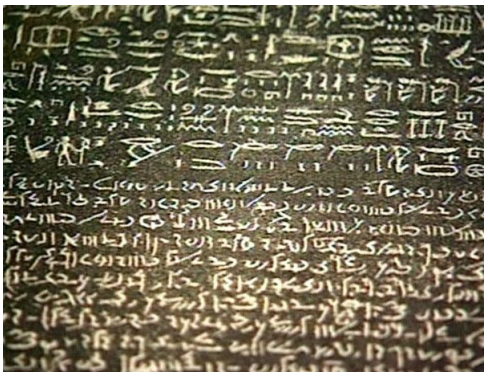
TextMine '19

Atelier sur la Fouille de Textes



Organisateurs :

Pascal Cuxac (INIST - CNRS),
Vincent Lemaire (Orange Labs),

Organisé conjointement à la conférence EGC
(Extraction et Gestion des Connaissances)
le 22 janvier 2019 à Metz

Editeurs :

Pascal Cuxac - INIST - CNRS
2 allée du Parc de Brabois, CS 10310, 54519 Vandoeuvre les Nancy Cedex
Email : pascal.cuxac@inist.fr

Vincent Lemaire - Orange Labs
2 avenue Pierre Marzin, 2300 Lannion
Email : vincent.lemaire@orange.com

Publisher:

Vincent Lemaire, Pascal Cuxac
2 avenue Pierre Marzin
22300 Lannion

Lannion, France, 2019

PRÉFACE

C'est une évidence que de dire que nous sommes entrés dans une ère où la donnée textuelle sous toute ses formes submerge chacun de nous que ce soit dans son environnement personnel ou professionnel : l'augmentation croissante de documents nécessaires aux entreprises ou aux administrations, la profusion de données textuelles disponibles via Internet, le développement des données en libre accès (OpenData), les bibliothèques et archives en lignes, les medias sociaux ne sont que quelques exemples illustrant l'évolution de la notion de texte, sa diversité et sa prolifération.

Face à cela les méthodes automatiques de fouille de données (data mining), et plus spécifiquement celles de fouille de textes (text mining) sont devenues incontournables. Récemment, les méthodes de deep learning ont créées de nouvelles possibilités de recherche pour traiter des données massives et de grandes dimensions. Cependant, de nombreuses questions restent en suspens, par exemple en ce qui concerne la gestion de gros corpus textuels multi-thématiques. Pouvoir disposer d'outils d'analyse textuelle efficaces, capables de s'adapter à de gros volumes de données, souvent de nature hétérogène, rarement structurés, dans des langues variées, des domaines très spécialisés ou au contraire de l'ordre du langage naturel reste un challenge.

La fouille de textes couvre de multiples domaines comme, le traitement automatique des langues, l'intelligence artificielle, la linguistique, les statistiques, l'informatique et les applications sont très diversifiées, que ce soit la recherche d'information, le filtrage de spam, le marketing, la veille scientifique ou économique, la lutte antiterroriste...

Le but de cet atelier est de réunir des chercheurs sur la thématique large de la fouille de textes. Cet atelier vise à offrir une occasion de rencontres pour les universitaires et les industriels, appartenant aux différentes communautés de l'intelligence artificielle, l'apprentissage automatique, le traitement automatique des langues, pour discuter des méthodes de fouille de texte au sens large et de leurs applications.

P. Cuxac — INIST-CNRS
V. Lemaire — Orange Labs

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Guillaume Cabanac (IRIT, Toulouse)

Mariane Clausel (Université de Lorraine, Nancy)

Vincent Claveau (IRISA, Rennes)

Guillaume Cleuziou (LIFO, Orléans)

Dominique Gay (U. Réunion, Saint Denis de la Réunion)

Natalia Grabar (STL - Lille3, Lille)

Mustapha Lebbah (LIPN, Paris)

Denis Maurel (Université F. Rabelais, Tours)

Patrick Paroubeck (LIMSI, Orsay)

David Reymond (Université du Sud, Toulon - Nice)

Julien Velcin (Université de Lyon, Lyon)

TABLE DES MATIÈRES

Exposé Invité

Session Exposés

Présentation des activités de recherche sur le traitement des textes en langues naturelles chez Orange

Frédéric Herlédan
Chef de projet Recherche
Orange Labs
2 avenue Pierre Marzin
22300 Lannion

frederic.herledan@orange.com

Frédéric Herlédan anime chez Orange un projet de recherche sur le traitement automatique des langues naturelles et le dialogue naturel.

Résumé : Depuis plus de 20 ans, Orange mène des activités de recherche dans le domaine du traitement automatique des langues naturelles. Avec l'évolution des technologies, il est devenu possible de traiter des textes de plus en plus variés. Les métiers et les activités des chercheurs ont aussi changé. Nous nous proposons de retracer rapidement ces évolutions et de lister quelques sujets qui intéressent aujourd'hui la recherche de l'opérateur.

Extraction de cartes d'inondation à partir de réseaux sociaux par apprentissage actif et production participative

Etienne Brangbour*, Pierrick Bruneau*
Stéphane Marchand-Maillet**

* LIST
Esch-sur-Alzette, Luxembourg
prenom.nom@list.lu,
** Université de Genève
Genève, Suisse
stephane.marchand-maillet@unige.ch

Résumé. Les réseaux sociaux sont beaucoup utilisés pendant les catastrophes naturelles pour rendre compte de situations difficiles. Extraire cette information peut servir à l'amélioration de modèles de prédiction de propagation, et aider les décisions relatives aux secours. Le projet Publimape vise à exploiter le contenu du réseau social Twitter en cas d'inondations de grande échelle en milieu urbain. Ce problème comporte de multiples aspects comme la géo-localisation, la temporalité, ou le traitement d'image. Dans ce papier nous présentons le travail en cours relatif au texte des statuts Twitter qui sont le point central de ce moyen de communication. Nous abordons la question de la représentation du texte issu de réseaux sociaux, la classification en relation avec la catastrophe naturelle, et la constitution d'un corpus grâce à la production participative (*Crowdsourcing*).

1 Contexte

Twitter est un système populaire de micro-blogs basé sur le texte. Ses utilisateurs peuvent y partager leurs opinions, expériences ou humeurs. Sa portabilité sur appareil mobile permet de réagir en temps réel à une catastrophe naturelle comme un feu de forêt, un tremblement de terre, ou une inondation. Les prédictions d'inondation sont traditionnellement obtenues grâce à des simulations physiques de l'écoulement de l'eau sur une région. Ces simulations sont générées à partir de mesures des niveaux d'eau et des précipitations relevées sur le terrain. Récemment, une méthode d'assimilation a été proposée pour injecter des cartes de probabilités d'inondation générées à partir d'images SAR (*Synthetic Aperture Radar*) dans le modèle de simulation [9]. L'objectif de notre travail est de générer des cartes d'inondation à partir de statuts Twitter pour avoir une source d'information complémentaire à assimiler.

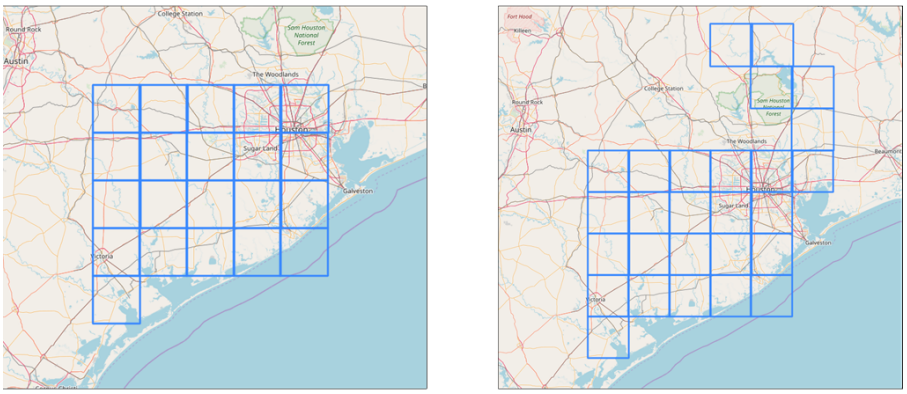


FIG. 1 – Gauche : *Collecte par API Twitter.* Droite : *Achat de données à Twitter*

2 Cas d'étude et collecte de données

Afin de valider les futures contributions du projet, nous avons choisi d'étudier les inondations qui ont eu lieu après le passage de l'ouragan Harvey dans la région de Houston (Texas). Nous avons collecté un premier corpus de tweets pendant l'événement grâce à l'API Twitter. La collecte s'est déroulée du 30 Août 1017 au 13 Septembre 2017 en sélectionnant les tweets dont la géo-localisation chevauche la zone d'intérêt décrite sur la figure 1. Il en a résulté un corpus d'environ 900K tweets. L'API ayant une vitesse de collecte limitée et des restrictions dans les filtres applicables, un second corpus a été acheté à Twitter. Pour celui ci, nous avons étendu la fenêtre temporelle ainsi que la zone d'intérêt en tenant compte du passage des cours d'eau (les zones environnantes étant plus susceptibles d'avoir des inondations) comme illustré sur la figure 1. Nous avons également étendu le filtre aux tweets des utilisateurs ayant déclaré leurs localisation dans la zone d'intérêt. Ce second corpus comporte environ 7,6M tweets.

Dans la littérature concernant la gestion et la détection d'événements en utilisant les réseaux sociaux, les statuts sont filtrés par mots-clés [16] [8] [15]. Nous pensons qu'un tel filtrage est trop restrictif et qu'une partie importante de l'information est perdue. Par exemple, dans : *"The Intersection of Asford Pkwy and Dairy Ashford Rd is singificantly higher than yesterday"*, aucun mot-clé associé aux inondations ne figure, pourtant le tweet contient de l'information exploitable. Pour ces raisons, nous avons choisi un filtrage par géo-localisation [1].

3 Représentation Textuelle

Afin de classifier les statuts, nous devons choisir une représentation adéquate. L'état de l'art en détection et gestion d'événement et en analyse du sentiment mentionne l'utilisation de mots-clés [16] ou des représentations classiques comme Tf-Idf ou Bag-of-Word [8] [10] [2]. Ces méthodes basées sur les occurrences de mots ont été développées pour l'étude de texte structuré et correctement orthographié. À l'opposé, les statuts des réseaux sociaux contiennent

beaucoup d'abréviations, d'argot, et de fautes d'orthographe. Une solution consisterait à pré-traiter les statuts, cependant un pré-traitement trop lourd peut mener à perdre de l'information.

En guise de solution, nous avons choisi d'utiliser Tweet2Vec, une représentation basée sur les caractères pour le contenu de réseaux sociaux[6]. L'avantage d'une représentation au niveau caractère est qu'elle sera plus robuste au texte informel et court qu'une représentation au niveau mot. Cette méthode utilise des réseaux de neurones récurrents pour tenter de prédire les hashtags attachés aux tweets. Le vecteur d'état final est utilisé comme représentation. [6] ont publié une implémentation sous licence BSD 2 avec un modèle pré-entraîné sur un ensemble de 2829 caractères (e.g. lettres, nombres, ponctuation, émoticones) et retournant un vecteur de 500 dimensions. Ce modèle a été utilisé lors de nos premières expériences, cependant nous avons collecté un corpus d'entraînement générique afin de pouvoir contrôler les paramètres décrits précédemment dans de futures expériences.

4 Entités nommées

Extraire les entités nommées, spécifiquement les noms de lieu, peut se révéler important pour la gestion d'évènement. En effet, en cas de catastrophe naturelle, on peut relever une proportion importante de tweets faisant mention d'une tierce personne subissant un sinistre en un lieu précis [13](e.g. *"My sister flooded Lumberton, Texas. Walmart on 69.. need help she is stranded and the family from Houston can not get to her"* dans le corpus). Dans ce cas, la géo-localisation du tweet doit être remplacée par la position associée à l'entité nommée. À l'aide d'outils comme Open Street Maps il est possible de retrouver un emplacement dans une zone d'intérêt à partir d'une entité nommée.

5 Modalité de Classification

Nous voulons classifier les tweets en trois classes déjà utilisées dans un contexte équivalent [3] :

1. *Non Pertinent* : Le statut ne contient aucune information concernant la crue
2. *Pertinent positif* : Le statut témoigne d'une crue à l'endroit ou se trouve l'utilisateur, ou en un autre lieu identifiable.
3. *Pertinent négatif* : Le statut témoigne d'une absence de crue à l'endroit ou se trouve l'utilisateur, ou en un autre lieu identifiable.

Compte tenu de la taille des corpus que nous avons collectés, et du coût de l'annotation manuelle d'un statut, nous avons choisi d'utiliser l'apprentissage actif [5] afin de réduire le coût total de l'annotation d'un corpus d'entraînement.

Nous avons réalisé une première expérience à partir d'un échantillon de 421 statuts étiquetés à la main par nos soins (316 en entraînement et 105 en test). Pour équilibrer ce sous ensemble, nous avons introduit un biais de mot-clef : les statuts ont été sélectionnés aléatoirement parmi ceux contenant le mot-clef *"flood"*.

L'objectif est de comparer plusieurs stratégies de requête dans le contexte de notre système de représentation. Cette expérience a été implémentée dans le langage Python avec la bibliothèque libact [17] qui expose les stratégies les plus communes. Nous avons comparé

les stratégies *Uncertainty sampling* et *Hierarchical Sampling*, ainsi qu'une stratégie aléatoire pour avoir une référence. Nous avons observé la précision d'un classifieur SVM durant l'apprentissage actif. Le résultat de cette expérience n'a pas permis de mettre en avant une stratégie spécifique. En effet, la précision maximale dépasse à peine 0.5 quelque soit la stratégie. Une explication possible est que la quantité de statuts annotés n'est pas suffisante pour entraîner un modèle de classification supervisé sur un espace d'une telle dimensionnalité.

6 Crowdsourcing

Afin de tester et valider nos travaux nous avons besoin d'un corpus étiqueté selon notre problème de classification ainsi qu'un relevé des entités nommées des lieux où l'inondation est constatée. Pour approcher cet objectif voulons utiliser une plateforme de production participative telle que *Mechanical Turk* [4]. Compte tenu de la taille de notre corpus (8M éléments), il nous faut sélectionner un sous ensemble représentatif à faire étiqueter. L'idée est d'avoir recours à l'*Active Learning* pour proposer une méthode sélectionnant les statuts à envoyer sur la plateforme afin de construire un corpus d'entraînement et un corpus de test.

Comme le montre la littérature, la production participative entraîne des problèmes de fiabilité des étiquetés données [14]. Des études ont été réalisées dans le contexte d'annotation d'images [14] [11], de l'analyse de média sociaux pour des applications proche de la notre [12], ou de la recherche d'entités nommées [7]. Notre travail sera entre autres d'évaluer le meilleur format de question afin de guider au mieux les travailleurs dans le contexte de notre application.

Références

[1] Geo objects - twitter developers. `https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/geo-objects.html`.

[2] R. Batool, A. Khattak, J. Maqbool, and S. Lee. Precise tweet classification and sentiment analysis.

[3] B. Bischke, P. Helber, Z. Zhao, J. de Bruijn, and D. Borth. The Multimedia Satellite Task at MediaEval 2018. 2018.

[4] M. Buhrmester, T. Kwang, and S. Gosling. Amazon's mechanical turk : A new source of inexpensive, yet high-quality, data ? *Perspectives on Psychological Science*, 6(1) :3–5, 2011.

