1877 ;The center shall make the compiled ridership data available to the system operator.	F129 ;'Transit Center Passenger Counting' receives and processes transit vehicle loading data using two-way communications from equipped transit vehicles.

Cas d'étude	liste des liens utilisés comme exemple dans le document	
EVA	160003 User shall authenticate to get into the booked vehicle.	User UserId password name location acountBalance setDestination getDestination
EVA	160001 User shall get all necessary information to reach a given destination.	User UserId password name location acountBalance setDestination getDestination
EVA	150020 The SoS shall prevent vandalization measured in reduction of costs of repairing.	150021 The SoS shall prevent failures measured in reduction of costs of repairing.
EVA	User UserId password name location acountBalance setDestination getDestination	Dispatch Request (u,s, d) usersToAuthentify u ID_USER u AuthentifyUser if success_authentification(u) then 1 u else empty AuthentifiedUser() ID_USER UserToAutorize() ID_USER autorizeUser if success_authorization(u) then 1 u else empty authorized User ID_USER u
EVA	650003 The System of System must be able to detect an obstacle.	600007 The System of System must be able to detect an obstacle when the obstacle is at 14m on a dry road (19m on a wet road) of an autonomous vehicle.
EVA	650001 The SoS shall prevent any damage on pedestrians.	600016 The SoS must be able to detect a pedestrian when a pedestrian is at 14m on a dry road (19m on a wet road) from an autonomous vehicle.
EVA	Reach destination operationalCapacity User Dispatch < Dispatch2User	init setDestination sendDestination ReceiveTrajectories NoTrajectoryReceived selectTrajectory available trajectory NoTrajectory confirm Trajectory Receive trajectory confirmation NoTrajectoryConfirmation
HIPAA	1245 : System will implement session timeouts and use cookies to terminate an electronic session	113 : Automatic logoff Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.
HIPAA	59104 : Electronic session must terminate after a pre determined period of inactivity Administrator must be able to specify this period	113 : Automatic logoff Implement electronic procedures that terminate an electronic session after a predetermined time of inactivity.
HIPAA	437 : Without data standards that promote compatibility and interoperability longitudinal patient medical records may be incomplete or of questionable integrity	111 Access Control Implement technical policies and procedures for electronic information systems that maintain electronic protected health information to allow access only to those persons or software programs that have been granted access rights as specified in 164 308 a 4.
Icebreaker	R9045 A list of trucks in the fleet shall be maintained.	Truck Repair Log getTruckID setTruckID getMaintenanceDate getMaintenanceHistory truckID setMaintenanceDate setRepairType getRepairType date setDate repairType getDate
ARC-IT2	R2539 The vehicle shall present information to the driver in audible or visual forms without impairing the driver 's ability to control the vehicle in a safe manner	F405 VehicleControlWarning Monitors areas around the vehicle and provides warnings to a driver so the driver can take action to recover and maintain safe control of the vehicle
CM1-NASA	51234 The DATA PROCESSING UNIT Command and Control Module Computer Software Component shall be able to count a consecutively reported error When the the count for a particular error ID exceeds 250 for a particular reporting period the error code will be replaced with a error code sequence which shall include the original error code and the number of times the error was reported	51222 Public Functions This routine is called by any Computer Software Component in order to report an error or event that should be included in DATA PROCESSING UNIT housekeeping. If this routine is called from interrupt context a static global variable Command and Control Module Computer Software Component Error is set so that the error can be place in a queue later (see Command and Control Module Computer Software Component Control Task). This is done since the error event queue is semaphore protected and a semaphore cannot be taken in. The error queue semaphore has priority inversion set to reduce conflicts between multiple callers should a priority inversion situation arise. This routine also replaces frequently occuring errors with a special repeat error code. The repeat error code is a special error code that follows a normally reported error code to indicate that the normally reported error code previously reported has occurred more than once in the last high rate reporting period
ARC-IT2	R6396 The center shall aggregate updates to rules, regulations, and statutes in order to define updates to be sent to vehicles and other mobile devices.	F474 TransportationInformationCenter Traffic Regulation Dissemination disseminates rules, regulations, and statutes that govern motor vehicle operation.
Easyclinic	70111 Login test case by Patient Data C 61 Card Hospital recorded 20 06 2003 in SIO and PIN proper higher figures Version 0 02 000 Use Case He performs the functions Use CaseLogPatient necessary to authenticate an patient High Priority The Hospital set up 00 001 Card is recorded in ISO Description test Hospital Input Card 00 001 PIN 65323555 Oracle Invalid Input PIN Classes cover valid CE2 Classes invalid CE9.	3025 Validation patient He works to perform the functions necessary to authenticate a patient. It s Interest to log on to see your information. The patient must be in possession of Hospital Card. The patient is not recognized system Success. The patient is recognized by system and access to its section Patient access to Box Tower View the mask for the operation login input the code Hospital Card and the code PCS Confirm validation Verify that the couple code Hospital Card and PCS identifies a patient within SIO Patientcancels function validation ends the execution of use case with failure Validation Failed Show appropriate message and Reinserts running from step 1.
ARC-IT2	R255 The center shall provide the collected border activities statistics data to archived data and planning systems.	F188 BorderInspectionAdministration performs administrative functions relating to the inspection of goods and vehicles at the border.

