

Modèle de données générique et cartographie
des données du marché financier(carnet d'ordres)

par

Franceska DORVAL

RAPPORT DE PROJET PRÉSENTÉ À L'ÉCOLE DE TECHNOLOGIE
SUPÉRIEURE COMME EXIGENCE PARTIELLE À L'OBTENTION DE
LA MAÎTRISE EN GÉNIE DES TECHNOLOGIES DE L'INFORMATION

MONTREAL, LE 25 MARS 2019 AU BUREAU DES CYCLES
SUPÉRIEURS

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC



Franceska Dorval, 2019



Cette licence [Creative Commons](https://creativecommons.org/licenses/by-nc-nd/4.0/) signifie qu'il est permis de diffuser, d'imprimer ou de sauvegarder sur un autre support une partie ou la totalité de cette œuvre à condition de mentionner l'auteur, que ces utilisations soient faites à des fins non commerciales et que le contenu de l'œuvre n'ait pas été modifié.

PRÉSENTATION DU JURY

CE RAPPORT DE PROJET A ÉTÉ ÉVALUÉ

PAR UN JURY COMPOSÉ DE :

Professeur Alain April, directeur de projet
Département de génie logiciel et TI à l'École de technologie supérieure

Professeur Abdelaoued Gherby, jury
Département de génie logiciel et TI à l'École de technologie supérieure

AVANT-PROPOS

L'intégration de données se révèle une tâche ardue, car elle nécessite de longues heures d'analyses et de traitements. Pour effectuer cette tâche, il faut comprendre, en détail, les données qui se trouvent dans chacun des champs et ensuite faire une correspondance avec différents formats en sortie. Les entreprises ne disposent pas souvent de dictionnaire de données qui permettent d'expliquer la structure, les relations et sens de chaque donnée. Ainsi, le modèle conceptuel des données reste dans la tête des membres d'équipes qui ont initié le projet et les noms des champs ne sont pas souvent significatifs pour d'autres intervenants.

Dans le domaine de la finance, il n'existe aucune norme de publication des flux de données de la bourse. Ces données sont produites indépendamment par chaque bourse et sont hétérogènes. Rassembler les données boursières de plusieurs bourses est un défi important d'intégration de données.

Ce projet de recherche s'intéresse à la conception d'un prototype logiciel qui permet d'effectuer la correspondance avec peu ou sans interventions humaines et concevoir un modèle de données générique. À cet effet, ce prototype logiciel fait usage de techniques provenant du traitement du langage naturel afin de déterminer automatiquement les propriétés morphologiques des mots. Ces algorithmes se basent sur divers types d'approches dans le but d'effectuer la correspondance de différents schémas. Ensuite, il définit et implémente l'architecture du processus d'extraction, de transformation et du chargement des données (ETC) dans le but de les charger dans le modèle générique de données et visualiser le carnet d'ordres boursier.

REMERCIEMENTS

Je tiens d'abord à remercier le professeur Alain April d'avoir accepté d'être mon directeur de projet de recherche. Je lui adresse encore de vifs remerciements de m'avoir offert cette opportunité de travailler sur ce projet industriel. Celui-ci m'ouvre de nouveaux horizons, en me permettant d'utiliser des outils et de travailler dans des secteurs porteurs d'opportunités et de défis notamment « le traitement de données massives en informatique, intégration de données, gestion des ordres en finance. »

Mes remerciements vont aussi à Mr Tony Bussièrès, de TickSmith, qui m'a fait confiance et s'est rendu disponible afin de répondre à mes multiples interrogations.

J'adresse aussi des remerciements à Mme Christine Richard, une source d'énergie intarissable, toujours prête à orienter la communauté étudiante de l'ÉTS concernant les bonnes méthodes de rédaction, via ses ateliers de rédactions scientifiques, et de m'avoir aidée particulièrement avec ce projet.

Merci aussi à Stéphanie Barthélemy pour ses judicieux conseils et de m'avoir aidée à me sentir toujours bien entourée au Canada en dépit que ma famille soit très loin de moi.

À tout ce beau monde, mes profonds et sincères remerciements

MODÈLE DE DONNÉES GÉNÉRIQUE ET CARTOGRAPHIE DES DONNÉES DU MARCHÉ FINANCIER(CARNET D'ORDRES)

Franceska DORVAL

RÉSUMÉ

Le marché boursier canadien génère environ 100 millions d'évènements par jour. Par contre, la structure actuelle des données générées ne permet pas de les utiliser ni de les exploiter, parce que les bourses qui les produisent ne disposent pas d'une approche commune pour engendrer des flux de données normalisés. Chaque bourse publie donc ses données dans un format propriétaire avec des spécifications différentes des autres bourses.

L'entreprise TickSmith a été fondée à Montréal en 2012 par des experts de l'industrie financière, mieux connue sous le nom de FinTech. Elle offre des services aux participants du marché des capitaux qui cherchent à travailler efficacement avec une grande quantité de données financières hétérogènes en constante croissance. Aujourd'hui, TickSmith fournit des systèmes de gestion de données aux leaders de l'industrie. Il s'agit en fait d'une très grande quantité de données présentées dans un tableur Excel provenant de chaque bourse. Chaque tableur Excel contient de multiples classeurs pour chaque bourse. La structure et le contenu de ces sources de données varient d'une bourse à une autre. Conséquemment, l'étape de traitement de ces fichiers est une tâche complexe, car les flux de données en temps réel provenant de chaque bourse nécessitent un traitement rapide et efficace de chaque champ dans le but d'établir la correspondance entre le modèle de donnée interne de TickSmith et la structure des données boursières. À ce jour, la mise à jour de la structure d'un des flux de données du marché ou du modèle de données interne de TickSmith nécessite de modifier les codes sources de l'application qui contient les règles d'affaires concernant la position de chaque information. Ceci entraîne des coûts de maintenance de plus en plus élevés .

Dans le but de réduire ou d'éliminer le temps nécessaire pour effectuer cette correspondance entre les différents champs et de comprendre le sens de chaque donnée enregistrée dans ces champs, l'implémentation d'un modèle générique de données qui comprendra l'ensemble de ces données pourrait permettre de simplifier les échanges est fortement nécessaire. Ainsi tous

les acteurs du marché pourront exploiter et s'échanger des données à l'aide de cette nouvelle structure du modèle générique.

Ce projet de recherche appliquée, de 15 crédits , à la maîtrise en génie des technologies de l'information, vise à réaliser un prototype logiciel suivi d'une étude de cas qui consiste à créer un modèle générique où les données des différentes bourses peuvent être fédérées dans une seule structure, c'est-à-dire un schéma de base de données normalisé. Il vise aussi à convertir la structure actuelle des données, qui est complexe, en un schéma de base de données plat, simple, structuré et documenté. Ce schéma résultant permettra d'exporter les données vers un fichier en format texte contenant des délimiteurs qui faciliteront l'importation et l'interrogation à l'aide de requêtes SQL. Les utilisateurs pourront ainsi l'exploiter à partir d'autres outils pour les rendre accessibles en temps réel. Finalement, un dernier objectif de ce projet de recherche appliquée est de s'assurer de concevoir un processus de conversion qui pourra être facilement utilisable sur d'autres marchés financiers que celui expérimenté dans cette étude de cas et ainsi permettre à TickSmith d'automatiser la correspondance et l'intégration de ces données hétérogènes.

Mots-clés : ETC, correspondance de données, fédération, modèle de données, marché financier.

GENERIC DATA MODEL AND MAPPING OF FINANCIAL MARKET DATA (ORDER BOOK)

Franceska DORVAL

ABSTRACT

The Canadian stock market generates about 100 million events per day. On the other hand, the current structure of the generated data does not make it possible to use or exploit easily that data because the stock exchanges do not use a common approach to standardize it. Each stock market publishes its data in a proprietary format where specifications differ from other stock exchanges.

TickSmith was founded in 2012 and is located in Montreal. Founders are experts from the financial industry also known as FinTech. TickSmith provides services to capital market customers seeking to work effectively with large and growing amount of heterogeneous financial data. Today, TickSmith provides data management systems to industry leaders. The source of the TickSmith data is a very large amount of data originating from each exchange, in the format of an Excel spreadsheet. Each Excel spreadsheet contains multiple workbooks for each market. As a result, the processing of these files is a complex task because the real-time data feeds from each exchange require fast and efficient processing of each field to match the internal data model used by TickSmith. To date, updating the structure of one of the market data feeds or of the internal TickSmith data model requires modifying the source codes of the application that contains the business rules regarding the position of each data item. This leads to higher and higher maintenance costs.

In order to reduce the time required to perform this correspondence between the different fields and understand the meaning of each data recorded in these fields, the implementation of a generic data model that will include all of these data could help to simplify the exchanges is strongly needed. In this way, all market players will be able to exploit and exchange data using this new structure of the generic model.

This applied research project, of 15 credits, in the master's degree in information technology engineering, aims at producing a software prototype followed by a case study which consists in creating a generic model where the data of the various exchanges can be federated into a single structure, that is, a standard database schema. This applied research project also aims to convert the current complex data structure into a flat, simple, structured and documented database schema. This resulting schema will export the data to a text file containing delimiters that will facilitate importing and querying with SQL queries. Users will be able to use it in other tools to make them accessible in real time. Finally, a final goal of this applied research project is to ensure that we design a conversion process that can be easily used in other financial markets than the one experienced in this case study and thus allow TickSmith to automate the process of correspondence and integration of these heterogeneous data.

Keywords: ETL, data matching, federation, data model, financial market.

TABLE DES MATIÈRES

	Page
INTRODUCTION	1
CHAPITRE 1 REVUE DE LITTERATURE	5
1.1 Introduction.....	5
1.2 Définitions de concepts.....	6
1.3 Approches d'alignements.....	6
CHAPITRE 2 MÉTHODOLOGIE	15
2.1 Introduction.....	15
2.2 Principes de la méthodologie	15
2.2.1 Principes de la méthodologie de mise en correspondance.....	16
2.2.2 Sélection des alignements	16
2.2.3 Validation et expérimentations	16
2.3 Développement	17
2.3.1 Processus de conception	17
2.3.2 Présentation du prototype logiciel	19
2.4 Modules du prototype logiciel	20
2.4.1 Module de chargement.....	20
2.4.2 Module de traitement	21
2.4.3 Module de transformation.....	21
2.4.3.1 Principes algorithmiques du module de transformation	22
2.4.3.2 Représentation schématique de la similarité Cosinus	23
2.5 Module de similarité sémantique	24
2.5.1 Principes algorithmiques du module sémantique	25
2.6 Module de sélection	26
2.7 Écran graphique du prototype.....	26
2.7.1 Fenêtre principale du prototype (application de mise en correspondance).....	27
2.7.2 Fenêtre similarité Cosinus.....	28
2.7.3 Fenêtre similarité BabelNet	29
2.7.4 Fenêtre similarité Word2Vec	30
2.8 Représentation globale du système	30
2.9 Conclusion	32
CHAPITRE 3 RÉSULTATS	33
3.1 Introduction.....	33
3.2 Expérimentations et jeu de données.....	33
3.3 Discussions	35
3.3.1 Comparaison des mesures de similarité du prototype du projet versus d'autres outils de mise en correspondance.....	36
3.3.2 Comparaison de performance du prototype du projet versus d'autres outils de mise en correspondance	37

3.3.3	Synthèse des discussions.....	38
CONCLUSION		39
RECOMMANDATIONS	41
ANNEXE I	Approches d'alignements.....	45
ANNEXE II	Représentation modèle générique.....	46
LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES	49

LISTE DES FIGURES

	Page
Figure 1.1.1 Représentation de la problématique	2
Figure 2.3.1 Diagramme de Séquence	18
Figure 2.4.3.2.1 Représentation similarité Cosinus	23
Figure 2.7.1 Fenêtre principale du prototype.....	27
Figure 2.7.2 Fenêtre de correspondance similarité Cosinus	28
Figure 2.7.3 Fenêtre mise en correspondance BabelNet.....	29
Figure 2.7.4 Fenêtre mise en correspondance Word2Vec	30
Figure 2.8.1 Représentation globale du système	31
Figure 3.2.1 Similarité Cosinus	34
Figure 3.2.2 Similarité sémantique	35
Figure 3.3.1 Résultats d'autres outils.....	37

LISTE DES ABRÉVIATIONS, SIGLES ET ACRONYMES

AWS	Amazon Web Services,ou en français services web d'Amazon.
ETC	Extraction Transformation Chargement
ETL	Extract Transform Load, en français ce sigle devient ETC
F-Mesure	Frequencies/Differential Weighting for precision and recall, en français Moyenne harmonique pondérée des métriques rappels et précision
GB	Gygabyte, en français gigaoctet, est une unité de mesure en informatique.
i-7	Intel Core 7, en français processeur 7 ^e génération de la marque déposée d'Intel.
N-grammes	Séquences de mots. La lettre N peut prend des valeurs allant de 1 à 3 dans ce projet de recherche.
NoSQL	Not Only SQL, ou en français pas seulement SQL
SQL	Structured Query Language, en français langage de requête structurée
TF/IDF	Terme Frequency/Inverse documentfrequency, en français fréquence du terme/fréquence inverse du document.
TDM	Traitements de données massives
THF	High Frequency Trading, en français titre à haute fréquence
TLN	Traitement du langage naturel
UML	Unified Modeling Language ou en français langage de modélisation des données.