[5] A. Dawod. Active learning survey.

[6] B. Dhingra, Z. Zhou, D. Fitzpatrick, M. Muehl, and W. Cohen. Tweet2vec : Character-Based Distributed Representations for Social Media. *arXiv :1605.03481 [cs]*.

[7] Tim Finin, William Murnane, Anand Karandikar, Nicholas Keller, Justin Martineau, and Mark Dredze. Annotating Named Entities in Twitter Data with Crowdsourcing. page 9.

[8] Y. Gao, S. Wang, A. Padmanabhan, J. Yin, and G. Cao. Mapping spatiotemporal patterns of events using social media : a case study of influenza trends. *International Journal of Geographical Information Science*, 32(3) :425–449, 2018.

[9] R. Hostache, M. Chini, L. Giustarini, J. Neal, D. Kavetski, M. Wood, G. Corato, R. Pelich, and P. Matgen. Near-real-time assimilation of sar-derived flood maps for improving flood forecasts. *Water Resources Research*, 54(8) :5516–5535, 2018.

[10] E. Kouloumpis, T. Wilson, and J. Moore. Twitter sentiment analysis : The good the bad and the omg ! 2011.

[11] B. Loni, J. Hare, M. Georgescu, M. Riegler, X. Zhu, M. Morchid, R. Dufour, and M. Larson. Getting by with a Little Help from the Crowd : Practical Approaches to Social Image Labeling. In *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia - CrowdMM '14*, pages 69–74. ACM Press, 2014.

[12] Boulos M., B. Resch, D. Crowley, J. Breslin, G. Sohn, R. Burtner, W. Pike, E. Jezierski, and K. Chuang. Crowdsourcing, citizen sensing and sensor web technologies for public and environmental health surveillance and crisis management : trends, OGC standards and application examples. *International Journal of Health Geographics*, 10(1) :67, 2011.

[13] S. Middleton, L. Middleton, and S. Modafferi. Real-time crisis mapping of natural disasters using social media. *IEEE Intelligent Systems*, 29(2) :9–17, 2014.

[14] S. Nowak and S. Rüger. How reliable are annotations via crowdsourcing : a study about inter-annotator agreement for multi-label image annotation. In *MIR*, page 557, 2010.

[15] H. Packer, S. Samangooei, J. Hare, N. Gibbins, and P. Lewis. Event detection using twitter and structured semantic query expansion. In *Proceedings of the 1st international workshop on Multimodal crowd sensing*, pages 7–14. ACM, 2012.

[16] T. Sakaki, M. Okazaki, and Y. Matsuo. Earthquake shakes twitter users : real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web*, pages 851–860. ACM, 2010.

[17] Y. Yang, S. Lee, Y. Chung, T. Wu, S. Chen, and H. Lin. libact : Pool-based active learning in python. *arXiv preprint arXiv :1710.00379*, 2017.

Summary

Social networks are widely used during natural disasters to account for difficult situations. Extracting this information can be used to improve propagation prediction models, and help rescue decisions. The Publimape project aims to exploit the contents of the Twitter social network in case of large-scale flooding in urban areas. This problem has multiple facets such as geo-location, temporality, or image processing. In this paper we present work in progress on Twitter text statuses, which are central in this medium. We consider the problem of representating text from social networks, their classification in relation with the natural disaster, and the constitution of an annotated corpus using crowdsourcing.

Etude expérimentale de classification textuelle multi-étiquette pour la relation client

Gil Francopoulo*, Léon-Paul Schaub**
Lynda Ould Younes***

*AKIO
gfrancopoulo@akio.com,
http://www.akio.com
**AKIO + LIMSI-CNRS
lpschaub@akio.com
***AKIO
louldyounes@akio.com

Résumé. La gestion de la relation avec les clients (GRC ou CRM selon le sigle anglais) est l'analyse des interactions des clients. Notre étude porte sur l'analyse du sens des textes pour en synthétiser les opinions et les sujets abordés par des clients qui s'expriment en plusieurs langues. L'approche de cette classification consiste à annoter les documents en différentes langues avec le même jeu de catégories, sachant que l'annotation est faite en une langue dite source ou native et qu'ensuite des algorithmes d'apprentissage automatique sont appliqués aux autres langues qui sont désignées comme les langues cibles ou non natives. Nous avons essayé différentes stratégies et comparé les options avec ou sans traitements linguistiques, de même que les différents algorithmes qu'ils soient neuronaux ou non. Les résultats de notre étude prouvent l'efficacité de notre approche quand elle est appliquée à des logiciels opérationnels.

1 Introduction

La gestion de la relation client est l'analyse des données des interactions des clients que l'on désigne sous le vocable de "voix du client'. Dans les systèmes modernes, le client a le choix entre différentes langues et canaux de communication comme le courrier électronique, le téléphone, le tchat, les médias sociaux et différentes enquêtes de satisfaction. À l'autre extrémité de l'analyse des données, les managers, les data-analystes et personnels du marketing qui consultent et paient pour le service veulent aussi avoir le choix entre une vaste gamme d'options sur le niveau de détail depuis le tableau de bord synthétique jusqu'à une sélection portant sur un sujet précis avec une classification à grains fins. Toutes ces navigations doivent être implémentées de façon à réagir à la volée en temps réel. Cela signifie qu'à la différence d'autres travaux, notre objectif n'est pas de catégoriser un document comme étant globalement bon ou mauvais (Pang et Lee, 2005). Ce n'est pas suffisant. Nous avons besoin de scruter profondément le sens des textes afin d'en extraire des indices sémantiques relativement fins comme le

sujet sur lequel porte l'option. Cela ne signifie pas que tous ces détails seront nécessairement présentés au lecteur final en toutes circonstances ; en fait, ces catégories peuvent être agrégées en des catégories plus abstraites, mais ces actions sont réalisées au sein d'un système de "business intelligence" qui est hors du champ du présent article. Notre étude se focalise donc sur la classification à grains fins. La ligne directrice de nos utilisateurs est : il faut comprendre pour pouvoir agir.

La classification monolingue de texte est l'affectation automatique d'une ou plusieurs catégories sémantiques dans une langue donnée. La catégorisation interlingue (Bel et al., 2003) est l'affectation des catégories à un texte d'une autre langue. Le raisonnement sous-jacent est de réduire le coût d'annotation à une unique langue considérée comme la langue native, et puis, à partir de cette langue, d'appliquer des algorithmes d'apprentissage automatique à d'autres langues qui sont considérées comme non natives ou langues cibles.

2 Contexte industriel

Notre société AKIO opère dans différents secteurs d'activité et les ressources associées à la personnalisation peuvent varier d'un secteur à l'autre. Ces domaines sont bien identifiés : 1) hôtels-restaurants 2) transport aérien 3) banque 4) sites de rencontres 5) e-commerce et boutiques 6) assurance. Les langues couvertes sont le français (considéré comme la langue native), l'anglais (natif mais moins développé donc non utilisé lors de l'apprentissage), l'espagnol, l'allemand, le portugais et l'italien. Le logiciel s'appelle AKIO Analytics. Toutes les paires secteur-langue sont implémentées selon la même stratégie avec des résultats similaires, mais pour la simplicité de l'exposé, nous nous focaliserons sur la paire e-commerce et boutiques pour l'espagnol.

3 Travaux connexes

Un certain nombre de classifications interlingues soit nécessitent des corpus parallèles soit ont besoin de documents annotés à la fois dans la langue source et cible (Xiao et Guo, 2013). Le problème principal est le manque de ressources à la fois fiables, diverses et volumineuses. La catégorisation dans de multiples langues peut être résolue en transférant la connaissance depuis une langue bien dotée vers une langue peu dotée. C'est pourquoi la plupart des systèmes emploient des ressources lexicales anglaises telles que SentiWordNet comme décrit dans l'état de l'art de (Dashtipour et al., 2016) et transfèrent vers d'autres langues. Certaines méthodes ne sont pas flexibles dans le choix des catégories tout au long de leur cycle de vie : en changeant les catégories, l'annotation sera à refaire ou bien nécessitera de gros efforts de transcodage manuel. Certaines approches sont basées sur des pivots entre la source et la cible. Ces pivots peuvent agir comme des filtres pour la classification et peuvent être utilisées dans du co-apprentissage comme dans (Wei et Pal, 2010). Une autre stratégie consiste à utiliser le texte traduit terme à terme avec un dictionnaire bilingue et ensuite à augmenter une sélection de documents de la langue source. Puis, un algorithme comme LSA (Latent Semantic Analysis) est appliqué pour obtenir une représentation interlingue (Gliozzo et Strapparava, 2006). Mais, pour autant que l'on sache, ces systèmes sont principalement appliqués à l'analyse d'opinions où la phrase est facilement transformable dans des pivots au sein de ces différentes langues. La

difficulté pour nous est de prendre en compte les phrases mal formées, les formes idiomatiques, l'ironie et les insultes, ce qui est très fréquent. La même remarque peut être faite à propos de CLESA (Cross-lingual Explicite Semantic Analysis) (Song et al., 2016), (Sorg et Cimiano, 2012). Nous faisons différemment, comme nous allons le voir par la suite.

4 Prétraitement et corpus

Concernant le prétraitement, nous avons développé en interne un pipeline linguistique robuste comprenant un tokeniseur, un correcteur orthographique et grammatical, un tagger-chunker statistique (Francopoulo, 2008), un analyseur syntaxique en dépendance, un annotateur de la négation et un détecteur d'entités nommées associé à un résolveur de co-référence. L'entrée peut être de niveau grand public avec d'importantes variations par rapport à la façon standard de s'exprimer. L'objectif est de normaliser l'entrée autant que possible. Tous ces outils sont optionnels dans le sens où, lors de nos expérimentations nous avons l'option de les utiliser ou non en fonction de l'évaluation globale. Ainsi, l'entrée peut être de quatre types : a) la chaîne brute d'origine, b) une suite de formes fléchies corrigées, c) une suite pleine de formes lemmatisées, encore appelées lemmes, d) une suite filtrée de lemmes corrigés. Précisons que les caractères tabulation et guillemet sont nettoyés pour tous les niveaux d'entrée car ce sont des caractères qui posent des problèmes de format pour certains logiciels et comme nous ne voulons pas introduire de biais en faveur d'un algorithme, nous les avons enlevés pour tous les logiciels. Le filtrage des lemmes consiste à ne prendre que les parties du discours comme les noms ou les adverbes de négation et d'ignorer d'autres mots comme les déterminants. On notera que le lemme est désambiguïsé car il est le résultat de la correction, tagging, chunking et de la résolution des entités nommées.

Concernant le système natif français, la classification est un système fondé sur des règles qui est accroché à la sortie du pipeline. En partant des lemmes corrigés, l'objectif est d'annoter le texte avec un ensemble de catégories. Le classifieur n'est pas un simple jeu de règles à plat, mais plutôt un système complexe fondé sur une organisation hiérarchique de composants sémantiques. Il y a trois types de composants : a) des composants transversaux qui sont valides pour tous les secteurs, b) une vingtaine de composants factorisables et c) six composants spécifiques aux secteurs. Un composant factorisable est par exemple le composant de la livraison qui est importé par le secteur du ecommerce (pour recevoir une robe) et par les restaurants (pour recevoir un repas) mais pas par le secteur des sites de rencontre. L'objectif est de partager les règles entre les secteurs afin de faciliter l'évolution et la maintenance. Le système est doté d'un dictionnaire de synonymes fondé sur CRISCO [1]. En prenant en compte l'héritage (donc en aplatissant virtuellement les composants pour un secteur), le secteur du e-commerce mobilise 9 000 règles. Le système est complexe mais bien organisé donc il est gérable et évolutif. La totalité du système, tous secteurs confondus comporte 17 500 règles.

Nous n'avons pas trouvé de corpus public pour notre étude, de ce fait, nous avons collecté un corpus par nous-mêmes. Notre corpus est constitué de nombreux textes venant de différents clients et canaux. Aucune modification manuelle n'est effectuée sur le contenu originel. La taille de chaque verbatim est petite, typiquement d'une longueur de cinq lignes. La plupart des documents sont des plaintes, le deuxième type de documents étant des questions qui sont sou-

1. http://crisco.unicaen.fr/des

vent des demandes de renseignement. La source étant le grand public, les textes contiennent de nombreuses fautes d'orthographe, des idiomes informels avec fréquemment une combinaison de tournures ironiques et d'insultes. Dit en d'autres termes, l'entrée est de mauvaise qualité et quelquefois, même pour un lecteur humain, le texte est difficile à comprendre. Les textes sont très différents de la Presse ou de Wikipedia tels qu'on peut les trouver dans des études comme (Zaid et al., 2017) par exemple. Le coût de l'annotation manuelle étant très élevé, nous adoptons une stratégie hybride afin de collecter une grande masse de données annotées. Une partie est annotée manuellement et le reste est engendré automatiquement à la manière des préparations de reconnaissance d'images au sein desquelles les images sont retournées pour faire grossir le corpus annoté. Nous gérons trois sous-corpus. La première sous-partie (appelée le corpus gold) est annotée par le système des règles et est constamment maintenue à l'aide d'un outil de vérification des tests de non régression. Le corpus gold comporte lui-même deux sous-parties : 90% pour la partie apprentissage et 10% pour le test. Le deuxième corpus (le corpus bronze) est automatiquement engendré depuis le corpus gold avec substitution des synonymes les plus fréquents en utilisant CRISCO. Pour éviter un biais dans le sens positif, nous ne créons pas de textes automatiquement pour les inclure dans le corpus de test car ils sont trop similaires aux textes du gold, donc seuls les textes de la partie apprentissage sont engendrés. Pour les corpus gold et bronze, le prétraitement et la classification ont été développés de manière à obtenir une annotation parfaite, ainsi la F-mesure est de 100% [2]. Le troisième corpus est un corpus du même domaine après application du système de règles. Le système français étant aux alentours d'une F-mesure de 85% sur les documents inconnus du même domaine, l'objectif est d'agrandir la taille du corpus même s'il existe une petite pénalité sur la qualité. Pour conclure, nous avons ainsi constitué un corpus d'une taille moyenne qui est représentative de l'activité de la relation client, comme présenté dans le tableau 1.

sous-corpus	mécanisme de constitution en français	FM	verbatim#	mots#
gold	développement manuel et écriture de règles	100	8 757	439 201
bronze	expansion automatique de synonymes depuis gold	100	15 269	1 299 180
silver	application automatique des règles	85	15 028	636 826
total			39K	2,4M

TAB. 1 – *Corpus.*

5 Catégories

Nous gérons trois types de catégories : les modalités d'expression, les thèmes et les opinions. Ces types ont été spécifiés après des études approfondies des flux de nos clients en complétant par les lectures de (Liu, 2012) et SentiWordNet (Esuli et Sebastiani, 2006), (Cambria et al., 2010). Comme il peut être observé dans le tableau 2, les catégories sont précises et nombreuses. La liste est relativement stable dans le temps, mais elle peut varier légèrement

