EvalECD'09

Évaluation des méthodes d'Extraction de Connaissances dans les Données

EGC

Organisateurs :

Fatiha Saïs (LRI, INRIA-Saclay, Université Paris-Sud – CNRS UMR8623)
Nicolas Béchet, Mathieu Roche (LIRMM, Université Montpellier 2 – CNRS UMR5506)

INRIA

LIRMM

UM2 UNIVERSITÉ MONTPELLIER 2

Atelier EvalECD'09 : Évaluation des méthodes d'Extraction de Connaissances dans les Données

Organisateurs : Fatiha Saïs[1], Nicolas Béchet[2], Mathieu Roche[2]

[1]LRI, INRIA-Saclay, Université Paris-Sud – CNRS UMR8623
[2]LIRMM, Université Montpellier 2 – CNRS UMR5506

Présentation

De plus en plus de méthodes sont proposées pour évaluer les approches dans divers domaines, que ce soit, en Fouille de Données, en Intégration de Données, en Ingénierie des Connaissances ou encore en Traitement Automatique des Langues. De manière générale, le but des méthodes d'évaluation est d'identifier la (ou les) méthode(s) les plus appropriée(s) à un problème donné. Ces évaluations consistent par exemple à comparer les différentes méthodes, en terme d'effort humain nécessaire que ce soit en amont (préparation des données) ou en aval (validation manuelle des résultats), en termes de pertinence, de temps d'exécution, de résistance au bruit ou encore de robustesse aux changements de données et/ou de domaine d'application. Notons que de nombreuses méthodes d'évaluation ont des similitudes (mesures de précision, rappel, F-Mesure) mais peuvent se révéler spécifiques aux domaines et/ou aux données traitées.

Si l'on se focalise sur le cas particulier de la fouille de données la pertinence et la cohérence des méthodes de validation actuelles peuvent être discutées. Nous noterons principalement (1) la validation humaine (validation confirmant (ou infirmant) des hypothèses fournies par un (ou des) expert(s)) et (2) la validation automatique (confrontation du modèle à des modèles de référence), qui ont leurs limites.

(1) pose le problème de la subjectivité de ce type de validation. En effet, des experts d'un même domaine auront des avis subjectifs et à fortiori, n'auront que très rarement des avis similaires à un problème donné. (2) pose le problème du choix des modèles de référence qui ne sont pas toujours adaptés à une approche. De plus, pour certaines tâches spécifiques, aucun modèle de référence n'existe et seule la validation humaine permet de mesurer la qualité d'un modèle. Par ailleurs, certains domaines d'application possèdent peu de données expertisées (par exemple dans le domaine biomédical), ce qui peut engendrer des difficultés majeures dans la phase d'évaluation. Enfin, les résultats sont souvent sensibles aux différents paramètres utilisés qui doivent être rigoureusement pris en compte dans les différents modèles utilisés.

L'objectif de cet atelier est de discuter des techniques d'évaluation utilisées dans différents domaines et de montrer leurs qualités et leurs limites. Un tel atelier ouvert aux différentes communautés d'EGC permet de compléter et/ou généraliser les travaux présentés dans des ateliers très spécifiques (par exemple, ateliers de TIA'05, COLING'08 en Traitement Automatique des Langues).

Les quatre articles présentés dans le cadre de cet atelier de la conférence EGC'09 permettent de proposer un début de solutions aux problèmes occasionnés par la diversité des évaluations en ECD. Les trois premiers articles présentés traitent la problématique de l'évaluation des données textuelles dans un processus de fouille de textes : évaluation des outils d'acquisition de la terminologie (H. Zargayounda et A. Nazarenko), jugement humain et segmentation thématique (Y. Benstgen), évaluation automatique de la pertinence de relations syntaxiques par l'utilisation de corpus (N. Béchet). Le dernier article fait quant à lui référence à des mesures d'évaluation de motifs séquentiels (H. Saneifar *et al.*). L'atelier clôturera sur le domaine de recherche de la SATisfiabilité de formules booléennes en IA grâce à une présentation sur le retour d'expérience des compétitions SAT qui sera donnée par L. Simon.

Notons que les travaux sur l'évaluation des méthodes d'ECD possèdent des liens avec les études sur la Qualité des Données (métriques de qualité des données, validation de modèles de fouille de données, etc). Ainsi, naturellement, notre atelier s'est associé à l'atelier QDC'09 (Qualité des Données et des Connaissances) organisé par Jérôme Azé et Sylvie Guillaume dans le cadre d'EGC'09.

Thèmes (liste non limitative)

- Protocoles d'évaluation (automatiques, semi-automatiques, manuels)
- Mesures d'évaluation (Précision, Rappel, F-Mesure, Courbes ROC, ...)
- Tuning : spécifications des paramètres
- Construction de benchmarks
- Validation de connaissances en ECD
- Expertise humaine en TAL et en IC
- Evaluation en Intégration de Données et en Réconciliation de Données
- Evaluation des outils d'IHM (ergonomie, visualisation)

Comité de Programme

- Bernd Amann (LIP6)
- Jean-Yves Antoine (LI)
- Jérôme Azé (LRI)
- Zohra Bellahsène (LIRMM)
- Catherine Berrut (LIG)
- Patrice Buche (INRA - Unité Mét@risk)
- Bich-Liên Doan (Supélec)
- Stéphane Lallich (ERIC)
- Anne Laurent (LIRMM)
- Nathalie Pernelle (LRI, INRIA Saclay)
- Pascal Poncelet (LIRMM)
- Chantal Reynaud (LRI, INRIA Saclay)
- Christophe Roche (LISTIC)
- Bernard Rothenburger (IRIT)
- Laurent Simon (LRI)
- Maguelonne Teisseire (LIRMM)
- Alexandre Termier (LIG)
- Fabien Torre (LIFL)
- Juan Manuel Torres-Moreno (LIA)

Peut-on évaluer les outils d'acquisition de connaissances à partir de textes ?

Haïfa Zargayouna, Adeline Nazarenko

LIPN, Université Paris 13 - CNRS UMR 7030,
99 av. J.B. Clément, 93440 Villetaneuse, France.
prenom.nom@lipn.univ-paris13.fr

Résumé. Malgré les années de recul et d'expériences accumulées, il est difficile de se faire une idée claire de l'état d'avancement des recherches en acquisition de connaissances à partir de textes. Le manque de protocoles d'évaluation ne facilite pas la comparaison des résultats. Nous développons, dans cet article, la question de l'évaluation des outils d'acquisition de terminologies et d'ontologies en soulignant les principales difficultés et en décrivant nos premières propositions dans ce domaine.

1 Introduction

Malgré les années de recul et d'expériences accumulées, il est difficile de se faire une idée claire de l'état d'avancement des recherches en acquisition de connaissances. Ces connaissances sont généralement utilisées comme ressources sémantiques pour des applications et le fait que ces travaux soient très orientés vers des applications ne facilite pas la comparaison des résultats.

Nous mettons l'accent sur deux types de ressources : les terminologies et les ontologies. Même si les premières sont souvent utilisées pour l'acquisition des secondes, nous considérons ici les deux problèmes d'acquisition séparément. Nous considérons que du point de vue de l'évaluation, il y a une parenté entre les deux tâches et qu'il est intéressant de faire le parallèle. Les difficultés sont les mêmes en effet. A ce jour, les outils d'acquisition terminologique et ontologique ont souvent été développés en fonction d'une application particulière sans qu'on sache apprécier leur portabilité à d'autres corpus (genre et taille), à d'autres domaines scientifiques ou techniques, et à d'autres tâches. On ne sait pas évaluer les outils terminologiques et ontologiques parce qu'on a du mal à définir ce que devraient être leurs résultats (qu'est-ce qu'une "bonne" terminologie ? sur quel critère comparer deux ontologies ?). Nous faisons l'hypothèse que les deux questions peuvent s'enrichir l'une l'autre : les solutions terminologiques peuvent aider à résoudre le problème ontologique et vice-versa[1].

Les deux premières parties de cet article présentent un état des lieux de l'évaluation et souligne les difficultés de l'évaluation des outils d'acquisition terminologique et ontologique. La troisième partie présente les pistes que nous suivons pour surmonter ces difficultés. Le volet terminologique du travail est plus avancé mais nous montrons qu'il éclaire le problème ontologique.

2 Premières expériences

Même si de nombreuses recherches ont porté sur l'acquisition de connaissances à partir de textes, il y a eu peu d'effort collectif pour définir un cadre d'évaluation adapté à ce type de travaux, à la différence de ce qui est fait dans d'autres sous-domaines du traitement automatique des langues. Nous présentons dans ce qui suit quelques expériences d'évaluation dans le cadre de campagnes d'évaluation ou plus ponctuellement au travers des applications.

2.1 Les campagnes d'évaluation

Les campagnes d'évaluation visent généralement à évaluer un ensemble de systèmes, sur une tâche clairement spécifiée et à partir d'un jeu de données commun, en classant les résultats obtenus par les différents systèmes ou en les comparant à une référence. Au final, on obtient les mesures de performances des systèmes et leur classement pour la tâche considérée.

[1] Nous nous intéressons à ces questions dans le cadre du programme Quaero (http ://www.quaero.org/modules/movie/scenes/home) qui considère l'évaluation de la contruction de terminologies et de l'acquisition d'ontologies comme deux tâches distinctes. Nous présentons dans cet article, les débuts de nos travaux effectués dans ce cadre.

2.1.1 Les campagnes d'évaluation des outils d'acquisition terminologique

L'acquisition de terminologie a déjà fait l'objet de campagnes d'évaluation qui ont permis de débroussailler les questions liées à l'évaluation des outils d'acquisition.

CESART c'est probablement la plus aboutie des campagnes dans le domaine de la terminologie computationnelle. Elle devait comporter initialement trois sous-tâches : l'extraction terminologique dans le but de construire des terminologies, l'indexation contrôlée et l'extraction de relations sémantiques entre termes (Mustafa el Hadi et al., 2006). Malheureusement CESART a réuni peu de participants avec quatre systèmes participant à la première tâche, aucun à la deuxième et seulement un pour la troisième. C'est le protocole élaboré pour la première tâche qui est le plus intéressant. Les résultats des extracteurs de termes sont évalués par comparaison avec une liste de termes établie par avance par des experts (en pratique dans CESART, ce sont des terminologies préexistantes qui ont été utilisées). Un corpus d'acquisition relevant du même domaine que la terminologie de référence est fourni aux systèmes comme données d'entrée, et ceux-ci doivent en extraire une liste de termes. Cette liste de termes candidats est comparée à la liste de référence. L'une des originalités de CESART a été de considérer la pertinence d'un candidat terme sur une échelle à cinq valeurs plutôt que comme une valeur booléenne. Un terme est considéré comme pertinent s'il apparaît tel que dans la terminologie de référence mais aussi, à un moindre degré, s'il est composé de mots qui relève du vocabulaire de la terminologie de référence. Pour chaque degré de pertinence, une mesure de précision spécifique est calculée. Cette campagne a également permis de mettre en évidence l'hétérogénéité des résultats fournis par les différents systèmes, du point de vue de la longueur des listes de candidats termes fournies. Les extracteurs ont été évalués sur des échantillons de 10 000 termes mais certains ont fourni 20 fois plus de candidats.

CoRRecT Bien que ce ne soit pas une campagne à proprement parler, CoRReCT propose un jeu de test et un protocole intéressants (Enguehard, 2003). L'objectif est d'évaluer la tâche de reconnaissance de termes en corpus, qui se rapproche de l'indexation contrôlée de documents. Les systèmes prennent en entrée un corpus et une terminologie relevant du même domaine et ils doivent indexer le corpus avec les termes de la liste fournie. Comme les termes de la références n'apparaissent pas toujours sous la même forme dans le corpus, les systèmes doivent souvent reconnaître les termes sous des formes variantes. CoRRecT a la particularité de reposer sur une construction incrémentale du corpus annoté de référence. Au départ le corpus de référence est le corpus brut qui est fourni aux participants mais il s'enrichit d'annotations de référence chaque fois qu'un nouveau système participe à l'évaluation. Lorsqu'un nouveau système soumet sa liste d'annotations, celles-ci confrontées avec l'état courant du corpus de référence. Les nouvelles propositions d'annotation sont évaluées manuellement et celles qui sont validées viennent enrichir le corpus de référence. Le corpus de référence comporte donc l'union des propositions d'annotations des différents systèmes une fois celles-ci validées. Les résultats du i^{eme} système sont comparés (en termes de précision et de rappel) avec l'union des résultats validés des i premiers systèmes. Malheureusement, là encore, seulement trois systèmes ont participé à CoRRecT.

NTCIR-TERMREC En 1999, La campagne japonaise d'évaluation en recherche d'information (NTCIR) a intégré une tâche d'acquisition terminologique qui se décomposait elle-même en trois sous-tâches distinctes : l'extraction de termes, l'extraction de mots-clefs et l'analyse de rôles (Kageura et al., 2000). Les résultats étaient évalués sur la base d'une comparaison avec une référence. Peu d'information est disponible sur cette expérience, mais l'organisateur lui-même s'est déclaré déçu par l'interprétation essentiellement quantitative qui a été faite des résultats. A noter également que les tâches terminologiques n'ont pas été reprises dans les éditions ultérieures de NTCIR.