INTRODUCTION

Contexte

En 1971, Fischer Black, dans sa vision futuriste, voyait déjà le marché financier comme un ensemble d'ordinateurs interconnectés sans la présence de spécialistes sur le plancher pour recevoir les ordres d'achats ou de vente (Black, 1971). Cependant, il a fallu attendre deux décennies pour comprendre et voir la concrétisation de cette vision. Les progrès de la technologie et de la communication ont transformé profondément le marché financier. L'émergence de plateformes de négociation électronique a poussé le marché des actions et des produits dérivés à s'organiser autour de carnets d'ordres électroniques. Ces plateformes ont tellement gagné en popularité que certains marchés boursiers ont été convertis en purs systèmes de négociation électronique. Plusieurs marchés ont aussi fusionné avec des marchés offrant ce type de plateforme et sont considérés comme des marchés hybrides.

L'essor de la technologie a aussi fourni graduellement l'infrastructure informatique et communicationnelle permettant le déploiement de la négociation de titres à haute fréquence (THF). Cette mise en œuvre très rapide des stratégies de négociations automatisées permet l'exécution d'un grand nombre d'opérations en de laps de temps très courts (Barker. W, Pomenarats. A, 2011). Les THF bouleversent le marché financier notamment par le volume élevé de messages, la faible latence, l'automatisation, la courte durée des positions. Ils entraînent des coûts technologiques importants en raison des volumes élevés de messages qu'ils génèrent. Les acteurs financiers qui doivent suivre cette évolution rapide doivent ainsi investir en technologie, puis consacrer une part importante de leurs ressources au traitement des messages pour assurer l'intégration de ces différentes sources de données.

Problématique et objectifs

Le marché financier canadien génère plus d'un million de données par jour. On n'y trouve aucun protocole d'échange de données normalisé. Chaque bourse fournit ses données sous

forme de messages dans un format qui leur est propre, c'est-à-dire avec des noms de colonnes différents pour une même information considérée. Dans le but de pallier l'absence de norme, nous aimerions concevoir un modèle générique facilitant l'intégration des données de différentes bourses du marché canadien et même d'autres marchés externes.

Il existe deux méthodes publiées pour l'intégration des données : l'entrepôt (Kimball, 1998) et le médiateur (Huang et coll. 2000). L'approche par entrepôt crée un schéma particulier qui sert à construire une base de données centralisée, mais ne fournit aucune règle pour résoudre le problème de mise à jour des données qui sont en constante évolution. L'approche du médiateur crée un schéma global à partir des correspondances des schémas locaux ou définit trois éléments : les schémas des sources locales, le schéma global et les correspondances entre le schéma global et les schémas locaux.

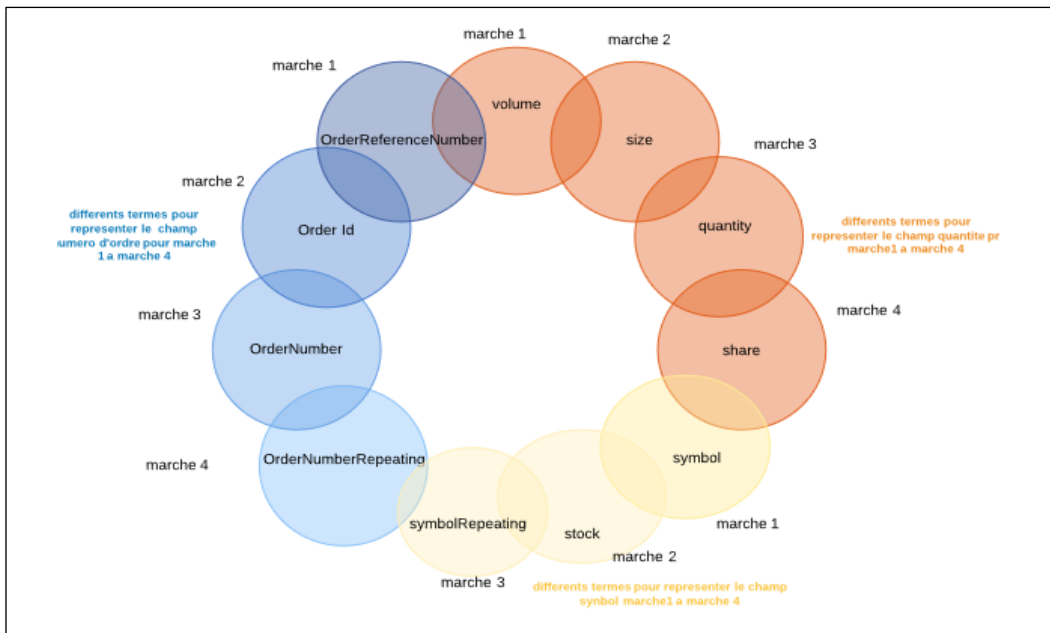


Figure 1.1.1 Représentation de la problématique

Les publications du domaine classent en trois grandes catégories les méthodes d'automatisation de correspondance des schémas qui visent à bâtir un modèle générique :

- 1) les connaissances à partir des schémas de données ;
- 2) les connaissances à partir du contenu ;
- 3) les connaissances à partir des ontologies (Rahm et coll. 2000).

Un modèle générique peut être conçu à partir de ces approches en vue d'établir la correspondance entre les champs des différents schémas extraits dans les fichiers du marché. La correspondance obtenue à partir de l'alignement des champs constitue la structure d'un modèle générique. Ainsi, dans une première étape, il est possible d'afficher cette correspondance sur une interface graphique afin de permettre à l'utilisateur de faire la validation de la relation bidirectionnelle des données. Cet affichage bidirectionnel des données permet, à partir du fichier de format .csv contenant les données dans la structure du modèle générique, de voir les données dans la structure initiale du marché correspondant. Ainsi, à l'aide de ce modèle générique, la structure du fichier peut être modifiée plus facilement. Les développeurs de TickSmith n'auront plus à passer des heures à comprendre la sémantique des champs et le type des données. Pour TickSmith, cette recherche pourra même faire l'objet d'un nouveau service. Ainsi, ce fichier pourrait être aussi vendu aux firmes de négociation ou d'investissement, car les THF nécessitent de scruter le carnet d'ordres en permanence pour confronter l'offre et la demande et calculer le cours d'équilibre afin d'envoyer les ordres automatiquement aux plateformes de négociation.

Ce projet doit aussi répondre à des exigences non fonctionnelles, autrement dit, le prototype logiciel doit être conçu de manière évolutive, facile à maintenir (c.-à-d. de faible couplage), avoir une faible latence lors de son exécution et permettre de monter en charge facilement (c.-à-d. être élastique). L'élasticité est la mise à l'échelle nécessaire de l'infrastructure, car le volume de données du marché financier s'accroît constamment et les régulateurs imposent une disponibilité en tout temps.

Organisation du travail

Ce rapport est divisé en trois parties :

- I) La première, qui comprend le chapitre 1, présente la revue de littérature du domaine. Elle définit les concepts essentiels de correspondance de schémas. Cette revue contient aussi les approches publiées sur les petits schémas (c.-à-d. traitements et calcul sur les chaînes de caractères) et les schémas à large échelle (c.-à-d. apprentissage machine, fouilles de données) qui visent à effectuer la correspondance de schémas, établir une correspondance et créer le modèle générique. Elle expose aussi la synthèse des études et pratiques publiées qui visent à implémenter un processus d'intégration des données pour les environnements à large échelle dans le but de les charger dans un fichier plat ;
- II) La deuxième décrit, à travers le chapitre 2, la méthodologie de recherche. Ce chapitre présente les décisions de conception et d'expérimentation du cas d'étude qui est divisé en quelques sous-sections :
 - L'analyse du modèle de données actuel ;
 - La transformation des données brutes sur l'intervalle de temps disponible avec les cadres (de l'anglais « frameworks ») d'Apache Spark et Scikit-learn ;
 - Les choix lors de la conception du modèle générique, les résultats comparatifs issus de différentes méthodes de correspondance et outils utilisés pour générer le modèle générique ;
 - Les méthodes utilisées pour générer le fichier final et la qualité des données fournies dans ce fichier.

La dernière partie, qui inclut le chapitre 3, présente les résultats de l'expérimentation ainsi qu'une conclusion qui discute des points forts et des limites du projet. Une discussion sur les recommandations est aussi présentée afin de guider les travaux futurs de l'élaboration de ce modèle générique. Finalement des informations additionnelles sont ajoutées en annexe.

CHAPITRE 1

REVUE DE LITTERATURE

1.1 Introduction

L'intégration ou l'échange des données ainsi que la recherche d'informations ont le besoin constant de consolider les données de sources disparates vers un schéma global afin de faciliter la transformation du format des données sources vers le format du système cible. Le processus d'alignement des schémas est souvent utilisé comme étape préalable à l'intégration ou l'échange de données. Cette revue élabore une synthèse de ce qui se fait dans l'industrie pour atteindre le premier objectif ciblé de notre projet à savoir concevoir un modèle de données générique pour le marché financier.

Dans le but d'atteindre cet objectif, la revue de littérature a été divisée en trois grandes parties :

- 1) La première introduit le lecteur aux concepts clés de similarité de schémas en vue d'aboutir à la correspondance. Elle présente les techniques de base, puis les techniques évolutives et finalement les principes d'optimisation pour effectuer la similarité des schémas à large échelle ;
- 2) La deuxième approfondit les approches d'implémentation des activités d'extraction, de transformation et de chargement des données (ETC) dans des environnements de données massives ;
- 3) La dernière partie illustre les méthodes et les outils sélectionnés, parmi ceux présentés, pour implémenter une preuve de concept.

1.2 Définitions de concepts

Les publications de la littérature sur les différentes méthodes d'alignements regroupent les concepts suivants : l'appariement de schémas et la correspondance de schéma mieux connus sous leurs noms anglais respectifs « schema matching » et « schema mapping ».

Néanmoins, ces concepts sont souvent confondus, les chercheurs (Shaivko et Euzenat, 2005) proposent des définitions pour clarifier et différencier ces deux concepts. Selon eux, l'appariement de l'anglais « match » est une opération de manipulation des données qui consiste à trouver des relations de correspondances sémantiques entre les schémas de données. La relation de correspondance « matching » indique qu'un certain élément du schéma S1 est relié à un autre élément du schéma S2 et la relation entre ces deux éléments peut être décrite par une expression de transformation appelée « correspondance » et l'ensemble des liens d'appariement forment l'alignement.

Un schéma, à son tour, est une notation qui permet de modéliser la notion d'éléments et de structure qui peut se présenter sous différents formats (Rahm et Bernstein, 2001).

1.3 Approches d'alignements

La littérature présente un grand nombre d'approches et techniques pour l'alignement automatique des schémas. Une catégorie de chercheurs (Bernstein et coll. 2011) présente une classification suivant deux approches pour la résolution de la problématique d'alignement : d'un côté celles basées sur un alignement individuel et de l'autre, celles basées sur une combinaison d'alignement. Un autre courant de pensées (Shvaiko et Euzenat, 2005) propose une classification suivant trois approches : les éléments de schémas, les instances de schémas ou données et les structures (voir ANNEXE I).

L'accroissement des données et la prolifération des schémas de données, au niveau des entreprises, portent les chercheurs à étudier de nouvelles approches pour réaliser l'alignement

entre des schémas de plus en plus volumineux (c.-à-d. de plus de 100 champs de données). Ces approches suivent la méthode « diviser pour régner » (de l'anglais « divide and conquer ») pour, dans un premier temps, partitionner les larges schémas en entrée de plus petits blocs (ou grappes) et, dans un deuxième temps, pour trouver les alignements au niveau de ces blocs. Parmi lesquels, citons l'approche deux à deux (de l'anglais « pair wise ») qui aligne les éléments des schémas entre deux paires d'éléments pour déterminer leur correspondance (Smiljanic Marko, 2006) ; l'approche holistique combine les données de plusieurs interfaces web pour produire les correspondances ; et l'approche ontologique qui crée des ancrs ou alignements en déterminant les relations entre les termes suivant un domaine spécifique (Hu. W, 2006).

1.4 Techniques des alignements

Les techniques d'alignements sont divisées en deux catégories (Bernstein et coll. 2011) : les techniques classiques et les techniques récentes. Une taxonomie de ces différentes techniques a été élaborée comme suit par Rahm et Bernstein (Rahm et Bernstein, 2001). Les techniques classiques regroupent les techniques terminologiques qui exploitent les méthodes fondés sur les chaînes de caractères tels que les préfixes ou les suffixes, et les libellés des termes sont considérés comme des caractères. Les techniques linguistiques elles, vont compléter les techniques terminologiques en exploitant les méthodes d'analyse de textes (telles que la lemmatisation, la représentation en sac de mots, le bouturage) pour déduire la similarité entre les éléments des schémas. Elles sont complétées à leur tour par les techniques basées sur les ressources auxiliaires. Ces dernières utilisent les dictionnaires pour déterminer les synonymes, les métonymies et aussi les homonymes pour résoudre la désambiguïsation des termes, c'est-à-dire des mots qui peuvent avoir plusieurs sens possibles et seulement la clarté du contexte permet de sélectionner le sens approprié. (Navigli, 2009).