2. Cela peut sembler inhabituel dans le cadre d'une annotation de corpus, mais le gold qualifie le système au sens industriel du terme dans la mesure où il fait fonction de test de non régression tout au long du cycle de vie. Ainsi, si des modifications sont apportées, le système ne sera pas mis en production si l'un des tests échoue

après une étude éditoriale prudente. Chaque texte peut être annoté par zéro (ce qui est rare), une ou plusieurs catégories. Il est à noter qu'il existe des opinions spéciales qui s'appellent NoSpecificTopicNeg et NoSpecificTopicPos utilisées pour annoter les rares situations où le locuteur ne donne aucune justification, thème ou détail comme "c'est nul" mais exprime clairement une opinion. Ajoutons aussi que l'opinion sur un thème est souvent désignée comme ABSA (aspect-based sentiment analysis) (Liu, 2012) dans la littérature scientifique et plusieurs études ont été menées sur les revues de films (Thet Tun et al., 2010), les produits électroniques (Hu et Liu, 2004), (Brody et Elhabad, 2010), les services (Long et al., 2010) et les restaurants (Ganu et al., 2009). Comparativement à SemVal sur ABSA, ce que nous nommons "thème" est appelé "Aspect Category Detection" et ce que nous appelons "opinion" est nommé "Aspect Category Polarity" (Pontiki et al., 2014).

type	définition	exemple	catégories#
modalité	forme générale de l'expression	injonction, question	4
thème	sujet de l'expression du locuteur	StoreDelivery, BankTransfer	117
opinion	jugement personnel du locuteur portant sur un thème et exprimé selon une polarité négative ou positive	MissingItemNeg, PricePos	58
total			179

TAB. 2 – *Catégories.*

6 Architecture du transfert interlingue

Concernant le flux de données, rappelons que nous n'avons pas de corpus parallèle. En outre, nous ne voulons pas passer du temps à annoter les corpus non natifs. Au moins deux architectures sont possibles : la première option consiste à traduire le texte espagnol en français lors de l'exploitation, et ensuite, à appliquer immédiatement le catégoriseur natif. La seconde option est de traduire un corpus de développement (lors de la phase de développement) puis d'apprendre et personnaliser un modèle de classification. Lors de l'exploitation, ce modèle est appliqué, cette phase étant nommée phase d'inférence. Le volume des messages est plutôt élevé : il peut être de 100K par jour, ainsi le coût en termes de service de traduction et de consommation CPU étant trop élevé avec la première option, nous adoptons la seconde option. Donc en résumé, nous faisons de la classification monolingue (en espagnol) au sein d'un système interlingue (français espagnol). Notre architecture de transfert est présentée dans la figure 1 avec Google Translation pour la traduction vers l'espagnol.

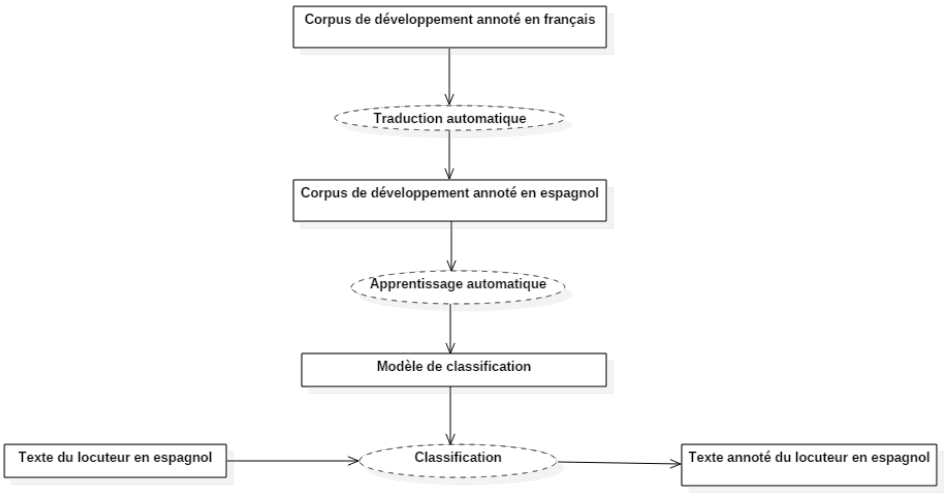


FIG. 1 – *Flux de transfert.*

7 Format d'entrée et algorithmes

Les options sur lesquelles nous pouvons agir sont :
— le niveau linguistique : chaîne brute, formes fléchies corrigées, lemmes corrigés, lemmes filtrés corrigés,
— le choix de l'algorithme.

Les algorithmes peuvent avoir les propriétés intrinsèques suivantes :
— sac de mots (bag of words / BOW) versus plongement de mots (word embedding / WE),
— la capacité de multi-étiquette (à la fois multi-classe et multi-label) : soit un modèle unique est capable de prendre en charge plusieurs catégories durant la phase d'apprentissage, soit il n'en est pas capable.

Nous avons étudié les algorithmes suivants : NB, classifieur SGD, SVM, SMO, FastText, CNN et BiLSTM.

NB, classifieur SGD, SVM et SMO sont des algorithmes linéaires d'apprentissage automatique. Nous utilisons l'implémentation de Weka [3]. NB est utilisé comme option de base avec ses hyper-paramètres par défaut. Ainsi, à la différence des autres algorithmes, aucune personnalisation des hyper-paramètres n'est effectuée pour NB. Le classifieur SGD de Weka implémente la descente de gradient stochastique avec ses paramètres par défaut : la fonction hinge loss avec un taux d'apprentissage de 0,01 est la meilleure combinaison pour nos données. SVM est une encapsulation de LIBSVM (Dashtipour et al., 2016), après réglage, la fonction noyau linéaire et un type C-SVM est le meilleur choix. SMO implémente l'optimisation séquentielle minimale de John Platt pour ensuite appeler un noyau SVM. Nous adoptons l'option de normalisation et un paramètre de tolérance de 0,001. Le classifieur SGD, SVM et SMO ne permettant

3. https://www.waikato.ac.nz/ml/weka

nom	nom complet	bibliothèque	version	origine
NB	Naive Bayes	Weka	3-8-3	Univ. de Waikato
classifieur SGD	Stochastic Gradient Descent	Weka	3-8-3	Univ. de Waikato
SVM	Support Vector Machine	Weka+LIBSVM	3-8-3	Univ. de Waikato
SMO	Sequential Minimal Optimization	Weka	3-8-3	Univ. de Waikato
FastText	FastText	FastText	0-3	Facebook
CNN	Convolutional Neural Network	TensorFlow	1-10	Google
BiLSTM	Bi-directional Long Short-Term Memory	TensorFlow	1-10	Google

TAB. 3 – *Algorithmes.*

pas le traitement multi-étiquette, nous effectuons un apprentissage binaire et de multiples inférences sont appliquées en séquence, donc la distribution des catégories est considérée comme indépendante. Une description complète de ces algorithmes peut être consultée dans le livre Weka (Witten et al., 2016).

FastText [4] est une régression logistique multi-classe à partir des moyennes des enchâssements des n-gram de caractères (Joulin et al., 2017). Ces enchâssements sont appris au préalable avec l'extension morphologique de skip-gram avec échantillonnage négatif (Bojanowski et al., 2017). L'apprentissage préalable est effectué sur un sur-ensemble deux fois plus vaste de nos corpus avec des textes du secteur d'activité concerné. Nous utilisons les paramètres par défaut car nous n'avons pas observé d'améliorations en faisant varier les réglages.

CNN est un réseau de neurones profond comprenant différentes couches comme les couches de convolution, des couches complètement connectées et des couches de normalisation (Kim, 2014). Nous utilisons TensorFlow [5] avec 300 époques pour une taille de batch de 128.

BiLSTM est un algorithme neuronal profond basé sur des unités de réseaux neuronaux récurrents (Hochreiter et Schmidhuber, 1997). Nous utilisons la fonction sigmoïd, la cross-entropie comme fonction de perte, ADAM pour l'optimisation et 300 époques pour une taille de batch de 128.

CNN et BiLSTM sont présentés selon deux formes : la première sans corpus pré-entraîné et la seconde (notée CNN-W2V et BiLSTM-W2V) avec le corpus pré-entraîné sur le Wikipedia espagnol [6].

8 Méthodologie d'expérimentation

Le corpus gold est découpé en 90% pour l'apprentissage et 10% pour le test. Les corpus bronze et silver sont utilisés à 100% pour l'apprentissage comme indiqué dans le chapitre sur le corpus. Pour déterminer les valeurs des hyper-paramètres mentionnées au paragraphe précédent, nous avons effectué une "grid selection" sur le corpus gold. Nous le faisons uniquement

4. https://fasttext.cc
5. https://www.tensorflow.org
6. https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md

sur le gold car nous n'avons pas d'autre choix du fait du nombre trop élevé de calculs que cela nécessite. Nous ne faisons varier que les paramètres les plus significatifs en développant toutes les valeurs pour les énumérations symboliques, en énumérant les puissances de 2 pour les grandes valeurs et par puissance de 10 pour les petites valeurs. Nous limitons chaque session à une semaine de calcul maximum pour chaque algorithme. Une fois fixés les hyper-paramètres, nous évaluons sur le gold, bronze et silver.

9 Comparaison des temps d'apprentissage et d'inférence

Les vitesses des différents algorithmes sont extrêmement variables. D'un point de vue scientifique, le temps de calcul n'est pas essentiel, mais il est important au plan pratique car plus l'algorithme est rapide, plus on peut faire d'essais de réglage. D'autre part, cela facilite la mise en place industrielle car nous avons 6 secteurs d'activité pour 4 langues, ce qui fait 24 sessions. Certaines implémentations utilisent les cartes graphiques basées sur les coeurs CUDA alors que d'autres ne le font pas. Tous les apprentissages CUDA sont calculés sur NVIDIA GTX-1080. Les apprentissages sont réalisés sur un Xeon W2155 en utilisant 12 fils d'exécution. Toutes les inférences sont effectuées sur un seul fil d'exécution. Le tableau 4 donne les temps pour deux niveaux linguistiques d'entrée : les chaînes brutes et les lemmes filtrés corrigés. Les mesures des autres niveaux ne sont pas présentées mais figurent entre ces deux extrêmes. Deux considérations expliquent ces différences : premièrement, quand on procède avec des lemmes corrigés, la diversité des indices pour l'apprentissage automatique est plus petite, et deuxièmement, après filtrage pour chaque phrase le nombre de mots est plus petit.

nom	multi-étiquette	ordre	CUDA	temps d'appr. chaînes brutes	temps d'appr. lemmes corrigés	temps d'inférence
NB	non	BOW	non	2 h 30	50 mn	7 mn
SGD	non	BOW	non	6 h	2 h	31 s
SVM	non	BOW	non	4 h 50	1 h 44	39 s
SMO	non	BOW	non	5 jours 16 h	15 h	21 s
FastText	oui	WE	non	15 mn	15 mn	2 s
CNN	oui	WE	oui	1 h 3	37 mn	2 s
BiLSTM	oui	WE	oui	27 h	14 h 30	10 s

TAB. 4 – *Temps de traitement.*

10 Comparaison de la qualité selon le niveau linguistique et choix des options

Les mesures de qualité des différentes sessions de calcul sont présentées dans le tableau 5 en fonction du niveau linguistique de l'entrée de la catégorisation, avec les conventions que R signifie rappel, P précision et FM F-mesure, définie comme la moyenne harmonique de R et P. Pour faciliter la lecture, les FM au dessus de 70 sont entourées.

nom	chaînes brutes			formes fléchies corrigées			lemmes corrigés non filtrés			lemmes corrigés filtrés		
	R	P	FM	R	P	FM	R	P	FM	R	P	FM
NB	68	19	30,0	74	21	32,6	72	22	34,2	73	25	37,4
SGD	68	79	72,9	70	76	73,3	69	73	71,0	69	72	70,4
SVM	58	87	69,8	57	87	68,7	50	88	64,0	48	87	61,6
SMO	68	83	74,4	70	81	75,2	67	78	72,2	65	79	71,3
FastText	45	61	51,6	45	55	49,7	44	47	45,7	45	48	46,4
CNN	67	40	50,8	65	36	46,8	70	38	49,4	69	32	44,7
BiLSTM	74	36	48,7	76	37	50,0	77	38	51,4	78	40	53,2
CNN-W2V	72	32	45,2	72	29	41,8	73	31	44,3	72	30	42,4
BiLSTM-W2V	79	47	59,6	78	48	59,8	78	47	59,1	81	48	60,4

TAB. 5 – *Qualité.*

Tout d'abord, après avoir écarté NB, nous observons que SGD, SVM, SMO, FastText donnent une meilleure précision en comparaison du rappel. C'est le contraire pour CNN et BiLSTM. Maintenant, concernant strictement la FM pour SVM, SMO, FastText, CNN, l'entrée brute donne de meilleures résultats que les lemmes filtrés. Pour BiLSTM, c'est le contraire. Pour SGD, les valeurs sont très proches. Concernant la différence avec ou sans corpus préentraîné, la différence de l'ajout est positive pour BiLSTM mais elle est négative pour CNN. On observe aussi que SMO est toujours meilleur que SVM même si la différence interne entre les deux n'est que l'ajout d'un pré-traitement avec ensuite l'appel à un noyau SVM. Il faut maintenant évaluer le surapprentissage pour déterminer si certaines options produisent des modèles qui sont trop étroitement liés aux données d'apprentissage et ne prévoient pas les observations futures du même secteur d'activité. Nous procédons selon une intervalidation de 10 plis. Ainsi, les jeux de tests sont déplacés par fractions d'un dixième du corpus gold et l'apprentissage est recalculé sur les 90% restants en ajoutant le bronze et le silver. Ensuite nous faisons la moyenne que nous comparons avec la valeur initiale du tableau 5. Comme cela prend 10 fois plus de temps que l'évaluation en un pli, nous n'avons pas eu le temps de refaire les calculs pour la totalité des options. Nous l'avons fait pour SGD, car il est rapide à la fois pour les chaînes brutes et les lemmes filtrés. Pour les chaînes brutes, la FM passe de 72,9 à 69,8, pour les lemmes filtrés, la FM passe de 70,4 à 70,0. On observe donc que l'apprentissage sur les chaînes brutes a tendance à se comporter en surapprentissage, tout ceci avec un temps de calcul trois fois plus long.

Concernant le choix des options pour la mise en production, si on se focalise sur les FM au dessus de 70, l'option des formes brutes avec SMO n'est pas réaliste car le temps d'apprentissage est trop long pour les moyens de calcul dont nous disposons actuellement. En l'état actuel de nos évaluations, nous optons pour le classifieur SGD avec les lemmes corrigés filtrés, étant entendu que ce choix pourrait être remis en question à la lumière de futurs développements.

11 Discussion

Jusqu'à présent, nous n'avons pas parlé des variations que nous avons testées et qui n'ont pas donné de bons résultats comme la prise en compte des entités nommées, soit en les enlevant, soit en les remplaçant par leur type, par exemple : nom de société, de ville ou de personne. Cela ne donne pas une différence statistique significative. Actuellement, nous n'inscrivons que la valeur "nom propre" dans le filtrage des lemmes. D'autre part, nous avons essayé de créer des corpus via des allers-retours de traduction pour construire automatiquement des corpus similaires au gold en bénéficiant des annotations du français. Nous avons essayé vers l'espagnol ainsi que vers l'anglais. Cela n'a pas donné de bons résultats car même si l'on pouvait comprendre le sens, ce ne sont pas des formes qu'un francophone écrirait. A contrario, pour le corpus bronze (celui des synonymes) et silver (celui qui est automatique) : les gains ont été respectivement de 6% et 5%, ce qui explique que nous combinons maintenant gold, bronze et silver. Concernant les algorithmes, nous avons essayé de combiner BiLSTM avec deux couches de CNN, mais la qualité n'était pas bonne.