2.1.2 Les campagnes d'évaluation des outils de construction d'ontologies

Avec l'essor des recherches visant à promouvoir le web sémantique, une réflexion sur l'évaluation des ontologies a également vu le jour, l'évaluation systématique des outils devant permettre d'atteindre un niveau homogène de qualité et de faciliter l'adoption des technologies du web sémantique par le monde industriel. L'atelier EON (Evaluation of Ontologies for the Web) a mis l'accent sur les langages du web sémantique proposés par le W3C et a cherché à mettre en place des procédures d'évaluation des ontologies adaptées aux besoins du Web sémantique et à la problématique d'hétérogénéité, d'évolutivité et d'incomplétude du web.

EON2002[2], associé à la conférence EKAW (European Conference on Knowledge Acquisition and Management), s'est interrogé sur la manière dont il convient d'évaluer les technologies liées aux ontologies et les environnement d'ingénierie des ontologies. Le protocole d'évaluation proposé comporte deux aspects principaux (Maynard et al., 2006) : (i) l'évaluation des caractéristiques syntaxiques et sémantiques de l'ontologie produite, (ii) l'évaluation technologique (passage à

[2]http ://km.aifb.uni-karlsruhe.de/ws/eon2002

l'échelle, allocation mémoire, interopérabilité, etc.). La compétition a réuni une douzaine de participants (dont le LIPN avec Terminae) mais il n'y a pas eu d'évaluation globale des systèmes en compétition, chaque participant présentant les limites et les atouts de son système.

Les ateliers suivants ont proposé une série d'expériences pour évaluer les outils de fusion et d'alignement d'ontologies (EON2004[3]), les outils d'annotation d'ontologies et les technologies de web services sémantiques (EON2008[4]). EON2006 a le mérite d'avoir cherché à comparer les approches d'évaluation sur 4 ontologies différentes. Il a débouché sur des propositions de méthodes et de métriques.

Au total, même s'il n'existe encore aucune méthode d'évaluation globale et intégrée, les ateliers EON ont mis l'accent sur l'importance et la difficulté de l'évaluation des ontologies et des outils d'ingénierie ontologique et ils ont permis de faire émerger des propositions (Brank, 2006; Sabou et al., 2006; Obrst et al., 2006).

2.2 Evaluation au travers des applications

Au-delà des tentatives d'évaluation centrées sur les outils d'acquisition, on a également cherché à évaluer l'apport de leurs résultats dans diverses applications. Cette approche de l'évaluation est d'autant plus importante qu'elle permet de mettre en évidence l'intérêt des terminologies ou des ontologies dans des applications diverses.

Terminologie pour l'analyse syntaxique Les résultats de l'acquisition de termes peuvent être exploités pour améliorer la qualité de l'analyse syntaxique. Elle permet notamment de réduire les ambiguïtés de rattachements prépositionnels fréquents dans les corpus spécialisés. Cette idée initialement introduite par Bourigault (1993) a été reprise et testée par Aubin et al. (2005) pour adapter le Link Grammar Parser (LGP) (Grinberg et al., 1995) au domaine de la biologie. Les premiers résultats ont montré que la connaissance des termes diminue considérablement le temps d'analyse du LGP qui reflète la complexité de la phrase. Le nombre d'analyses produites par phrase diminue de manière très significative avec l'introduction des termes qui permettent de simplifier la phrase et le nombre d'analyses erronées baisse de 40% environ. Même si ces résultats montrent l'intérêt de l'analyse terminologique pour l'analyse syntaxique, l'évaluation réelle de la terminologie au travers de l'analyse syntaxique est difficile à conduire : les résultats dépendent de l'analyseur utilisé et de ses caractéristiques, du sous-langage considéré dans lequel les termes peuvent avoir plus ou moins d'importance et il faudrait comparer l'apport de différentes terminologies pour avoir une idée un peu précise de la qualité de la liste des termes exploitée.

Terminologie dans les index de fin de livre Les outils d'analyse terminologique étant exploités pour construire des index de fin de livre, leur apport et leur qualité ont pu être évalués dans ce contexte. IndDoc est par exemple un outil d'aide à l'indexation qui repose sur des outils d'analyse terminologique. Il permet de construire des ébauches d'index à partir desquels les indexeurs peuvent travailler pour produire des index finaux. Dans le cas d'un outil interactif, l'évaluation vise à apprécier la charge de travail qui reste à la charge de l'indexeur qui doit retravailler l'ébauche d'index produite automatiquement. Les expériences d'évaluation présentées dans (Ait El Mekki et Nazarenko, 2006) mettent clairement en valeur l'apport de l'analyse terminologique (extraction de termes, tri des termes par ordre de pertinence, hiérarchisation de la terminologie produite) mais, là non plus, elles ne constituent pas des expériences complètes d'évaluation des outils d'analyse terminologique. Il faudrait pour cela comparer les résultats produits avec différents outils d'analyse terminologique, ce qui représente un travail d'intégration et de validation non négligeable.

Terminologie et traduction automatique Dans les domaines spécialisés, la traduction s'appuie depuis longtemps sur des ressources terminologiques bilingues que l'on cherche aujourd'hui à intégrer dans les systèmes de traduction automatique. Lors du projet CESTA sur l'évaluation des systèmes de traduction automatique, une tâche originale a été proposée aux participants en leur permettant d'adapter leur système au domaine spécifique de la santé (Hamon et al., 2007). La comparaison des résultats obtenus avant et après cet enrichissement terminologique met en lumière l'apport de la terminologie sur les systèmes de traduction automatique[5] : sur cinq systèmes évalués, trois ont obtenus des résultats légèrement meilleurs et deux autres ont nettement amélioré leurs performances. (Langlais et Carl, 2004) présente une expérience d'évaluation intéressante dans ce contexte. Les auteurs cherchent à mesurer l'apport d'une terminologie bilingue dans l'adaptation d'un système de traduction automatique générique. Ce système repose sur une approche statistique et les auteurs montrent comment la terminologie peut être prise en compte dans le modèle de langage : elle est utilisée comme un faisceau de contraintes pour élaguer l'espace des traductions possibles. Même si le bilan est fortement dépendant de

[3]Qui a donné l'amorce de la campagne OAEI (Ontology Alignment Evaluation Initiative).

[4]http ://sws-challenge.org/wiki/index.php/EON-SWSC2008

[5]Ceci en dépit d'un protocole d'évaluation difficile : la campagne a été réalisée sur une durée relativement courte et sur un corpus d'adaptation de faible volume, environ 20 000 mots.

la manière dont la terminologie est prise en compte et du système de traduction automatique dans laquelle elle est exploitée, le bilan est globalement positif : les auteurs font état d'un accueil favorable des traducteurs et d'une amélioration significative du NIST[6]. Ils montrent également que cet impact dépend assez fortement de la couverture de la ressource terminologique.

Ontologie et recherche d'information La prise en compte de l'ontologie dans un système de recherche d'information peut intervenir essentiellement à deux niveaux : à l'étape d'indexation des documents et des requêtes et au moment de recherche elle-même, c'est-à-dire, lors de l'appariement requêtes-documents (Zargayouna, 2005). Les ontologies utiles en recherche d'information sont des ontologies lexicales ou temino-ontologies. La majorité des travaux utilisent WordNet en utilisant les liens lexicaux (des synonymes) et conceptuels (hiérarchiques ou méronymiques) (Baziz et al., 2003). Même si on voit bien en quoi les ontologies peuvent être utiles pour l'expansion ou le raffinement sémantique des requêtes, l'apport des ontologies dans les systèmes de recherche d'information reste difficile à déterminer. Les expérimentations effectuées sont généralement difficilement reproductibles, ce qui rend plus difficiles les évaluations comparatives. Selon les cas, les résultats sont liés à un domaine spécialisé et donc peu généralisables ou ils restent de portée limitée.

2.3 Discussion

La problématique de l'évaluation n'a pas le même poids dans les communauté d'acquisition terminologique et ontologique. Les campagnes d'évaluation en terminologie ont souffert du nombre restreint de participants, ce qui s'explique sans doute en partie par un déficit d'intérêt pour l'évaluation. Du côté de l'acquisition d'ontologie, la problématique de l'évaluation est reconnue comme importante en revanche. Les ateliers EON ont rassemblé des chercheurs autour de cette problématique et abouti à la proposition de protocoles et de métriques, ce qui constitue un point de départ intéressant.

Dans les deux cas, il reste difficile de faire émerger un cadre fédérateur. Le faible succès des campagnes en terminologie s'explique aussi par le fait que les systèmes qui avaient été conçus dans des perspectives différentes ont eu du mal à "entrer dans le cadre" de l'évaluation. L'atelier EON n'a pas permis de faire émerger un protocole d'évaluation global. Les travaux autour de l'évaluation des ontologies contournent souvent cette difficulté en proposant d'évaluer les ontologies par des critères intrinsèques, le plus souvent formels (pour vérifier par exemple la cohérence ou la consistence), sans que la corrélation de ces critères formels avec ceux de qualité globale de l'ontologie soit prouvée.

Il est également important d'évaluer les outils d'acquisition dans leur contexte applicatif et de mesurer leur valeur ajoutée dans ce cadre. Il s'agit alors de mesurer la différence de qualité de l'application elle-même selon qu'elle incorpore ou non des outils d'acquisition de connaissances. Ce type d'évaluations est indéniablement intéressant mais, il est difficile à mettre en oeuvre : on a du mal à dissocier ce qui relève de la qualité des ressources utilisées et du fonctionnement de l'application.

3 Difficultés

Nous avons décrit les expériences d'évaluation, dans le cadre de campagnes ou ponctuellement. Il en sort que ces expériences n'ont pas encore permis de faire émerger un cadre fédérateur d'évaluation comme c'est le cas dans d'autres disciplines. Nous présentons dans cette section quelques éléments qui expliquent, à notre sens, la difficulté de mise en œuvre d'un tel cadre pour les outils d'acquisition de connaissances.

3.1 Une tâche d'acquisition difficile à définir

Complexité des artefacts produits La principale difficulté tient à la nature même des terminologies et des ontologies. Ce sont des artefacts complexes. Considérons le cas des terminologies. Les termes eux-mêmes sont souvent des unités "complexes", composés de plusieurs mots, de longueur très variable et obéissant à des règles de variation multiples en corpus. Ensuite une terminologie ne se réduit souvent pas à une liste de termes, aussi complexes soient-ils. Des relations existent entre ces termes, et ces relations sont elles-mêmes diverses, depuis la variation morphologique (*véhicule d'occasion, véhicules d'occasion*) jusqu'aux relations de synonymie (*voiture d'occasion, automobile d'occasion*) ou d'hyperonymie (*véhicule d'occasion, véhicule*). La complexité même de ces artefacts constitue un frein à leur évaluation. On ne peut pas évaluer à la fois la qualité des termes extraits et des différentes relations entre ces termes. De même, il est difficile dans le cas d'une ontologie d'évaluer à la fois la qualité de ses concepts, sa structuration hiérarchique (granularité, densité, etc.), sa cohérence et sa consistence.

[6] Score utilisé comme critère de qualité en traduction automatique

Importance du rôle de l'application L'application pour laquelle la terminologie ou l'ontologie ont été développées joue aussi un rôle dans les contours de ces artefacts que l'on cherche à produire ainsi que sur les critères de qualité qui peuvent être exigés. Par exemple, toutes les terminologies ne doivent pas être évaluées selon les mêmes critères : si on a besoin d'une liste de termes bien formés pour la présenter à l'utilisateur (*logement étudiant* plutôt que *logement de l'étudiant*), de simples associations de mots statistiquement pertinentes comme *étudiant-logement* suffisent quand il s'agit de mesurer le poids des termes dans un document. Cette variété dans les critères de qualité selon l'application empêche d'avoir un cadre global qui pourrait s'appliquer à tout type de terminologie ou ontologie.

Place de l'interaction Dans la mesure où le résultat des outils d'acquisition de connaissances est dépendant de l'application, du domaine considéré, du niveau de qualité attendu et du point de vue adopté, le processus d'acquisition est rarement vu comme un processus entièrement automatique. Le plus souvent les outils d'acquisition sont des outils d'aide à l'acquisition qui intègre la participation du terminologue ou de l'ontologue dans le processus de construction de la ressource sémantique. Cette part d'interaction nécessite, au moment de l'évaluation, de départager ce qui relève du système d'acquisition et la part de travail manuel.

Absence de référence stable Au-delà de l'hétérogénéité formelle des artefacts produits, on observe également une grande hétérogénéité d'un point de vue sémantique, ce qui constitue une troisième difficulté pour l'évaluation. Pour un même domaine, on peut produire plusieurs terminologies ou ontologies différentes qui ne décrivent pas le domaine modélisé avec la même granularité, qui ne reflètent pas le même point de vue sur le domaine et qui peuvent même traduire des parti pris terminologiques et ontologiques différents. Le fait de partir de corpus permet de restreindre quelque peu l'espace de ces choix mais seulement partiellement. Ceci se traduit par l'absence de ressources standard utilisables pour l'évaluation. Autant de facteurs d'hétérogénéité qui sont difficiles à isoler et qui compliquent donc un peu plus l'évaluation : il n'y a pas de référence stable mais au contraire un grand choix de "solutions possibles".