C – Liste des abréviations

Abbréviations	Définition
MBSE	Model-Based System Engineering
RI	Recherche d'information
ATLaS	Aggregation Trace Links Support
EVA	Éco-mobilité par Véhicules Autonomes
IRT SystemX	Institut de recherche SystemX
UML	Unified Modeling Language
OSLC	Open Services for Lifecycle Collaboration
IDM	Ingénierie Dirigée par les Modèles
Pré-RS	Pré-Requirement Specification
Post-RS	Post-Requirement Specification
CMMI	Capability Maturity Model Integration
TF-IDF	Term frequency - inverse document frequency
VSM	Vector Space Model
PN	Probabilistic Network
EBT	Event-based traceability
RTOM	Requirement-to-object-model
IREQ	Inter-Requirements
EMF	Eclipse Modeling Framework
LSI	Latent Semantic Indexing
LDA	Latent Dirichlet Allocation
RNN	Recurrent Neural Network
LSTM	Long Short Term Memory
GRU	Gated Recurrent Unit
BI-GRU	Bidirectional Gated Recurrent Unit
TAL	traitement automatique de la langue
HIPAA	Healthcare Insurance Portability and Accountability Act
COEST	Center of Excellence for Software & Systems Traceability
CBOW	continuous bag of words
GLoVe	Global Vectors
WMD	Word Mover's Distance
SIF	Smooth Inverse Frequency
SVD	Singular Value Decomposition
PCA	Principal Component Analysis
DBOW	Distributed Bag of Words
DM	Distributed Memory
RDF	Resource Description Framework
ARC-IT	Architecture Reference for Cooperative and Intelligent Transportation
NASA	National Aeronautics and Space Administration
STI	Intelligent Transportation Systems
INCOSE	International Council on Systems Engineering

Bibliographie

[1] S. Winkler & J. von Pilgrim ; «A survey of traceability in requirements engineering and model-driven development» ; Software & Systems Modeling **9**, p. 529–565 (2010). vii, 18, 19, 20, 22, 23

[2] K. Pohl ; *Requirements engineering : fundamentals, principles, and techniques* (Springer Publishing Company, Incorporated) (2010). vii, 21

[3] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol & J. Maletic ; «Software and Systems Traceability» ; p. 343–409 (Springer) (2012). vii, 34, 35

[4] J. Pennington, R. Socher & C. Manning ; «Glove : Global vectors for word representation» ; dans «Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)», p. 1532–1543 (2014). vii, 53, 54, 55, 81

[5] wikipedia Community. vii, 67

[6] «ISO/IEC 15288 :2008, Systems Engineering – System Life-Cycle Processes» ; (2008). http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=63711. 3

[7] A. Ziani, B. Hamid & J. Bruel ; «A Model-Driven Engineering Framework for Fault Tolerance in Dependable Embedded Systems Design» ; dans «Proc. 38th Euromicro Conf. Software Engineering and Advanced Applications», p. 166–169 (2012) ; ISSN 1089-6503. 4

[8] F. Scippacercola, R. Pietrantuono, S. Russo & A. Zentai ; «Model-driven engineering of a railway interlocking system» ; dans «Proc. 3rd Int. Conf. Model-Driven Engineering and Software Development (MODELSWARD)», p. 509–519 (2015). 4

[9] S. Chong, C. Wong, H. Jia, H. Pan, P. Moore, R. Kalawsky & J. O'Brien ; «Model Driven System Engineering for vehicle system utilizing Model Driven Architecture approach and hardware-in-the-loop simulation» ; dans «Proc. IEEE Int. Conf. Mechatronics and Automation», p. 1451–1456 (2011) ; ISSN 2152-744X. 4

[10] M. Loew ; «Engineering Where the Most Opportunity Exists» ; (2013). http://www.engineering.com/DesignSoftware/DesignSoftwareArticles/ArticleID/5762/Engineering-Where-the-Most-Opportunity-Exists.aspx. 4

[11] S. GROUP *et al.* ; «The CHAOS Report into Project Failure, The Standish Group International Inc» ; (2014) ; available on-line at http://www.standishgroup.com/visitor/chaos.htm. 4

[12] S. Jayatilleke & R. Lai ; «A systematic review of requirements change management» ; Information and Software Technology **93**, p. 163–185 (2018). 6

[13] N. Heumesser & F. Houdek ; «Experiences in managing an automotive requirements engineering process» ; dans «Proceedings. 12th IEEE International Requirements Engineering Conference, 2004.», p. 322–327 (IEEE) (2004). 7

[14] W. Stefan & J. Pilgrim ; «A Survey of Traceability in Requirements Engineering and Model-driven Development» ; Softw. Syst. Model. **9**, p. 529–565 (2010). ISSN 1619-1366. http://dx.doi.org/10.1007/s10270-009-0145-0. 7, 39, 61

[15] S. Nair, J. L. De La Vara & S. Sen ; «A review of traceability research at the requirements engineering conference re@ 21» ; dans «2013 21st IEEE International Requirements Engineering Conference (RE)», p. 222–229 (IEEE) (2013). 7

[16] G. Spanoudakis & A. Zisman ; «Software traceability : a roadmap» ; dans «Handbook Of Software Engineering And Knowledge Engineering : Vol 3 : Recent Advances», p. 395–428 (World Scientific) (2005). 7, 22

[17] N. Mustafa & Y. Labiche ; «The Need for Traceability in Heterogeneous Systems : A Systematic Literature Review» ; dans «2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)», , tome 01p. 305–310 (2017) ; ISSN 0730-3157. doi.ieeecomputersociety.org/10.1109/COMPSAC.2017.237. 7

[18] S. Maro, J.-P. Steghöfer & M. Staron ; «Software traceability in the automotive domain : Challenges and solutions» ; Journal of Systems and Software **141**, p. 85–110 (2018). 7, 8

[19] F. Blaauboer, K. Sikkel & M. N. Aydin ; «Deciding to adopt requirements traceability in practice» ; dans «International Conference on Advanced Information Systems Engineering», p. 294–308 (Springer) (2007). 7

[20] A. Kannenberg & H. Saiedian ; «Why software requirements traceability remains a challenge» ; CrossTalk The Journal of Defense Software Engineering **22**, p. 14–19 (2009). 7

[21] E. A. Holbrook, J. H. Hayes, A. Dekhtyar & W. Li ; «A study of methods for textual satisfaction assessment» ; Empirical Software Engineering **18**, p. 139–176 (2013). 7, 23, 24, 25, 26

[22] B. Ramesh, C. Stubbs, T. Powers & M. Edwards ; «Requirements traceability : Theory and practice» ; Annals of Software Engineering **3**, p. 397–415 (1997). ISSN 1573-7489. https://doi.org/10.1023/A:1018969401055. 7, 22

[23] R. M. Parizi, S. P. Lee & M. Dabbagh ; «Achievements and challenges in state-of-the-art software traceability between test and code artifacts» ; IEEE Transactions on Reliability **63**, p. 913–926 (2014). 8