A cause du manque de place, nous ne pouvons pas justifier en détail toutes nos choix d'architecture. L'une de celles-ci concerne la flexibilité de choisir l'espace d'annotation qui n'est pas complètement figé et nécessite d'être adapté légèrement de temps en temps. Avec notre architecture, tout est contrôlé par le système natif associé avec des tests stricts de non régression. Comparé avec d'autres architectures où les catégories sont réparties sur plusieurs langues ou sur de nombreux documents de manière incontrôlée, notre système est facilement gérable car il est localisé au sein du système natif, la distribution en direction des systèmes non natifs étant entièrement automatique. En contre-partie, une des faiblesses de notre approche est que si dans une langue, les clients ont des préoccupations qui ne sont pas exprimées dans la version française pour des raisons culturelles, nous ne traitons pas ces préoccupations, puisque tout part du français. Jusqu'à présent, ce n'est pas un problème que nous avons rencontré. Si tel était le cas, alors nous devrions inclure ces formes dans la langue source et ensuite, relancer les calculs. Une autre limitation est que la qualité globale dépend de la qualité de la traduction automatique. A ce propos, lors des tests, nous avons détecté une dizaine de documents avec une mauvaise traduction. Nous avons implémenté un mécanisme d'exception pour prendre en compte la traduction manuelle rectificative. Nous n'avons pas eu le temps de travailler sur ce sujet qui concerne principalement la traduction des injures et des formes idiomatiques pour lesquelles Google Translation n'est pas très bon. Nous avons simplement observé qu'il était préférable de fournir à Google Translation non pas la forme initiale, mais la forme corrigée dans la mesure où nous disposons d'un correcteur orthographique bien adapté à notre domaine.

12 Futurs développements

Il se pourrait que les algorithmes neuronaux donnent de meilleurs résultats avec plus de données. De ce fait, nous allons continuer à faire grossir les corpus tout en continuant à comparer les options. D'autre part, il va falloir mesurer le surapprentissage pour les meilleures options, mais les temps de calculs sont très importants. Pour CNN et BiLSTM, nous prévoyons aussi de comparer la qualité avec un modèle construit à partir des données dont nous disposons dans le domaine de la relation client au lieu d'utiliser Wikipedia. Pour les systèmes neuronaux, nous avons fait quelques essais avec un mécanisme d'attention servant de première couche de

chaque réseau et d'entrée pour la première couche que ce soit pour CNN ou BiLSTM, mais pour l'instant les essais ne sont pas très concluants : d'autres tentatives sont nécessaires. Enfin, suite à nos recherches, nous avons remarqué que plusieurs systèmes désormais utilisaient en dernière couche d'activation des CRF afin d'avoir des probabilités markoviennes et ainsi une meilleure représentation du texte. Concernant les canaux d'entrée, actuellement l'entrée est purement textuelle en différents genres. Pour l'instant, nous traitons tous les canaux de la relation client sauf le téléphone mais nous prévoyons de travailler sur l'intégration d'un module de parole vers texte (STT) car la voix reste un canal majeur en termes de volume d'interactions.

13 Conclusion

Dans la présente étude, nous avons donc décrit une série d'expériences qui montrent qu'il est possible de classifier les documents en de multiples langues à partir d'un espace d'annotation français. Nos résultats ne sont pas très loin de l'état de l'art qui est de 75% comme présenté dans (Banea et al., 2010), malgré le fait que la qualité en entrée de nos textes est bien moindre que la leur, ce qui rend la tâche plus difficile. Notre étude soulève une question importante pour le traitement automatique des langues : a t-on besoin d'effectuer des prétraitements linguistiques avec les outils traditionnels, sachant que l'apprentissage automatique profond peut constuire des couches internes qui sont plus ou moins équivalentes à ce qui est construit par les outils traditionnels ? Ces idées sont mentionnées par exemple dans (Dias et al., 2018). Nous pensons qu'il est un peu trop tôt pour trancher.

Références

Banea, C., R. Mihalcea, et J. Wiebe (2010). Multilingual subjectivity : Are more languages better ? *Proceedings of COLING*.

Bel, N., C. Koster, et M. Villegas (2003). Cross-lingual text categorization. *Research and Advanced Technology for Digital Libraries*.

Bojanowski, P., E. Grave, A. Joulin, et T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*.

Brody, S. et N. Elhabad (2010). An unsupervised aspect-sentiment model for online reviews. *Proceedings of NAACL*.

Cambria, E., R. Speer, C. Havast, et A. Hussain (2010). Senticnet : A publicly available semantic resource for opinion mining. *Proceedings of AAAI Fall Symposium*.

Dashtipour, K., S. Poria, A. Hussain, E. Cambria, A. A. Hawalah, A. Gelbukh, et Q. Zhou (2016). Multilingual sentiment analysis : State of the art and independent comparison of techniques. *Cognitive Computation*.

Dias, C.-E., C. Gainon de Forsan de Gabriac, V. Guigne, et P. Gallinari (2018). RNN et modèles d'attention pour l'apprentissage de profils textuels personnalisés. *Proceedings of CORIA*.

Esuli, A. et F. Sebastiani (2006). Sentiwordnet : A publicly available lexical resource for opinion mining. *Proceedings of LREC*.

Francopoulo, G. (2008). Tagparser : well on the way to ISO-TC37 conformance. *Proceedings of the International Conference on Global Interoperability*.

Ganu, G., N. Elhadad, et A. Marian (2009). Beyond the stars : Improving rating predictions using review text content. *Proceedings of WebDB*.

Gliozzo, A. et C. Strapparava (2006). Exploiting comparable corpora and bilingual dictionaries for cross-language text categorization. *Proceedings of the ICCL-ACL*.

Hochreiter, S. et J. Schmidhuber (1997). Long short-term memory. *Neural Computation*.

Hu, M. et B. Liu (2004). Mining and summarizing customer reviews. *Proceedings of AAAI*.

Joulin, A., E. Grave, P. Bojanowski, et T. Mikolov (2017). Bag of tricks for efficient text classification. *Proceedings of the European Chapter of ACL*.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Proceedings of EMNLP*.

Liu, B. (2012). *Sentiment Analysis and Opinion Mining*. Morgan and Claypool.

Long, C., J. Zhang, et X. Zhu (2010). A review selection approach for accurate feature rating estimation. *Proceedings of COLING*.

Pang, B. et K. Lee (2005). Seeing stars : Exploiting class relationships for sentiment categorization with respect to rating scales. *Proceedings of ACL*.

Pontiki, M., D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, et S. Manandhar (2014). Semeval-2014 task 4 : Aspect bases sentiment analysis. *Proceedings of the 8th International Workshop on Semantic Evaluation at COLING 2014*.

Song, Y., S. Upadhyay, H. Peng, et D. Roth (2016). Cross-lingual dataless classification for many languages. *Proceedings of IJCAI*.

Sorg, P. et P. Cimiano (2012). Exploiting wikipedia for cross-lingual and multilingual information retrieval. *Data and Knowledge Engineering*.

Thet Tun, T., J.-C. Na, et C. Koo (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Information Science*.

Wei, B. et C. Pal (2010). Cross lingual adaptation : An experiment on sentiment classifications. *Proceedings of ACL*.

Witten, I., E. Frank, M. Hall, et C. Pal (2016). *Data Mining : Practical Machine Learning Tools and Techniques, 4th edition*. Morgan Kaufmann.

Xiao, M. et Y. Guo (2013). Semi supervised representation learning for cross-lingual text classification. *Proceedings of the EMNLP*.

Zaid, E., T. Lehinevych, et A. Glybovets (2017). Cross-language text classification with convolution neural networks from scratch. *Computer Sciences and Mathematics*.

Summary

The paper deals with cross-language classification for customer relationship management within a commercial running system. The aim is to start from a source language where resources are avalaible, then to translate automatically and to learn within one target language. Various linguistic options and modern algorithms are presented and compared.

Mining Sequential Patterns for Hypernym Relation Extraction

Ahmad Issa Alaa Aldine[*,***], Mounira Harzallah[**]
Giuseppe Berio[*] Nicolas Bechet[*]
Ahmad Faour[***]

[*]University Bretagne Sud, IRISA Lab, France - Vannes
[**]Nantes University, LS2N Lab, France - Nantes
[***]Lebanese University, Lebanon - Beirut

Abstract. Hearst's patterns are popular patterns for hypernym relation extraction from text. In this work, we propose to apply sequential pattern mining to learn Sequential Hearst's Patterns (SHP), a new formalization of Hearst's patterns as sequential patterns using lexical and syntactical information. A comparison between the three types of Hearst's patterns (the original Hearst's patterns, an extended set of Hearst's patterns and Hearst's patterns reformulated by using dependency relations) is provided by using a music domain corpus. The results show that precision and recall are improved by SHP.

1 Introduction

Hypernym relations are very useful for building taxonomies that are considered the backbone of ontologies. In addition, they are useful for other tasks such as automated translation and information retrieval. Pattern-based approaches have been pioneering for extracting hypernym relations. Patterns are either manually defined or automatically extracted. Hearst (1992) earlier introduced a list of patterns over lexical informations extracted from the text, suggesting when hypernym relations are stated by that text between noun phrases. For instance, the pattern "NP such as NPs" means that a noun phrase (NP) must be followed by term "such", term "as", and then by a NP or a list of NPs. Hearst's patterns suffer from low recall because they are few (5 patterns) and there are several ways to express the same relation in one text. Furthermore, traditional Hearst's patterns are defined as regular expressions. Regular expressions are restricted to match exactly the right arrangement, thus any intermediate occurrence of a word prevents successfully matching the text to the patterns. For instance, let's consider the sentence "piano (pianoforte) is a musical instrument"; occurrence of "(pianoforte)" prevents Hearst's pattern "NP is a NP" to match the sentence and extract hypernym relation (piano, musical instrument). In addition, Hearst's patterns are prone to make errors due to the ambiguity of sentences. For instance, sentence "Musical instruments in Spain such as a guitar" matches the pattern "NP such as NP" extracting a wrong hypernym relation (Spain, guitar). Such errors reduce pattern precision.

Various approaches have been proposed to increase recall and precision of Hearst's patterns. Most of these approaches extend Hearst's patterns by manually defining new patterns

(Jacques and Aussenac-Gilles, 2006; Orna-Montesinos, 2011; Seitner et al., 2016) or automatically extracting new patterns (Snow et al., 2005; Nakashole et al., 2012). Other approaches, especially for increasing precision, add extra information to the patterns. These information are either syntactic (using dependency parsing) (Snow et al., 2005; Nakashole et al., 2012) or semantic (using given semantic relations) (Ponzetto and Strube, 2011). More recently, Aldine et al. (2018) propose to increase precision and recall by manually redefining Hearst's patterns as set of grammatical dependencies instead of regular expressions. Dependency based patterns should increase precision because they include extra syntactic information (dependency relations) relating much better the words in a sentence. Additionally, they should allow achieving better recall because they match sentences without arrangement restriction. However, results show that adding dependency relations improves a little bit precision while recall is dramatically improved.

In this work, we propose a semi-supervised approach to learn patterns correspond to Hearst's patterns by using a sequential pattern mining algorithm. The learned sequential patterns comprise lexical and syntactical information to increase precision. In addition, we use gap constraints to learn and match sequential patterns for recall increasing. The sequential Hearst's patterns (SHP) are learned, evaluated, and compared to traditional Hearst's patterns (HP), extended set of Hearst's patterns (extHP), and dependency Hearst's patterns (DHP) by using one corpus made available by Camacho-Collados et al. (2018) describing knowledge relevant to the music domain. The results confirm that SHP yields better performance than HP, extHP, and DHP in terms of precision and recall.

2 Sequential Hearst's Patterns Learning

In this section, we provide some details about the proposed approach. The following are the workflow steps to learn one or more sequential patterns correspond to one of the Hearst's pattern. First, we extract a set of learning sentences from the corpus using a dataset of given known hypernym relations. Learning sentences are sentences truly matched with one of the Hearst's patterns: the pattern matches the sentence and identifies true hypernym relation according to the dataset. Second, we build the sequence database from the set of learning sentences by representing each sentence as a sequence (more details about sequence representation are provided in the remainder). Third, we apply sequential patterns mining algorithm to extract frequent sequential patterns. We use CloSPEC (Béchet et al., 2015) an algorithm to extract frequent closed sequential patterns under multiple constraints such as minimum gap and maximum gap. A sequential pattern is frequent if its occurrence support is greater than minimum threshold. A sequential pattern is closed if it does not exist more general sequential patterns with equal support. Finally, we apply a selection step to keep the most relevant sequential patterns from a large set of mined frequent sequential patterns. For that purpose, we have used a set of validation sentences (it is equally divided between positive and negative). The positive sentences are sentences truly matched with Hearst's patterns, thus the number of expected true for each Hearst's pattern is known. Then, for each mined frequent sequential pattern, we evaluate its precision, recall, and f-beta measure. We select the k sequential patterns with the highest value of f-beta as the best relevant sequential patterns. β and k have to be tuned in order to select the best relevant sequential patterns. The workflow steps are repeatable to learn one or more sequential patterns for each Hearst's pattern.

To represent a sentence as a sequence, we perform the following processing: Tokenization, Lemmatization, POS tagging, and Dependency Parsing. As a result, we obtain for each token (word or multi-words) of the sentence it's POS tag, lemma, and dependency relation. Dependency relation is a binary grammatical relation between two tokens: Dependent token and Head token. We transform each dependency relation into a sequence element, comprising the following items: the Dependent token, the Dependent token lemma, the Dependent token POS tag, the grammatical relation, and the Head token.

A sequential pattern matches a sentence if the sequential pattern is a subsequence of the sequence representing the sentence. A sequential pattern SP is a subsequence of sentence sequence SS if each SP element is a subset of SS element and the subset of elements satisfies the minimum and maximum gap constraints. In other words, if SP element e_k is a subset of SS element e_i, SP element e_{k+1} must be a subset of SS element e_j, where $(i + min\_gap) < j \leq (i + max\_gap)$. An SP element e_i is a subset of SS element e_j, if all items of e_i are included in e_j.

3 Corpus and Evaluation

The corpus used for learning and validation is provided by SemEval2018 hypernym discovery task organizers (Camacho-Collados et al., 2018). A dataset of hypernym relations that should be found is also provided. This dataset is used to label sentences of the corpus as positive or negative. A sentence is positive if it comprises at least one couple hypernym/hyponym found in the dataset; otherwise, it is negative. After sentence labeling, we noticed the existence of positive sentences but they do not convey the expected meaning. Thus, we are obliged to apply corpus cleaning to remove such sentences. The final corpus used to evaluate our approach contains approximately 60,000 sentences, where half of them are cleaned positive sentences. Table 1 shows the performance of our approach (SHP) compared to the performance of traditional Hearst's Patterns (HP), extended set of Hearst's Patterns (extHP) (Seitner et al., 2016), and Dependency Hearst's Patterns (DHP) (Aldine et al., 2018). We show in the table the 3 best results achieved by SHP after tuning K and β parameters. The results confirm the better performance of the learned sequential Hearst's patterns compared to the performance of the Hearst's patterns defined as regular expressions (HP), the extended set of HP (extHP), and the Hearst's patterns defined as dependency patterns (DHP). In addition, we achieve a new benefit using our approach, which is the dynamic of learning patterns with high precision ($k = 1$ and β closer to 0.1) or learning patterns with high recall (greater k and β closer to 1).