Il est en effet assez difficile de disposer d'un référentiel établi pour un domaine et une application donnés. La construction de ces ressources est coûteuse et l'évaluation d'outil d'acquisition automatique ne serait pas aussi cruciale s'il était aisé de les construire manuellement. De plus, en admettant qu'on dispose d'un référentiel construit manuellement, le silence dans les résultats peut être dû à un manque dans le corpus d'extraction ou à une faille dans les systèmes d'acquisition.

Le référentiel peut être produit de différentes manières. On peut réutiliser une ressource existante mais elle risque de n'être que partiellement liée au corpus d'acquisition. On peut demander à un ou des experts de procéder à l'acquisition manuelle des connaissances à partir du même corpus d'acquisition qui est fourni aux systèmes. Cette seconde solution a l'avantage de la fiabilité mais elle est coûteuse. Une troisième approche consiste comme dans CoRRecT à fusionner la validation des résultats de différents systèmes. Même si elle manque d'exhaustivité, cette approche permet de comparer les résultats des différents systèmes entre eux et est *a priori* moins coûteuse que la précédente.

La nature même des connaissances, le lien à la langue et à une nécessaire modélisation explique la variation des références (même pour un domaine spécifique). Il est donc nécessaire de prendre en compte l'ensemble des solutions possibles, comme c'est le cas par exemple pour la traduction automatique.

3.2 Diversité des protocoles

Nous présentons dans la figure 1, différents scénarios d'évaluation.

Le premier consiste à comparer les sorties du système à une référence. Même si les sorties ainsi produites ne sont généralement pas utilisées sans être validées ou filtrées, il est important de pouvoir évaluer les sorties de manière indépendante. Elles offrent un bon point de comparaison entre différents outils.

Le second scénario vise à évaluer l'interaction. Quand les terminologies ou ontologies sont considérées comme ressources autonomes, elles font généralement l'objet d'un travail de validation par un expert avant d'être publiées. Ce type d'évaluation permet de mesurer l'effort fourni par l'expert pour aboutir à un produit final. Cette évaluation est évidemment liée à la qualité de l'interface de validation qui peut jouer un rôle important en facilitant le travail de l'expert et à la problématique de l'usage[7] mais nous mettons ici l'accent sur les opérations de validation elles-mêmes que l'on peut interpréter comme une distance d'édition.

Le troisième scénario vise à évaluer au travers d'une application. Comme nous l'avons souligné dans la section 2.2, une telle évaluation sert à quantifier la "plus-value" du produit (brut ou final) et à mesurer son apport à l'application. On a vu plus haut, cependant, que ce type d'évaluation n'est pas aisée car on a du mal à savoir si on est en train d'évaluer l'apport du produit ou la manière dont il est pris en compte dans le système.

[7] Des protocoles d'évaluation centrés sur l'usage sont proposés dans (Mustafa El Hadi et Chaudiron, 2007)

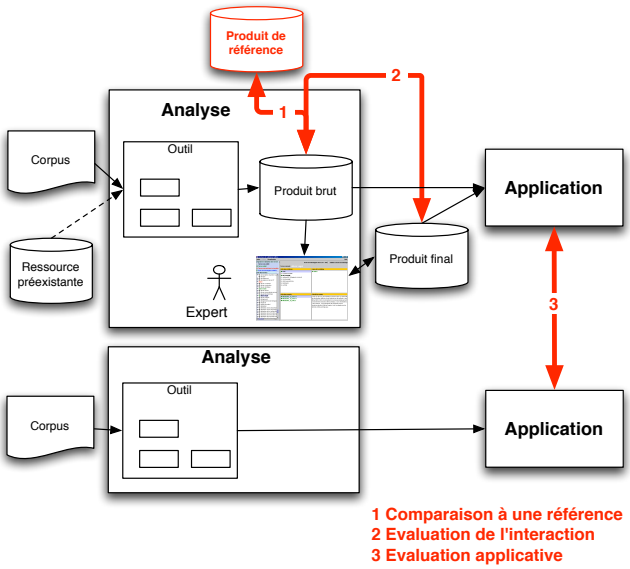

FIG. 1 – *Place des outils d'acquisition*

Les deux premiers scénarios d'évaluation permettent une évaluation comparative (entre deux ressources) qui est particulièrement adaptée aux évaluations de type "boîte noire" dans lesquelles on s'intéresse uniquement aux résultats fournis par les logiciels et non pas à leur mode de production. Ce type d'évaluation peut être représenté comme un triplet $< S, Corr, R) >$ où S est le produit à évaluer, R est une référence (indépendante ou construite à partir de la validation de S) et $Corr$ est une fonction de correspondance qui permet de mesurer la proximité entre les deux ressources. Les résultats de cette fonction de correspondance servent de base pour le calcul des différentes mesures d'évaluation.

Nous préconisons dans ce qui suit ce type d'évaluation en détaillant les difficultés inhérentes à ce cadre.

3.3 Limites des mesures actuelles

Les fonctions de correspondance entre le résultat produit et le résultat attendu doivent prendre en compte les caractéristiques de ces derniers. Les mesures les plus répandues pour rendre compte des résultats des systèmes sont très classiques : le rappel et la précision. Ce sont des mesures faciles à calculer, qui ont aussi le mérite d'être faciles à interpréter quel que soit le domaine de recherche où elles sont appliquées. Elles permettent de rendre compte du bruit et du silence des résultats produits par rapport à un résultat de référence.

Néanmoins, ces mesures ne sont pas applicables telles quelles pour juger de la qualité des connaissances parce qu'elles font l'hypothèse que la pertinence est une notion binaire (oui/non), ce qui dans la réalité n'est pas le cas. Cette pertinence binaire relève d'une logique du "tout ou rien" et ne tient pas compte du fait qu'un résultat s'approche plus de la solution qu'un autre. Toutes les erreurs ne devraient pas compter autant : certaines erreurs sont flagrantes, d'autres plus discutables, d'autres encore sont apparentées et donc plus facilement identifiables. Ce fait est d'autant plus problématique que la référence elle-même ne peut être unique comme expliqué plus haut. Les mesures de rappel et précision ne permettent ni de discriminer les résultats entre eux (différence entre un mauvais et un moins mauvais), ni de mesurer l'effort requis pour corriger une sortie.

On retrouve ce problème dans d'autres domaines de recherche telles que la recherche d'information structurée et l'alignement d'ontologies. Dans le cadre de la campagne INEX [8], il a été proposé dès le départ de graduer le jugement de pertinence en le quantifiant selon deux nouveaux critères (spécificité et exhaustivité) (Fuhr et al., 2002).. La campagne a constitué un excellent vivier de définition des mesures, telles que la mesure EPRUM (*expected precision-recall with user modelling*) proposée par Piwowarski et Dupret (2006), même si les mesures ne sont pas encore stabilisées (Fuhr et al., 2007). La question de l'évaluation des résultats des systèmes d'alignement d'ontologies est posée dans le cadre de la campagne OAEI[9]. Une solution proposée dans Ehrig et Euzenat (2005) est d'approximer la référence par similarité. Nous verrons dans la section 4.2 que nous adoptons une solution similaire pour l'évaluation de l'acquisition de terminologies.

[8] Initiative for the Evaluation of XML Retrieval
[9] Ontology Alignment Evaluation Initiative

4 Premières propositions

Nous proposons dans ce qui suit des solutions aux difficultés présentées plus haut. Nous proposons d'abord de décomposer le processus d'acquisition en fonctionnalités élémentaires pour mieux délimiter les objets d'étude. Nous proposons ensuite de définir des mesures qui tiennent compte de la variation des références. Il nous paraît aussi essentiel de mettre en œuvre une méta-évaluation pour vérifier l'adéquation des mesures aux tâches à évaluer.

Nous détaillons les deux premières propositions en rapportant les travaux effectués dans le cadre du programme Quaero sur l'acquisition des terminologies.

4.1 Découper pour mieux observer

Un des enjeux de l'évaluation des outils d'acquisition consiste à décomposer le processus d'acquisition global en fonctionnalités élémentaires. Il s'agit d'identifier, au-delà de la diversité des outils d'acquisition et de la complexité des résultats qu'ils produisent, et en faisant abstraction à la fois des méthodes utilisées par les outils et des applications pour lesquelles ils ont été conçus, d'identifier ce qu'ils ont en commun et de les découper en fonctionnalités élémentaires et indépendantes. Ces fonctionnalités ne fournissent pas nécessairement des résultats utilisables en tant que tels mais elles offrent des points de comparaison entre les outils. Ce ne sont pas forcément des sorties standard des systèmes à évaluer mais peu importe dès lors qu'on peut extraire des résultats de leurs sorties standard (voir figure 2).

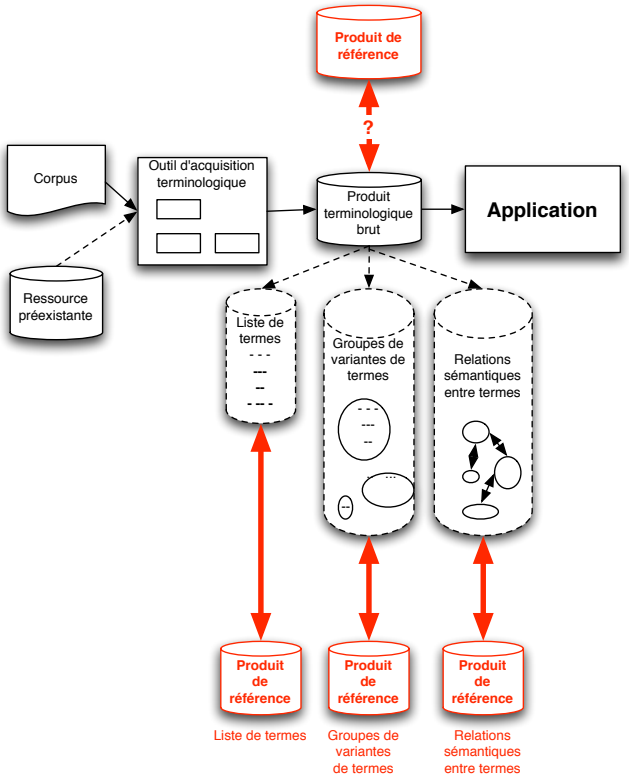

FIG. 2 – *Découpage et évaluation par fonctionnalité.*

Pour l'acquisition de terminologie, nous avons fait ce travail de "décomposition en facteurs premiers", première étape de la définition d'un protocole d'évaluation. En nous inspirant des sous-tâches définies dans les campagnes précédentes, nous avons proposé de distinguer trois fonctionnalités élémentaires génériques et d'évaluer les outils d'acquisition terminologiques selon ces trois dimensions (Zargayouna et al., 2007).

L'extraction terminologique représente la capacité d'un système à établir une liste de termes simples et complexes à partir d'un corpus d'acquisition. La liste plate des termes extraits du corpus est rarement utilisée en tant que telle dans les applications mais la plupart des outils d'acquisition terminologique passent par une étape d'extraction. Cette étape du processus d'acquisition constitue donc un point d'observation privilégié pour l'évaluation. L'évaluation, à ce niveau, peut consister à comparer la sortie d'un ou plusieurs systèmes avec une terminologie de référence, mais aussi à comparer

la sortie brute d'un système avec cette même sortie une fois validée. Nous avons pris le parti de dissocier l'extraction terminologique à proprement parler du tri des termes extraits, les critères de pertinence essentiellement dépendants de l'application cible dans laquelle la terminologie doit être exploitée.

Le calcul de variation terminologique consiste à regrouper les familles de termes qui sont des variantes les uns des autres, c'est-à-dire à établir des classes d'équivalence de termes. Pour s'en tenir à des fonctionnalités élémentaires, nous dissocions le fait de construire la classe (ou d'identifier des relations d'équivalence entre termes) et le choix du représentant de la classe, le terme canonique. Les systèmes qui font du calcul de variation (en lien avec l'extraction de termes, l'expansion de thesaurus ou l'indexation contrôlée) n'ont pas tous besoin d'identifier un terme canonique. Évaluer le calcul de variation va consister à comparer les relations d'équivalence proposées par les systèmes entre eux ou par rapport à une référence établie au préalable.

Cette fonction est cependant plus difficile à définir que la précédente dans la mesure où différentes relations d'équivalence peuvent être prises en compte. Certains auteurs distinguent différents niveaux de variation (variation morphologique, syntaxique ou sémantique) mais ce critère est lié aux méthodes de repérage en corpus plutôt qu'à la nature du résultat. En tenir compte biaiserait l'évaluation. Il faut au contraire tenir compte de la nature de la relation obtenue qui doit refléter une équivalence sémantique. On dira que deux termes sont les variantes l'un de l'autre s'ils sont synonymes. On considérera donc comme bien formée la classe de termes suivante : *expression de gène, expression génique, gènes exprimés* et *production de gène*, dès lors qu'un terminologue connaissant la biologie pose que *production de gène* et *expression de gène* peuvent être employés l'un pour l'autre.