[24] P. Rempel, P. Mäder, T. Kuschke & J. Cleland-Huang ; «Mind the gap : assessing the conformance of software traceability to relevant guidelines» ; dans «Proceedings of the 36th International Conference on Software Engineering», p. 943–954 (ACM) (2014). 8

[25] B. Dowdeswell, R. Sinha & E. Haemmerle ; «TORUS : Tracing complex requirements for large cyber-physical systems» ; dans «2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS)», p. 23–32 (IEEE) (2016). 8

[26] P. Arkley & S. Riddle ; «Overcoming the traceability benefit problem» ; dans «13th IEEE International Conference on Requirements Engineering (RE'05)», p. 385–389 (IEEE) (2005). 8

[27] B. Berenbach, D. Gruseman & J. Cleland-Huang ; «Application of just in time tracing to regulatory codes» ; dans «Proceedings of the conference on systems engineering research», (2010). 9

[28] R. A. Pierce ; «A requirements tracing tool» ; ACM SIGMETRICS Performance Evaluation Review **7**, p. 53–60 (1978). 17

[29] I. R. D. Family. 17

[30] B. Amar, H. Leblanc & B. Coulette ; «A traceability engine dedicated to model transformation for software engineering» ; dans «ECMDA Traceability Workshop (ECMDA-TW)», p. 7–16 (2008). 18

[31] F. Jouault ; «Loosely coupled traceability for ATL» ; dans «Proceedings of the European Conference on Model Driven Architecture (ECMDA) workshop on traceability, Nuremberg, Germany», , tome 91p. 2 (2005). 18

[32] I. Galvao & A. Goknil ; «Survey of traceability approaches in model-driven engineering» ; dans «11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)», p. 313–313 (IEEE) (2007). 18

[33] J.-R. Falleri, M. Huchard & C. Nebut ; «Towards a traceability framework for model transformations in kermeta» ; dans «ECMDA-TW'06 : ECMDA Traceability Workshop», p. 31–40 (Sintef ICT, Norway) (2006). 18

[34] V. Katta & T. Stålhane ; «Traceability of safety systems : approach, meta-model and tool support» ; Technical Report hwr-1053, Institute for Energy Technology (2012). 18

[35] M. Taromirad, N. D. Matragkas & R. F. Paige ; «Towards a Multi-Domain Model-Driven Traceability Approach.» ; dans «MPM@ MoDELS», p. 27–36 (2013). 18

[36] J. Guo, N. Monaikul, C. Plepel & J. Cleland-Huang ; «Towards an intelligent domain-specific traceability solution» ; dans «Proceedings of the 29th ACM/IEEE international conference on Automated software engineering», p. 755–766 (ACM) (2014). 18, 37

[37] S. J. Herzig & C. J. Paredis ; «A conceptual basis for inconsistency management in model-based systems engineering» ; Procedia CIRP **21**, p. 52–57 (2014). 18

[38] H. Giese, T. Levendovszky & H. Vangheluwe ; «Summary of the workshop on multi-paradigm modeling : concepts and tools» ; dans «International Conference on Model Driven Engineering Languages and Systems», p. 252–262 (Springer) (2006). 18

[39] O. Community. 19

[40] I. Board ; «Ieee standard ISO 24765 Systems and software engineering — Vocabulary» ; New York : The Institute of Electrical and Electronics Ehgineers (2010). 19

[41] I. Board ; «IEEE Std 1362-1998 (R2007) IEEE Guide for Information Technology-System Definition -Concept of Operation Document.3.24» ; C/S2ESC - Software and Systems Engineering Standards Committee (1998). 19

[42] O. C. Z. Gotel & A. C. W. Finkelstein ; dans «An Analysis of the Requirements Traceability Problem», p. 94–101 (1994). 19

[43] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin & Y. Shaham-Gafni ; «Model traceability» ; IBM Systems Journal **45**, p. 515–526 (2006). 19

[44] M. Edwards & S. L. Howell ; «A methodology for systems requirements specification and traceability for large real time complex systems» ; Rapport technique ; NAVAL SURFACE WARFARE CENTER SILVER SPRING MD (1991). 20

[45] G. O'Regan ; «Capability Maturity Model Integration Capability Maturity Model Integration» ; dans «Introduction to Software Process Improvement», p. 43–65 (Springer) (2011). 20

[46] sebokwiki. 21

[47] O. Gotel, J. Cleland-Huang, J. H. Hayes, A. Zisman, A. Egyed, P. Grünbacher, A. Dekhtyar, G. Antoniol, J. Maletic & P. Mäder ; «Traceability fundamentals» ; dans «Software and Systems Traceability», p. 3–22 (Springer) (2012). 22

[48] G. Spanoudakis, A. Zisman, E. Pérez-Minana & P. Krause ; «Rule-based generation of requirements traceability relations» ; Journal of Systems and Software **72**, p. 105–127 (2004). 22

[49] M. Thakur, B. J. Martens & C. R. Hurburgh ; «Data modeling to facilitate internal traceability at a grain elevator» ; Computers and electronics in agriculture **75**, p. 327–336 (2011). 22

[50] M. Taromirad, N. D. Matragkas & R. F. Paige ; «Towards a Multi-Domain Model-Driven Traceability Approach.» ; dans «MPM@ MoDELS», p. 27–36 (2013). 22, 37, 63

[51] S. Khouri, K. Semassel & L. Bellatreche ; «Managing data warehouse traceability : a life-cycle driven approach» ; dans «International Conference on Advanced Information Systems Engineering», p. 199–213 (Springer) (2015). 22

[52] A. Espinoza, P. P. Alarcon & J. Garbajosa ; «Analyzing and systematizing current traceability schemas» ; dans «Software Engineering Workshop, 2006. SEW'06. 30th Annual IEEE/NASA», p. 21–32 (IEEE) (2006). 22

[53] S. Lehnert ; «A taxonomy for software change impact analysis» ; dans «Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution», p. 41–50 (ACM) (2011). 22

[54] G. Salton, A. Wong & C.-S. Yang ; «A vector space model for automatic indexing» ; Communications of the ACM **18**, p. 613–620 (1975). 23, 38