Les techniques récentes, quant à elles, regroupent les techniques basées sur les instances ou qui utilisent des statistiques ou des classificateurs d'apprentissage machine pour déduire les

similarités entre les données des schémas. Une nouvelle vague de chercheurs a démontré des améliorations significatives en termes de précision du modèle d'apprentissage profond, grâce au paradigme de représentations vectorielles continues des mots c'est-à-dire de l'anglais le « word embedding » (Levy et coll. 2015). L'exemple de modèle le plus populaire, dans ces tâches de correspondances des éléments de schémas basés sur ce principe, est le Word2Vec.

Des techniques récentes comprennent aussi des techniques basées sur le contenu des documents. Celles-ci regroupent les éléments des schémas en différents documents et les comparent à l'aide de la technique de TF/IDF.

D'autres techniques, basées sur les graphes, transforment les éléments de schémas en graphes et leurs similarités sont déduites à partir des liens formés entre leurs arcs. Ces techniques peuvent être hybrides, c'est-à-dire qu'elles combinent différentes techniques. Certains chercheurs commencent à vulgariser l'utilisation des plateformes de traitements de données massives par exemple : Hadoop et Spark pour supporter ces différentes techniques afin d'améliorer leur performance et permettre une meilleure mise à l'échelle.

1.5 Mesure de similarité

Les techniques d'alignements, une fois utilisées pour aligner les schémas, nécessitent de déterminer le degré (de l'anglais « score ») de similarité pour chaque paire d'éléments de schémas en calculant leur indice de similarité.

Les méthodes de calcul de la similarité des textes sont groupées en deux grandes classes : les mesures textuelles (c.-à-d. lexicales et sémantiques) et les mesures structurelles (c.-à-d. similarité entre les graphes) (Cohen et coll.2003).

Les mesures de similarité sémantique utilisent des sources auxiliaires telles que BabelNet, Wikipédia, WordNet et autres, pour calculer le degré de similarité entre les termes. Les

mesures textuelles quant à elle se basent sur des formules mathématiques ou des fonctions de distance pour construire ou comparer les chaînes et ensuite évaluer leurs similarités par d'autres mesures telles que la technique TF/IDF et la similarité Cosinus. La technique TF/IDF détermine la similarité entre deux chaînes de caractères suivant leur fréquence et accorde une grande importance aux mots peu communs alors que la similarité Cosinus calcule le degré de l'angle formé par les chaînes de caractères des schémas transformées en vecteurs.

Les mesures textuelles peuvent être aussi hybrides c'est à dire elles font une combinaison de ces différentes mesures.

1.6 Outils d'alignement populaires

Ces différentes approches ont donné lieu à une multitude d'outils qui permettent d'effectuer l'alignement automatique ou semi-automatique des éléments de schémas. Parmi les plus populaires issues des revues scientifiques il y a : Cupid (Madhavan et coll. 2001) qui utilise des techniques hybrides (c.-à-d. des similarités linguistiques) et requiert par la suite que les éléments du schéma soient transformés en graphes afin d'en déduire les similarités. Un autre outil nommé Similarity Flooding (Melnik et coll. 2002) utilise aussi les techniques basées sur les graphes. L'outil COMA (Madhavan et coll. 2001) considéré comme le plus configurable permet à un utilisateur de combiner différentes techniques de son choix pour produire l'alignement. Il y a aussi l'outil SemInt (Li et coll. 2000) qui repose sur l'utilisation des réseaux neuronaux et la similarité entre les noms d'éléments pour produire la correspondance. Ensuite, l'outil V-Doc+ (Zhang et coll. 2012) utilise la plateforme Hadoop pour regrouper les termes en documents et les mesurer à l'aide de la technique TF/IDF et la similarité Cosinus. Finalement, les outils FAMER (Saaedi et coll. 2018) et S-DCS ++ (Demetrio et coll. 2017) utilisent la plateforme Spark pour effectuer la mise en correspondance des schémas.

1.7 Conclusion

L'objectif de cette revue littéraire consistait à présenter les différentes approches de correspondances sur les petits schémas ainsi que sur des schémas très volumineux, dans le but d'établir les correspondances entre les éléments des schémas et en dériver un modèle générique automatisable. Chacune des approches présentées utilise différentes techniques qui tentent d'améliorer la performance ainsi que la qualité des résultats des correspondances. Ce chapitre nous aide à faire des recommandations pour l'étude de cas de ce projet de recherche. A l'aide de ces connaissances, les recommandations sont les suivantes :

- 1) l'utilisation des techniques du traitement du langage naturel (TLN) aidera à effectuer un nettoyage en profondeur des éléments des schémas;
- 2) la détermination des relations entre les termes et leur poids pourra être effectuée à l'aide de la technique TF/IDF;
- 3) la similarité Cosinus permettra d'évaluer le degré de similarité entre les termes;
- 4) l'utilisation d'une ressource lexicale et du modèle de prolongement des vecteurs sera utile afin d'aligner les termes non mesurés par la similarité Cosinus. Mais avant d'effectuer l'expérimentation de toutes ces techniques, présentons une deuxième revue littéraire centrée sur le processus d'extraction de données à large échelle.

1.8 Revue littéraire – deuxième partie

1.8.1 Introduction

Beaucoup d'entreprises ont ce même défi de réaliser, par elles-mêmes, la tâche de trouver la correspondance entre des données. Au chapitre précédent, les différentes méthodes qui permettent d'établir les correspondances entre les schémas pour réaliser une correspondance

en vue de concevoir un modèle générique ont été présentées. De plus, les approches et techniques retenues pour expérimenter sur les données boursières ont été précisées.

Afin de démontrer la faisabilité de l'utilisation de cet ensemble de techniques de correspondance, ce projet de recherche appliquée doit aussi charger les données du marché financier, fournies par TickSmith, dans un fichier de format .csv. Cette deuxième revue littéraire présente l'état de l'art de l'implémentation d'un processus ETC, soit d'extraction, de transformation et de chargement des données, mieux connu sous l'acronyme anglais « ETL » à grande échelle. Cette synthèse de la littérature présente autant des ouvrages scientifiques que des suggestions de meilleures pratiques de l'industrie.

1.9 Catégorisation des ETC

La littérature des mégadonnées (de l'anglais « big data ») divise le processus d'ETC en deux catégories : ETC traditionnel et ETC à large échelle car il existe un changement important dans les méthodes d'alimentation des entrepôts de données et dans la manière d'arriver aux résultats. La transformation des données dans un ETC traditionnel intervient sur un serveur intermédiaire avant le chargement sur la cible car les outils d'ETC sont installés sur des serveurs distincts et tous les traitements de transformation se font par le biais du moteur de l'outil de l'ETC.

Les ETC à large échelle, de leur côté, permettent le chargement des données brutes directement sur la cible, où elles seront alors transformées et utilisent le moteur de transformation de la cible pour effectuer le processus de transformation des données. Les outils/plateformes des ETC à large échelle intègrent des concepts et des technologies du domaine des mégadonnées. Ce terme émergent est associé à des technologies et outils spécialisés pour traiter efficacement de très grands volumes de données, en temps réel.

1.9.1 Architecture des ETC traditionnels

Le pipeline des ETC traditionnels (c'est-à-dire la séquence des processus exécutés une à la suite de l'autre) d'une entreprise peut se construire à partir d'un logiciel d'ETC. Un logiciel d'ETC est utilisé pour effectuer la conversion de données et comprend typiquement trois phases :

- 1) l'extraction des données de/des sources ;
- 2) la transformation du format de ces données dans le format visé ;
- 3) le chargement de ces données dans la base de données visée.

Les données brutes sont d'abord extraites de multiples sources de données qui ont souvent des formats hétérogènes. Ensuite, ces données sont transformées et chargées dans des tables de bases de données temporaires constituant l'étape dite de « staging » où d'autres transformations de données peuvent encore être appliquées, par exemple l'harmonisation des formats de dates. Ces données, une fois transformées, sont chargées dans des bases de données (ou des entrepôts de données) afin de les rendre persistantes et de permettre leur visualisation sous des formes analytiques (Ralph Kimball, 1998).

La phase de transformation, quant à elle, est subdivisée en plusieurs sous tâches : analyse des données, définition du flux de transformation, du nettoyage des données et de la mise en correspondance (Raihm et Do, 2000).

1.9.2 Architecture ETC large échelle

L'architecture d'un ETC à large échelle regroupe typiquement un pipeline à quatre processus : 1) ingestion de données ; 2) traitement des données ; 3) requêtes sur les données et 4) exploration des données.

Le premier processus de ce pipeline, connu sous l'appellation d'ingestion de données, consiste à importer les données provenant de différentes sources et formats dans un système

pour une utilisation immédiate ou future. Ces données doivent s'ingérer, en temps réel ou en différé, et sont stockées dans des plateformes qui offrent divers bénéfices tels que : un fort taux de compression, la récupération de l'intégralité des informations en cas de panne ou la reprise en cas d'erreurs.

Ces plateformes de stockage utilisent typiquement un type système de fichier distribué, par exemple Hadoop ou un type objet tel que suggéré par Amazon Web Services (AWS).

Le deuxième processus du pipeline consiste à prendre la sortie du processus d'ingestion les types de données qui nécessitent une transformation. Ce processus est divisé en deux grands sous processus : 1) un processus de transformation qui effectue des calculs ou des transformations diverses sur les données ciblées de différentes sources ; et 2) un processus d'intégration qui fusionne les données transformées provenant de différentes sources. La plateforme requise à ce stage doit offrir les techniques de traitements massifs de données telles que (les traitements distribués, la communication inter-composants, le calcul en grappe). La technologie Big Data Spark répond à ces différents critères et représente l'outil de choix pour les ETC à large échelle pour l'industrie. Ces deux processus sont souvent combinés pour former un lac de données (de l'anglais « *data lake* »)

Les requêtes sur les données constituent le troisième processus du pipeline et permettent d'effectuer le traitement analytique sur les données à l'aide de requêtes et d'enregistrer ces données, soit dans une base de données NoSQL soit dans des entrepôts de données. À ce stade, les technologies de bases de données émergentes comme Impala ou Hive qui permettent d'emmagasiner de très grandes quantités de données et qui fournissent de très bonnes performances pour les requêtes sont très populaires et figurent parmi les outils de référence dans ce domaine.

Le dernier processus du pipeline, connu sous le terme d'exploration de données, permet de visualiser la valeur des données sous différents formats par la mise en place de différents types de rapports.

Dans le cadre de ce projet, le lac de données choisi comprend : Hadoop pour le stockage de fichiers distribués et la mise en place des fonctions nécessaires pour les stocker dans S3, Parquet pour compresser les fichiers et Avro pour faciliter l'évolution des schémas et les sérialiser. Les transformations et le traitement des données seront faits à l'aide des fonctions intégrées de Spark. Ces différents outils et technologies référencés sont en utilisation par TickSmith actuellement.

1.10 Conclusion

L'architecture des processus d'un ETC traditionnel et ceux d'un ETC à grande échelle ainsi que les différentes technologies reliées aux mégadonnées ont été présentées dans cette deuxième revue littéraire. Des technologies ont été retenues pour implémenter une preuve de concept d'ETC à large échelle et de permettre de visualiser les jeux de données financières de ce projet. Tout type d'ETC nécessite de faire la correspondance des schémas à la phase de transformation. La preuve de concept d'un modèle de données générique (voir ANNEXE II) implémenté suivant les recommandations formulées dans la revue, sera expérimentée afin d'évaluer sa capacité à atteindre les objectifs de cette recherche.

Le chapitre Méthodologie présentera les différentes démarches, les outils, les principes ainsi que les spécificités techniques articulées autour des techniques d'alignements et de correspondances en vue d'implémenter le prototype logiciel. Ce dernier permettra d'automatiser la mise en correspondance des schémas issus du marché financier.

CHAPITRE 2

MÉTHODOLOGIE

2.1 Introduction

Nous avons présenté dans la revue de littérature les différentes approches qui permettent de générer l'alignement et d'aboutir à la correspondance entre les éléments des schémas. Dans le cadre de notre approche, nous nous situons dans la première classe d'alignement proposée par Rahm Erhard (Rahm, E.2011), c'est-à-dire la mise en correspondance à partir des éléments de schéma en manipulant les en-têtes des fichiers du marché financier. Nous avons aussi identifié dans la revue les techniques retenues dans le cadre de ce projet pour créer un modèle générique pour des schémas qui sont nombreux et volumineux. Ce chapitre décrit de manière ordonnée et séquentielle toutes les étapes à suivre pour implémenter un prototype logiciel capable d'établir la correspondance automatique entre les éléments des schémas des termes de différents marchés et de créer un modèle générique après validation de l'utilisateur. Ce chapitre est divisé en deux grandes parties, la première décrit la méthodologie adoptée pour implémenter le prototype logiciel et la seconde présente les détails techniques, les outils et les principes algorithmiques pour atteindre cet objectif.

2.2 Principes de la méthodologie

La méthodologie retenue s'inspire des mêmes démarches adoptées par Rahm Erhard (Rahm, E.2011) pour bâtir l'alignement tout en intégrant les approches récentes de mesures sémantiques comme le modèle Word2Vec et l'utilisation d'un thésaurus pour effectuer la mise en correspondance pour les termes qui présentent des conflits sémantiques, c'est-à-dire où le nom peut être interprété différemment et aussi pour déterminer une mise en correspondance pour les éléments de schémas qui n'ont pas eu d'alignement.