L'extraction de relations terminologiques est moins souvent présente dans les outils d'acquisition. Elle consiste à élaborer un réseau de relations sémantiques entre termes ou classes de termes. Il peut s'agir de différents types de relations : relations hiérarchiques ou relations plus spécialisées. La diversité des relations considérées, leur dépendance au domaine d'application et les différences de granularité des descriptions sémantiques produites font que cette troisième fonction est sans doute la plus difficile à évaluer.

Pistes pour l'acquisition d'ontologies Nous préconisons la même démarche pour les outils d'acquisition d'ontologies. Nos premières propositions s'orientent vers la décomposition en : (i) acquisition de classes sémantiques, (ii) structuration hiérarchique de l'ontologie, (iii) extraction de relations sémantiques ou rôles et (iv) formalisation[10]..

4.2 Définir des mesures et des protocoles pour chaque tâche

Si l'on adopte le découpage proposé ci-dessus, il faut donc, pour l'évaluation des outils d'acquisition terminologique, proposer des métriques pour chaque sous-tâche considérée séparément. Nous mettons ici l'accent sur la première de ces sous-tâches, l'extraction de termes.

Il est essentiel que les mesures tiennent compte des caractéristiques des produits à évaluer mais, en même temps, ces mesures doivent autant que possible être indépendantes des méthodes qu'elles visent à évaluer et elles doivent permettre d'évaluer chaque sous-tâche indépendamment les unes des autres, une fois une décomposition en tâches établie.

Pour l'extraction de termes, nous définissons des métriques qui tiennent compte de la variation des références et du fait que la pertinence est plus graduée que booléenne. Il est aussi important d'avoir des mesures simples et adaptables. De ce point de vue les mesures de rappel et de précision sont bien connues et communément admises. C'est pourquoi nous proposons d'adapter les mesures classiques de précision et rappel rappelées ci-dessous plutôt que d'en adopter de nouvelles :

$$precision = \frac{|S \cap R|}{|S|}$$

$$rappel = \frac{|S \cap R|}{|R|}$$

où $|S \cap R|$ est le nombre d'éléments pertinents retournés par le système, $|S|$ est le nombre d'éléments retournés par le système et $|R|$ le nombre d'éléments dans la référence.

La tâche d'extraction produit une liste de termes plate, dite "de sortie" (S), à comparer à une liste de termes de référence (R). La référence peut être de longueur variable, même pour un domaine et un corpus donnés : cela dépend de la granularité de la description terminologique choisie. La liste produite par le système peut elle-même être de taille très hétérogène comme l'ont montré les expériences de CESART. Elle peut comporter des termes pertinents (présents dans

[10]Précisons que nous excluons du champ de l'évaluation ontologique tout ce qui concerne la population des ontologies par extraction des entités nommées.

la référence), des termes non pertinents (considérés comme du bruit) et des termes proches des termes de la référence sans être strictement identiques. Ces derniers sont à considérer comme "presque bons" ou comme redondants si le terme exact figure aussi dans la ressource proposée. En pratique, la liste produite par le système d'extraction peut comporter des variantes de termes mais pour ne pas interférer avec le calcul de la variation qui fait l'objet de la deuxième sous-tâche, nous la considérons comme de la pseudo-redondance ici. Nous voulons pouvoir quantifier les phénomènes suivants :

- un système parfait ($S = R$) doit avoir une valeur de qualité maximale (si on raisonne en rappel et précision, la valeur doit être à 1).
- un système qui renvoie une liste de termes avec variantes ($S = R \cup Var(R)$[11]) ne doit pas être pénalisé ou faiblement.
- un système qui renvoie une liste de non termes ($S \cap R = \emptyset$) doit avoir la valeur minimale (valeur proche de zéro).
- la qualité d'un système S doit augmenter quand il se rapproche globalement de la référence.

Considérons le cas où $R = \{$*base de données*$\}$ et les listes résultats suivantes : $S_1 = \{$*base de données, bases de données*$\}$, $S_2 = \{$*bases de données*$\}$, $S_3 = \{$*base de données, langage de requête*$\}$. On souhaite que S_1 et S_2 soient considérés comme sensiblement de même qualité mais que celle de $S3$ soit moindre au regard de R.

Prendre une mesure de pertinence graduée Nous proposons de reprendre les mesures de rappel et précision classique mais en considérant une mesure de pertinence graduée. Cette fonction de pertinence $Pert(S, R)$ doit vérifier

$$|S \cap R| \leq Pert(S, R) \leq min(|S|, |R|)$$

et rendre compte d'une proximité globale entre S et R fondée sur un calcul de similarités individuelles entre les éléments de la sortie et ceux de la référence. Nous définissons pour cela une similarité entre deux éléments de S et R, $sim(e_s, e_r)$ par :

$$sim(e_s, e_r) = 1 - dist(e_s, e_r)$$

où $dist$ est une distance terminologique du type de celle proposée par El Mouedden (2008). Cette distance repose sur la distance d'édition qui permet de calculer de manière homogène la distance entre termes simples et la distance entres termes complexes comme l'a souligné Tartier (2004). Notre approche diffère cependant de cette dernière parce que nous ne voulons pas faire appel à des outils de calcul linguistique, lesquels pourraient introduire des biais dans l'évaluation. La distance entre termes complexes repose sur le même principe de l'alignement et de la distance optimale, mais on fait cette fois les opérations de transformation sur les mots plutôt que sur les caractères. On calcule l'alignement optimal qui permet d'atteindre un terme en partant de l'autre, en ajoutant à chaque fois le coût de l'opération faite. Les opérations d'ajout et de suppression ont le même coût égal à 1, tandis que le coût de la substitution varie selon la nature des mots en question : il est égal à la distance entre termes simples séparant les deux mots mis en correspondance. La distance finale est égale au coût optimal de l'ensemble des opérations divisé par le nombre de mots du plus long terme. Cette normalisation par la longueur du plus long terme permet de dire que *base de donnée relationnelle avancée* est plus proche de *base de donnée relationnelle* que ce dernier ne l'est de *base de données*.

On retient pour chaque terme de S le terme de la référence avec qui il a une similarité maximale, ce qui permet de définir la pertinence d'un terme de la manière suivante :

$$pert(e_s) = max_{e_r \in R}(sim(e_s, e_r)) \; si \; max_{e_r \in R}(sim(e_s, e_r)) > \sigma$$

$$pert(e_s) = 0 \; sinon$$

où σ est un seuil de similarité en-deçà duquel on considère que deux termes ne sont pas comparables[12].

Adapter formellement la sortie à la référence Dans la mesure où la référence ne peut pas être considérée comme un absolu, il serait artificiel de comparer directement la sortie du système avec la référence. On favoriserait trop le système qui aurait "par hasard" fait les mêmes choix de granularité de description que la référence et on risquerait d'avoir des résultats d'évaluation trop dépendants du type de référence adopté. Nous proposons donc de transformer la sortie pour trouver sa correspondance maximale avec la référence, ce qui revient à adapter la sortie à la référence. La précision et le rappel sont calculés sur la sortie transformée plutôt que de la sortie brute.

De fait, comme plusieurs termes de la sortie peuvent correspondre au même terme de la référence, on regroupe dans certains cas les termes de la sortie. Il s'agit donc de calculer les mesures de précision et rappel non pas directement sur S mais sur une partition de S qui est définie relativement à R. Nous définissons cette partition $\mathcal{P}(\mathcal{S})$ telle que toute partie

[11] Où $Var(R)$ est un ensemble de variantes de termes de R.

[12] Nous le fixons arbitrairement à 0,5 mais ce seuil pourrait être fixé en fonction de la distance des termes de la référence entre eux.

p de $\mathcal{P}(\mathcal{S})$ correspond soit à un ensemble des termes de S qui ont une similarité maximale strictement positive avec un même terme de R, soit à un terme singleton. On peut dès lors définir la pertinence d'une partie p de $\mathcal{P}(\mathcal{S})$[13] :

$$pert(p) = max_{e_s \in p}(pert(e_s))$$

Les mesures de rappel et de précision terminologiques se définissent alors comme suit :

$$T - precision = \frac{Pert(S,R)}{|\mathcal{P}(\mathcal{S})|} = \frac{\sum_{p \in \mathcal{P}(\mathcal{S})}(pert(p))}{|\mathcal{P}(\mathcal{S})|}$$

$$T - rappel = \frac{Pert(S,R)}{|R|} = \frac{\sum_{p \in \mathcal{P}(\mathcal{S})}(pert(p))}{|R|}$$

Nous pouvons vérifier que dans le cas d'un système parfait $T - precision = T - rappel = 1$ et que dans le cas d'un système qui ne retournerait que du bruit les valeurs de $T - precision$ et $T - rappel$ tendent vers 0. La figure 3 montre ce qu'on obtient comme mesures de précision et de rappel pour les trois sorties mentionnées ci-dessous.

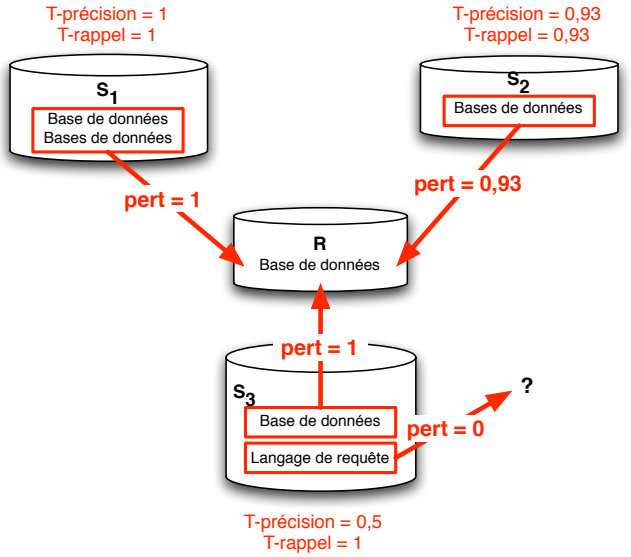

FIG. 3 – *Exemples de mesures de $T - precision$ et $T - rappel$.*

4.3 Méta-évaluer les métriques proposées

Une bonne évaluation nécessite que les efforts d'évaluation eux-même soient évalués. Il est nécessaire de vérifier l'évaluation pour détecter les problèmes tels que le biais, les erreurs techniques, les utilisations erronées (Stufflebeam, 1974).

Après avoir bien défini les spécifications des protocoles d'évaluation, la méta-évaluation consiste en une étape de vérification (comme en génie logiciel) qui revient à s'assurer que les spécifications ont été bien respectées. En d'autres termes, la méta-évaluation consiste à se doter de moyens pour vérifier qu'on est bien en train d'évaluer ce qui doit être évalué, de vérifier les biais, etc. En traduction automatique, la méta-évaluation a ainsi permis de bien cerner les limites des mesures utilisées. Les études reportées dans (Koehn et al., 2006) ont montré que la mesure Bleu sous-estime la qualité des systèmes à base de règles. Timimi (2007) rapporte un exemple de méta-évaluation qui a été proposé dans le cadre de la campagne d'évaluation CESART. L'étude s'intéresse au cadre global du protocole d'évaluation *i.e* choix du corpus de tests, statut de l'expert, etc.

Nous sommes actuellement en train de définir un cadre de méta-évaluation des mesures que nous avons présentées dans la section précédente. On peut facilement vérifier les critères de cohérence des mesures en vérifiant les bornes inférieures et bornes supérieures Popescu-Belis (1999). Nous voulons éprouver les mesures proposées expérimentalement sur des jeux de tests choisis pour vérifier le comportement des mesures dans certains cas particuliers. Ces mêmes tests devraient être exécutés sur des volumes significatifs, par exemple en introduisant progressivement du bruit représentatif et en suivant son influence sur les résultats.

[13] Des variantes de cette formule sont à étudier expérimentalement.

L'étude de corrélation des références, dans le cas où on dispose de plusieurs références ou plusieurs experts, peut être intéressante à mettre en regard avec les résultats des mesures.

L'évaluation étant un processus itératif, la méta-évaluation permet aussi de faire évoluer les mesures et les protocoles.

5 Conclusion

Peut-on évaluer les outils d'acquisition de connaissances à partir de textes ? Notre réponse est positive, mais il faut prendre en compte toutes les difficultés liées au domaine. Nous avons étayé notre réponse par des propositions concrètes dans le cadre de l'évaluation de l'acquisition terminologique. Nous proposons de nous affranchir dans un premier temps d'une vision globalisante des ressources produites pour décomposer la tâche d'acquisition et ses résultats en sous-tâches élémentaires et en résultats partiels. Il faut mettre l'accent sur les composantes de base qui sont communes à la plupart des outils d'acquisition de connaissance. Nous proposons également d'adapter à notre cadre les métriques existantes en essayant d'être le plus fidèle possible à cette part de variation caractéristique des systèmes qui manipulent la langue ou qui produisent des modèles de connaissances. Ces mesures sont implémentées et des expérimentations sont en cours pour valider nos propositions. Nous soulignons pour finir l'importance de la méta-évaluation pour la mise au point des métriques.