4 Conclusion

Hearst's patterns based approaches have been extensively used to extract hypernym relations. Traditional Hearst's patterns are defined manually as regular expressions. More recently, dependency relations have been used to manually redefine Hearst's patterns as dependency patterns. In this paper, we propose an approach to automatically learn sequential Hearst's patterns represented by using dependency relations and using sequential pattern mining. The evaluation results confirm that our learned patterns outperform the patterns found in the literaure for extracting hypernym relations.

TAB. 1: The performance of our approach and other approaches on music corpus.

Music	HP	extHP	DHP	SHP(k=3, β=1)	SHP(k=5, β=1)	SHP(k=1, β=0.1)
Pre	0.1235	0.1161	0.0967	0.1313	0.1154	**0.1919**
Rec	0.0305	0.042	0.0429	0.0532	**0.0627**	0.0259
F-m	0.0489	0.0617	0.0594	0.0757	**0.0813**	0.0456

References

Aldine, A. I. A., M. Harzallah, B. Giuseppe, N. Bechet, and A. Faour (2018). Redefining hearst patterns by using dependency relations. In *Proceedings of the 10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - Volume 2: KEOD,*, pp. 148–155. INSTICC: SciTePress.

Béchet, N., P. Cellier, T. Charnois, and B. Crémilleux (2015). Sequence mining under multiple constraints. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, SAC '15, New York, NY, USA, pp. 908–914. ACM.

Camacho-Collados, J., C. Delli Bovi, L. Espinosa-Anke, S. Oramas, T. Pasini, E. Santus, V. Shwartz, R. Navigli, and H. Saggion (2018). SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, United States. Association for Computational Linguistics.

Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. *In Proceedings of the 14th International Conference on Computational Linguistics*, 539–545.

Jacques, M.-P. and N. Aussenac-Gilles (2006). Variabilité des performances des outils de tal et genre textuel. cas des patrons lexico-syntaxiques. *47*, 11–32.

Nakashole, N., G. Weikum, and F. Suchanek (2012). Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '12, Stroudsburg, PA, USA, pp. 1135–1145. Association for Computational Linguistics.

Orna-Montesinos, C. (2011). Words and patterns: Lexico-grammatical patterns and semantic relations in domain-specific discourses. *24*.

Ponzetto, S. P. and M. Strube (2011). Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence 175*(9), 1737 – 1756.

Seitner, J., C. Bizer, K. Eckert, S. Faralli, R. Meusel, H. Paulheim, and S. P. Ponzetto (2016). A large database of hypernymy relations extracted from the web. In *LREC*.

Snow, R., D. Jurafsky, and A. Ng (2005). Learning syntactic patterns for automatic hypernym discovery. *MIT Press*, 1297–1304.

Résumé

Aide à la sauvegarde et aux évolutions du patrimoine logiciel

Françoise Deloule*

*Université Savoie Mont Blanc, Polytech Annecy-Chambéry, LISTIC
BP 80439 Annecy-le-Vieux
F-74944 ANNECY CEDEX, France
Francoise.Deloule@univ-smb.fr,
http://www.listic.univ-smb.fr

Résumé. Bon nombre de connaissances et de savoir-faire de nos entreprises sont mémorisés dans les codes informatiques de leurs logiciels métier. Mais en fonction des dates de création de ceux-ci, les connaissances renfermées sont loin d'être explicites, notamment car leur codification nécessitait de sérieuses abréviations. Ces logiciels ne peuvent être réécrits et doivent cependant être maintenus pour suivre les évolutions de l'entreprise. Ceci nécessite de la part de ceux qui gèrent ces programmes qu'ils en saisissent la sémantique. Nous proposons une aide basée sur une approche ontologique pour mieux comprendre et utiliser les connaissances et les savoirs faire contenus dans ces logiciels. En nous appuyant sur un cas industriel, nous construirons une ontologie du domaine et une ontologie des actions puis nous mettrons en lumière et exploiterons les liens qui peuvent exister entre les concepts de cette ontologie et les portions de code qui portent sur les mêmes connaissances.

1 Problématique générale

Dans les entreprises, le logiciel est présent à tous les niveaux. Seuls les codes métier se référant à la production spécifique d'une société nous intéressent ici. En effet, souvent présents dès la création de l'entreprise, ils contiennent une part importante de leur savoir-faire spécifique. Ils contiennent souvent plusieurs milliers de ligne de code, écrites dans différents langages de programmation. Le terme de patrimoine logiciel reprend bien d'une part cette notion historique et son aspect indispensable pour le cœur de métier de l'entreprise, et d'autre part une notion d'évolution dans le temps pour que la société puisse innover et maintenir sa place sur le marché économique.

La maintenance des codes sources et l'ensemble de la documentation de la conception aux tests représentent un coût important pour chaque projet (Erlikh (2000)). Ce coût très élevé s'explique notamment par le coût de la compréhension des programmes (Muller et al. (1993)). Celle-ci peut s'appuyer sur la perception des propriétés structurelles et dynamiques des programmes. De nombreux travaux et approches tentent de remonter au niveau d'une modélisation conceptuelle des programmes et montrent la complexité de cette tâche. Elle est souvent rendue difficile car les logiciels anciens respectaient peu les normes de qualité logicielle, leur documentation est obsolète voire absente, et les méthodes et langages de programmation ont évolué

au fil du temps. On constate souvent un réel fossé entre les développeurs initiaux programmant dans des langages dit procéduraux, et ceux en charge de leur maintenance baignant dans une culture des langages à objets.

La compréhension des programmes nécessite aussi de s'appuyer sur la réalité des processus métier du domaine d'application concerné (connaissances descriptives et fonctionnelles). Or, dans les langages plus anciens, la longueur des noms des variables utilisées était très limitée et obligeait les programmeurs à utiliser des abréviations. Celles-ci pouvaient à la fois porter sur la désignation de l'objet du métier et sur sa nature dans le code. Ceci complexifie encore la compréhension du code et ses liens avec les détails techniques de fabrication des produits de sa société.

De plus, de nombreuses entreprises, ont pour des raisons économiques, sous-traité et externalisé leurs développements informatiques. Mais pour pouvoir évoluer et se protéger de la concurrence, elles doivent retrouver la maitrise de pans entiers de leur patrimoine technique et scientifique. Les coûts de réappropriation et de maintenance de ces logiciels sont donc très élevés et l'intérêt pour une aide à leur compréhension s'en trouve accru.

Les travaux menés en rétro-ingénierie du logiciel (Eilam (2005)) s'intéressent plus particulièrement à la structure et aux comportements des logiciels, mais les outils restent insuffisants pour identifier les ressources métier implicitement contenues dans les logiciels métier. De ce fait, une partie importante des connaissances métier restent inaccessibles alors qu'elles sont présentes dans ces lignes de code constituant un patrimoine important. Les travaux portant sur la compréhension des programmes sont nombreux (O'Brien (2003)), (Storey (2006)) et une conférence internationale porte sur cette thématique (voir ICPC). Dès (Brooks (1983)) la compréhension des programmes est définie comme une « mise en correspondance » des éléments du code source avec les connaissances du domaine.

Les travaux présentés ici visent à faciliter la maintenance et l'évolution de logiciels existants. Nous nous intéressons à la mise en relation des éléments contenus dans les codes source avec les connaissances du domaine relatives aux objets manipulés en nous appuyant sur la construction d'ontologies sur un domaine. Leur construction permet d'éliciter bon nombre de connaissances implicites des experts, tant comme concepts que comme actions.

2 Démarche

Pour que le patrimoine logiciel d'une entreprise puisse survivre et évoluer, il convient d'apporter une aide complémentaire aux outils classiquement utilisés en retro-ingénierie du logiciel, notamment pour augmenter la prise en compte de la sémantique contenue dans les lignes de code, en relation avec les savoirs et savoir-faire du domaine.

Notre approche se décompose en 4 étapes. Tout d'abord, nous construisons une ontologie du domaine avec des experts. Si certaines connaissances sont formalisées dans un langage de programmation, elles ne le sont que partiellement. Ici, on ne s'intéresse pas directement à la rétro-ingénierie des structures de données ou des classes à partir du code. D'une part ces structures intègrent les connaissances et contraintes du (des) langage(s) de programmation, d'autre part la puissance des outils de retro-ingénierie actuels ne permet pas encore de remonter aux diagrammes UML de niveau conceptuel (Korshunova et al. (2006)), (Sutton et Maletic (2007)). De plus ces modèles conceptuels ne peuvent pas correspondre à l'ontologie du domaine, même s'il existe une forte corrélation. L'alignement des diagrammes de classes extraits

automatiquement des programmes avec l'ontologie du domaine figure parmi nos perspectives de travail. La définition d'un concept de l'ontologie peut être traduite non pas en une classe (ou structure de données) mais en de multiples attributs de dénominations différentes. Réciproquement, il existe des concepts de l'ontologie nécessaires à la compréhension des connaissances du domaine pour lesquels ne correspond aucun élément logiciel. On retrouve ici les résultats de travaux de recherche menés sur l'extraction d'ontologies à partir de textes (un programme pouvant être considéré comme un texte) où le réseau lexical ne se superpose pas au réseau conceptuel (Roche (2007)).

La deuxième étape consiste à construire la terminologie du domaine, terminologie qui sera ensuite mise en relation avec l'ontologie du domaine. Cela consiste à identifier la façon dont on « parle » des concepts de l'ontologie au sein des programmes dans le cadre d'une programmation dans un langage donné. Par exemple, en Fortran les termes pourront être les identifiants d'entités ou tables de la base de données associée, de structures, de types ou de variables. En C++, les termes seront les identifiants de classes ou d'attributs, de structures ou de types, des variables d'instances. Ils désigneront les concepts de l'ontologie. Le résultat est, d'après (Roche (2012)), une ontoterminologie, c'est-à-dire une terminologie dont le système conceptuel est une ontologie formelle. Ainsi, dans le cadre de notre application logicielle sur les articulations mécaniques, les termes « IrotulCF » et « idRotul » se réfèrent au concept <Rotule> de l'ontologie (voir figure 2). Ceci apporte la maitrise d'une partie du vocabulaire se rapportant aux concepts de l'ontologie utilisé dans les programmes.

Puis nous proposons d'élargir cette démarche aux connaissances décrivant la dynamique des processus métier traduits dans ces mêmes codes sources. Nous proposons de définir et de construire une onterminologie des actions présentes dans ces codes, appartenant au même sous-domaine de connaissances. Pour cela nous précisons ce que nous entendons par action, leurs propriétés et caractéristiques ainsi que les liens qu'elles entretiennent avec certains concepts de l'ontologie construite précédemment. Il ne s'agit pas ici de remonter à des modèles conceptuels ou de rivaliser avec les outils classiquement utilisés comme BPMN (White (2008)) mais seulement d'apporter un complément de compréhension des programmes que nous devons sauvegarder ou maintenir.

Enfin, avant de conclure, nous montrons comment ces connaissances métier sont liées aux codes sources, à l'aide de traces d'exécution. Des statistiques sur les apparitions des concepts en fonction du temps, les mises en jeu conjointes de différents concepts permettent de mieux comprendre ces liens implicites entre code et connaissances métier.

3 Concepts métier et ontologie

3.1 Contexte applicatif

Nous présentons rapidement un contexte industriel. Nous nous appuyons sur une petite partie des logiciels d'une entreprise française spécialiste des composants des articulations mécaniques. Ces éléments apparaissent notamment dans ce que nous appelons souvent des bras articulés, que ce soit par exemple le support d'une lampe dite d'architecte, le bras d'une pelle mécanique, les rotules permettant d'assurer la direction des véhicules, ou encore les articulations de robots de peinture. Historiquement, cette société s'est intéressée aux articulations d'éléments métalliques en se basant sur les propriétés des différents métaux. La plupart des

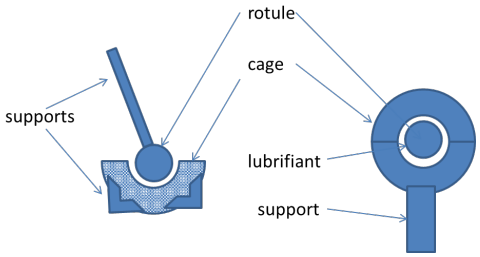


FIG. 1 – *exemples de schéma de deux types d'articulations simples à rotule*

logiciels datant de cette époque sont écrits en Fortran77, C et C++. Ces produits constituent toujours une large partie de la gamme proposée aux clients. Puis les évolutions ont introduit d'autres produits utilisant notamment les polymères. Pour ces produits, les parties du logiciel intégrant ces innovations ont été développés en C++, mais s'appuient aussi sur les programmes plus anciens. Enfin, les programmeurs actuels souhaitent utiliser des langages et méthodes de programmation plus récents ou plus performants (langages à objets, python, Java, NodeJS, ...).

Nous nous limitons ici au cas des articulations métalliques à rotules, permettant des mouvements dans 2 ou 3 dimensions. Certaines peuvent être décrites en 3 parties : une **tête** (ou **rotule**) s'articulant dans une **cage** contenant ou non un fluide **lubrifiant**, chacune de ces pièces pouvant se prolonger par un **support** (figure 1).La tête peut prendre différentes formes, d'un simple axe à une combinaison plus complexe d'axes, de roulements et de rotules. Une cage peut être un simple guide plein ou ajouré, de forme sphérique plus ou moins ouverte ou encore une sphère complète et étanche pouvant recevoir et garder un lubrifiant. Différents métaux peuvent être associés en fonction de leurs caractéristiques respectives et surtout de l'environnement dans lequel sera utilisée cette articulation. Ils peuvent ou non être traités, revêtus d'un autre matériau ou avoir subi des traitements de surface. Les lubrifiants nécessaires (pour les articulations fermées) doivent présenter des propriétés chimiques adaptées à l'utilisation de l'articulation.

Cette brève description doit être complétée, pour s'adapter à chaque client voire à chaque projet, d'une part par des caractéristiques dimensionnelles tant interne qu'externe, d'autre part par des caractéristiques de résistance liées au poids des éléments supportés, aux degrés de liberté à prendre en compte, à la nécessité d'introduire un fluide lubrifiant, à la fréquence d'utilisation sans arrêt de l'articulation et enfin aux supports associés (intégrés dans la masse, vissés, ... ou inexistants).

La grande variété des produits et de leurs utilisations, dans des environnements à chaque fois différents nécessitent de disposer de nombreux programmes utilisés pour définir les qualités des articulations demandées par les clients. Les programmes qui portent sur les produits RSMM (articulations simples à rotule entièrement métalliques) ont été identifiés, isolés et fournis par l'entreprise. Ils comportent plusieurs milliers de ligne de code source en Fortran, C ou C++. Il représente un patrimoine technique très important dont la préservation est vitale pour l'entreprise, les développeurs des débuts partant successivement à la retraite. L'entreprise est vivement intéressée par cette activité de rétro-ingénierie de logiciel, notamment pour en avoir une vue d'ensemble aussi complète que possible, pour faciliter l'innovation sans avoir à redévelopper les parties existantes. Cependant, l'absence de documentation de conception ou de programmation de ces lignes de code ainsi que la complexité et le volume des connaissances

métier à maitriser nous conduisent à mettre en œuvre la démarche proposée ci-dessus.