[55] G. A. Cysneiros, F. Andrea & Z. G. Spanoudakis ; «Traceability approach for i* and uml models» ; Citeseer (2003). 25, 37

[56] G. Spanoudakis, A. S. d. Garcez & A. Zisman ; «Revising Rules to Capture Requirements Traceability Relations : A Machine Learning Approach.» ; dans «SEKE», p. 570–577 (2003). 25, 37

[57] A. Zisman, G. Spanoudakis, E. Pérez-Miñana & P. Krause ; «Towards a traceability approach for product families requirements» ; dans «Proceedings of 3rd ICSE Workshop on Software Product Lines : Economics, Architectures, and Implications», p. 19–25 (2002). 25, 37

[58] «author=Goknil, Arda and Kurtev, Ivan and Van Den Berg, Klaas, booktitle=ECMDA Traceability Workshop (ECMDA-TW), pages=59–75, year=2008, organization=SINTEF Report» ; . 28

[59] S. Riddle & P. Arkley ; «Overcoming the Traceability Benefit Problem» ; dans «13th IEEE International Conference on Requirements Engineering (RE'05)(RE)», , tome 00p. 385–389 (2005). doi.ieeecomputersociety.org/10.1109/RE.2005.49. 34, 43

[60] P. Mäder & O. Gotel ; «Ready-to-Use Traceability on Evolving Projects» ; dans «Software and Systems Traceability», p. 173–194 (Springer) (2012). 34

[61] J. Cleland-Huang, O. C. Gotel, J. Huffman Hayes, P. Mäder & A. Zisman ; «Software traceability : trends and future directions» ; dans «Proceedings of the on Future of Software Engineering», p. 55–69 (ACM) (2014). 35, 36, 43

[62] P. Mäder & O. Gotel ; «Ready-to-use traceability on evolving projects» ; dans «Software and Systems Traceability», p. 173–194 (Springer) (2012). 35, 36

[63] J. H. Hayes & A. Dekhtyar ; «A framework for comparing requirements tracing experiments» ; International Journal of Software Engineering and Knowledge Engineering **15**, p. 751–781 (2005). 36

[64] Y. Shin & J. Cleland-Huang ; «A comparative evaluation of two user feedback techniques for requirements trace retrieval» ; dans «Proceedings of the 27th Annual ACM Symposium on Applied Computing», p. 1069–1074 (ACM) (2012). 36

[65] J. H. Hayes, A. Dekhtyar & S. K. Sundaram ; «Advancing candidate link generation for requirements tracing : The study of methods» ; IEEE Transactions on Software Engineering **32**, p. 4 (2006). 36, 106

[66] S. Winkler ; «Trace retrieval for evolving artifacts» ; dans «Proceedings of the 2009 ICSE Workshop on Traceability in Emerging Forms of Software Engineering», p. 49–56 (IEEE Computer Society) (2009). 36

[67] A. Ghabi & A. Egyed ; «Code patterns for automatically validating requirements-to-code traces» ; dans «Automated Software Engineering (ASE), 2012 Proceedings of the 27th IEEE/ACM International Conference on», p. 200–209 (IEEE) (2012). 36

[68] J. Cleland-Huang, C. K. Chang & M. Christensen ; «Event-Based Traceability for Managing Evolutionary Change» ; IEEE Trans. Softw. Eng. **29**, p. 796–810 (2003). ISSN 0098-5589. http://dx.doi.org/10.1109/TSE.2003.1232285. 37

[69] J. Cleland-Huang, R. Settimi, C. Duan & X. Zou ; «Utilizing supporting evidence to improve dynamic requirements traceability» ; dans «Requirements Engineering, 2005. Proceedings. 13th IEEE International Conference on», p. 135–144 (IEEE) (2005). 37, 44

[70] J. Cleland-Huang, C. K. Chang & M. Christensen ; «Event-based traceability for managing evolutionary change» ; IEEE Transactions on Software Engineering **29**, p. 796–810 (2003). 37

[71] A. Goknil, I. Kurtev & K. Van Den Berg ; «Change impact analysis based on formalization of trace relations for requirements» ; dans «ECMDA Traceability Workshop (ECMDA-TW)», p. 59–75 (SINTEF Report) (2008). 37

[72] J. C. Martins & R. J. Machado ; «Ontologies for product and process traceability at manufacturing organizations : a software requirements approach» ; dans «Quality of Information and Communications Technology (QUATIC), 2012 Eighth International Conference on the», p. 353–358 (IEEE) (2012). 37

[73] J. Guo, J. Cleland-Huang & B. Berenbach ; «Foundations for an expert system in domain-specific traceability» ; dans «Requirements Engineering Conference (RE), 2013 21st IEEE International», p. 42–51 (IEEE) (2013). 37, 44

[74] D. Kolovos, R. Paige & F. Polack ; «Detecting and repairing inconsistencies across heterogeneous models» ; dans «Software Testing, Verification, and Validation, 2008 1st International Conference on», p. 356–364 (IEEE) (2008). 37

[75] B. Ramesh & M. Jarke ; «Toward reference models for requirements traceability» ; IEEE transactions on software engineering p. 58–93 (2001). 37

[76] G. Zemont ; «Towards value-based requirements traceability» ; Department of Computer Science p. 80 (2005). 37

[77] A. Egyed, P. Grunbacher, M. Heindl & S. Biffl ; «Value-based requirements traceability : Lessons learned» ; dans «Requirements Engineering Conference, 2007. RE'07. 15th IEEE International», p. 115–118 (IEEE) (2007). 37

[78] J. Guo, J. Cheng & J. Cleland-Huang ; «Semantically enhanced software traceability using deep learning techniques» ; dans «Software Engineering (ICSE), 2017 IEEE/ACM 39th International Conference on», p. 3–14 (IEEE) (2017). 38, 40, 41, 43, 48, 50, 55

[79] T. Zhao, Q. Cao & Q. Sun ; «An Improved Approach to Traceability Recovery Based on Word Embeddings» ; dans «Asia-Pacific Software Engineering Conference (APSEC), 2017 24th», p. 81–89 (IEEE) (2017). 38, 55