2.2.1 Principes de la méthodologie de mise en correspondance

Elle se base sur une série d'étapes qui débute par l'importation des schémas en entrées et termine par la production d'un alignement en sortie, ensuite elle suit successivement trois étapes de prétraitement (représentation en sac de mots, élimination des mots vides, transformation des schémas en sac de mots), suivi de l'étape de l'exécution des techniques pour mesurer les similarités entre les sacs de mots : TF/IDF, la similarité Cosinus, Word2Vec. Ces mesures vont permettre de calculer le degré de similarité entre les termes et produire une matrice de similarité qui va être utilisée pour sélectionner les alignements suivant les résultats obtenus.

2.2.2 Sélection des alignements

La sélection de la stratégie basée sur les seuils a été retenue pour sélectionner les alignements et ce seuil peut être fixé par l'utilisateur lors de son processus de validation, car l'utilisation du seuil constitue une approche judicieuse (Melnik et coll. 2002), mais présente aussi un défi, celui d'indiquer la valeur du seuil optimal en vue d'accroître la précision sans pour autant discriminer les approches valides.

2.2.3 Validation et expérimentations

La méthodologie retenue pour valider le prototype repose sur deux grands axes. Le premier consiste à définir les différents scénarios d'alignement, à conduire différentes expérimentations et ensuite à compiler les résultats obtenus. Le deuxième consiste à évaluer la qualité du logiciel par les mesures de performance qui sont la précision et le rappel. Le client TickSmith nous a fourni un référentiel qui contient les différentes mises en correspondance pour chaque élément de schéma de différents marchés financiers. Ce référentiel servira de base pour jauger la performance attendue par rapport à la mise en correspondance produite.

Deuxième Partie

2.3 Développement

Ce chapitre regroupe les étapes de la méthodologie dans des modules logiciels et présente les concepts et les techniques qui y sont associés, les principes algorithmiques qui doivent être codés dans un langage de programmation pour concevoir le prototype logiciel. Ce dernier doit permettre d'effectuer la correspondance des éléments de schémas de façon automatique, avec faible couplage et capable d'assurer une mise à l'échelle. Ce chapitre se veut être un guide de référence technique à tous ceux qui voudront continuer le travail de mise en correspondance des éléments de schéma de TickSmith.

Les différents modules de ce prototype logiciel sont les suivants : module de chargement, module de traitement, module de transformation, module sémantique, module de sauvegarde. La suite de ce chapitre présente chacun des modules avec la définition des objectifs, la présentation des concepts ainsi qu'un tableau résumant les principes algorithmiques et les bibliothèques essentielles des langages de programmation pour aboutir à l'implémentation du prototype logiciel.

2.3.1 Processus de conception

De la phase d'analyse à la conception, les diagrammes du langage de modélisation unifiée, mieux connu sous l'acronyme anglais UML (« Unified Modeling Language ») permettent d'analyser les besoins et de montrer l'interaction de l'utilisateur avec le système, les différentes actions du système et de l'utilisateur. Dans notre prototype les acteurs principaux sont les suivants : l'utilisateur et le système.

L'utilisateur introduit dans le système les fichiers contenant les schémas afin de bénéficier des fonctionnalités du système. Il peut aussi changer certains paramètres (seuil de similarité)

qui sont déjà fixés avec des valeurs par défaut. Le système lui-même assure le bon déroulement et la bonne intégration des différents modules du système.

Un diagramme de séquence est proposé dans le but de comprendre et de donner une vue détaillée avec un ordre chronologique des différentes actions. Le diagramme de séquence est un diagramme UML qui représente la séquence de messages entre les objets au cours d'une interaction. Ils peuvent être utilisés à différents stades du processus de développement pour décrire les interactions entre objets dans un système. (IBM Rational)

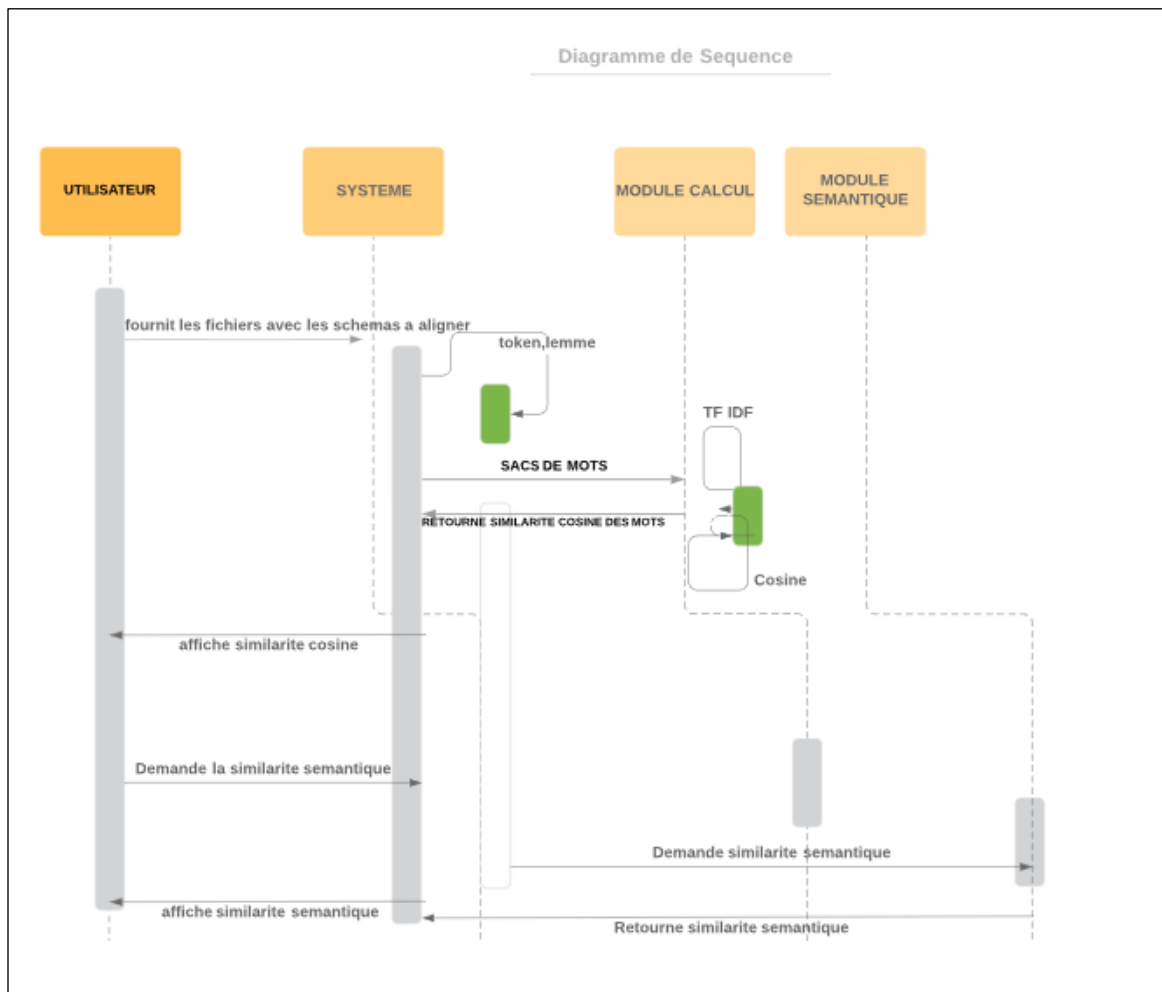


Figure 2.3.1 Diagramme de Séquence

Le diagramme de séquence ci-dessus montre les différentes actions qui se produisent dans le système. Ces actions sont les suivantes : l'utilisateur charge les fichiers dans le système et sélectionne les schémas à aligner pour générer la correspondance. Il valide les seuils de degré de similarité ou modifie d'autres paramètres de configuration suivant ses besoins.

Le système analyse les fichiers chargés et effectue des nettoyages sur le jeu de données. Il transforme les textes en une matrice composée de chiffres via la technique TF/IDF, réalise des opérations matricielles pour normaliser la longueur des documents, et détermine la similarité des textes de chacun des fichiers du marché via la similarité Cosinus ou sémantique. Ces différentes opérations aident à établir la correspondance entre les termes des différents fichiers et à afficher le résultat à l'utilisateur.

2.3.2 Présentation du prototype logiciel

Le prototype logiciel a été développé à l'aide de la plateforme Spark pour exploiter la possibilité d'effectuer les opérations de calcul et les traitements sur de larges volumes de données de manière distribuée sans avoir à les implémenter. Spark offre aussi des interfaces de programmation qui permettent d'utiliser différents langages de développement comme Scala, Java, ou Python, car chacun de ces langages offre des subtilités, de meilleures performances suivant un certain type de besoins.

Les éditeurs de développement intégré Eclipse et Spyder ont été utilisés pour développer les modules du logiciel. Le logiciel a été testé dans un ordinateur avec les spécificités suivantes : processeur intel septième génération (de l'anglais « i-7 »), mémoire (16 Gb).

L'architecture du système regroupe en module les différentes étapes présentées. Le logiciel prend en entrée un ensemble de schémas de différents formats qui peuvent être volumineux et nombreux. Les éléments de ces schémas subissent un processus de traitement grâce à un

parseur de schémas, ensuite sont décomposés en matrice de cooccurrence et mis en correspondance grâce au module de transformation. La sortie de notre plateforme est un ensemble de correspondances sémantiques et statistiques entre les schémas. Les termes similaires sont sauvegardés dans des fichiers ou dans une structure persistante où le terme générique est relié à un mot ou un groupe de mots suivant un degré de similarité. Cette représentation aide à toute réutilisation et à la mise en place d'un vocabulaire pour les termes du marché financier.

2.4 Modules du prototype logiciel

Cette section contient les différents modules du prototype. Cette section contient les différents modules du prototype.

2.4.1 Module de chargement

Ce module consiste à importer les données de toutes les sources et de différents formats dans le système. Ce module aide aussi l'utilisateur à charger des fichiers à partir d'une interface graphique, ou du moins suivant les paramètres de configuration, le système chargeant lui-même les fichiers qui se trouvent dans le chemin indiqué.

Le module de chargement collecte les fichiers des différentes sources, extrait leurs en-têtes et les fournit en entrée aux autres modules qui en ont besoin car les en-têtes des fichiers constituent les éléments des schémas.

Ce module requiert l'utilisation du langage Scala car celui-ci aide à compresser les fichiers en différents formats tels qu'Avro ou Parquet, et à les sauvegarder dans des espaces de stockage comme dans le nuage (Amazon S3), en grappe (Hadoop) et aussi en local. Ce module supporte la phase ingestion de données du pipeline d'ETC à large échelle.

2.4.2 Module de traitement

Ce module permet d'effectuer une série de prétraitements sur les fichiers, les nettoyer et les mettre dans un format attendu par les autres modules. Le nettoyage d'un fichier dépend de l'observation usuelle sur le contenu et de ce que l'on souhaite exploiter. Dans notre cas, la phase de nettoyage effectue l'analyse lexicale sur les éléments de schémas et ensuite la décomposition de ces schémas. L'analyse lexicale consiste à éliminer les caractères indésirables comme les espaces doubles, les signes de ponctuation, les mots vides (de l'anglais « stop words ») qui sont représentés par les prépositions, pronoms, articles ou verbes auxiliaires. Elle permet aussi de décomposer en deux les mots surlignés ainsi que les champs qui contiennent des sous-champs imbriqués, car ceci est un cas fréquent des schémas de notre échantillon qui sont en format Json.

Ce module utilise les bibliothèques du langage Python en vertu des techniques avancées du TLN mises à la disposition des développeurs pour nettoyer en profondeur les textes et appliquer sur les éléments de schémas les techniques linguistiques présentées dans la revue.

2.4.3 Module de transformation

Ce module constitue le cœur du projet car il permet de déterminer la similarité ou la non-similarité des éléments des schémas pour effectuer la mise en correspondance. Ce module permet de représenter en un sac de mots les en-têtes des différents fichiers traités et analysés par le module de traitement, pour être ensuite transformés en un tableau de données ou matrice. Cette transformation est effectuée par la technique TF/IDF et est nécessaire dans le but d'utiliser les mesures de similarité pour déduire la fréquence des termes et accorder un poids à chaque élément de schémas. Cette pondération des termes permet de bâtir une matrice de cooccurrence en relevant les termes les plus importants qui constituent le fichier ou corpus. Ensuite, ce module effectue le produit scalaire des vecteurs dans le but de normaliser le contenu des fichiers et de réduire davantage la dimensionnalité du vecteur en vue d'accroître la performance.

Le module évalue la similarité des éléments de schémas transformés en vecteurs par le calcul du cosinus de l'angle formé par ses représentations vectorielles pour afficher les alignements et effectuer par la suite la correspondance entre les champs.

Cette mesure (similarité Cosinus) considère que deux ou plusieurs documents sont similaires ou proches l'un de l'autre quand la différence du cosinus de l'angle qu'ils forment est petite.

2.4.3.1 Principes algorithmiques du module de transformation

Ce module implémente l'expression mathématique de la technique TF/IDF pour transformer les éléments de schémas en sac de mots, déterminer le poids des termes et construire la matrice de termes par fichiers.

L'algorithme reçoit en entrée un fichier et bâtit la matrice de fréquence en comptant les occurrences de chaque élément de schéma constituant ce fichier où le poids des termes augmente proportionnellement avec le nombre d'occurrences d'un terme dans le fichier. Ensuite, l'algorithme détermine pour chaque terme le logarithme de l'inverse de la proportion des fichiers qui contiennent ce terme dans l'ensemble des fichiers pour empêcher à un terme de recevoir trop de poids.