3.2 Les ontologies

La construction d'une ontologie ne peut se faire sans l'intervention des experts. Un groupe de travail associant mécaniciens, cogniticiens et programmeurs de l'entreprise a donc été constitué. Par phases successives, les connaissances nécessaires à la création de l'ontologie se rapportant aux programmes traités ont pu être élicitées et expliquées.

Dans un premier temps, nous avons construit l'ontologie des concepts permettant de comprendre ce que sont les différentes articulations, leurs propriétés et leur constitution. Cette construction a été guidée par les deux points de vue : mécanique et informatique, et deux objectifs. Le premier objectif est bien sûr de comprendre le contenu du code pour en faciliter l'indispensable maintenance. Le deuxième objectif est de pouvoir transférer les connaissances et savoir-faire dans un cadre de conservation et d'évolution du patrimoine industriel de l'entreprise. Celle-ci a besoin de s'appuyer sur ce patrimoine pour faire évoluer ses produits, augmenter sa compétitivité, capitaliser les connaissances et pérenniser les savoirs faire de l'entreprise, et donc innover.

Notre démarche globale reposant sur la mise en correspondance de l'ontologie du domaine avec les programmes informatiques, nous cherchons ce lien par les termes utilisés dans le code. Il est important de noter qu'il n'y a pas d'isomorphisme entre les deux structures. En effet, certains concepts de l'ontologie sont nécessaires à la compréhension du domaine mais ne sont pas directement traduits dans le code ou certains concepts seront traduits différemment (classe, attribut de classe, ...). Inversement, certaines parties du code apparaîtront sans corrélation directe avec les concepts de l'ontologie, mais seront nécessaire à l'exécution et aux interfaces des programmes. Par exemple, les concepts de l'ontologie « matière » doivent être modélisés car ils sont nécessaires à la compréhension des aspects mécaniques et techniques de la structure des articulations et de leurs utilisations (cœur de métier de l'entreprise). La constitution d'une articulation est directement liée au choix des matières disponibles car il existe une forte interaction entre ces matériaux et les utilisations de l'articulation. Ces connaissances n'apparaissent pas telles quelles dans les programmes. Cependant, certaines instances de l'ontologie correspondent à des valeurs de variables du programme, comme descriptif d'un composant ou comme paramètre de calcul pour évaluer la robustesse de l'articulation.

D'un autre côté, des structures informatiques de données ou des classes apparaissent dans les programmes sans corrélation avec l'ontologie. Celles-ci sont induites tant par le langage de programmation choisi que par la finalité des programmes (calculs). Des données telles que *resistlocal* et *résistglobal* jouent un rôle important dans la compréhension des choix d'articulations. En effet, la résistance globale se calcule en fonction des résistances plus localisées, par combinaison additive ou complétive des différents éléments de l'articulation. Certains éléments conceptuels sont pris en compte dans divers éléments de programmation. Par exemple, la résistance d'un matériau est prise en compte sous différentes formes et dans différentes structures de données ou classes (*attribut qui intervient pour vérifier la compatibilité des matériaux composant l'articulation, ainsi que dans les méthodes de calcul*).

Les concepts utiles de l'ontologie construite doivent être nommés à l'aide d'un identifiant unique dans l'ensemble de notre structure ontologique. Mais cette identification est souvent loin des termes utilisés par les différents intervenants. Aussi, pour chaque concept, nous cherchons tous les termes qui y font référence. Ce seront ici très souvent des termes du langage

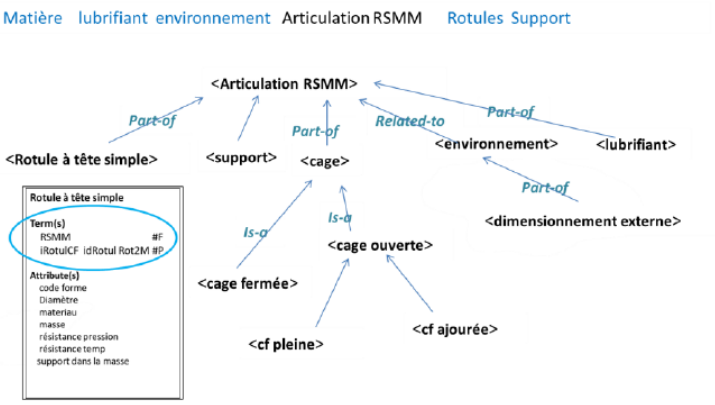


FIG. 2 – *Extrait de l'ontoterminologie métier et correspondance avec le code*

de programmation (il peut y avoir des termes dans les différentes langues naturelles aussi) car ce sont ceux qui désignent nos éléments de connaissance dans les programmes. Certains ont une formation historique intégrant des propriétés propres aux langages et contraintes informatiques. Certains reprennent l'initiale ou le début de tous les éléments concernés (figure 2 : IrotulCF désigne la valeur entière attribuée (code) à une rotule placée dans une Cage Fixe). Pour identifier ces termes, nous nous sommes ainsi intéressés aux méthodes de recherche de termes basées sur les similarités (Harth et Dugerdil (2015)).

La construction de l'ontologie a nécessité l'intervention des experts pour établir aussi rapidement que possible les correspondances entre les concepts et les attributs de ceux-ci dans l'ontologie et les noms de variables, de classes et d'attributs utilisés dans les programmes. Les informaticiens ont pu aussi soit rattacher les termes informatiques non référencés à un ou plusieurs éléments de l'ontologie, soit expliquer leur utilisation par l'activité de programmation proprement dite (i.e. les paramètres formels des méthodes qui ne peuvent au sens strict être considérés ici comme des termes).

L'ontoterminologie construite a permis de formaliser une soixantaine de concepts avec leur dénomination et leurs attributs. Cette spécification est complétée par les terminologies nécessaires pour désigner et manipuler ces concepts et ces attributs de concepts tant en français qu'en langages de programmation. Par exemple, le terme (en langue naturelle) de RSMM fait référence à une Rotule Simple Métal-Métal. Le terme Rot2M désigne ce même concept, en langage de programmation. La figure 2 permet de visualiser une partie de cette ontoterminologie construite, la représentation graphique des concepts de l'ontologie et l'affichage des termes associés aux concepts. Une liste de termes peut aussi être associée à chaque attribut.

4 Actions et ontologie

La construction de l'ontologie précédente permet de représenter les connaissances descriptives du domaine. Cependant, il peut être intéressant de formaliser aussi les actions dans lesquelles ces concepts sont nécessairement mis en jeu dans le savoir-faire de l'entreprise. De

nombreux travaux portent sur la représentation des processus métier au niveau global de l'entreprise (BPMN (White (2008)), ERP, . . .). Par contre, il est difficile de confronter ces modèles aux ontologies métier. Cependant, le savoir-faire d'une entreprise gagne à être modélisé pour en assurer la transmission, faciliter les évolutions, permettre les vraies innovations et éventuellement neutraliser les codes obsolètes.

Si les savoirs sont souvent décrits par des substantifs (ou formes nominales), les actions sont souvent traduites par des expressions verbales (verbes et complément). La langue française, comme beaucoup de langues, est riche en verbes d'action (en français, tous sauf 6 verbes d'état), certains entretiennent des relations de synonymie plus ou moins forte, comme pour les substantifs (Sagot et Fiser (2008)). On pourrait les organiser pour obtenir une ontologie de haut niveau sémantique. Mais ici, nous nous intéressons à un niveau plus interne, celui de l'ontologie métier précédente. En nous appuyant sur les connaissances des experts métier, nous retiendrons les expressions métier partagées et consensuelles traduisant ces actions, éléments du savoir-faire, qui vont nécessairement s'appuyer sur l'ontologie des concepts.

Comme pour les concepts, il n'y a pas de recouvrement fort entre structure du code et action. Une action peut faire appel à plusieurs procédures ou méthodes, une procédure peut associer une ou plusieurs actions. Le but de l'ontologie des actions n'est pas de représenter la dynamique d'un système, mais d'apporter aux systèmes dédiés à cette tâche une aide pour s'appuyer sur des connaissances pertinentes, stables et consensuelles, nécessairement complémentaire de la description des concepts de ce domaine. Dans le cas de la retro-ingénierie du logiciel, elle permettra d'améliorer la compréhension du logiciel.

Après avoir construit l'ontoterminologie des concepts, nous proposons de construire une ontologie des actions couvrant le même sous-domaine, par un raisonnement analogue. Puis nous définissons et associons à chaque action son environnement proche, comme les concepts mis en jeu, pour établir les premiers liens Concepts/Actions.

Nous commençons par donner quelques définitions. Une action est une mise en œuvre de connaissances par un agent ou une entité (ici un programme) pour atteindre un objectif. La combinaison des actions définit la capacité d'action d'un système donné.

Une action conceptuelle (désignée dans la suite du texte par le seul mot action) est une représentation des manipulations des connaissances métier de l'ontologie. Cette action peut mobiliser un ou plusieurs concepts, ou/et porter sur l'un ou plusieurs attributs d'un ou de plusieurs concepts. Une action est décrite par différents attributs ou propriétés. Les actions sont désignées de façon unique pour être manipulables dans la structure ontologique. Cette désignation sera une expression verbale, souvent un verbe à l'infinitif avec des adverbes et compléments si nécessaire (initialiser, choisir-cage, afficher . . .). On pourra ensuite lui associer tous les termes avec lesquels les différents intervenants en parlent.

Une action peut être rattachée à un type d'action (création, destruction, transformation et communication), et être caractérisée par l'effet de sa mise en œuvre (réversible ou définitive, répétitive, récursive ou à exécution unique, sélective, initiale ou terminale). L'action d'initialiser est une action de création, initiale, à exécution unique et définitive (au sens de l'occurrence de l'exécution d'un programme, bien entendu).

Une action peut être mise en relation avec une autre action conceptuelle par une relation de subsomption 'est-une-façon-de-faire' ou une relation partitive 'contribue-à'. Elle peut aussi être instanciée par la relation 'réalisation-de' pour une réalisation particulière et datée d'une action (exemple sur la figure 3).

La construction de l'ontologie d'actions métier nécessite l'élicitation d'actions dites « élémentaires » significatives et pertinentes au regard du métier concerné et au sens de la cognition, c.à.d. compréhensible pour les experts et les utilisateurs, et de ce fait favorisant les explications sur des actions de plus haut niveau sémantique. Nous pouvons nous appuyer sur les outils et méthodes existants pour la création et l'exploitation des ontologies. Certains éléments ont déjà pu être fournis lors de l'extraction des connaissances visant à construire l'ontologie des concepts. Mais comme nous prenons un point de vue différent, cela peut faciliter ou augmenter l'explicitation des connaissances implicites des experts. Cette nouvelle étape permettra de valider les rôles respectifs des concepts dans le système.

L'intérêt de cette ontologie des actions est de permettre de représenter les compétences nécessaires à la réalisation d'un l'objectif final, avec une mise en relation avec des concepts de l'ontologie de ce même domaine. Pour cela nous proposons d'enrichir la description de chaque action par la définition de son environnement proche (sans prise en compte des répercussions éventuelles de cette action sur d'autres éléments, concepts ou actions), c.à.d. l'expression claire des concepts ou propriétés de concepts qui sont mis en jeu lors de cette action. Nous nous inspirons de diverses approches (position centrale du verbe dans les langues européennes, actigrammes (Marca et McGowan (1987)) ou modèles UML (Rumbaugh et al. (2004)), actème (Vogel (1988)), . . .) pour préciser ce que nous définissons comme l'environnement d'exécution d'une action.

L'environnement proche d'une action est composé de 4 composantes dont la présence est facultative. Chaque composante peut être décrite par une liste de ce que nous nommons des « éléments de connaissance ». Ces éléments de connaissances peuvent être exprimés en langage naturel ou dans tout autre langage plus ou moins formel (code informatique, modèles ou outils UML par exemple). Cependant, seuls ceux exprimés sous la forme d'un identifiant (nom_concept, nom_d'attributs_concepts, nom_relation, ..) ou de termes associés à un élément de l'ontologie de concepts pourront être traités et exploités par le système ontologique. Ils permettront ainsi une première mise en relation exploitable entre les concepts et les actions décrivant notre domaine.

La première composante concerne les « actants » : qui est déclencheur de l'action, qui supporte l'action, ou encore celui au bénéfice ou au détriment duquel se fait l'action. Si nous regardons l'action choisir-cage, les actants sont *articulation_RSMM* qui déclenche l'action et *environnement* qui contribue par le dimensionnement souhaité.

La deuxième composante concerne le(s) « but(s) » ou résultats attendus de l'action et les concepts, attributs ou acteurs impactés. Dans l'exemple, choisir-cage fournira plusieurs éléments de la réponse attendue de cette action : cage ouverte ou fermée, et des indications sur les supports.

La troisième composante concerne « les instruments » nécessaires à la réalisation de l'action, les moyens dont l'action doit disposer pour se réaliser. Ici nous avons besoin de calculer la force résultante dans la cage choisie.

La quatrième composante concerne « les contraintes » qui limitent la réalisation de l'action et provoquent des impacts néfastes. Ici nous avons une contrainte temporelle (durée de fonctionnement de l'articulation sans arrêt), une contrainte de lieu (positions des supports) et une contrainte logique (lubrifiant si cage fermée). Nous pouvons remarquer que certains éléments de connaissance jouent différents rôles dans la description d'une action.

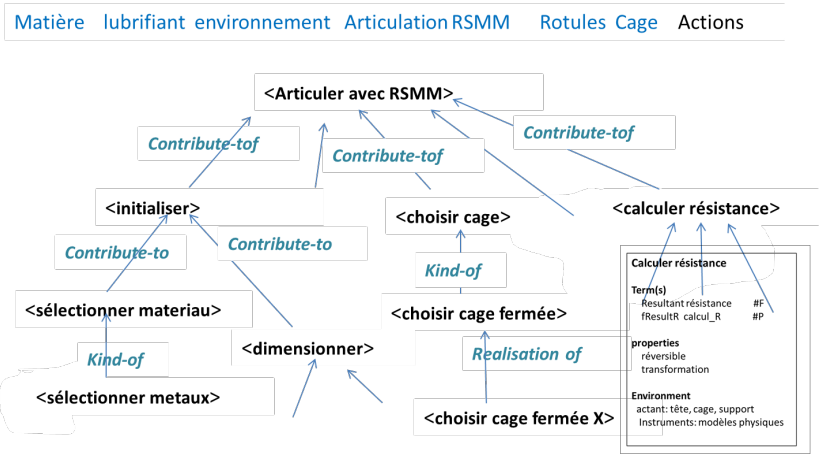


FIG. 3 – *Un sous-ensemble de l'ontologie des actions*

Une action est donc définie de façon unique dans l'ontologie par une expression verbale. Elle possède des propriétés (répétitive, unique, réversible, définitive, terminale, initiale) et un environnement (actants, but, instrument, contrainte). Elle peut entretenir des relations de subsomption 'est-une-façon-de', d'instance 'réalisation-de', partitive 'contribue-a' ou tout autre type de relations à définir selon les cas. Comme pour les concepts, nous associerons les terminologies aux éléments de cette ontologie des actions. L'action choisir-cage aura les termes ou expressions associés comme imposer-cage choisir-cageF/O, ChoiCage() auxquels on associera la langue #P (noms des fonctions ou méthodes dans le code), ou encore sélectionner-un-guide a laquelle on peut associer #F pour une expression en français.