[80] B. Xu, D. Ye, Z. Xing, X. Xia, G. Chen & S. Li ; «Predicting semantically linkable knowledge in developer online forums via convolutional neural network» ; dans «Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering», p. 51–62 (ACM) (2016). 38, 55

[81] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer & R. Harshman ; «Indexing by latent semantic analysis» ; Journal of the American society for information science **41**, p. 391–407 (1990). 38, 61

[82] M. Kleffmann, S. Röhl, V. Gruhn & M. Book ; «Establishing and navigating trace links between elements of informal diagram sketches» ; dans «Software and Systems Traceability (SST), 2015 IEEE/ACM 8th International Symposium on», p. 1–7 (IEEE) (2015). 38, 62

[83] V. I. Levenshtein ; «Binary codes capable of correcting deletions, insertions, and reversals» ; dans «Soviet physics doklady», , tome 10p. 707–710 (1966). 38

[84] R. A. Wagner & M. J. Fischer ; «The string-to-string correction problem» ; Journal of the ACM (JACM) **21**, p. 168–173 (1974). 38

[85] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand & F. Zimmer ; «Change impact analysis for natural language requirements : An NLP approach» ; dans «Requirements Engineering Conference (RE), 2015 IEEE 23rd International», p. 6–15 (IEEE) (2015). 38, 39

[86] S. Hotomski & M. Glinz ; «GuideGen : a tool for keeping requirements and acceptance tests aligned» ; dans «Proceedings of the 40th International Conference on Software Engineering : Companion Proceeedings», p. 49–52 (ACM) (2018). 38

[87] M. Borg, P. Runeson & A. Ardö ; «Recovering from a decade : a systematic mapping of information retrieval approaches to software traceability» ; Empirical Software Engineering **19**, p. 1565–1616 (2014). 39

[88] M. Anas & D. Carver ; «Exploiting online human knowledge in Requirements Engineering» ; dans «Proceedings of the 23rd IEEE International Requirements Engineering Conference (RE)», (2015). 39

[89] J. Sreedhar, S. V. Raju, A. V. Babu, A. Shaik & P. P. Kumar ; «Word sense disambiguation : An empirical survey» ; International Journal of Soft Computing and Engineering (IJSCE) ISSN p. 2231–2307 (2012). 39, 41

[90] H. U. Asuncion, A. U. Asuncion & R. N. Taylor ; «Software traceability with topic modeling» ; dans «Software Engineering, 2010 ACM/IEEE 32nd International Conference on», , tome 1p. 95–104 (IEEE) (2010). 39, 63, 64

[91] L. V. Galvis Carreño & K. Winbladh ; «Analysis of user comments : an approach for software requirements evolution» ; dans «Proceedings of the 2013 International Conference on Software Engineering», p. 582–591 (IEEE Press) (2013). 39, 63, 64

[92] A. Panichella, B. Dit, R. Oliveto, M. Di Penta, D. Poshyvanyk & A. De Lucia ; «How to effectively use topic models for software engineering tasks ? an approach based on genetic algorithms» ; dans «Proceedings of the 2013 International Conference on Software Engineering», p. 522–531 (IEEE Press) (2013). 39, 44, 63, 64

[93] X. Chen, J. Hosking & J. Grundy ; «A combination approach for enhancing automated traceability :(NIER track)» ; dans «Proceedings of the 33rd IEEE International Software Engineering (ICSE)», (2011). 39, 61

[94] N. Niu & A. Mahmoud ; «Enhancing candidate link generation for requirements tracing : the cluster hypothesis revisited» ; dans «2012 20th IEEE International Requirements Engineering Conference (RE)», p. 81–90 (IEEE) (2012). 39, 61, 63, 76

[95] K. Canini, L. Shi & T. Griffiths ; «Online inference of topics with latent Dirichlet allocation» ; dans «Artificial Intelligence and Statistics», p. 65–72 (2009). 39, 64

[96] J. MacQueen *et al.* ; «Some methods for classification and analysis of multivariate observations» ; dans «Proceedings of the fifth Berkeley symposium on mathematical statistics and probability», , tome 1p. 281–297 (Oakland, CA, USA) (1967). 39

[97] D. Bahdanau, K. Cho & Y. Bengio ; «Neural machine translation by jointly learning to align and translate» ; arXiv preprint arXiv :1409.0473 (2014). 40

[98] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado & J. Dean ; «Distributed representations of words and phrases and their compositionality» ; dans «Advances in neural information processing systems», p. 3111–3119 (2013). 40, 44, 50, 80

[99] S. Hochreiter & J. Schmidhuber ; «Long short-term memory» ; Neural computation **9**, p. 1735–1780 (1997). 40

[100] J. Chung, C. Gulcehre, K. Cho & Y. Bengio ; «Empirical evaluation of gated recurrent neural networks on sequence modeling» ; arXiv preprint arXiv :1412.3555 (2014). 40

[101] X. J. Zhu ; «Semi-supervised learning literature survey» ; Rapport technique ; University of Wisconsin-Madison Department of Computer Sciences (2005). 40

[102] M. Seeger ; «Learning with Labeled and Unlabeled Data» ; Rapport technique (2001). 40, 41, 43, 77

[103] M. Osborne & C. MacNish ; «Processing natural language software requirement specifications» ; dans «Requirements Engineering, 1996., Proceedings of the Second International Conference on», p. 229–236 (IEEE) (1996). 43

[104] J. Cleland-Huang, M. Rahimi & P. Mäder ; «Achieving Lightweight Trustworthy Traceability» ; dans «Proceedings of the 22Nd ACM SIGSOFT International Symposium on Foundations of Software Engineering», FSE 2014 ; p. 849–852 (ACM) (2014) ; ISBN 978-1-4503-3056-5. http://doi.acm.org/10.1145/2635868.2666612. 43

[105] A. Mahmoud, N. Niu & S. Xu ; «A semantic relatedness approach for traceability link recovery» ; dans «2012 20th IEEE international conference on program comprehension (ICPC)», p. 183–192 (IEEE) (2012). 44