Remerciements

Ce travail a été partiellement réalisé dans le cadre du programme Quaero,financé par OSEO, l'agence française pour l'innovation. Nous remercions Mourad El Moueddeb, Olivier Hamon et Laurent Audibert pour leurs discussions et contributions à ce travail.

Références

Ait El Mekki, T. et A. Nazarenko (2006). An application-oriented terminology evaluation : the case dof back-of-the-book indexes. In R. Costa, F. Ibekwe-SanJuan, S. Lervad, M.-C. L'Homme, A. Nazarenko, et H. Nilsson (Eds.), *Proceedings of the Workshop "Terminology Design : Quality Criteria and Evaluation Methods" (TerEval) associated with the Language Ressource and Evaluation Conference (LREC)*, Genova, Italy, pp. 18–21.

Aubin, S., A. Nazarenko, et C. Nédellec (2005). Adapting a general parser to a sublanguage. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'05)*, Borovets, Bulgaria, pp. 89–93.

Baziz, M., N. Aussenac-Gilles, et M. Boughanem (2003). Désambiguisation et expansion de requêtes dans un sri, etude de l'apport des liens sémantiques. *Revue des Sciences et Technologies de l'Information (RSTI) série ISI 8*(4), 113–136.

Bourigault, D. (1993). An endogeneous corpus-based method for structural noun phrase disambiguation. In *Proceedings of the 6th European Chapter of the Association for Computational Linguistics (EACL'93)*, pp. 81–86.

Brank, J. (2006). Gold standard based ontology evaluation using instance assignment. In *Proceedings of the Evaluation of Ontologies for the Web (EON 2006) Workshop*.

Ehrig, M. et J. Euzenat (2005). Relaxed precision and recall for ontology matching. In *K-Cap 2005 workshop on Integrating ontology, Banff (CA)*, pp. 25–32.

El Moueddeb, M. (2008). Définition d'un protocole d'évaluation des outils d'analyse terminologique. Master's thesis, Université Paris-Nord, France.

Enguehard, C. (2003). Correct : Démarche coopérative pour l'évaluation de systèmes de reconnaissance de termes. In *Actes de la 10ème conférence annuelle sur le Traitement Automatique des Langues (TALN 2003)*, Nancy, pp. 339–345.

Fuhr, N., N. Gövert, G. Kazai, et M. Lalmas (Eds.) (2002). *Proceedings of the First Workshop of the INitiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, Germany, December 9-11, 2002*.

Fuhr, N., J. Kamps, M. Lalmas, S. Malik, et A. Trotman (2007). Overview of the inex 2007 ad hoc track. In *INEX*, pp. 1–23.

Grinberg, D., J. Lafferty, et D. Sleator (1995). A robust parsing algorithm for link grammars. In *Proceedings of the Fourth International Workshop on Parsing Technologies*.

Hamon, O., A. Hartley, A. Popescu-Belis, et K. Choukri (2007). Assessing Human and Automated Quality Judgments in the French MT Evaluation Campaign CESTA. In *Proceedings of MT Summit XI*, Copenhagen, Danmark.

Kageura, K., T. Fukushima, N. Kando, M. Okumura, S. Sekine, K. Kuriyama, K. Takeuchi, M. Yoshioka, T. Koyama, et H. Isahara (2000). Ir/ie/summarisation evaluation projects in japan. In *LREC2000 Workshop on Using Evaluation within HLT Programs*, pp. 19–22.

Koehn, P., N. Bertoldi, O. Bojar, C. Callison-Burch, A. Constantin, B. Cowan, C. Dyer, M. Federico, E. Herbst, H. Hoang, C. Moran, W. Shen, et R. Zens (2006). Factored translation models. In J. H. University (Ed.), *CLSP Summer Workshop Final Report WS-2006*.

Langlais, P. et M. Carl (2004). General-purpose statistical translation engine and domain specific texts : Would it work ? *Terminology 10*(1), 131–152.

Maynard, D., W. Peters, et Y. Li (2006). Metrics for evaluation of ontology-based information extraction. In *Proceedings of the Evaluation of Ontologies for the Web (EON 2006) Workshop*.

Mustafa El Hadi, W. et S. Chaudiron (2007). L'évaluation des outils d'acquisition de ressources terminologiques : problèmes et enjeux. In *Actes de la Conférence TOTh (Terminologie et Ontologie : Théories et Applications)*, pp. 163–179.

Mustafa el Hadi, W., I. Timimi, M. Dabbadie, K. Choukri, O. Hamon, et Y. Chiao (2006). Terminological resources acquisition tools : Toward a user-oriented evaluation model. In *Proceedings of the Language Resources and Evaluation Conference (LREC'06)*, Genova, Italy, pp. 945–948.

Obrst, L., T. Hughes, et S. Ray (2006). Prospects and posiibilities for ontology evaluation : The view from ncor. In *Proceedings of the Evaluation of Ontologies for the Web (EON 2006) Workshop*.

Piwowarski, B. et G. Dupret (2006). Evaluation in (xml) information retrieval : expected precision-recall with user modelling (eprum). In *SIGIR*, pp. 260–267.

Popescu-Belis, A. (1999). L'évaluation en génie linguistique : un modèle pour vérifier la cohérence des mesures. *Langues (Cahiers d'études et de recherches francophones) 2*(2), 151–162.

Sabou, M., V. Lopez, E. Motta, et V. Uren (2006). Ontology selection : Ontology evaluation on the real semantic web. In *Proceedings of the Evaluation of Ontologies for the Web (EON 2006) Workshop*.

Stufflebeam, D. L. (1974). Metaevaluation. In *Occasional Paper Series, Kalamazoo MI : Western Michigan University Evaluation Center*.

Tartier, A. (2004). *Analyse automatique de l'évolution terminologique : variations et distances*. Ph. D. thesis, Université de Nantes.

Timimi, I. (2007). Peut-on faire confiance aux outils de terminologie ? ou l'évaluation entre un souci de normalisation et une complexité de modélisation. In *Actes du Colloque Terminologie et Ontologie Théorie et Applications (Toth)*.

Zargayouna, H. (2005). *Indexation sémantique de documents XML*. Ph. D. thesis, Uniersité Paris-Sud, France.

Zargayouna, H., O. Hamon, et A. Nazarenko (2007). Evaluation des outils terminologiques : état des lieux et propositions. In *Actes des 7èmes rencontres Terminologie et Intelligence Artificielle*.

Summary

A large effort has been devoted to the development of knowledge acquisition tools, but it is still difficult to assess the progress that have been made.. The lack of well-accepted evaluation protocols and data leads to a gap in comparing results. In this article, we raise the question of evaluation for acquisition from text of terminologies and ontologies. We underline the major difficulties and describe first propositions we have made so far in this context.

Jugements humains et évaluation d'algorithmes de segmentation thématique : application de WindowDiff

Yves Bestgen*

*Université catholique de Louvain, Centre for English Corpus Linguistics,
Place du Cardinal Mercier, 10, B-1348 Louvain-la-Neuve, Belgique
yves.bestgen@psp.ucl.ac.be,
http://www.psor.ucl.ac.be/personal/yb/

Résumé. L'objectif de cette recherche est d'étudier les problèmes qui se posent lors du recours à des juges pour déterminer la norme de référence nécessaire à l'évaluation de procédures de segmentation thématique. Il s'agit tout particulièrement d'évaluer la possibilité d'utiliser la mesure d'erreurs WindowDiff pour estimer la fiabilité de juges en distinguant la fiabilité moyenne d'un juge de celle d'un ensemble de juges qui ont effectué la même tâche. Pour ce faire, 32 textes ont été segmentés par des groupes de 13 juges. Les performances des juges sont comparées à celle d'un algorithme de segmentation fréquemment employé comme point de repère lors de la mise au point de nouveaux algorithmes. Dans la conclusion, les avantages et limites de la mesure WindowDiff sont discutés.

1 Introduction

La segmentation thématique de textes a pour objectif de localiser les changements de thème dans des documents. Ce type d'informations peut permettre l'amélioration de nombreuses applications en traitement automatique des langues naturelles comme l'extraction d'informations, le résumé automatique ou encore la navigation à l'intérieur de longs textes. Une série de recherches ont par exemple mis en évidence l'intérêt de segmenter des textes en fonction des thèmes qu'ils abordent afin d'améliorer les résultats de procédures d'extraction d'informations (Hearst, 1997 ; Prince et Labadié, 2007). Ces dernières années, de nombreux algorithmes de segmentation thématique, basés principalement sur la cohésion lexicale, ont été proposés (p.ex., Brants et al., 2002 ; Choi, 2000 ; Ferret, 2002 ; Fragnou et al., 2004 ; Hearst, 1997 ; Ponte et Croft, 1997 ; Utiyama et Isahara, 2001) rendant encore plus important le double problème que pose leur évaluation. Il est, en effet, nécessaire de disposer non seulement d'indices adéquats pour mesurer l'efficacité, mais également d'une norme à laquelle la segmentation proposée par l'algorithme peut être comparée. Ceci est évidemment vrai pour toute évaluation, mais ces problèmes sont particulièrement manifestes dans le champ de la segmentation thématique comme la suite de cette section l'expose.

1.1 Indices pour mesurer l'efficacité

En ce qui concerne la mesure de l'efficacité, l'indice Pk proposé par Beeferman et al. (1999) et ultérieurement amélioré par Pevzner et Hearst (2002) sous le nom de WindowDiff semble faire actuellement l'objet d'un consensus. Cette mesure d'erreur compare une segmentation hypothétique (par exemple, celle d'une procédure automatique) à une segmentation de référence. WindowDiff est défini par Pevzner et Hearst comme suit

$$WindowDiff(ref, hyp) = \frac{1}{N-k}\sum_{i=1}^{N-k}(|b(ref_i, ref_{i+k}) - b(hyp_i, hyp_{i+k})| > 0)$$

où *b(i, j)* représente le nombre de frontières entre les positions *i* et *j*, *N* représente le nombre de positions et *k* correspond à la moitié de la longueur moyenne d'un segment dans l'annotation de référence. De manière moins formelle, on peut décrire le fonctionnement de WindowDiff de la manière suivante. Une fenêtre de taille *k* est déplacée tout au long des unités minimales de segmentation d'un texte (habituellement les phrases). Pour chaque position de la fenêtre, on compare le nombre de frontières de segments que celle-ci englobe selon la norme de référence au nombre de frontières détectées par l'algorithme. Celui-ci est pénalisé d'un point chaque fois que ces nombres sont différents. Le dénominateur permet

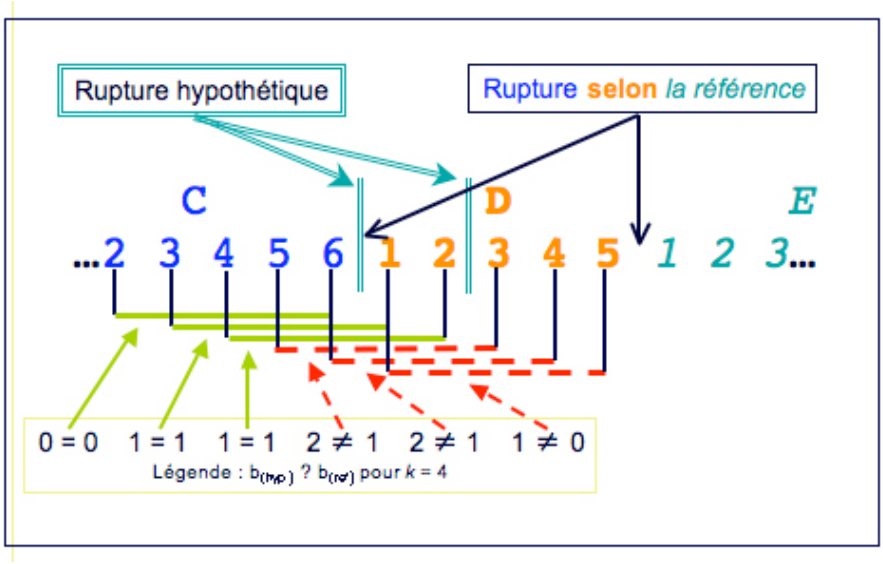

FIG. 1 – *Exemple de calcul de l'indice WindowDiff (adapté de Pevzner et Hearst, 2002).*

d'obtenir un score WindowDiff compris entre 0 et 1[1]. Comme il s'agit d'une mesure d'erreur, plus sa valeur est proche de 0 et meilleure est la performance de l'algorithme.