Le poids final s'obtient par le produit des deux matrices : $tf(t, d) * idf(t)$ (Luhn, H.P, 1957)

La formule de représentation TF/IDF s'énonce par les expressions suivantes :

$$tf(t, d) = \sum_{x \in d} fr(x, t)$$

$tf(t, d)$: Fréquence du terme t dans le document d qui s'obtient par la somme des termes t dans chaque document x .

$$idf(t) = \log\left(\frac{N}{d \in D: t \in d}\right)$$

N : représente le nombre total de documents ou fichiers

$d \in D: t \in d$: représente l'ensemble des documents qui contiennent le terme t .

La matrice de termes une fois construite, un ensemble de méthodes des bibliothèques du cadre Scikit-Learn et Numpy de Python ont été utilisés pour trouver le cosinus de l'angle formé par les termes. Ces fonctions implémentent l'expression mathématique de la similarité Cosinus (voir figure 3.4.1) et présentent les termes qui peuvent être mis en correspondance l'un avec l'autre. Ces mots sont affichés dans une interface graphique sous la rubrique mots similaires et sont regroupés entre eux avec leur degré de similarité qui est un nombre situé dans l'intervalle 0 et 1. Suivant la configuration, le système établit la correspondance entre les termes de deux manières :

1. Les mots similaires sont regroupés et sont alignés entre eux. L'utilisateur choisit un terme générique pour représenter chaque groupe de termes similaire où le système choisit le mot générique par lui-même suivant sa configuration;
2. L'ensemble de ces termes génériques constitue le modèle générique de données et représente le dictionnaire de données des agrégateurs du marché financier avec lesquels TickSmith a un partenariat d'affaires.

2.4.3.2 Représentation schématique de la similarité Cosinus

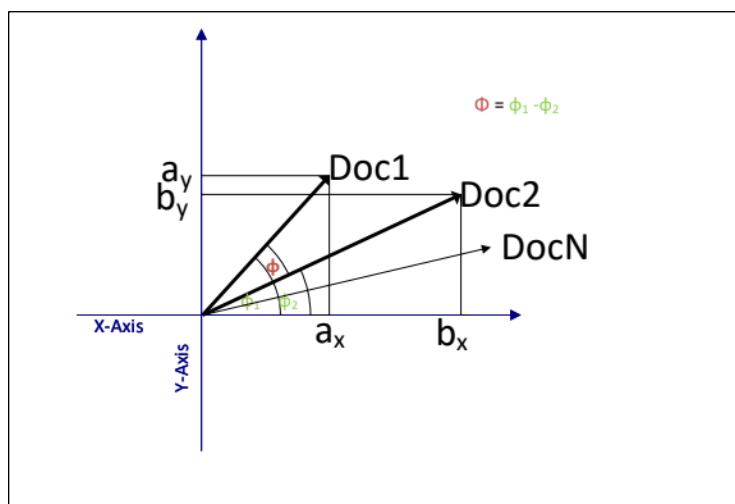


Figure 2.4.3.2.1 Représentation similarité Cosinus

Cette figure indique que le document Doc1 va former un angle θ_1 et le document Doc2 va former un angle θ_2 . La différence entre les angles θ_1 et θ_2 va former un angle final θ . Si la valeur de θ ou le cosinus de l'angle formé par les documents est proche ou plus grand que 90 degrés, les documents n'ont aucune similarité, c'est-à-dire qu'aucune correspondance ne peut être effectuée entre ces termes.

Le terme « document » dans la représentation graphique est similaire à « fichier ».

Le document peut être remplacé par nos échantillons de fichiers du marché financier.

Le schéma de la figure 2.4.3.2.1 permet aussi de déduire que la similarité Cosinus s'obtient par la formule suivante :

$$\vec{a} \cdot \vec{b} = \|\vec{a}\| \cdot \|\vec{b}\| \cos \theta$$

$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$

2.5 Module de similarité sémantique

Le module de similarité Cosinus ne capture pas le sens véhiculé par les mots et les relations potentielles entre eux, il s'avère donc nécessaire d'intégrer d'autres techniques en vue d'expérimenter de nouvelles observations. Le module sémantique du prototype utilise deux techniques sémantiques pour capturer les relations sémantiques entre les termes qui ont affiché un faible score de similarité ou qui sont absents dans les résultats de la similarité par le calcul du cosinus. Ces techniques sont les suivantes : l'utilisation des bibliothèques de Babelnet (Navilgi, R., 2012) qui est une lexicographie et le prolongement lexical avec le modèle Word2Vec (Mikolov et coll, 2013).

Les deux techniques modélisent les concepts ou groupes de mots sous forme de vecteurs. BabelNet évalue la proximité sémantique de ces vecteurs en calculant leur distance par le chemin le plus court et détermine leur niveau de ressemblance par des relations lexicales.

Word2Vec est un modèle d'apprentissage profond basé sur les réseaux de neurones à deux couches qui permettent de capturer la sémantique des mots. À chaque mot il y a deux vecteurs à apprendre : un vecteur d'entrée et un vecteur de sortie, et le modèle calcule ensuite la probabilité reliant chaque mot aux mots du vocabulaire par des expressions mathématiques. Le modèle, à chaque entraînement, tente d'obtenir d'autres vecteurs de mots permettant d'exprimer le même sens que le mot en cours, c'est-à-dire le mot pour lequel on cherche d'autres mots proches par rapport au contexte.

2.5.1 Principes algorithmiques du module sémantique

Le module sémantique du prototype qui se base sur le modèle Word2Vec utilise la librairie Gensim (Rehurek et Sojka, 2010) en Python et effectue lui-même les calculs de probabilité de similarité sémantique entre les termes. Le modèle Word2Vec nécessite beaucoup de données lors de son entraînement et la librairie *BeautifulSoup* de Python a été utilisée pour moissonner le contenu des sites web (du terme anglais « web scraping ») afin de lui fournir autant de termes dans le contexte de la finance. Les fonctions intégrées de *BeautifulSoup* récupèrent la définition des noms des éléments de schémas dans le contexte de la finance à partir de Wikipédia, qui est une encyclopédie, et d'Investopedia, un site web dédié à la finance.

Dans un premier temps, le contenu tiré de ces sites a été transformé en utilisant les fonctions du module traitement du prototype pour nettoyer le contenu et éliminer les mots vides. Ensuite, diverses fonctions de la librairie Gensim (Rehurek et Sojka, 2010) ont permis d'entraîner le corpus et de bâtir le vocabulaire pour chaque terme et ensuite d'effectuer une correspondance d'alignement complexe, c'est-à-dire une correspondance de cardinalité multiple au lieu d'une cardinalité unaire.

Le module sémantique qui se réfère à la version 3.5 de Babelnet, à partir de ces librairies, a permis de bâtir un graphe avec l'ensemble des voisins pour sélectionner la paire de concepts

similaires ayant la profondeur maximale et d'effectuer une correspondance de cardinalité unaire pour chaque élément de schéma qui lui a été fourni.

2.6 Module de sélection

Le module de sauvegarde combine les résultats des mesures de la similarité Cosinus et des mesures sémantiques et sélectionne les résultats les plus pertinents en utilisant des seuils de filtrage. Il retourne les similarités entre les éléments des différents schémas liés suivant ces deux mesures. Ce module bâtit un dictionnaire où les termes synonymes ainsi que le mot générique auquel ils sont mappés sont enregistrés dans une base de données. Le module génère aussi la correspondance entre les termes dans un fichier d'extension csv pour des besoins de réutilisation et d'affichage.

2.7 Écran graphique du prototype

La mise en application des différentes étapes de la méthodologie à travers des outils de développement logiciel a permis d'implémenter le prototype logiciel et de présenter les interfaces suivantes à l'utilisateur afin de lui faciliter la tâche de mise en correspondance. Les menus (voir figure 2.7.1) ont des noms significatifs afin de mieux guider l'utilisateur et de faciliter sa compréhension et son apprentissage du prototype logiciel. Les options de menu nommées « Lemmatisation » et « Représentation en sac de mots » permettent à l'utilisateur d'effectuer ces différents types de traitements sur les fichiers sans obtenir la mise de correspondance entre les termes semblables. Néanmoins, si l'utilisateur clique sur le menu Similarité, le module effectue toutes les étapes pour aboutir à la mise en correspondance.

2.7.1 Fenêtre principale du prototype (application de mise en correspondance)

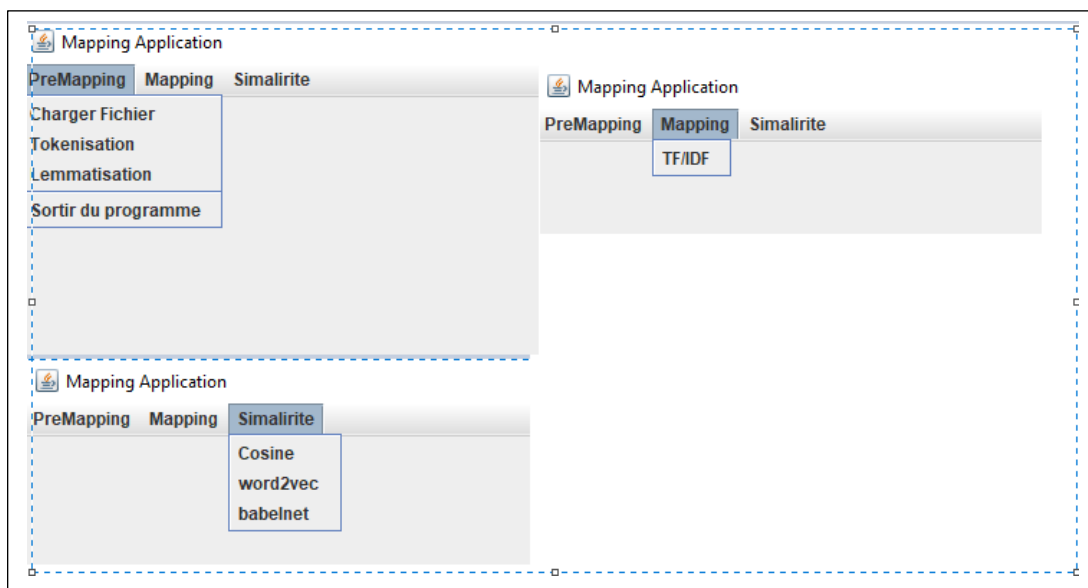
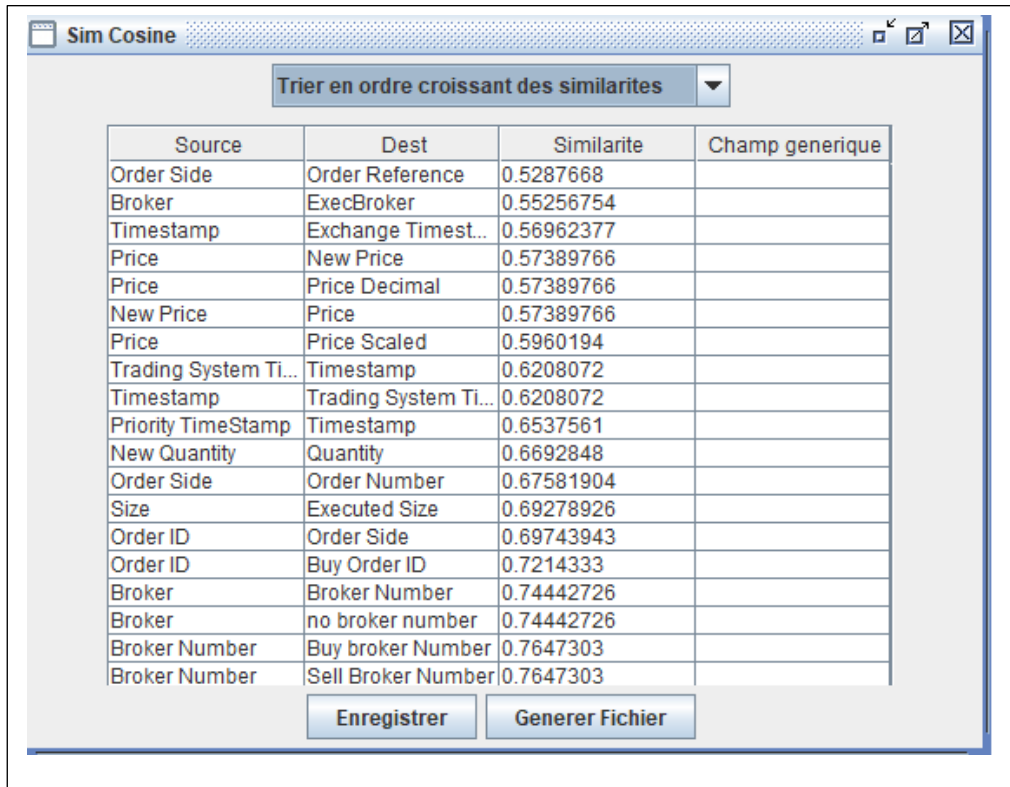


Figure 2.7.1 Fenêtre principale du prototype

2.7.2 Fenêtre similarité Cosinus



The screenshot shows a window titled 'Sim Cosine' with a dropdown menu set to 'Trier en ordre croissant des similarites'. Below the menu is a table with four columns: 'Source', 'Dest', 'Similarite', and 'Champ generique'. The table contains 20 rows of data. At the bottom of the window, there are two buttons: 'Enregistrer' and 'Generer Fichier'.