[106] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia & E. Merlo ; «Recovering traceability links between code and documentation» ; IEEE transactions on software engineering **28**, p. 970–983 (2002). 44, 61

[107] S. Lohar, S. Amornborvornwong, A. Zisman & J. Cleland-Huang ; «Improving trace accuracy through data-driven configuration and composition of tracing features» ; dans «Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering», p. 378–388 (ACM) (2013). 44

[108] C. Mills & S. Haiduc ; «A Machine Learning Approach for Determining the Validity of Traceability Links» ; dans «Proceedings of the 39th International Conference on Software Engineering Companion», ICSE-C '17 ; p. 121–123 (IEEE Press) (2017) ; ISBN 978-1-5386-1589-8. https://doi.org/10.1109/ICSE-C.2017.86. 44

[109] K. Divya, R. Subha & S. Palaniswami ; «Similar Words Identification Using Naive and TF-IDF Method» ; International Journal of Information Technology and Computer Science (IJITCS) **6**, p. 42 (2014). 44

[110] L. Kof, R. Gacitua, M. Rouncefield & P. Sawyer ; «Concept mapping as a means of requirements tracing» ; dans «Managing Requirements Knowledge (MARK), 2010 Third International Workshop on», p. 22–31 (IEEE) (2010). 44

[111] A. De Nicola, M. Missikoff & R. Navigli ; «A software engineering approach to ontology building» ; Information systems **34**, p. 258–275 (2009). 44

[112] R. Gacitua, P. Sawyer & P. Rayson ; «A flexible framework to experiment with ontology learning techniques» ; dans «Research and Development in Intelligent Systems XXIV», p. 153–166 (Springer) (2008). 44

[113] M. Kusner, Y. Sun, N. Kolkin & K. Weinberger ; «From word embeddings to document distances» ; dans «International Conference on Machine Learning», p. 957–966 (2015). 44, 54

[114] J. H. Martin & D. Jurafsky ; *Speech and language processing : An introduction to natural language processing, computational linguistics, and speech recognition* (Pearson/Prentice Hall Upper Saddle River) (2009). 47

[115] C. Manning, P. Raghavan & H. Schütze ; «Introduction to information retrieval» ; Natural Language Engineering **16**, p. 100–103 (2010). 48, 63, 79

[116] R. Baeza-Yates, B. d. A. N. Ribeiro *et al.* ; *Modern information retrieval* (New York : ACM Press ; Harlow, England : Addison-Wesley,) (2011). 48

[117] S. Pandanaboyana, S. Sridharan, J. Yannelli & J. H. Hayes ; «Requirements tracing on target (retro) enhanced with an automated thesaurus builder : An empirical study» ; dans «2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)», p. 61–67 (IEEE) (2013). 48

[118] M. Starets ; *Théories syntaxiques du français contemporain* (Presses Université Laval) (2000). 49

[119] W. H. Gomaa & A. A. Fahmy ; «A survey of text similarity approaches» ; International Journal of Computer Applications **68**, p. 13–18 (2013). 49

[120] C. Arora, M. Sabetzadeh, A. Goknil, L. C. Briand & F. Zimmer ; «Change impact analysis for natural language requirements : An NLP approach» ; dans «2015 IEEE 23rd International Requirements Engineering Conference (RE)», p. 6–15 (IEEE) (2015). 49

[121] R. Navigli ; «Word sense disambiguation : A survey» ; ACM computing surveys (CSUR) **41**, p. 10 (2009). 50

[122] Z. S. Harris ; «Distributional structure» ; Word **10**, p. 146–162 (1954). 50

[123] T. Mikolov, K. Chen, G. Corrado & J. Dean ; «Efficient estimation of word representations in vector space» ; arXiv preprint arXiv :1301.3781 (2013). 50, 51

[124] X. Ye, H. Shen, X. Ma, R. Bunescu & C. Liu ; «From word embeddings to document similarities for improved information retrieval in software engineering» ; dans «Proceedings of the 38th international conference on software engineering», p. 404–415 (ACM) (2016). 51, 80

[125] Y. Kim ; «Convolutional neural networks for sentence classification» ; arXiv preprint arXiv :1408.5882 (2014). 51

[126] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami & C. Dyer ; «Neural architectures for named entity recognition» ; arXiv preprint arXiv :1603.01360 (2016). 51

[127] Y. Qi, D. S. Sachan, M. Felix, S. J. Padmanabhan & G. Neubig ; «When and why are pre-trained word embeddings useful for neural machine translation ?» ; arXiv preprint arXiv :1804.06323 (2018). 51

[128] A. Abad, A. Ortega, A. J. S. Teixeira, C. García-Mateo, C. D. Martínez-Hinarejos, F. Perdigão, F. Batista & N. J. Mamede (rédacteurs) ; *Advances in Speech and Language Technologies for Iberian Languages - Third International Conference, IberSPEECH 2016, Lisbon, Portugal, November 23-25, 2016, Proceedings* ; *Lecture Notes in Computer Science*, tome 10077 (2016) ; ISBN 978-3-319-49168-4. https://doi.org/10.1007/978-3-319-49169-1. 53

[129] M. Baroni, G. Dinu & G. Kruszewski ; «Don't count, predict ! A systematic comparison of context-counting vs. context-predicting semantic vectors» ; dans «Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)», , tome 1p. 238–247 (2014). 55

[130] T. Mikolov, W.-t. Yih & G. Zweig ; «Linguistic regularities in continuous space word representations» ; dans «Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies», p. 746–751 (2013). 55

[131] S. Arora, Y. Liang & T. Ma ; «A simple but tough-to-beat baseline for sentence embeddings» ; (2016). 55, 56, 81, 88

[132] J. Wieting, M. Bansal, K. Gimpel & K. Livescu ; «Towards universal paraphrastic sentence embeddings» ; arXiv preprint arXiv :1511.08198 (2015). 55

[133] A. Conneau, D. Kiela, H. Schwenk, L. Barrault & A. Bordes ; «Supervised learning of universal sentence representations from natural language inference data» ; arXiv preprint arXiv :1705.02364 (2017). 55

[134] Q. Le & T. Mikolov ; «Distributed representations of sentences and documents» ; dans «International conference on machine learning», p. 1188–1196 (2014). 56