La figure 1, adaptée de Pevzner et Hearst (2002), illustre le fonctionnement de cet indice d'erreur. Les unités minimales d'un texte y sont représentées par des chiffres qui traduisent la segmentation selon la norme de référence. Selon celle-ci, le segment C contient six unités minimales et le D cinq. Ces segments sont également mis en évidence par la mise en forme des caractères et les couleurs. La segmentation hypothétique met en évidence deux ruptures, l'une identique à une de celles mises en évidence par la norme de référence et l'autre non (celle entre D2 et D3). Pour un paramètre k égal à 4, les traits horizontaux traduisent le déplacement de la fenêtre mobile. Les trois premiers traits (verts et continus) signalent des fenêtres pour lesquelles les deux segmentations sont d'accord puisqu'elles marquent le même nombre de ruptures (0 ou 1). Les trois traits discontinus et rouges signalent des points de désaccord : les deux extrémités de la fenêtre mobile ne sont pas séparées par le même nombre de ruptures selon les deux segmentations. Elles pénalisent donc la segmentation hypothétique.

L'intérêt majeur de cet indice par rapport aux mesures plus classiques de rappel, de précision et de F-score est qu'il pénalise moins les erreurs légères, comme le fait de placer une frontière juste à côté de la position attendue, par comparaison aux erreurs plus graves, comme manquer une frontière ou ajouter une frontière, alors que les indices classiques ne font aucune différence entre ces types d'erreurs.

1.2 Norme de référence pour évaluer l'efficacité

Comment déterminer les véritables changements de thème à identifier ? Cette seconde question est encore plus problématique dans le champ de la segmentation thématique. Si quelques études ont choisi d'évaluer les performances d'un système sur la base des bénéfices qu'il apporte à l'application pour laquelle il a été conçu (Bellot et El-Bèze, 2001 ; Prince et Labadié, 2007), la majorité des chercheurs procèdent en comparant la segmentation postulée à une norme censée correspondre à la vraie segmentation du texte ("gold standard"). Pour déterminer cette norme, deux approches sont principalement employées. La première consiste à demander à des juges d'effectuer la même tâche que l'algorithme et donc à segmenter des textes de diverses origines (Hearst, 1997 ; Kozima, 1993 ; Passonneau et Litman, 1997). La seconde s'appuie sur un matériel artificiel obtenu en concaténant des textes, les changements de thème à identifier correspondant aux frontières entre ceux-ci. Depuis quelques années, cette seconde approche s'est très largement imposée en raison de sa simplicité et de son faible coût ainsi que de l'existence d'un matériel de référence (Choi, 2000), qui permet de comparer les performances de tout nouvel algorithme à celles des algorithmes considérés comme les plus efficaces selon la littérature (Bestgen, 2006).

[1]Une autre formule, prenant en compte le nombre de différences entre les deux segmentations et non la dichotomisation de celui-ci, est parfois employée, même si elle ne correspond pas à la formule originale de Pevzner et Hearst. Sa principale limitation est de ne pas être bornée entre 0 et 1.

Évaluer un algorithme de segmentation au moyen de textes concaténés est une procédure parfaitement justifiée lorsque la fonction pour laquelle il a été développé consiste à segmenter des séquences continues de brefs textes (Allan et al., 1998 ; Ponte et Croft, 1997). Elle est, par contre, beaucoup plus discutable lorsque l'objectif est d'identifier les changements de thèmes à l'intérieur de textes (Ferret, 2006). En effet, ceux-ci sont liés entre eux et au thème du texte en général. Prétendre qu'un algorithme efficace dans une situation le sera aussi dans l'autre est pour le moins imprudent comme le souligne Hearst (1997). Or, nombre d'applications de la segmentation automatique visent la mise au jour de la structure thématique d'un texte. Ce second objectif semble être aussi le plus pertinent pour l'extraction de connaissances à partir de textes parce que, non seulement, les frontières thématiques à l'intérieur d'un texte sont rarement marquées explicitement contrairement aux frontières entre les textes (Hearst, 1997), mais aussi parce que, dans le cas des procédures de segmentation basées sur la cohésion lexicale, on peut penser qu'une procédure efficace pour segmenter un texte devrait être également efficace pour segmenter entre les textes alors que l'inverse est loin d'être évident. Ces différentes raisons m'ont conduit à étudier les problèmes qui se posent lors du recours à des juges pour déterminer la norme de référence et tout particulièrement la question de la fiabilité de leurs jugements.

1.3 Évaluation de la fiabilité de juges dans une tâche de segmentation

Comme l'ont récemment souligné Artstein et Poesio (2008) dans leur revue de question à propos de l'évaluation de l'accord interjuges en linguistique computationnelle, "*The analysis of discourse structure—and especially the identification of discourse segments—is the type of annotation that, more than any other, led CL researchers to look for ways of measuring reliability and agreement, as it made them aware of the extent of disagreement on even quite simple judgments...*". Ces difficultés sont, par exemple, mises en évidence dans une étude de Bestgen et Piérard (2006) dont l'objectif était de développer un matériel de référence pour l'évaluation de procédures de segmentation thématique. Trente-deux articles du journal *Le Monde* ont été segmentés par des groupes de 15 juges. L'analyse de leurs réponses confirme le manque de fiabilité des juges, puisque le coefficient kappa moyen, classiquement employé dans ce contexte (Carletta, 1996) est inférieur à 0.45 alors qu'on considère qu'une valeur minimale de 0.67 et même de 0.80 est nécessaire.

La réponse de Artstein et Poesio (2008) au manque de fiabilité des données issues de juges pour l'évaluation des procédures de segmentation est de proposer l'emploi, non du coefficient kappa, mais de la mesure WindowDiff (Pevzner et Hearts, 2002) déjà mentionnée ci-dessus. Celle-ci permet en effet de prendre en compte le fait que les juges, comme les procédures automatiques, peuvent détecter les différents thèmes tout en se trompant sur la localisation exacte de leurs frontières. Il s'agit sans aucun doute d'une amélioration puisqu'on voit difficilement pourquoi on admettrait que les procédures automatiques soient moins pénalisées pour des erreurs légères que pour des erreurs graves, alors que les réponses des juges seraient évaluées en termes de tout ou rien. Artstein et Poesio (2008) ne font toutefois que suggérer l'emploi de cet indice sans discuter les modalités pratiques de son application. Un des objectifs de la présente recherche est d'évaluer les conséquences de l'emploi de celui-ci.

1.4 Fiabilité moyenne d'un juge et fiabilité effective de plusieurs juges

Une remise en cause de la mesure de la fiabilité plus profonde que celle présentée par Artstein et Poesio (2008) semble cependant nécessaire. Pour une raison toute différente de celle qu'ils avancent, il est en effet loin d'être évident que des indices comme kappa ou même WindowDiff, calculés de la manière classique soient les mesures les plus adéquates dans ce genre de tâche. En effet, même lorsque plus de deux ou trois juges sont interrogés, l'indice de fiabilité calculé, qu'il s'agisse de la moyenne des indices obtenue en appariant les juges de toutes les manières possibles ou qu'il résulte d'une formule spécifique, mesure ce que Rosenthal (1982) appelle la fiabilité moyenne d'un juge. Elle répond à la question suivante : deux juges donnés segmentent-ils les textes aux mêmes endroits ? Si cette fiabilité est élevée, cela signifie qu'il n'est pas nécessaire d'interroger un grand nombre de juges puisque chacun d'entre eux apporte toute l'information utile. Une fiabilité moyenne d'un juge élevée est indispensable dans toutes les études qui procèdent en présentant à chaque juge un matériel spécifique à annoter auquel s'ajoute une fraction du matériel qui est commune à l'ensemble des juges afin d'estimer cette fiabilité. La situation est toute autre lorsque le matériel est systématiquement annoté par plusieurs juges comme c'est habituellement le cas dans le domaine de la segmentation thématique (Hearst, 1997 ; Passoneau et Litman, 1997). Dans ce cas, ce n'est pas la fiabilité d'un juge qui est la plus importante, mais celle de l'indice qui peut être dérivé de l'ensemble des juges qui ont traité une même portion du matériel. Cet indice est obtenu en comptant le nombre de juges qui ont segmenté entre chaque paire contiguë de phrases. Le calcul de cet indice global de segmentation (IG) est illustré à la figure 2 pour les segmentations hypothétiques par 4 juges (J1 à J4) d'un texte composé de 20 unités minimales (UM). Si on choisit d'interroger plusieurs juges, c'est parce qu'on pense a priori qu'ils ne seront pas systématiquement d'accord, que chaque juge percevra la structure des textes imparfaitement, mais que les erreurs des uns seront compensées par celles des autres et qu'au total, l'indice global reflétera adéquatement la segmentation des textes. C'est, au moins implicitement, la position prise par Hearst (1997) ou Passonneau et Litman (1997) lorsqu'ils ont décidé de ne prendre en compte que les

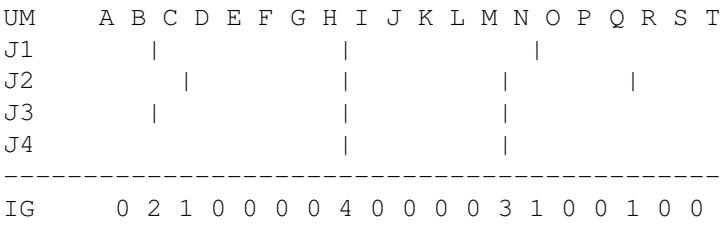

FIG. 2 – *Exemple de calcul de l'indice global de segmentation (IG).*

segments marqués par plusieurs juges. Cette fiabilité effective, comme l'appelle Rosenthal, est plus élevée que la fiabilité d'un juge puisque les erreurs propres à chaque juge, qui sont la source du manque de fiabilité de leur jugement, tendent à s'annuler les unes les autres. Conformément à la théorie classique des scores (Leveault et Grégoire, 1997), le gain dépend du nombre de juges (plus il est grand, plus la fiabilité effective sera élevée) et de la fiabilité moyenne d'un juge (si celle-ci est très élevée, on gagne peu à multiplier le nombre de juges). Comme la fiabilité effective sera, sauf situation extrême, toujours supérieure à la fiabilité moyenne d'un juge, ne pas la prendre en compte pose deux problèmes. D'une part, elle risque de laisser croire à tort qu'un domaine (comme la segmentation de textes) ne peut pas être étudié sur la base de normes de segmentation obtenues à partir de tâche de jugements puisque les accords entre les juges sont très faibles. D'autre part, la fiabilité des juges dans une tâche de segmentations, mais également dans d'autres tâches de jugements, peut être employée pour fournir une limite supérieure pour jauger l'efficacité de procédures automatiques en comparant l'efficacité de celles-ci à celle des juges eux-mêmes (Hearst, 1997 ; Sporleder et Lapata, 2006). Cette limite supérieure sera différente selon qu'elle est calculée sur la base de la fiabilité moyenne d'un juge ou de la fiabilité effective.

Pour quantifier cette fiabilité effective, différents indices ont été développés en psychométrie (Shrout et Fleiss, 1979), comme le coefficient alpha de Cronbach[2] (voir Bestgen et Piérard (2006) pour une utilisation de ce coefficient). Appliqués à la segmentation thématique, ces indices présentent toutefois la même limitation que le coefficient kappa : ils pénalisent de manière équivalente les erreurs légères et les erreurs graves. Il est donc souhaitable de suivre, comme pour l'estimation de la fiabilité moyenne d'un juge, la recommandation de Artstein et Poesio (2008) et d'employer la mesure WindowDiff. Toutefois, aucune procédure permettant de mettre en pratique ce type d'analyse n'a, à ma connaissance, été proposée jusqu'à présent. Le deuxième objectif de cette étude est de répondre au moins partiellement à ce manque. Pour ce faire, des groupes de 13 juges ont reçu pour tâche d'identifier les changements de thème qu'ils percevaient dans 32 textes. La présentation de ce matériel fait l'objet de la section suivante. Une série d'analyses visant à établir la fiabilité d'un juge, mais aussi celle de l'indice moyen qui peut être dérivé de plusieurs juges, sont ensuite présentées. Enfin, comme ce travail a pour objet majeur de contribuer au développement de procédures pour l'évaluation d'algorithmes de segmentation thématique, ce même matériel est utilisé dans la quatrième section pour mesurer l'efficacité de la procédure de segmentation C99 (Choi, 2000) qui sert fréquemment de point de repère pour évaluer de nouveaux algorithmes.

2 Constitution du matériel d'évaluation

La procédure utilisée pour construire le matériel est inspirée de celle employée dans l'étude de Hearst (1997) et appliquée par Bestgen et Piérard (2006). Elle a été légèrement modifiée afin de remédier à une limitation de cette dernière étude. Leur étude ne portait que sur un seul genre de textes, des articles du journal *Le Monde*. Analyser la fiabilité de juges sur la base de ce seul matériel est d'autant plus discutable qu'il s'agit d'un des journaux français les plus difficiles à comprendre. Pour cette raison, la présente étude est basée sur des textes extraits de quatre sources différentes.