Source	Dest	Similarite	Champ generique
Order Side	Order Reference	0.5287668	
Broker	ExecBroker	0.55256754	
Timestamp	Exchange Timest...	0.56962377	
Price	New Price	0.57389766	
Price	Price Decimal	0.57389766	
New Price	Price	0.57389766	
Price	Price Scaled	0.5960194	
Trading System Ti...	Timestamp	0.6208072	
Timestamp	Trading System Ti...	0.6208072	
Priority TimeStamp	Timestamp	0.6537561	
New Quantity	Quantity	0.6692848	
Order Side	Order Number	0.67581904	
Size	Executed Size	0.69278926	
Order ID	Order Side	0.69743943	
Order ID	Buy Order ID	0.7214333	
Broker	Broker Number	0.74442726	
Broker	no broker number	0.74442726	
Broker Number	Buy broker Number	0.7647303	
Broker Number	Sell Broker Number	0.7647303	

Figure 2.7.2 Fenêtre de correspondance similarité Cosinus

La fenêtre de similarité Cosinus (Figure 2.7.2) affiche les correspondances obtenues à partir de cette technique en utilisant le contenu des en-têtes de fichiers fourni par l'utilisateur. La figure montre aussi le degré de similarité entre les éléments de schémas et confirme que cette valeur se situe toujours à l'intérieur d'un intervalle entre 0 et 1. L'utilisateur est aussi invité à saisir le terme générique dans le champ de texte nommé « Champ générique ».

2.7.3 Fenêtre similarité BabelNet

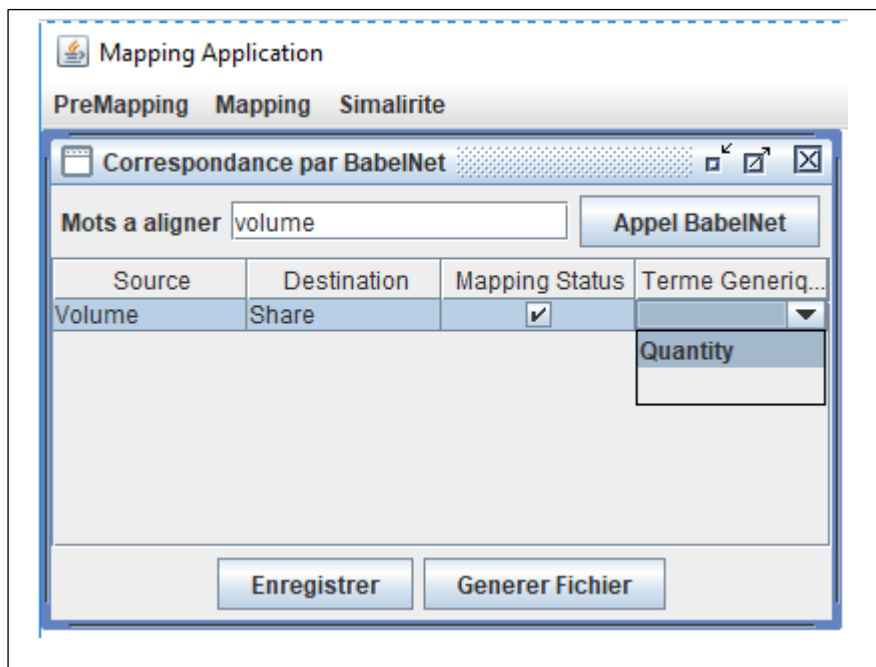


Figure 2.7.3 Fenêtre mise en correspondance BabelNet

La fenêtre BabelNet (figure 2.7.3) invite l'utilisateur à saisir un terme pour déterminer avec quels autres termes le mot original peut être mis en correspondance, et à choisir dans la liste déroulante un terme générique pour représenter ces deux termes dans le modèle générique de données.

2.7.4 Fenêtre similarité Word2Vec

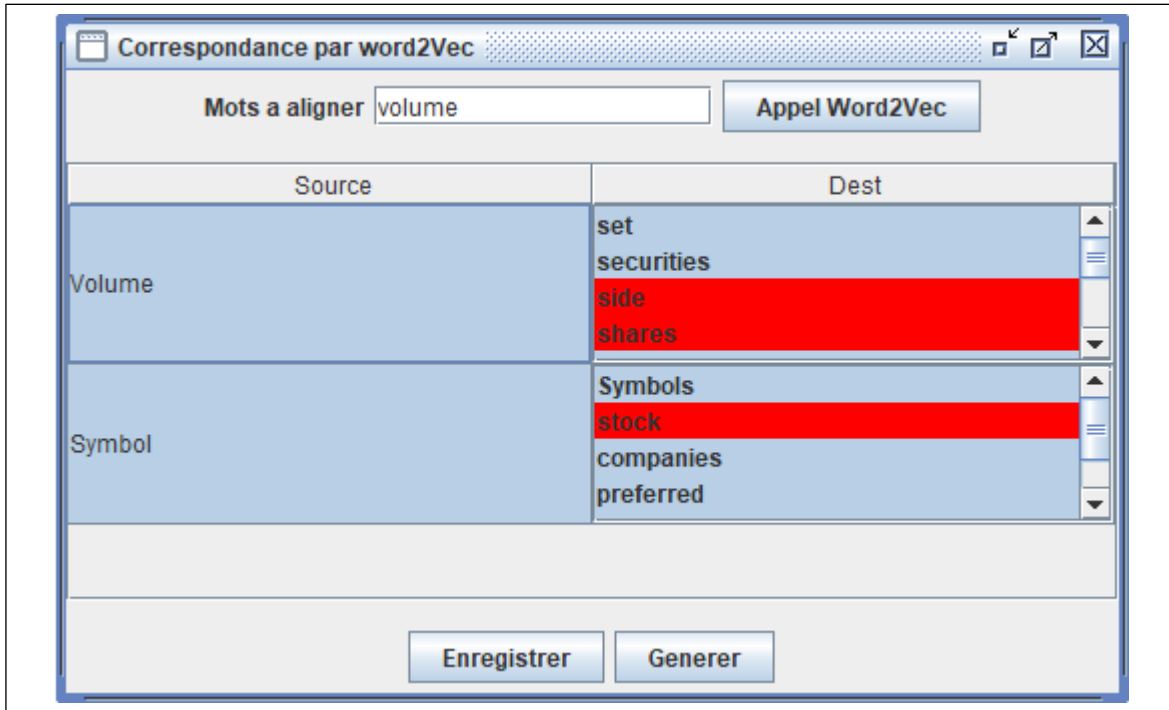


Figure 2.7.4 Fenêtre mise en correspondance Word2Vec

La fenêtre Word2Vec (voir figure 2.7.4) affiche les résultats du modèle Word2Vec pour les éléments de schémas saisis par l'utilisateur. Ce dernier valide aussi la correspondance en sélectionnant l'ensemble des termes avec lesquels un élément de schéma peut être mis en correspondance. Cette fenêtre montre l'alignement complexe via une cardinalité multiple telle que mentionnée.

2.8 Représentation globale du système

L'objectif de ce projet de recherche appliquée consistait à créer un modèle de données générique et à charger les données du marché financier dans ce modèle. Un prototype logiciel a été implémenté afin de combler ces objectifs au cours d'un cycle de développement logiciel et une vue globale du système avec les différentes phases du pipeline des ETC à large échelle

est illustrée à travers un schéma qui résume toutes les fonctionnalités implémentées dans ce prototype.

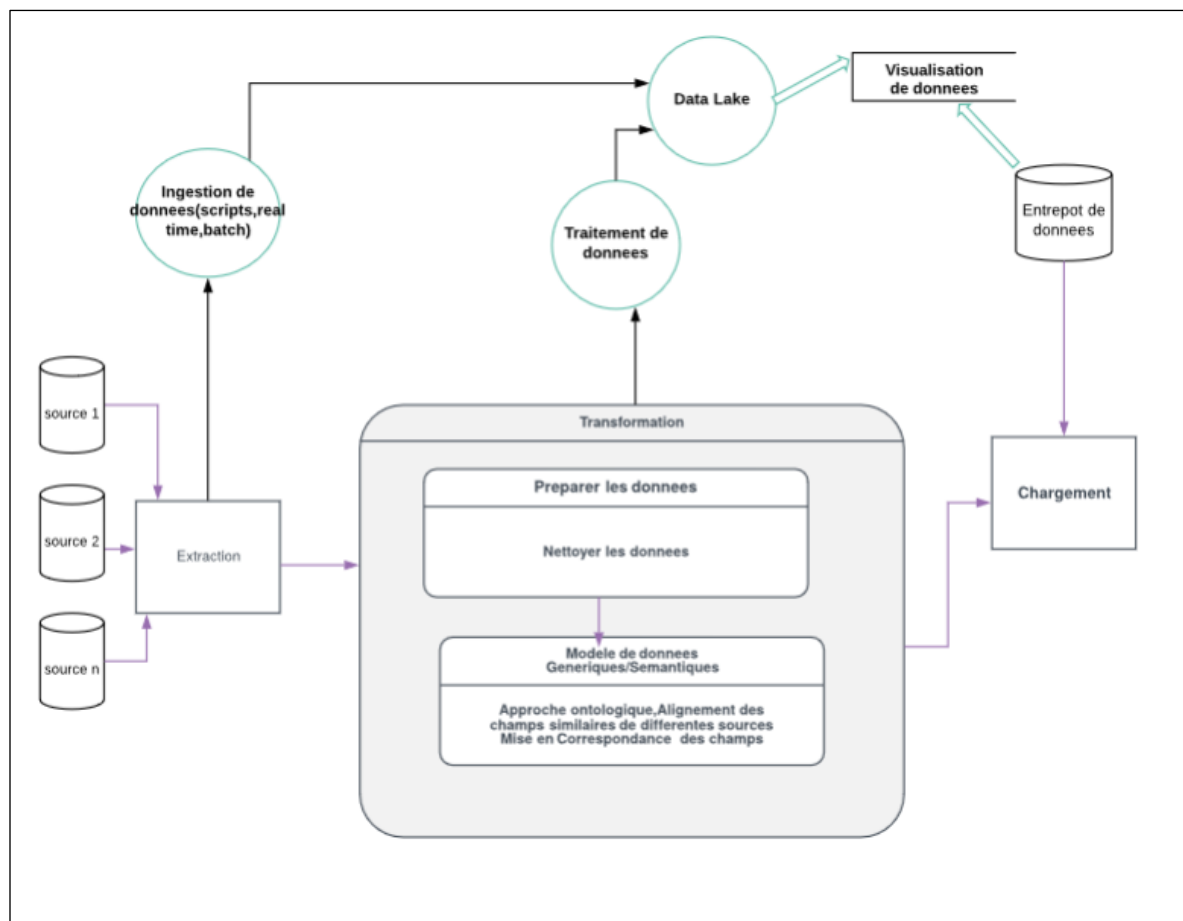


Figure 2.8.1 Représentation globale du système

Le schéma regroupe les différents modules de l'application. Il comprend les parties d'un ETC traditionnel et d'un ETC à large échelle. La partie *ingestion de données* récupère les fichiers dans les différentes sources, les importe pour utilisation immédiate ou les stocke dans des endroits appropriés. Le module de chargement de l'application supporte la partie Ingestion et Extraction du schéma (voir figure 2.8.1) en chargeant les fichiers bruts dans des structures appropriées pour les autres modules. La partie Transformation du schéma nécessite le module traitement pour nettoyer les données et les schémas suivant les conditions

indiquées, ensuite le module transformation effectue toutes les étapes pour produire le modèle générique ou le modifier en ajoutant ou en supprimant des champs au besoin. Ensuite les données transformées sont chargées dans la structure du module générique. Ces données transformées sont exploitées par l'icône entrepôt de données du schéma pour produire les dimensions ou les tables de faits, ou encore sont chargées dans les bases de données NoSQL de la partie exploration de données pour produire des rapports analytiques suivant les besoins explicites.

2.9 Conclusion

La méthodologie adoptée pour la mise en place et la validation du prototype logiciel a été définie. Les concepts ainsi que les techniques utilisés pour implémenter et aboutir à un prototype qui répond aux exigences définies par TickSmith ont été décrits. Les écrans graphiques du prototype logiciel ont été aussi présentés. Le chapitre suivant, nommé Résultats, enchaînera par une description détaillée des résultats obtenus et entamera aussi une discussion sur les résultats de performance de notre prototype versus les résultats affichés par certains outils de mise en correspondance publiés dans les revues scientifiques et présentés dans la revue de littérature.

CHAPITRE 3

RÉSULTATS

3.1 Introduction

Ce chapitre présente les résultats obtenus après la mise en œuvre de notre méthodologie et discute aussi des perspectives qu'offrent ces différents résultats pour la mise en correspondance des éléments de schéma. Il effectue aussi une comparaison des résultats de notre prototype avec d'autres outils de mise en correspondance existants. Il décrit aussi les différentes expérimentations qui ont été conduites à travers le prototype pour aboutir à ces résultats. Ces expérimentations ont permis de valider l'efficacité et l'efficience du prototype, c'est-à-dire que les mises en correspondance effectuées par le prototype répondent aux attentes des utilisateurs. Cela a aussi permis de mesurer la performance du prototype à travers deux métriques qui sont très utilisées en alignement et en recherche d'informations qui sont la performance et le rappel.