5 Liens entre code et ontologies

Un des objectifs du projet est la mise en correspondance des connaissances métier avec des portions de code. Pour prendre en compte non seulement le « texte » des logiciels mais aussi leur mise en œuvre, les experts ont proposé des scénarios d'exécution reflétant les principales utilisations de ces programmes et pour suivre le déroulement du code, les fichiers sources sont instrumentés afin de fournir des traces exploitables. A partir du code instrumenté et des scénarii proposés, il est alors possible d'établir les corrélations qui existent entre les identifiants apparaissant dans les traces et les termes de l'ontologie, permettant d'une part d'identifier les concepts, d'autre part les actions conceptuelles mis en jeu. Par exemple, la figure 4 permet de montrer ces liens entre code et termes. Ces termes désignant des concepts ou des actions ontologiques, ils donnent accès également à l'environnement ontologique de ces éléments de connaissance, ce qui augmente d'autant leur sémantique. Nous pouvons ainsi retrouver tous les éléments du code qui « parlent » ou « utilise» un concept donné, avec sa position dans la structure ontologique.

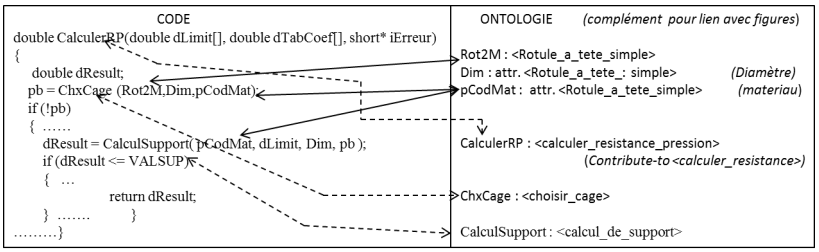


FIG. 4 – *Corrélations code – ontologie (concepts/ actions)*

A partir de cette identification, nous pouvons suivre, lors d'une exécution, la fréquence et la chronologie des apparitions d'un concept et étudier les interactions entre les concepts. Par exemple le suivi de deux concepts particuliers peut montre que ces concepts sont fortement liés dans certains programmes. Ces liens entre les concepts peuvent être des liens de l'ontologie des concepts (spécialisation par exemple) ou encore de contribution à une même activité ou action. Certains se lisent directement dans le code (signature des procédures/méthodes). D'autres sont plus difficiles à déceler, notamment dans les programmes plus anciens ou les termes utilisés sont des contractions incluant un typage de la variable. Lorsqu'il s'agit d'une contribution à une action, le rôle du concept peut alors être éclairé par la définition ontologique de cette action. Par exemple dans l'action <choisir_cage >, la cage est le but puisque le résultat de ce calcul doit nous apporter des éléments pour choisir une cage fermée ou une cage ouverte, nous aurons un attribut *pCodeMat* désignant le matériau (actant) à utiliser pour obtenir la rotule à tête simple (actant) souhaitée. Le diamètre est un élément imposé. Le concept *rotule_a_tête_simple* intervient dans deux rôles d'actant par chacun de ses deux attributs mis en jeu ici, l'un étant imposé, l'autre résultant des calculs de résistance (instrument) ainsi que de l'usage ou non de lubrifiant en fonction de la cage retenue.

La construction des ontologies nécessite un effort conséquent et la présence d'experts du domaine. Pour traiter environ 4000 lignes de code (dont 2000 en Fortran), nous avons identifié une soixantaine de concepts et une soixantaine d'actions. Cependant, la connaissance « retrouvée » lors de cette expérience a permis d'améliorer de façon conséquente la compréhension de ces codes. Même si cela ne suffit pas à expliquer tout le code, cela a permis à l'entreprise d'en (re-)commenter une partie de façon satisfaisante pour un meilleur transfert des connaissances et sans doute une maintenance facilitée. En effet, les impacts d'une modification ou d'une évolution du code seront plus aisément contrôlables, car sa sémantique sera plus accessible.

Ces résultats sont obtenus alors qu'aussi bien modélisation (de type UML par exemple) que documentation sont quasi inexistantes pour ces codes (Fortran et C++). L'effort de rétro-ingénierie de ces codes rebutait les informaticiens car ils restent sceptiques quant au pouvoir d'explication des outils de ce domaine, surtout sur les programmes en Fortran, langage qu'ils n'utilisent plus. L'approche ontologique les a motivés, car nouvelle pour eux. Cependant, il serait intéressant de comparer l'apport des ontologies et celui des modèles reconstruits avec les outils de rétro-ingénierie, surtout sur les programmes plus récents écrits en C++ pour lesquels les résultats d'une rétro modélisation UML devraient être probants et pour lesquels il existe maintenant une résistance moindre dans l'entreprise.

6 Conclusion

La construction des ontologies métier, dans un processus de retro-ingénierie de logiciel, permet d'une part de connaître et de comprendre les connaissances métier mises en jeu dans les programmes à exploiter et d'autre part de faciliter la capitalisation et le transfert de connaissances dans une entreprise. Ce dernier point revêt une grande importance dans le cas du départ des experts. L'association de l'ontologie métier et de la terminologie (associée aux langages de programmation) fournit un dictionnaire des identifiants du programme. Ces identifiants ont une définition donnée par l'élément ou les éléments (polysémie) de l'ontologie du domaine. De plus, on dispose de données contextuelles relatives à l'usage de ces identifiants par les références aux éléments de code qui les contiennent (traces, suivi des fréquences et chronologies d'apparition).

En facilitant le rapprochement entre les concepts métier de l'ontologie et les termes qui s'y réfèrent dans le code, l'ontoterminologie et les outils d'analyse de code permettent de rapprocher les connaissances métier des portions de codes étudiées. Cela en facilite la compréhension notamment en vue de la maintenance des patrimoines logiciel et industriel. Lorsque le code sera appelé à évoluer, l'ontoterminologie pourra et devra elle aussi être mise à jour. Dans le cadre de la rétro-ingiénerie des logiciels, les résultats obtenus sur les concepts ont été validés par les différents acteurs du projet.

Pour aller plus loin dans la compréhension des programmes, il serait sans doute pertinent d'une part d'utiliser les outils existants pour extraire des diagrammes de conception de type UML pour les concepts et BPMN pour les processus et d'étudier leur alignement sur les concepts et actions de nos ontologies.

Remerciements

Ces travaux s'appuient et prolongent les résultats obtenus dans le cadre du projet Interreg IV Franco-Suisse Ontoreverse (2013-2015), projet auquel ont participé notamment mes collègues Y. Dumond et C. Roche.

Références

Brooks, R. (1983). Towards a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies Vol. 18*, 543–554.

Eilam, E. (2005). *Reversing : Secrets of Reverse Engineering*. Wiley ed.

Erlikh, L. (2000). Leveraging legacy system dollars for ebusiness. *International Transaction Professional Vol. 2 no 3*, 17–23.

Harth, E. et P. Dugerdil (2015). Document retrieval metrics for program understanding. *In proceedings of the Seventh meeting of the Forum for Information Retrieval Evaluation, Gandhinagar, India, December*, 4–6.

ICPC, . *IEEE International Conference on Program Comprehension - http://www.program-comprehension.org/.*

Korshunova, E., M.Petrovic, G. Van Den Brand, et al. (2006). Cpp2xmi: reverse engineering of uml class, sequence, and activity diagrams from c++ source code. *In 13th Working Conference on Reverse Engineering, IEEE*, 297–298.

Marca, D.-A. et C. McGowan (1987). *SADT: structured analysis and design technique*. McGraw-Hill, Inc.

Muller, A.-H., S.-R. Tilley, et K. Wong (1993). Understanding software systems using reverse engineering technology perspectives from the rigi project. *In proceedings of the conference of the Centre for Advanced Studies on Collaborative research: software engineering. Toronto, Ontario, Canada, October*, 24–28.

O'Brien, M.-P. (2003). Software comprehension – a review research direction. Technical report ul-csis-03-3, Department of Computer Science Information Systems University of Limerick, Ireland.

Roche, C. (2007). Saying is not modelling. *In Proceedings of the 4th International Workshop on Natural Language Processing and Cognitive Science, International Conference on Enterprise Information Systems, Funchal, Portugal, June*, 12–16.

Roche, C. (2012). Ontoterminology: how to unify terminology and ontology into a single paradigm. *In proceedings of the 8th International Conference on Language Resources and Evaluation. Istanbul, Turkey, May*, 21–27.

Rumbaugh, J., I. Jacobson, et G. Booch (2004). *Unified modeling language reference manual*. the Pearson Higher Education.

Sagot, B. et D. Fiser (2008). Construction d'un wordnet libre du français à partir de ressources multilingues. *In TALN 2008-Traitement Automatique des Langues Naturelles*, 21–27.

Storey, M. (2006). Theories, tools and research methods in program comprehension: past, present and future. *Software Quality Journal, September Vol. 14, Issue 3*, 187–208.

Sutton, A. et J.-I. Maletic (2007). Recovering uml class models from c++: A detailed explanation. *Information and Software Technology vol. 49, no 3*, 212–229.

Vogel, C. (1988). *Génie cognitif*. Masson.

White, S.-A. (2008). *BPMN modeling and reference guide: understanding and using BPMN*. Future Strategies Inc.

Summary

Many of the knowledge and know-how of our companies are stored in the computer codes of their business software. But according to the creation dates of these, the enclosed knowledge is far from explicit, especially because their codification required serious abbreviations. This software can't be rewritten and must however be maintained to follow the evolution of the company. This requires those who manage these programs to enter their semantics. We offer help based on an ontological approach to better understand and use the knowledge and know-how contained in this software. Based on an industrial case, we will build an ontology of the domain and an ontology of the actions then we will highlight and exploit the links which can exist between the concepts of these ontologies and the portions of code which relate to the same knowledge.

Une nouvelle approche d'analyse non supervisée des données textuelles basée sur la combinaison du clustering, de la maximisation des traits et des graphes de contraste: application à l'analyse de l'évolution de sujets de recherche en Science de la Science en Chine durant 40 ans

Jean-Charles Lamirel*,**

*Equipe Synalp - LORIA, BP 239, 54506 VANDOEUVRE CEDEX, FRANCE
lamirel@loria.fr,
https://www.researchgate.net/profile/J-C_Lamirel
**Laboratoire WISELAB - DUT, DALIAN, CHINE

Résumé. L'analyse diachronique de corpus de données volumineux exploitant des approches non supervisées, si l'on s'attend à ce qu'elle fournisse des résultats suffisamment précis et fiables pour être exploités par des spécialistes des domaines concernés par les données traitées, reste encore un challenge très ouvert de nos jours. Les méthodes basées sur l'analyse latente de Dirichlet qui pourraient s'avérer de bonnes candidates dans ce contexte peinent cependant à fournir des résultats cohérents et s'avèrent de plus très sensibles au paramétrage, comme nous l'avons montré dans des travaux antérieurs. Nous montrons néanmoins dans ce papier qu'un tel type de challenge peut être relevé en exploitant une combinaison originale de méthodes : clustering neuronal basé sur les gaz de neurones croissants, métrique basée sur la maximisation des traits, développée récemment comme alternative aux métriques usuelles, et graphes de contraste dérivées de cette métrique.

1 Introduction

L'analyse diachronique de corpus de données volumineux exploitant des approches non supervisées, si l'on s'attend à ce qu'elle fournisse des résultats suffisamment précis et fiables pour être exploités par des spécialistes des domaines concernés par les données traitées, reste encore un challenge très ouvert de nos jours. Les méthodes basées sur l'analyse latente de Dirichlet (Blei et al. (2003)) qui pourraient s'avérer de bonnes candidates dans ce contexte peinent cependant à fournir des résultats cohérents et s'avèrent de plus très sensibles au paramétrage, comme nous l'avons montré dans des travaux antérieurs (Lamirel et al. (2015)). Nous montrons néanmoins dans ce papier qu'un tel type de challenge peut être relevé en exploitant une combinaison originale de méthodes : clustering neuronal basé sur les gaz de neurones croissants, métrique basée sur la maximisation des traits, développée récemment comme alternative aux métriques usuelles, et graphes de contraste dérivées de cette métrique.

Pour démontrer l'efficacité de notre approche, nous effectuons une analyse du contenu d'articles de revues académiques sélectionnées dans le domaine Science of Science in China au cours des 40 dernières années (autrement dit depuis la création reconnue du domaine en Chine) et construisons une carte globale de la structure des thèmes de recherche. De plus, nous mettons en évidence l'évolution des thèmes par l'exploitation des dates de publication et faisons un usage supplémentaire des informations liées aux auteurs afin de clarifier le rôle de ceux ci dans le domaine.

Dans les sections suivantes, nous présentons la méthode employée, puis nous décrivons les données utilisées et les résultats obtenus, puis, nous terminons par nos conclusions.

2 Maximisation des traits et exploitation dans le processus d'analyse

La maximisation des traits est une métrique sans biais qui peut être utilisée pour estimer la qualité d'une classification, qu'elle soit supervisée ou non supervisée. En classification non supervisée (i.e. clustering), cette mesure exploite les propriétés (i.e. les traits ou variables descriptives) des données associées aux clusters à différentes fins (étiquetage et mise en évidence du contenu des clusters, détection du modèle optimal de clustering, visualisation globale des résultats de clustering sous forme de graphe d'interaction tel que présenté dans ce travail, ...) Ses principaux avantages sont d'être sans paramètres, d'être totalement indépendante de la méthode de clustering et de son mode opératoire, de travailler convenablement dans des espaces fortement multidimensionnels et de représenter un meilleur compromis entre discrimination et généralisation que les métriques habituelles (Euclidienne, Cosinus, Chi2, etc.).

Considérons une partition C qui résulte d'une méthode de clustering appliquée à un ensemble de données D représenté par un groupe de traits F. La F-mesure de trait $FF_c(f)$ d'un trait f associé à un cluster c est définie comme la moyenne harmonique du rappel de traits $FR_c(f)$ et de la prédominance de traits $FP_c(f)$, qui sont elles-mêmes définies comme :

$$FR_c(f) = \frac{\Sigma_{d \in c} W_d^f}{\Sigma_{c \in C} \Sigma_{d \in c} W_d^f} \tag{1}$$

$$FP_c(f) = \frac{\Sigma_{d \in c} W_d^f}{\Sigma_{f' \in F_c, d \in c} W_d^{f'}} \tag{2}$$

où W_d^f représente le poids du trait f pour la donnée d et F_c représente l'ensemble des traits présents dans les données associées au cluster c. La prédominance de trait mesure la capacité du trait f à décrire le cluster c. De façon complémentaire, le rappel de trait permet de caractériser f selon sa capacité à distinguer c des autres clusters.

Dans un contexte supervisé, la métrique de maximisation des traits peut être exploitée pour générer un puissant processus de sélection de variables (Lamirel et al. (2015)). Dans un contexte non supervisé, le processus de sélection de variables peut être utilisé pour décrire ou étiqueter les clusters selon les caractéristiques les plus typiques et les plus représentatives. Il

s'agit d'un processus non paramétré qui utilise à la fois la capacité de la métrique à discriminer entre les clusters (indice $FR_c(f)$) et sa capacité à représenter fidèlement les données du cluster (indice $FP_c(f)$).