Le matériel présenté aux juges est composé de 32 textes qui ont été extraits de deux genres : des articles de journaux et des articles d'encyclopédie. Pour chacun de ces deux genres, une source était destinée à un public jeune (le journal *Le Petit Ligueur* et l'*Encyclopédie Multimédia Hachette*) et l'autre à un public d'adultes (le journal *Le Monde* et l'*Encyclopédie Universalis*). Les textes ont été sélectionnés par une procédure aléatoire parmi les articles respectant les conditions suivantes : avoir une longueur comprise entre 30 et 50 phrases, être compréhensibles par des non-experts et ne pas inclure d'illustrations nécessaires à la compréhension du texte. Comme le montre le tableau 1, les textes des quatre sources comptent en moyenne le même nombre de phrases, mais les textes destinés à un public adulte contiennent plus de mots parce que les phrases y sont plus longues.

[2]Le coefficient alpha est une mesure de la consistance interne d'un ensemble de juges. Une manière particulièrement illustrative de le présenter est de considérer qu'il correspond à la moyenne de toutes les corrélations qui peuvent être calculées en divisant aléatoirement les juges en deux groupes et en corrélant l'indice de segmentation qui est obtenu, comme illustré à la figure 2, pour chaque groupe de juges (méthode de bissection ou *split-half correlation*).

	Journaux		Encyclopédies	
	Jeune	Adulte	Jeune	Adulte
Phrases				
Moyenne	36.8	37.3	36.6	36.8
Écart-type	4.0	5.7	7.2	6.6
Mots				
Moyenne	587	1025	847	1165
Écart-type	52	214	169	257

TAB. 1 – *Longueur moyenne des textes en phrases et en mots.*

Il est à noter que segmenter un texte en fonction des thèmes abordés est une tâche nettement plus complexe, tant pour les juges que pour les algorithmes, que de segmenter une série d'extraits de textes différents concaténés. C'est d'autant plus le cas dans la présente expérimentation en raison de la brièveté des textes, certains faisant à peine trente phrases. Leur lecture met cependant en évidence la présence de différents thèmes. Par exemple, l'article du *Le Petit Ligueur "Etre esclave encore aujourd'hui"* introduit d'abord la problématique en général avant d'aborder l'esclavage domestique, la servitude pour dette et la traite des êtres humains pour conclure sur les raisons de la persistance de ces pratiques.

Pour la tâche de jugement, ces textes ont été mis en forme de manière à ce que chaque phrase soit séparée de la suivante par un retour à la ligne et qu'aucun indice typographique ne signale la présence d'un alinéa dans le texte original. Le titre du texte a été conservé et placé dans un cadre afin de le distinguer du texte à segmenter. Les 32 textes ont été répartis en huit carnets de quatre textes (un texte de chaque source) de manière à ce qu'au total chaque carnet contienne un nombre comparable de phrases. Pour chaque carnet, deux ordres de présentation des textes ont été établis, l'un étant l'inverse de l'autre.

Cent quatre participants (13 par carnet) ont pris part à cette recherche. Ils étaient tous étudiants en Bac 2 à l'Université catholique de Louvain (âgé de plus ou moins 20 ans) et participaient à cette étude dans le cadre de travaux pratiques. Il leur était demandé de segmenter les textes en fonction des changements de thèmes qu'ils percevaient en traçant des lignes entre les phrases. Ils n'ont pas reçu d'indication quant au nombre minimal ou maximal de segments à marquer. L'emploi d'instructions très générales et l'absence d'entraînement correspondent à la procédure habituellement employée dans ce domaine de recherche (Hearst, 1997 ; Passoneau et Litman, 1997 ; Sporleder et Lapata, 2006).

3 Analyse de la fiabilité des jugements

En moyenne, les juges ont segmenté toutes les 5.42 phrases, soit un peu moins de sept fois par texte. On observe des différences selon la source du texte, les juges segmentant en moyenne toutes les 6.15 phrases du *Petit Ligueur* alors qu'ils segmentent toutes les 4.87 phrases de l'*Encyclopédie Universalis*, les textes des deux autres sources étant segmentés approximativement toutes les 5.35 phrases. Pour mesurer la fiabilité des juges, la mesure WindowDiff a été employé conformément à la recommandation de Artstein et Poesio (2008). Afin de disposer d'une vue générale des résultats des diverses analyses, le tableau 2 reprend l'ensemble de ceux-ci.

3.1 Estimation de la fiabilité moyenne d'un juge par WindowDiff

Une première procédure pour estimer la fiabilité moyenne d'un juge au moyen de WindowDiff consiste à calculer la moyenne de toutes les valeurs de cet indice que l'on peut obtenir en comparant les segmentations définies par toutes les paires possibles de juges. Cette procédure est celle qui est classiquement appliquée pour calculer par exemple le kappa moyen pour n juges. Il faut cependant noter que, contrairement au kappa pour lequel les deux juges comparés ont le même statut, WindowDiff impose de considérer un des juges comme fournissant la segmentation de référence et l'autre la segmentation hypothétique. En effet, Pevzner et Hearst (2002), à la suite de Beeferman et al. (1999), donne comme valeur au paramètre k la moitié de la longueur moyenne d'un segment selon la segmentation de référence. Si les deux juges n'ont pas segmenté le texte un même nombre de fois, la valeur du paramètre k sera différente suivant le juge qui est choisi comme référence, ce qui affectera le score WindowDiff. On notera qu'un problème similaire se pose si l'on emploie les scores de précision et de rappel, mais non le F-score (lorsque précision et rappel y reçoivent le même poids) ou le coefficient kappa (Hripcsak et Rothschild, 2005). Cette difficulté peut être résolue de différentes manières, comme en calculant la moyenne des deux k possibles ou en calculant deux scores WindowDiff pour chaque paire de juges, de sorte que chaque juge serve une fois de référence. C'est cette seconde solution qui a été employée ici parce qu'elle permet

la mise en évidence de l'impact du facteur k sur le score WindowDiff (voir conclusion). La valeur moyenne obtenue est de 0.3914 (ligne WD-1/1 dans le tableau 2).

Pour estimer la fiabilité moyenne d'un juge, une seconde procédure peut être employée. Elle consiste à comparer chaque juge à l'ensemble des autres juges en considérant que ceux-ci ont placé conjointement une rupture lorsqu'un nombre minimal de ceux-ci l'ont placée individuellement. Comme l'ont fait Hearst (1997) et Passonneau et Litman (1997) pour décider des ruptures définies par des groupes de juges, un seuil juste inférieur à une majorité de juges, soit 6 sur 12, a été employé. Dans cette seconde procédure, c'est évidemment l'ensemble des autres juges qui sert à chaque fois de référence ; un seul score WindowDiff doit donc être calculé par juge et par texte. La valeur moyenne obtenue (0.3867 — ligne WD-1/12 dans le tableau 2) est très proche de celle produite par la première procédure.

Indices	Scores WindowDiff
WD-1/1	0.39
Fiabilité moyenne d'un juge (comparaison de pairs de juges)	
WD-1/12	0.39
Fiabilité moyenne d'un juge (comparaison de chaque juge aux 12 autres)	
WD-7/6	0.28
Fiabilité effective de groupes de 6 juges	
WD-C99	0.45
Performance de C99 pour la segmentation interne à un texte	
WD-Base-1	0.61
Performance d'une procédure qui place aléatoirement le nombre adéquat de segments	
WD-Base-2	0.51
Performance d'une procédure qui place un nouveau segment toutes les n phrases	
WD-Base-3	0.46
Performance d'une procédure qui ne signale aucun segment	

TAB. 2 – *Mesures d'erreurs WindowDiff pour toutes les analyses.*

3.2 Estimation de la fiabilité effective par WindowDiff

La seconde approche décrite ci-dessus peut être utilisée pour estimer au moyen de WindowDiff la fiabilité de l'indice global qui peut être dérivé de plusieurs juges en comparant les segments définis par un groupe de juges à ceux définis par un autre groupe de juges. Concrètement, ceci est effectué de la manière suivante :

1. diviser les 13 juges qui ont segmenté un texte donné en deux groupes, l'un de 7 et l'autre de 6 juges ;
2. considérer que chaque groupe a conjointement placé une rupture entre deux phrases si au moins 3 d'entre eux l'ont fait individuellement ;
3. comparer au moyen de WindowDiff ces deux segmentations.

Cette procédure rencontre deux problèmes. Le premier découle de ce qu'il existe de nombreuses manières de diviser 13 juges en deux groupes de 7 et 6 juges et qu'il est difficile de se contenter d'une seule division, même choisie aléatoirement. Il est donc souhaitable de procéder, pour chaque texte, soit à un grand nombre de tirages aléatoires, soit à l'ensemble des divisions possibles et de calculer la moyenne des scores WindowDiff ainsi obtenus. La seconde option a été employée ici. De nouveau, deux scores WindowDiff ont été calculés selon le groupe de juges qui est employé comme référence. La valeur obtenue est de 0.2777 (WD-7/6 dans le tableau 2)[3]. Comme attendu (voir la section 2), cette mesure de la fiabilité effective d'un groupe de six juges est nettement supérieure à la fiabilité moyenne d'un juge.

Le second problème est que le score WindowDiff ainsi obtenu est valable pour des décisions conjointes prises par la moitié des juges et non pour celles qui peuvent être dérivées de l'ensemble des juges. Il s'agit donc d'une sous-estimation de sa valeur réelle dans le même sens qu'un coefficient de fiabilité obtenu par la méthode de bissection (split-half), qui procède en divisant un test en deux parts égales et en corrélant les scores à chacune de ces deux moitiés, est une sous-estimation de la fiabilité du test entier. Comme la méthode de bissection a été développée dans le cadre de la théorie classique des scores, une formule a pu être proposée pour calculer la fiabilité du test entier sur la base du coefficient de

[3]En raison des effectifs inégaux des deux groupes de juges, la même analyse a été effectuée en ne prenant en compte à chaque fois que 12 des 13 juges ; le score WindowDiff obtenu est de 0.2834.

fiabilité obtenu pour les deux moitiés (formule de Spearman-Brown, Rosenthal, 1982 par exemple). WindowDiff ayant été développé en dehors de toute théorie de ce genre, on ne dispose pas d'une telle formule. Une solution à ce problème consiste à demander à deux fois plus de juges (*2N*) de segmenter un échantillon aléatoire de textes (par exemple en ajoutant aux textes propres à chaque groupe de juges un texte communs avec un autre groupe). Cette méthode n'est, en fait, rien d'autre que l'approche classique pour évaluer la fiabilité d'un juge lorsqu'on s'attend a priori à ce que celle-ci soit élevée (voir section 1.4).

3.3 Analyse complémentaire : impact de la source des textes sur la fiabilité

Afin de déterminer si le genre des textes et le public auquel ceux-ci sont destinés affectent le degré d'accord entre les juges, les moyennes des différents scores WindowDiff calculés par texte ont été analysées par une ANOVA 2 * 2. On observe un effet du public statistiquement significatif à p=0.05 pour la fiabilité moyenne d'un juge estimée par la première méthode (WD-1/1 : F(1,28)=5.05 ; p=0.0264) et un effet presque significatif de ce même facteur pour la deuxième méthode (WD-1/12 : F(1,28)=5.85 ; p=0.0596).Par contre, les différences ne sont pas statistiquement significatives pour la fiabilité effective. Ces résultats indiquent que les textes destinés à un public jeune semblent donner lieu à un accord inter-juges légèrement meilleur. Les différences sont toutefois assez faibles et la variance entre les différents textes d'une même source assez élevée. Contrairement à mon hypothèse, aucune différence statistiquement significative liée au genre de textes n'est observée. On notera néanmoins que, comme attendu, les articles du journal *Le Monde* donnent lieu aux taux d'erreurs les plus élevés.

	Journaux		Encyclopédies	
	Jeune	Adulte	Jeune	Adulte
WD-1/1				
Moyenne	0.37	0.42	0.37	0.41
Écart-type	0.06	0.05	0.06	0.04
WD-1/12				
Moyenne	0.33	0.44	0.37	0.40
Écart-type	0.09	0.11	0.10	0.11
WD-7/6				
Moyenne	0.27	0.33	0.24	0.27
Écart-type	0.06	0.09	0.11	0.06

TAB. 3 – *Mesures d'erreurs WindowDiff par type de source*

4 Évaluation d'un algorithme de segmentation

Le matériel de test a été employé pour évaluer l'efficacité d'un algorithme de segmentation thématique basé sur la cohésion lexicale : C99 développé par Choi (2000). Cet algorithme est fréquemment utilisé comme point de repère pour évaluer de nouveaux algorithmes sur la base de matériels composés de textes concaténés. Il présente les trois étapes classiques des procédures de segmentation basées sur la cohésion lexicale. Lors de la première étape, le document à segmenter est divisé en unités textuelles minimales correspondant aux phrases. La seconde étape consiste en l'estimation des similarités entre les unités minimales basée sur l'indice du cosinus entre des vecteurs de mots, calculé pour toutes les paires de phrases, qu'elles soient ou non contiguës. Enfin, la segmentation proprement dite est effectuée au moyen d'une procédure d'analyse en grappes (clustering) qui segmente répétitivement le document selon les frontières entre les unités minimales maximisant la similarité moyenne à l'intérieur des segments ainsi constitués.

Avant toute analyse, le matériel a été lemmatisé au moyen du programme TreeTagger de Schmid (1994) et un ensemble de mots fonctionnels ou très fréquents ont été supprimés. Lemmatiser le matériel est une procédure classique en segmentation automatique de textes parce que cela permet d'accroître la proportion de mots identiques.