[135] G. Salton & C. Buckley ; «Term-weighting approaches in automatic text retrieval» ; Information processing & management **24**, p. 513–523 (1988). 59

[136] G. Salton, A. Wong & C.-S. Yang ; «A vector space model for automatic indexing» ; Communications of the ACM **18**, p. 613–620 (1975). 60

[137] R. Tsuchiya, H. Washizaki, Y. Fukazawa, K. Oshima & R. Mibe ; «Interactive recovery of requirements traceability links using user feedback and configuration management logs» ; dans «International Conference on Advanced Information Systems Engineering», p. 247–262 (Springer) (2015). 61

[138] X. Wang, G. Lai & C. Liu ; «Recovering relationships between documentation and source code based on the characteristics of software engineering» ; Electronic Notes in Theoretical Computer Science **243**, p. 121–137 (2009). 62

[139] H.-Y. Jiang, T. N. Nguyen, X. Chen, H. Jaygarl & C. K. Chang ; «Incremental latent semantic indexing for automatic traceability link evolution management» ; dans «Proceedings of the 2008 23rd IEEE/ACM International Conference on Automated Software Engineering», p. 59–68 (IEEE Computer Society) (2008). 62

[140] N. Alhindawi, O. Meqdadi, B. Bartman & J. I. Maletic ; «A tracelab-based solution for identifying traceability links using LSI» ; dans «2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)», p. 79–82 (IEEE) (2013). 62

[141] O. Chapelle, J. Weston & B. Schölkopf ; «Cluster kernels for semi-supervised learning» ; dans «Advances in neural information processing systems», p. 601–608 (2003). 63

[142] X. Zhu, Z. Ghahramani & J. D. Lafferty ; «Semi-supervised learning using gaussian fields and harmonic functions» ; dans «Proceedings of the 20th International conference on Machine learning (ICML-03)», p. 912–919 (2003). 63

[143] A. Leuski ; «Evaluating document clustering for interactive information retrieval» ; dans «Proceedings of the tenth international conference on Information and knowledge management», p. 33–40 (ACM) (2001). 63

[144] X. Chen & J. Grundy ; «Improving automated documentation to code traceability by combining retrieval techniques» ; dans «Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering», p. 223–232 (IEEE Computer Society) (2011). 63, 76

[145] C. Duan & J. Cleland-Huang ; «Clustering support for automated tracing» ; dans «Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering», p. 244–253 (ACM) (2007). 63, 76

[146] S.-i. Amari & H. Nagaoka ; *Methods of information geometry* ; tome 191 (American Mathematical Soc.) (2007). 64

[147] D. Zhou, O. Bousquet, T. N. Lal, J. Weston & B. Schölkopf ; «Learning with local and global consistency» ; dans «Advances in neural information processing systems», p. 321–328 (2004). 65

[148] T. M. Mitchell ; «Machine learning and data mining» ; Communications of the ACM **42** (1999). 74

[149] D. Jurasky & J. H. Martin ; «Speech and Language Processing : An introduction to natural language Processing» ; Computational Linguistics and Speech Recognition, Prentice Hall : San Francisco (2000). 79

[150] E. A. Holbrook, J. H. Hayes, A. Dekhtyar & W. Li ; «A study of methods for textual satisfaction assessment» ; Empirical Software Engineering **18**, p. 139–176 (2013). 81

[151] K. Divya, R. Subha & S. Palaniswami ; «Similar Words Identification Using Naive and TF-IDF Method» ; International Journal of Information Technology and Computer Science (IJITCS) **6**, p. 42 (2014). 81

[152] C. A. Petri ; «Communication with automata» ; (1966). 87

[153] L. Wouters, S. Creff, E. E. Bella & A. Koudri ; «Collaborative systems engineering : Issues & challenges» ; dans «2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)», p. 486–491 (IEEE) (2017). 87

[154] L. Wouters, S. Creff, E. E. Bella & A. Koudri ; «Towards Semantic-Aware Collaborations in Systems Engineering» ; dans «2017 24th Asia-Pacific Software Engineering Conference (APSEC)», p. 719–724 (IEEE) (2017). 87

[155] J. Cleland-Huang, A. Czauderna, A. Dekhtyar, O. Gotel, J. H. Hayes, E. Keenan, G. Leach, J. Maletic, D. Poshyvanyk, Y. Shin *et al.* ; «Grand challenges, benchmarks, and TraceLab : developing infrastructure for the software traceability research community» ; dans «Proceedings of the 6th international workshop on traceability in emerging forms of software engineering», p. 17–23 (ACM) (2011). 88

[156] S. M. Stigler ; «Francis Galton's account of the invention of correlation» ; Statistical Science p. 73–79 (1989). 94

[157] E. Michael & S. L. Howell ; «A Methodology for Systems Requirements Specification and Traceability for Large Real Time Complex Systems» ; (1992).

[158] E. Effa Bella, M.-P. Gervais, R. Bendraou, L. Wouters & K. Ali ; «Semi-supervised Approach for Recovering Traceability Links in Complex Systems» ; dans

«Proceedings of the 23rd IEEE International Conference In the Engineering of Complex Computer Systems (ICECCS)», (2018).

[159] X. Blanc, A. Mougenot, I. Mounier & T. Mens ; «Incremental Detection of Model Inconsistencies Based on Model Operations» ; dans P. van Eck, J. Gordijn & R. Wieringa (rédacteurs), «Advanced Information Systems Engineering», p. 32–46 (Springer Berlin Heidelberg, Berlin, Heidelberg) (2009) ; ISBN 978-3-642-02144-2.

[160] N. Aizenbud-Reshef, B. T. Nolan, J. Rubin & Y. Shaham-Gafni ; «Model Traceability» ; IBM Syst. J. **45**, p. 515–526 (2006). ISSN 0018-8670. http://dx.doi.org/10.1147/sj.453.0515.

[161] I. Galvão & A. Goknil ; «Survey of Traceability Approaches in Model-Driven Engineering» ; dans «EDOC», p. 313–326 (IEEE Computer Society) (2007).