3.2 Expérimentations et jeu de données

Les deux grands axes de notre méthodologie de validation peuvent être décrits comme suit : la définition des scénarios de mise en correspondance, ensuite la conduite des expérimentations puis la compilation des résultats. La définition des scénarios prend en compte les considérations suivantes :

- Un scénario où le prototype utilise en entrée tous les fichiers de notre échantillon, c'est-à-dire que tous les éléments de schémas des fichiers vont se confronter pour produire un alignement et effectuer la mise en correspondance;
- Un scénario où l'on fournit les fichiers deux à deux au prototype;
- Des scénarios où l'on fournit seulement des termes ambigus ou des termes choisis au hasard dans les éléments de schémas de notre échantillon. Les mêmes termes vont être fournis aux deux ressources sémantiques du prototype, c'est-à-dire Babel et

Word2Vec, afin de déterminer laquelle génère les similarités attendues et orienter nos perspectives d'études.

La mise en correspondance des deux premiers scénarios se fait suivant la similarité Cosinus. Celle du troisième scénario s'effectue par le biais des deux mesures sémantiques. Ces scénarios sont répétés en modifiant à chaque fois le seuil de similarité. Les tests de validations sont considérés à succès quand l'association des termes affichés figure dans la liste de validation fournie par le client.

Les expérimentations conduites par notre prototype ont permis d'aboutir aux résultats suivants selon les différentes méthodes de calcul de similarité :

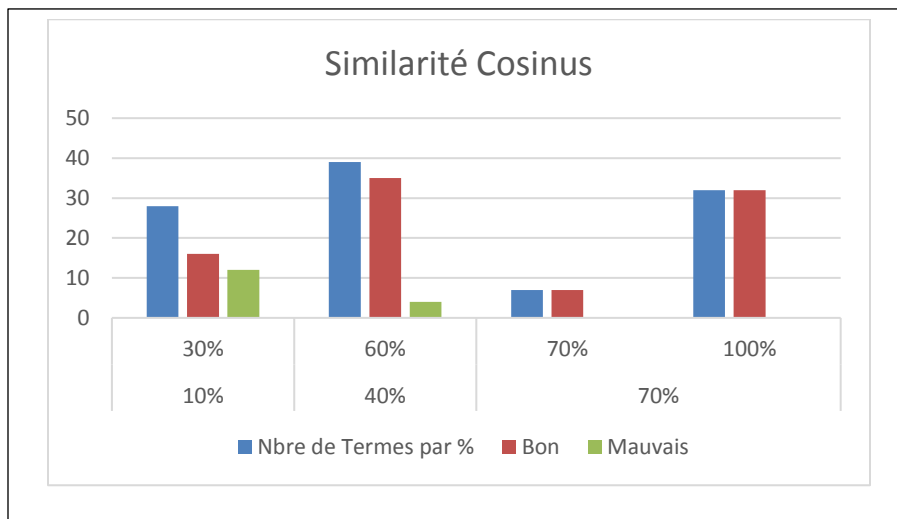


Figure 3.2.1 Similarité Cosinus

La figure ci-dessus affiche les résultats produits par le module de similarité Cosinus. Le degré de similarité des termes est découpé en quatre intervalles pour un ensemble de 124 termes, dont 90 ont effectué une correspondance correcte, 16 une mauvaise correspondance, et 18 ont été rejetés par le module.

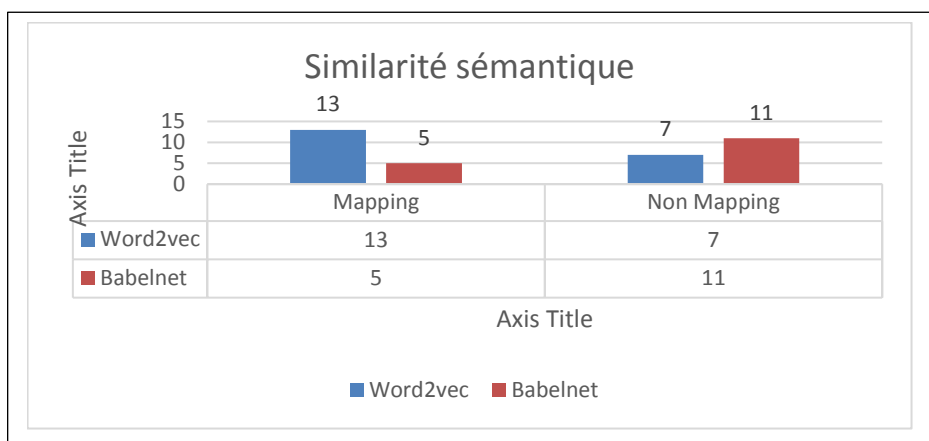


Figure 3.2.2 Similarité sémantique

Les deux figures ci-dessus reportent les résultats du module sémantique qui utilisent les 18 termes rejetés par la similarité Cosinus. Le modèle Word2Vec a fourni des correspondances correctes pour 13 de ses termes. Il a aussi fourni plusieurs termes qui peuvent être mis en correspondance par le terme source (cardinalité multiple). Par contre, BabelNet a effectué une bonne mise en correspondance pour sept termes et les autres correspondances ne sont pas conformes aux résultats attendus.

3.3 Discussions

À la lumière des résultats obtenus, nous discutons dans cette section de la performance du prototype au regard des deux aspects suivants : la précision et le rappel. Nous comparons nos résultats avec ceux publiés dans la littérature. La précision, le rappel, la moyenne harmonique (de l'anglais « F-measure ») sont les mesures les plus utilisées pour évaluer la qualité des outils de mises en correspondances (Euzenat. J, 2007). La précision représente l'ensemble des vraies correspondances retournées automatiquement par le prototype du projet. Le rappel mesure la proportion de tous les résultats corrects que le prototype n'a pas réussi à trouver.

La précision des résultats de la similarité Cosinus est de 84 %, c'est-à-dire que 84 termes sur les 106 restants ont généré une bonne correspondance. La similarité sémantique améliore le rappel ainsi que la précision car 13 termes sur les 18 rejetés ont été alignés par le modèle Word2Vec et les mises en correspondances effectuées figurent aussi dans le référentiel de TickSmith. Autre constat, le seuil de degré de similarité du prototype était fixé à une valeur supérieure ou égale à 0.8, mais durant les tests de validation nous avons constaté qu'il existe 35 termes dont le degré de similarité est nettement inférieur à ce seuil et dont la mise en correspondance effectuée est correcte. Cette observation a confirmé la sélection de stratégie de Melnik (Melnik et coll.2012) et nous a conduit à créer un écran graphique permettant à l'utilisateur de configurer autant de fois souhaité le seuil de similarité lors du processus de mise en correspondance.

3.3.1 Comparaison des mesures de similarité du prototype du projet versus d'autres outils de mise en correspondance

Les éléments de schéma de l'échantillon de données du projet ayant les mêmes graphies ont affiché un degré de similarité égal à 100% et celui des autres oscille dans l'intervalle de 20% à 70%. Ce résultat est conforme à l'ensemble des expériences de ce type dans la littérature. Les chercheurs qui ont implémenté l'outil V-Doc+ ont aussi obtenu ces mêmes seuils de similarité lors de leur processus de mise en correspondance.

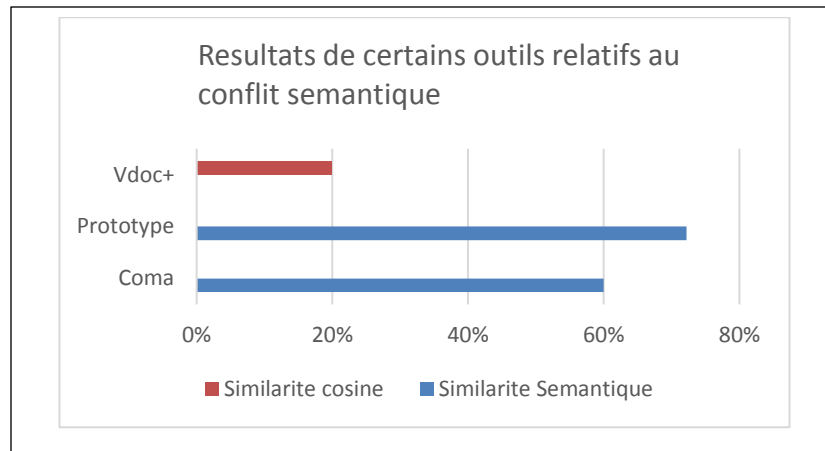


Figure 3.3.1 Résultats d'autres outils

La figure 3.3.1 montre que la similarité Cosinus ne traite pas les termes qui ont des conflits sémantiques en témoignent les résultats de l'outil V-Doc+; l'outil COMA et le prototype du projet ont résolu cette problématique en utilisant une approche hybride. De plus, les chercheurs de l'outil V-Doc+ ont aussi indiqué que l'usage d'une autre mesure leur permettrait de remédier à l'interopérabilité sémantique.

3.3.2 Comparaison de performance du prototype du projet versus d'autres outils de mise en correspondance

Le prototype a été implémenté à l'aide de la plateforme de Spark qui a permis d'obtenir un temps d'exécution en moins de 10 minutes pour effectuer la mise en correspondance des éléments de schéma de l'échantillon. La distribution des mots vides et des calculs matriciels des mesures de similarité sur chacun des exécuteurs de Spark ont aussi contribué à avoir une performance optimale. Tout en utilisant les mêmes spécifications techniques présentées au point 2.3.2, le nombre de termes des éléments de schémas a été augmenté de 600 en vue de simuler et valider la montée en charge du prototype. Ce dernier a réalisé la mise en correspondance des 600 termes en 20 minutes avec une bonne précision.

Les outils de mise en correspondance de la littérature, qui ont utilisé les plateformes adaptées aux mégadonnées, ont eu aussi une bonne performance et un temps de réponse supérieur aux autres technologies qui utilisent des plateformes et des langages plus traditionnels. Les chercheurs de l'outil V-Doc+ ont démontré que la plateforme Hadoop leur a permis d'effectuer en moins d'une heure l'alignement de 85000 classes dans une infrastructure en grappe de dix nœuds et les outils référencés dans leur étude comparative n'ont pas pu égaler ce résultat. Les chercheurs de FARMER ont aussi démontré que grâce à l'utilisation des fonctions internes de la plateforme Spark, les éléments de leurs schémas ont été partitionnés et les calculs sur différents exécuteurs ont été distribués de manière optimale.

3.3.3 Synthèse des discussions

Les différents résultats affichés par notre prototype prouvent que les algorithmes de mise en correspondance par le modèle Word2Vec et l'utilisation des plateformes de traitements de données massives peuvent fournir des résultats exploitables. Les principes d'apprentissage profond du modèle Word2Vec ont permis d'aboutir à un alignement complexe, comportement souhaité et recherché par les utilisateurs dans un outil de mise en correspondance. Ces techniques pourraient être étendues avec succès et la qualité des résultats obtenus par les mesures sémantiques est directement liée à la richesse lexicale et structurelle des ressources mises en œuvre.

Cette expérience a démontré que les éléments des schémas sont souvent spécifiques à un domaine et ils existent des termes qui ne sont pas définis par les ressources lexicographiques. À cet effet, l'utilisateur doit être en mesure de fournir les définitions relatives pour les termes de son choix et le module sémantique se baserait sur ces définitions pour effectuer la mise en correspondance. Ensuite, le prototype actuel n'affiche pas à l'utilisateur final les termes rejetés par les modules de similarité, il éclate les n-grammes en plusieurs mots et il ne valide pas le sens des termes génériques fourni par l'utilisateur.

CONCLUSION

L'objectif de ce projet de recherche appliquée consistait à générer un modèle de données génériques et une cartographie du marché financier, c'est-à-dire de trouver des approches qui permettent d'aligner les termes provenant des en-têtes des différents fichiers du marché financier et d'établir des correspondances entre les termes similaires pour bâtir un modèle de données générique. Ensuite, les données provenant de ces différents marchés devraient être chargées dans ce modèle générique dans un souci d'uniformisation, car il n'existe aucun protocole d'échange de données normalisé sur le marché financier. Ainsi, les acteurs du marché pourront exploiter et s'échanger des données à l'aide de cette nouvelle structure car les modèles de données des fournisseurs du marché financier sont développés indépendamment les uns des autres et ne partagent pas le même vocabulaire.

Une revue de la littérature a été effectuée pour identifier les méthodes et techniques de recherche qui ont été déjà utilisées pour traiter le problème de mise en correspondance pour les petits schémas et les schémas volumineux à large échelle, puis nous avons établi un lien entre les cadres théoriques et leurs applications à travers un prototype.

Le marché financier génère de plus en plus de grandes quantités de données, de ce fait, la méthodologie retenue pour implémenter le prototype se réfère autour des approches et techniques de correspondance de schémas volumineux et hybrides. Elle a aussi intégré les techniques récentes de TLN pour aboutir à la mise en correspondance des éléments de schémas après détection de conflits sémantiques.

Ce projet de recherche appliquée répond à la problématique initiale et permet aussi d'automatiser la correspondance et l'intégration de données hétérogènes, non seulement pour d'autres marchés financiers que celui expérimenté, mais aussi pour d'autres domaines. Ce modèle de données est aussi dynamique car le prototype implémenté permet de le confronter à différents fichiers lors de la phase de transformation d'un processus ETC pour procéder à sa mise à jour.