L'ensemble S_c des traits qui sont caractéristiques d'un cluster c issu de la partition C défini par :

$$S_c = \left\{ f \in F_c \mid FF_c(f) > \overline{FF}(f) \text{ and } FF_c(f) > \overline{FF}_D \right\} \quad (3)$$

où

$$\overline{FF}(f) = \Sigma_{c' \in C} \frac{FF_{c'(f)}}{|C_{/f}|} \text{ et } \overline{FF}_D = \Sigma_{f \in F} \frac{\overline{FF}(f)}{|F|} \quad (4)$$

et $C_{/f}$ représente le sous-ensemble de C dans lequel le trait f apparait.

Un concept spécifique de contraste $G_c(f)$ peut être défini pour calculer la performance d'un trait f retenu comme caractéristique d'un cluster donné c. C'est une valeur d'indicateur qui est proportionnelle au rapport entre la F-mesure du trait $FF_c(f)$ du trait f pour le cluster c et la F-mesure moyenne de trait $\overline{FF}$ de f pour la partition entière. Ainsi, le contraste $G_c(f)$ d'un trait f pour un cluster c est exprimé comme :

$$G_c(f) = FF_c(f)/\overline{FF}(f). \quad (5)$$

Les traits actifs d'un cluster sont ceux pour lesquels le contraste est supérieur à 1. De plus, plus le contraste d'un trait est élevé pour un cluster, meilleures sont ses capacités à décrire le contenu du cluster. Une façon simple d'exploiter les traits est d'utiliser ceux dont le constraste est supérieur à 1 (traits actifs) dans les clusters pour étiqueter ces derniers comme nous l'avons proposé dans Lamirel et al. (2015) et comme nous l'exploitons également dans cette approche.

Dans le domaine de la théorie des graphes, un graphe bipartite (ou bigraphe) est un graphe dont les sommets peuvent être divisés en deux ensembles disjoints et indépendants U et V de sorte que chaque arête relie un sommet de U à un sommet de V. Les graphes de contraste peuvent être considérés comme des graphes bipartites basés sur les relations entre un ensemble de traits S et un ensemble de labels L (Cuxac et Lamirel (2013)). Théoriquement, l'ensemble de labels L peut représenter n'importe quel type d'information à laquelle les traits peuvent être associés, notamment des catégories ou des clusters, et l'ensemble de traits S peut lui-même représenter un sous-ensemble d'un ensemble global de traits F qui a été obtenu par un processus de sélection de variables, comme celui basé sur la maximisation des traits présentée ci-avant. Dans le cas de l'utilisation de la maximisation des traits, le poids $c_{(u,v)}$ d'une arête $(u, v), u \in S, v \in S$ représente alors le contraste du trait u pour une étiquette v tel qu'il est défini par l'équation 5.

La gestion des résultats de clustering avec une approche basée sur un graphe de contraste permet à la fois de réduire la surcharge cognitive qui devrait résulter de la représentation des interactions entre clusters à partir de l'exploitation directe de leurs caractéristiques dans l'espace de description des données, en général fortement multidimensionnel, et de déterminer avec précision les dépendances entre les sujets extraits par le processus de clustering grâce à

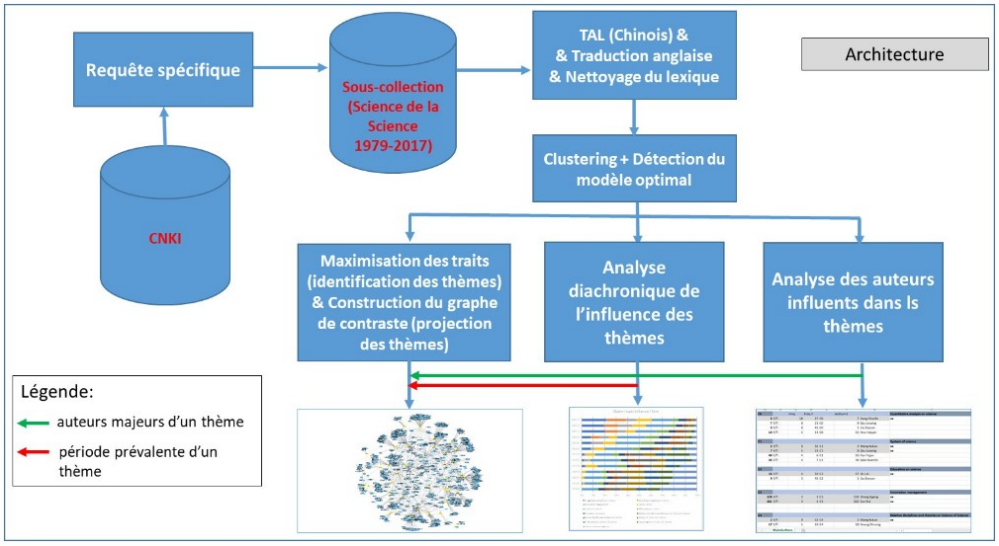


FIG. 1 – *Vue générale du processus expérimental.*

des caractéristiques (traits) communes à contraste élevé. Nous illustrons clairement cela dans l'expérience suivante.

3 Protocole expérimental

Pour initier notre processus expérimental, nous avons d'abord effectué une interrogation de la base de données de la China National Knowledge Infrastructure (CNKI) en utilisant "Science of Science" comme terme thématique. Ainsi, 2401 articles publiés dans les revues principales de l'Université de Pékin et les revues du CSSCI [1] (couvrant une période de recherche allant jusqu'au 2017-10-22) ont été extraits. Dans une deuxième phase, les titres, résumés et mots-clés des 2790 articles ont été extraits. Pour la segmentation des mots et le balisage des titres et des résumés des articles, nous avons utilisé NLPIR-ICTCLAS [2], une boîte à outils spécifique pour le traitement de la langue chinoise. La traduction anglaise a ensuite été appliquée au dictionnaire de mots chinois obtenu. A titre de post-traitement, le seuillage de fréquence a été opéré sur les documents réindexés avec le lexique traduit en anglais pour supprimer les mots de basse fréquence. Il en est résulté un dictionnaire final de 1576 termes avec lesquels les articles on été réindexés. Aucun document n'a finalement été supprimé par ce processus.

A l'étape suivante, nous avons exploité le clustering en combinaison avec la maximisation des traits pour extraire les principaux sujets de recherche de l'ensemble des données extraites et réindexées. Nous avons récemment montré (Lamirel et al. (2015)) que la combinaison

1. Chinese Social Sciences Citation Index
2. http ://ictclas.nlpir.org/

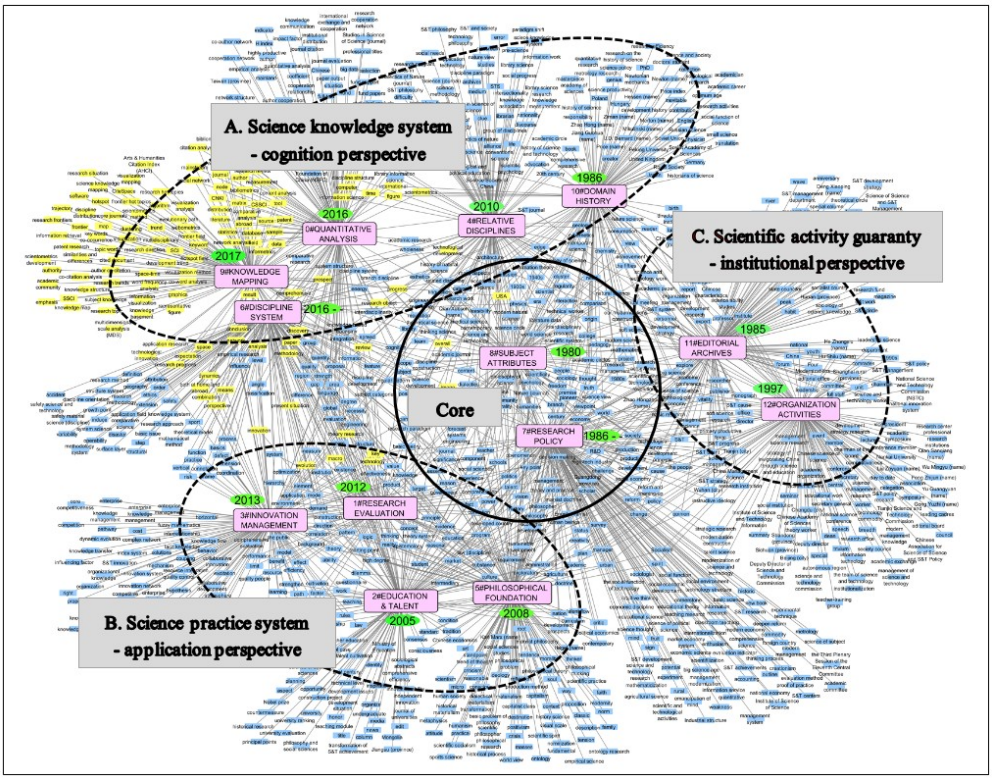


FIG. 2 – *Graphe de contraste global représentant les principaux thèmes de recherche et la structure du domaine de la Science de la Science en Chine. (Le thème récent du « Knowledge Mapping » (Cartographie des connaissances) est présenté en surbrillance).*

d'une approche de clustering appropriée, comme le clustering neuronal (Fritske (1995)), avec la maximisation des traits offre des performances supérieures à celles des approches alternatives pour l'extraction de sujets, comme la LDA (Blei et al. (2003)), à la condition de pouvoir identifier correctement un modèle de clustering optimal (c'est-à-dire un nombre approprié de clusters) à partir des données analysées. Nous nous sommes donc proposé d'exploiter une de nos approches récentes et efficace également basée sur la maximisation des traits pour la détection du modèle de clustering optimal (Lamirel et al. (2016)).

Selon cette dernière méthode, le modèle optimal que nous avons retenu a été un modèle à 13 clusters. Nous avons donc construit un graphe de contraste à partir de ce modèle par l'exploitation des traits obtenus sur celui-ci en appliquant le processus de maximisation des traits et le calcul du contraste, comme décrits à la section 2.

La dernière partie de notre approche a consisté à exploiter les étiquettes externes [3] des données associées à des clusters. Premièrement, la date de publication est utilisée pour effectuer une analyse diachronique de l'activité des clusters (i.e des thèmes de recherche identifiés) et deuxièmement, l'information sur les auteurs a été utilisée pour souligner les auteurs les plus influents dans les différents thèmes. Les informations relatives à la date et à l'auteur sont également reportées sur le graphe de contraste. La figure 1 résume l'ensemble du processus. La distribution spatiale de 13 sujets dans le graphe de contraste à retour de force obtenu est présenté à la figure 2.

Selon l'avis des experts du domaine, ce graphe met en évidence la structure très clairement interprétable du domaine de la Science de la Science en Chine. Dans un tel modèle, les sujets natifs fortement inter-connectés ont tendance à apparaître au centre de la représentation. De fait, autour du domaine cœur (Core) qui à servi de base de construction de la recherche en Science de la Science en Chine, apparaissent 3 grands domaines périphériques qui sont cohérents avec les activités scientifiques observées, à savoir, « A. Science knowledge system » (Système de connaissances scientifiques), « B. Science practice system » (Système de pratique scientifique), et, « C. Scientific activity guaranty » (Système de garantie de l'activité scientifique). Ces trois derniers domaines forment la structure logique complète de la Science of Science en Chine du point de vue de la connaissance, des applications et du fonctionnement institutionnel, respectivement.

Les dates prépondérantes associées aux clusters du graphe (qui représentent leur période d'activité majoritaire calculée à partir des dates des publications qui leur sont associées) permettent de mettre en évidence le chemin d'évolution des thèmes.

Une représentation globale de l'influence de chaque cluster (i.e. thème) dans les différentes périodes (en utilisant ici des blocs de 3 ans) peut être également dérivée de la distribution des années dans les publications associées à chaque cluster. Cette représentation présentée à la figure 3 peut ensuite être utilisée pour mieux comprendre les lois de développement de la Science de la Science en Chine. Ce point de vue peut surtout aider à distinguer entre les « sujets chauds accidentels » qui ont une certaine chance de se développer à court terme et les « sujets chauds rationnels » qui jouent un rôle majeur à long terme dans la construction et le développement du domaine. Une nouvelle fois, toujours selon les experts, cette représentation se coordonne parfaitement avec le développement historique du domaine. Un exemple est donné par les thèmes "Analyse quantitative de la science", "Cartographie des connaissances scientifiques" et "Gestion de l'innovation" qui ne sont pas apparus au début de la recherche scientifique en Science de la Science en Chine, mais ont seulement émergés ces dernières années, en devenant cependant de plus en plus prépondérants dans ce domaine.

4 Conclusion

Nous montrons expérimentalement dans cet article que la combinaison de la méthode de maximisation des traits, de la représentation par graphe de contraste associée, et, de l'appren-

3. Les étiquettes externes représentent des traits, ou caractéristiques, qui ne sont pas exploités lors du processus de clustering.

FIG. 3 – *Evolution de l'influence des thèmes de recherche en Science de la Science en Chine au cours du temps.*

tissage neuronal non supervisé peut révéler efficacement les thèmes de recherche ainsi que leur évolution dans un domaine de recherche donné. Il est également facile de montrer qu'il est plus adapté à l'analyse de données à grande échelle dans un contexte de représentation fortement multidimensionnel des données que les méthodes de regroupement usuelles basées sur la co citation ou la concurrence de mots-matières, ou encore que les méthodes d'extraction de sujets usuelles, comme la méthode LDA. Dans l'analyse des changements dans les thèmes de la science chinoise de la science, cette méthode reflète objectivement le processus du changement du domaine, et, ce changement est également conforme à la loi du développement de la science chinoise.

Références

Blei, D., A. Ng, et M. Jordan (2003). Latent dirichlet allocation. *Journal of Machine Learning Research 3*, 993–1022.

Cuxac, P. et J.-C. Lamirel (2013). Analysis of evolutions and interactions between science fields: the cooperation between feature selection and graph representation. *14th COLLNET Meeting*.

Fritske, B. (1995). A growing neural gas network learns topologies. *Advances in neural information processing systems*, 625–632.

Lamirel, J.-C., N. Dugué, et P. Cuxac (2015). Performing and visualizing temporal analysis of large text data issued for open sources: past and future methods. *Beyond Databases, Architectures and Structures. Advanced Technologies for Data Mining and Knowledge Discovery*, 56–76.

Lamirel, J.-C., N. Dugué, et P. Cuxac (2016). New efficient clustering quality indexes. *International Joint Conference on Neural Networks*, 3649–3657.

Summary

Diachronic analysis of large data corpuses using unsupervised approaches, while expected to provide results that are sufficiently accurate and reliable to be used by specialists in the fields concerned by the processed data, remains a very open challenge today. The methods based on Dirichlet's latent analysis, which could prove to be good candidates in this context, however, struggle to provide consistent results and are also very sensitive to parameter settings, as we have shown in previous work. Nevertheless, we show in this paper that such a challenge can be met by exploiting an original combination of methods: neural clustering based on growing neural gas (GNG), specific metric based on feature maximization, recently developed as an alternative to the usual metrics, and contrast graphs derived from this metric.

Index