[162] Y. Li & J. Cleland-Huang ; «Ontology-based trace retrieval» ; dans «2013 7th International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE)», , tome 00p. 30–36 (2013). doi.ieeecomputersociety.org/10.1109/TEFSE.2013.6620151.

[163] J. Cleland-Huang, M. Rahimi & P. Mäder ; «Achieving lightweight trustworthy traceability» ; dans «Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering», p. 849–852 (ACM) (2014).

[164] M. Kleffmann, S. Röhl, V. Gruhn & M. Book ; «Establishing and navigating trace links between elements of informal diagram sketches» ; dans «Software and Systems Traceability (SST), 2015 IEEE/ACM 8th International Symposium on», p. 1–7 (IEEE) (2015).

[165] P. Mäder, O. Gotel & I. Philippow ; «Rule-based maintenance of post-requirements traceability relations» ; dans «International Requirements Engineering, 2008. RE'08. 16th IEEE», p. 23–32 (IEEE) (2008).

[166] B. Ramesh & M. Edwards ; «Issues in the development of a requirements traceability model» ; dans «Requirements Engineering, 1993., Proceedings of IEEE International Symposium on», p. 256–259 (IEEE) (1993).

[167] R. F. Paige, G. K. Olsen, D. S. Kolovos, S. Zschaler & C. Power ; «Building model-driven engineering traceability classifications» ; Citeseer (2008).

[168] M. Heindl & S. Biffl ; «A case study on value-based requirements tracing» ; dans «Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering», p. 60–69 (ACM) (2005).

[169] S. Bendriss & A. Benabdelhafid ; «Enabling goods traceability through data modeling and semantic web service ontologies» ; dans «Logistics (LOGISTIQUA), 2011 4th International Conference on», p. 385–390 (IEEE) (2011).

[170] C. Mills & S. Haiduc ; «A machine learning approach for determining the validity of traceability links» ; dans «Proceedings of the 39th International Conference on Software Engineering Companion», p. 121–123 (IEEE Press) (2017).

[171] A. N. Lam, A. T. Nguyen, H. A. Nguyen & T. N. Nguyen ; «Combining deep learning with information retrieval to localize buggy files for bug reports (n)» ; dans «Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference on», p. 476–481 (IEEE) (2015).

[172] R. Oliveto, M. Gethers, D. Poshyvanyk & A. De Lucia ; «On the equivalence of information retrieval methods for automated traceability link recovery» ; dans «Program Comprehension (ICPC), 2010 IEEE 18th International Conference on», p. 68–71 (IEEE) (2010).

[173] R. Navigli & M. Lapata ; «An experimental study of graph connectivity for unsupervised word sense disambiguation» ; IEEE transactions on pattern analysis and machine intelligence **32**, p. 678–692 (2010).

[174] G. Bavota, L. Colangelo, A. De Lucia, S. Fusco, R. Oliveto & A. Panichella ; «TraceME : traceability management in eclipse» ; dans «Software Maintenance (ICSM), 2012 28th IEEE International Conference on», p. 642–645 (IEEE) (2012).

[175] J. I. Maletic, E. V. Munson, A. Marcus & T. N. Nguyen ; «Using a hypertext model for traceability link conformance analysis» ; dans «Proc. of the Int. Workshop on Traceability in Emerging Forms of Software Engineering», p. 47–54 (2003).

[176] L. G. Murta, A. van der Hoek & C. M. Werner ; «Continuous and automated evolution of architecture-to-implementation traceability links» ; Automated Software Engineering **15**, p. 75–107 (2008).

[177] P. Mäder & O. Gotel ; «Towards automated traceability maintenance» ; Journal of Systems and Software **85**, p. 2205–2227 (2012).

[178] S. K. M. Wong & Y. Y. Yao ; «On modeling information retrieval with probabilistic inference» ; ACM Transactions on Information Systems (TOIS) **13**, p. 38–68 (1995).

[179] S. Maro & J.-P. Steghofer ; «Capra : A configurable and extendable traceability management tool» ; dans «2016 IEEE 24th International Requirements Engineering Conference (RE)», p. 407–408 (IEEE) (2016).

[180] M. Y. Haouam & D. Meslati ; «Towards Automated Traceability Maintenance in Model Driven Engineering» ; IAENG International Journal of Computer Science **43**, p. 147–155 (2016).

[181] N. Drivalos-Matragkas, D. S. Kolovos, R. F. Paige & K. J. Fernandes ; «A state-based approach to traceability maintenance» ; dans «Proceedings of the 6th ECMFA Traceability Workshop», p. 23–30 (ACM) (2010).

[182] J. Sztipanovits & G. Karsai ; «Model-integrated computing» ; Computer **30**, p. 110–111 (1997).

[183] X. Zou, R. Settimi & J. Cleland-Huang ; «Improving automated requirements trace retrieval : a study of term-based enhancement methods» ; Empirical Software Engineering **15**, p. 119–146 (2010).

[184] W. B. Frakes & R. Baeza-Yates ; *Information retrieval : Data structures & algorithms* ; tome 331 (prentice Hall Englewood Cliffs, NJ) (1992).

[185] M. Buckland & F. Gey ; «The relationship between recall and precision» ; Journal of the American society for information science **45**, p. 12–19 (1994).

[186] R. Bellman ; «Dynamic programming» ; Science **153**, p. 34–37 (1966).

[187] E. Effa Bella, S. Creff, M.-P. Gervais & R. Bendraou ; «ATLaS : A Framework for Traceability Links Recovery Combining Information Retrieval and Semi-supervised

Techniques» ; dans «Proceedings of the 23rd IEEE International Conference In the Enterprise Computing Conference (EDOC)», (2019).

[188] C. M. Chang ; *Service systems management and engineering : Creating strategic differentiation and operational excellence* (John Wiley & Sons) (2010).

[189] L. Wouters ; *Multi-domain expert-user modeling infrastructure* ; Thèse de doctorat ; Paris 6 (2013).

[190] cenotelie.

[191] E. M. Voorhees ; «The effectiveness and efficiency of agglomerative hierarchic clustering in document retrieval» ; Rapport technique ; Cornell University (1985).