Les résultats obtenus par les métriques de précision et de rappel ont permis de confirmer que la tâche de mise en correspondance effectuée par le prototype est fiable. De plus, un mini dictionnaire a été bâti où le mot générique contient un ensemble de termes auxquels il peut être mappé. Ce dictionnaire aide à avoir une représentation bidirectionnelle des termes à mettre en correspondance.

En dépit du fait que des techniques et des outils récents (apprentissage profond) aient été utilisés sur des données non supervisées pour générer la correspondance entre les termes, il reste des points à améliorer dans ce premier prototype et ce projet ouvre aussi la voie à beaucoup d'autres perspectives de recherche.

RECOMMANDATIONS

L'utilisation d'un modèle d'apprentissage profond avec le modèle Word2Vec a déjà été entreprise pour bâtir la correspondance entre les termes qui étaient en rappel. Ceci a permis d'effectuer une correspondance de cardinalité multiple entre les termes et de diminuer le nombre de termes en rappel. Afin d'améliorer ces résultats, il est d'abord nécessaire de mesurer tous les éléments de schémas par une approche sémantique qui se base sur l'apprentissage profond des réseaux de neurones multicouches et ensuite de bâtir la matrice des termes par la similarité Cosinus. Ces modèles doivent être capables de traiter des n-grammes puisqu'il existe beaucoup de termes formés de mots composés dans les schémas de données boursières. Il sera ainsi nécessaire que les nouvelles versions du prototype puissent collecter par elles-mêmes plusieurs définitions pour un même terme afin d'alimenter le modèle d'apprentissage profond afin que ce dernier puisse s'autoévaluer et détecter par lui-même les correspondances erronées.

Ceci permettrait de réduire l'effort nécessaire pour valider les correspondances effectuées par le prototype ainsi que la création d'un référentiel pour comparer les résultats. Il faudrait également étudier d'autres techniques de pondérations des termes comparées au TF-IDF standard, ainsi que d'autres méthodes de calcul pour obtenir les similarités entre les termes et évaluer chacune d'elles. En effet, le choix de la méthode de calcul pourrait changer les performances du système ainsi que la fiabilité des résultats de mise en correspondance.

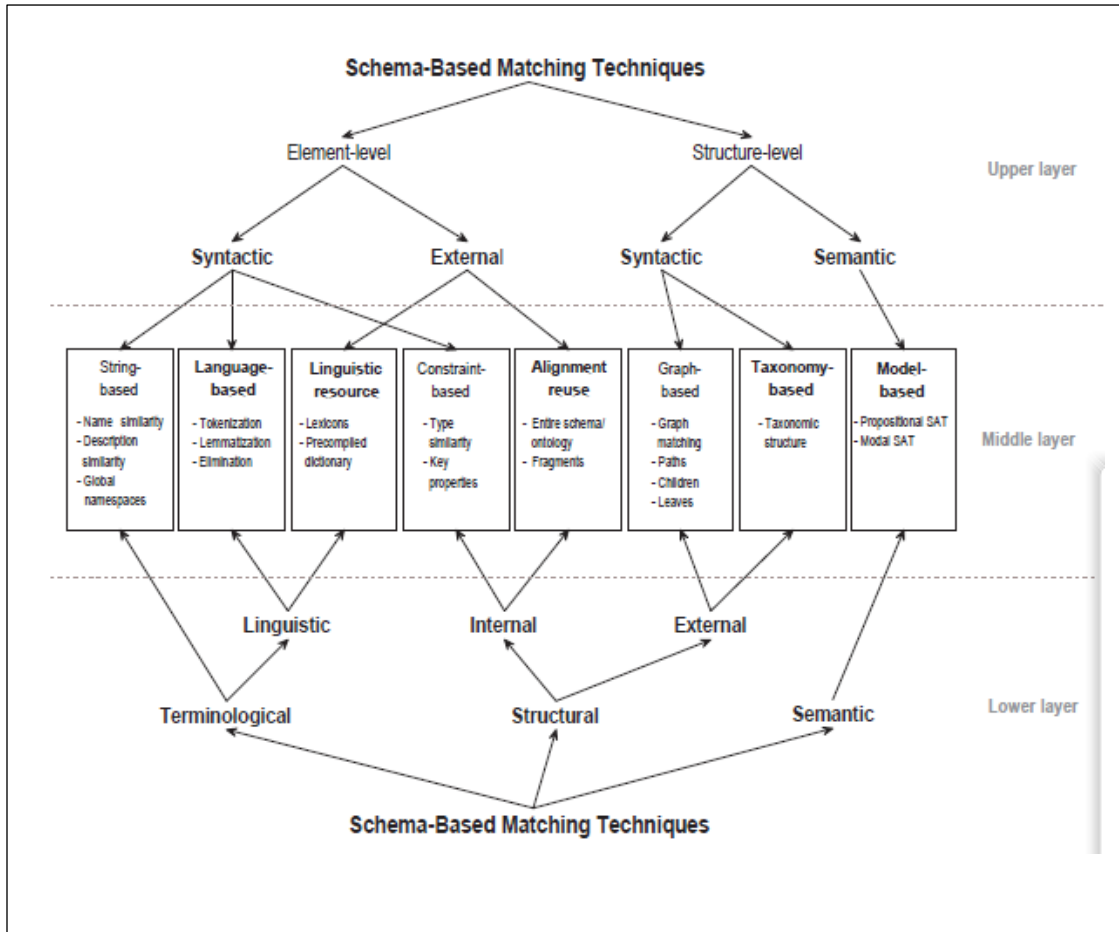
Du point de vue de l'expérience utilisateur, l'interface graphique du prototype actuel utilise un tableau pour afficher les correspondances mais l'utilisation de ce composant pour des schémas ayant plus de 300 éléments peut affecter l'ergonomie et rendre l'affichage illisible. Il conviendrait d'illustrer les similarités et les relations entre les mots avec des graphes. Le composant *Jgraphx* de Java et *Igraph* de Python seraient de bons candidats, car ils permettent de créer des graphes avec plus de 1 million de relations entre les nœuds.

Réutiliser le résultat de cette recherche et son prototype ainsi que concrétiser les différentes perspectives énoncées permettront de réaliser une mise en correspondance encore plus complexe pour les éléments des schémas provenant de diverses sources.

ANNEXE I

Approches d'alignements

Représentation schématique des approches d'alignements de Shvaiko et Euzenat (Shvaiko et Euzenat, 2005)



ANNEXE II

Représentation modèle générique

Le schéma ci-dessus représente une portion de la structure du modèle générique dans la base de données NoSQL d'Amazon en l'occurrence DynamoDB. La structure contient les noms des champs et leur type. Ce dernier peut être des chaînes (de l'anglais string), un numérique (de l'anglais number), des collections (de l'anglais stringset(SS), map(M)).

Le prototype crée la structure du modèle générique qui contient le terme générique saisi par l'utilisateur, les différentes mises en correspondance qui y sont associées ainsi que le marché boursier d'où proviennent les termes sources.

```

 DynamoDB JSON
{
  "champgenerique": {
    "S": "quantity"
  },
  "positionchampgenerique": {
    "N": "13"
  },
  "Correspondances": {
    "M": {
      "chix": {
        "S": "side"
      },
      "pure": {
        "SS": [
          "shares",
          "side"
        ]
      },
      "tmx": {
        "SS": [
          "lastqty",
          "volume"
        ]
      }
    }
  }
}

```


LISTE DE RÉFÉRENCES BIBLIOGRAPHIQUES

- Aumueller, D., Hong -Hai, D., Massmann, S., Rahm.E. (2005). Schema and ontology matching with COMA++. Proceedings of the ACM SIGMOD International Conference on Management of Data, Baltimore, Maryland, USA. New York, NY: ACM Press, 2005, pp 906–908.
- Baeza-Yates, R. A., Ribeiro-Neto, B. (1999).Modern Information Retrieval, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Bansal, S.K. (2014).Towards a semantic extract-transform-load (ETL) framework for big data integration. In Proceedings of International Congress on Big Data, pp. 522–529.
- Barker, W. Pomenarats, A. (2011). Banque du Canada. A Financial System Review, 53 p.
- Bernstein, P., Madhavan, J., Rahm, E. (2001). Generic schema matching with Cupid in the proceedings of VLDB, pp. 49–58.
- Black, F. (1971). Toward a fully automated exchange. Financial Analyst Journal July/August: 29–35.
- Big Data Pipeline: <https://www.xenonstack.com/blog/big-data-engineering/ingestion-processing-big-data-iot-stream/>
- Demetrio, M, Pires. C, Nascimento. D, DeQueiroz. A, Santos .V, Araujo,T. (2017).An efficient Spark-based adaptive windowing for entity matching, Journal of Systems and Software, vol. 128, pp. 1–10.
- Doan, A., Domingos, P., Halevy, A. (2001). Reconciling schemas of disparate data sources: A Machine-learning approach. In Proceedings of the International Conference on Management of Data (SIGMOD), pp. 509–520.
- Euzenat, J. (2007) .Semantic Precision and Recall for Ontology Alignment Evaluation. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, pp 348–353
- Golfarelli, M., Rizzi, S. (2009). Data warehouse design: modern principles and methodologies. Columbus: McGraw-Hill.
- Huang, H, Kerridge, J. M.,Chen, S.L. (2000). A query mediation approach to interoperability of heterogeneous databases. In Australasian Database Conference, pp.41–48.
- IBM Rational Sequence Diagram: <https://www.ibm.com/support/knowledgecenter/>

- Inmon, W. H. (1992). *Building the Data Warehouse*. New York, NY, USA: John Wiley & Sons, Inc.
- Kimball, R. (1998). The operational data warehouse. *DBMS* 11(1), pp 14–16.
- Kornacker, M., Behm, A., Bittorf, V., Bobrovitsky, C., Ching, A. Choi, Y.M.(2015). Impala: A modern, open-source SQL engine for Hadoop. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR '15)*.
- Levy, O., Goldberg, Y., Dagan, I. (2015). Improving Distributional Similarity with lessons learned from Word Embeddings. *Transactions of the Association for Computational Linguistics*, 3 (0): 211–225.
- Li W., Clifton C. (2000). SemInt: a tool for identifying attribute correspondences in heterogeneous databases using neural network. *Data Knowledge Eng* 33(1) pp 49–84.
- Luhn, Hans Peter (1957). A statistical approach to mechanized Encoding and Searching of Literary Information. *IBM Journal of research and development* 1 (4):315. doi:10.1147/rd.14.0309.
- Mikolov T., Quock, V. Le., Sutskever, I. (2013). Exploiting similarities among languages for machine translation.
- Millan, T., Mulatéro, F., Lamolle, M. (1998). Design Share and Re-use of Data and Applications into a Federate Database System. *Onzièmes Journées Internationales le Génie Logiciel et ses Applications*, Paris, France.
- Milo, T., Zohar, S. (1998). Using Schema Matching to Simplify Heterogeneous Data translation. *Proceedings of International Conference on VLDB*.
- Navigli, R (2009). Word Sense disambiguation a Survey. *ACM Computing Surveys*, ACM Press, 41(2): 1–69.
- Nothaft, F., Massie, M., Danford .T, Zhang, Z., Laserson, U., Yeksigian, C. Patterson, D.A. (2015). Rethinking Data-Intensive Science Using Scalable Analytics Systems. *SIGMOD '15 – Proceedings of the SIGMOD International Conference on Management of Data*.
- Pei, J, Hong, J., Bell, D., A. (2006). A Robust Approach to Schema Matching over Web Query Interfaces. *Proceedings of the 22nd International Conference on Data Engineering Workshops*, Atlanta, GA, USA. IEEE Computer Society, pp 46.
- Rahm, E., Bernstein, P. (2001). A survey of approaches to automatic schema matching, *VLDB Journal*: 334–350.

- Rahm, E. Do, H.H. (2001). COMA a system for flexible combination of schema matching approaches. In Proceedings of the Very Large Data Bases Conference (VLDB), pp. 610–621.
- Rahm, E. (2011). Towards large-scale schema and ontology matching. Schema matching and mapping, pp. 3–27.
- Rehurek, R., Sojka, P. (2001). Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Citeseer.
- Saeedi, A., Nentwig, M., Peukert, E., Rahm, E. (2018). Scalable matching and clustering of entities with FAMER. Complex Syst Informatics Model Q (CSIMQ), pp.61–83. <https://doi.org/10.7250/csimq.2018-16.04>.
- Scikit-learn: <https://scikit-learn.org/stable/index.html>
- Shvaiko, P., Euzenat, J. (2005). A Survey of Schema-based Matching approaches. Journal on Data Semantics IV, vol.3730: pp.146–171.
- Siméon, J. (2000). Data Integration with XML: A Solution for Modern Web Applications. Lecture at Temple University.
- Smiljanic, M. (2006). XML Schema Matching Balancing Efficiency and Effectiveness By means of clustering. Informatique. Netherlands: Center for Telematics and Information Technology (CTIT), 194 p.
- Wen-Syan, Li. Clifton, C. (1994). Semantic integration in heterogeneous databases using neural networks. In the proceedings of 20th International Conference on Very Large Data Bases, pp. 1–12.
- Wilks, Y., Stevenson, M (1996). The grammar of sense: Is word sense tagging much more than part-of-speech tagging? Technical Report CS-96-05, University of Sheffield, Sheffield, United Kingdom.
- Zhu, D., (2017). Degree projects in Computer science and engineering, Second cycle, Large Scale ETL Design, Optimization and Implementation Based On Spark and AWS Platform, KTH, Stockholm Sweden.