C99 a été soumis à deux tests. Le premier consiste à identifier le début de chaque article lorsque ceux-ci sont placés les uns à la suite des autres dans un ordre aléatoire. Le second test consiste à identifier les segments marqués par au moins 6 juges sur 13. Afin de simplifier les analyses, le nombre de segments à identifier était indiqué à l'algorithme. Les paramètres nécessaires au fonctionnement de l'algorithme ont été fixés en fonction des indications données par son auteur (Choi, 2000).

En ce qui concerne l'identification des frontières des articles concaténés, C99 produit un score WindowDiff de 0.06. Cette performance, pratiquement parfaite, résulte probablement de la nature très diversifiée des textes composant le matériel.

Pour l'identification des segments délimités par les juges, WindowDiff vaut 0.45 (WD-C99 dans le tableau 2). Pour jauger cette performance, elle a été comparée à différents niveaux de base et aux limites supérieures basées sur les degrés d'accord entre les juges obtenus à la section précédente. La performance de C99 est nettement meilleure que celle obtenue par une procédure qui introduit en des points choisis aléatoirement le nombre attendu de ruptures dans un texte (WindowDiff moyen pour 5000 tirages aléatoires = 0.61 — WD-Base-1 dans le tableau 2). Elle est cependant à peine meilleure que deux autres niveaux de base (Sporleder and Lapata, 2006) : un premier obtenu en plaçant un nouveau segment toutes les n phrases, n étant choisi pour obtenir le nombre de segments selon la segmentation de référence (WindowDiff = 0.51 — WD-Base-2) et le second qui n'introduit aucune rupture thématique, plaçant donc toutes les phrases dans la catégorie la plus fréquente (WindowDiff = 0.46 — WD-Base-3).

Le tableau 2 permet de comparer la performance de C99 à différentes limites supérieures basées sur les degrés d'accord entre les juges. Il permet aussi de comparer ces mêmes degrés d'accord entre les juges au niveau de base. C99 est moins efficace que toutes les limites supérieures calculées. Comparée à la fiabilité effective de groupes de 6 juges, sa performance est très faible. Par contre, les juges sont systématiquement plus performants que tous les niveaux de base.

5 Discussion et conclusion

L'objectif principal de cette recherche est d'étudier les problèmes qui se posent lors du recours à des juges pour déterminer la norme de référence nécessaire à l'évaluation de procédures de segmentation thématique. Il s'agissait tout particulièrement d'évaluer la possibilité d'utiliser la mesure WindowDiff pour estimer la fiabilité de juges en distinguant la fiabilité d'un juge de celle d'un ensemble de juges qui ont effectué la même tâche. Pour ce faire, 32 textes ont été segmentés par 8 groupes de 13 juges. Les analyses confirment que les juges considérés individuellement présentent un degré de désaccord important. En effet, si l'on interprète WindowDiff comme indiquant la proportion de phrases (séparées par *k-1* phrases) entre lesquelles deux procédures placent un nombre différent de segments, on observe qu'en moyenne 39% des phrases répondent à cette condition. Lorsqu'on analyse la segmentation effectuée par des groupes de 6 juges, le degré d'accord est nettement plus élevé puisqu'on observe qu'en moyenne 28% des phrases ne sont pas séparées par un même nombre de segments. C99, un algorithme de segmentation fréquemment employé comme système de référence, obtient un résultat très faible puisqu'à peine supérieur de 1% à celui d'un niveau de base basé sur une procédure n'introduisant aucune segmentation dans les textes.

Dans la suite de cette section, les différents avantages et limitations de l'emploi de WindowDiff dans le cadre du développement d'une norme de référence en segmentation thématique sont discutés. L'avantage principal de WindowDiff est qu'il n'est autre que l'indice employé pour évaluer les performances des procédures de segmentation automatique. Si l'on considère, comme cela est le cas dans ce secteur, qu'il est nécessaire de pénaliser différemment les erreurs légères et les erreurs graves, on ne voit pas comment on pourrait faire l'impasse sur l'utilisation d'une telle mesure. De plus, employer WindowDiff pour évaluer tant les juges que la procédure automatique permet de disposer de limites supérieures d'efficacité auxquelles comparer les procédures automatiques. Toutefois, WindowDiff présente plusieurs limitations qui sont autant de pistes pour de futures recherches. La première, mentionnée par Artstein et Poesio (2008), est que, contrairement aux indices plus classiques de fiabilité comme kappa, WindowDiff n'est pas corrigé pour l'effet du hasard. On notera cependant que les mesures classiques de performances comme la précision et le rappel ne sont pas plus corrigées pour ce facteur et que l'emploi de niveaux de base (bien choisis) permet de contourner partiellement ce problème. Toujours en comparaison avec kappa, WindowDiff présente une deuxième limitation : l'absence de normes généralement acceptées pour décider si la valeur obtenue est satisfaisante. Il s'agit ici d'un problème très général en évaluation qui se pose pour de nombreux indices. Une difficulté supplémentaire pour WindowDiff est que la signification de ce taux d'erreur est plus difficile à comprendre que, par exemple, celle de la précision ou du rappel.

Une troisième limite de WindowDiff, qu'il a hérité de l'indice Pk (Beeferman et al., 1999), est sa dépendance au paramètre k. Celui-ci définit la taille de la fenêtre dans laquelle on enregistre les désaccords entre les deux segmentations analysées. Ce problème a été signalé à la section 3. Logiquement, plus cette fenêtre est petite, moins nombreux seront les désaccords. L'impact de ce paramètre sur les résultats peut être mis en évidence par une analyse complémentaire de certaines données recueillies dans la tâche de segmentation. Si on reprend, par exemple, l'estimation de la fiabilité moyenne d'un juge basée sur la comparaison de toutes les paires possibles de juges (ligne WD-1/1 dans le tableau 2) et qu'on décide de considérer comme segmentation de référence celle fournie par le juge qui segmente le moins souvent des deux (ce qui correspond au plus grand des deux k possibles), on obtient un WindowDiff de 0.4513 (2196 cas). Si par contre, on considère comme référence le juge qui a segmenté le plus souvent (le plus petit k possible), WindowDiff vaut 0.3530 (2196 cas). Enfin, il y a 600 cas dans lesquels les deux juges ont segmenté un même nombre de fois, donnant lieu

à un WindowDiff de 0.31, la moyenne, pondérée par le nombre de cas, de ces trois valeurs étant rapportée dans le tableau 2[4]. Ceci implique que les performances estimées au moyen de WindowDiff auront tendance à s'améliorer à mesure que les segments selon la norme de référence seront de plus petites tailles. Cette observation a un impact, par exemple, sur le choix du nombre de juges devant avoir segmenté en un même point pour décider de la présence d'une frontière selon la norme de référence. Plus ce seuil est bas, plus *k* sera petit, meilleur sera en moyenne WindowDiff.

Pour conclure, il paraît nécessaire de reconsidérer la question initiale de la procédure à employer pour évaluer des algorithmes de segmentation thématique. La comparaison des performances de C99 lorsqu'il doit trouver les frontières des articles (WD=0.06) et lorsqu'il doit trouver les frontières entre des segments marqués par des juges (WD=0.45) indique très nettement que les conclusions tirées du premier test ne peuvent pas être généralisées au second. Lorsque l'objectif de la procédure est de segmenter un texte en fonction des thèmes qu'il aborde, le recours à des juges se justifie pour autant qu'un nombre suffisant de juges segmentent chaque texte. Il faut reconnaître qu'une limitation de la présente recherche est qu'elle n'apporte pas de réponse à la question de savoir ce qu'il faut entendre par un nombre suffisant de juges. Cette limitation découle de l'absence d'un seuil permettant de décider quand un score WindowDiff est suffisamment bon. Ce problème affecte d'autres indices d'efficacité et de fiabilité comme l'ont encore montré récemment Reidsma et Carletta (2008) par la remise en cause des seuils pourtant bien établis pour le kappa.

Remerciements

Yves Bestgen est chercheur qualifié du F.R.S.—FNRS. Il tient à remercier Sophie Piérard pour son aide lors du recueil des données de segmentation.

Références

Allan J., J. Carbonell, G. Doddington, J. Yamron and Y. Yang (1998). Topic detection and tracking pilot study. Final report. *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*.

Artstein, R. and M. Poesio (2008). Inter-Coder agreement for computational linguistics. *Computational Linguistics 34*, 555–596.

Beeferman, D., A. Berger and J. Lafferty (1999). Statistical models for text segmentation. *Machine Learning 34*, 177–210.

Bellot, P. et M. El-Bèze (2001). Classification et segmentation de textes par arbres de décision. *Application à la recherche documentaire. Technique et science informatiques 20*, 107–134.

Bestgen, Y. (2006). Improving text segmentation using Latent Semantic Analysis : A Reanalysis of Choi, Wiemer-Hastings and Moore (2001). *Computational Linguistics 32*, 5–12.

Bestgen, Y. et S. Piérard (2006). Comment évaluer les algorithmes de segmentation automatique ? Essai de construction d'un matériel de référence. *Actes de TALN 2006 : Verbum ex machina*, 407–414.

Brants, T., F. Chen and I. Tsochantaridis (2002). Topic-based document segmentation with probabilistic latent semantic analysis. *Proceedings of CIKM'02*, 211–218.

Carletta, J. (1996). Assessing agreement on classification tasks : The Kappa statistic. *Computational Linguistics 22*, 249–254.

Choi, F. (2000). Advances in domain independent linear text segmentation, *Proceedings of NAACL-00*, 26–33.

Ferret, O. (2002). Using collocations for topic segmentation and link detection. *Proceedings of COLING 2002*, 260–266.

Ferret, O. (2006). Approches endogène et exogène pour améliorer la segmentation thématique de documents. *Traitement Automatique des Langues 47*, 111–135.

Fragnou, P., V. Petridis and A. Kehagias (2004). A dynamic programming algorithm for linear text segmentation. *Journal of Intelligent Information Systems 23*, 179–197.

Hearst, M. (1997). TextTiling : Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics 23*, 33–64.

Hripcsak, G. and A.S. Rothschild (2005). Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association 12*, 296–298.

[4]Il est à noter que ce facteur n'affecte pas négativement l'évaluation de l'efficacité de C99 puisque le nombre adéquat de segments à identifier lui était fourni.

Kozima, H. (1993). Text segmentation based on similarity between words. *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, 286–288.

Laveault, D. et J. Grégoire (1997). *Introduction aux théories des tests en sciences humaines*. De Boeck.

Passoneau, R., and D. Litman (1997). Discourse segmentation by human and automated means. *Computational Linguistics 23*, 103–139.

Pevzner, L. and M. Hearst (2002). A Critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics 28*, 19–36.

Ponte, J. and W. Croft (1997). Text segmentation by topic. *Proceedings of the 1st European Conference on Research and Advanced Technology for Digital Libraries*, 120–129.

Prince, V., and A. Labadié (2007). Text segmentation based on document understanding for information retrieval. *Proceedings of NLDB 2007*, 295–304.

Stemler, S. E. (2004). A comparison of consensus, consistency, and measurement approaches to estimating interrater reliability. *Practical Assessment, Research and Evaluation 9*, 1–19.

Reidsma, D. and J. Carletta (2008). Reliability measurement without limits. *Computational Linguistics 34*, 319–326.

Rosenthal, R. (1982). Conducting judgement studies. In K.R. Scherer and P. Ekman (Eds.), *Handbook of methods in nonverbal behavior research* (pp. 287–361). Cambridge : Cambridge University Press.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. *Proceedings of the International Conference on New Methods in Language Processing*.

Shrout, P.E. and J.L. Fleiss (1979). Intraclass correlations : Uses in assessing reliability. *Psychological Bulletin 86*, 420–428.

Sporleder, C. and M. Lapata (2006). Broad coverage paragraph segmentation across languages and domains. *ACM Transactions on Speech and Language Processing 3*, 1–35.

Utiyama, M. and H. Isahara (2001). A Statistical model for domain-independent text segmentation. *Proceedings of ACL'2001*, 491–498.

Summary

The objective of this research is to study the difficulties which arise when judgment are used to determine the gold standard necessary to evaluate thematic segmentation procedures. It specifically questions the use of the WindowDiff metric to estimate the average reliability of a judge and the effective reliability the reliability of an index that can be obtain from several judges). With this aim, 32 texts were segmented by groups of 13 judges. The performances of the judges are compared with a segmentation algorithm frequently used as a benchmark. In the conclusion the advantages and limits of the WindowDiff index are discussed.

Description d'un protocole d'évaluation automatique comme alternative à l'évaluation humaine.

Application à la validation de relations syntaxiques induites

Nicolas Béchet*

*LIRMM - UMR 5506, CNRS - Univ. Montpellier 2 - 34392 Montpellier Cedex 5 - France
bechet@lirmm.fr

Résumé. Cet article propose un protocole d'évaluation afin de valider la qualité d'approches, visant à évaluer et à ordonner automatiquement des relations syntaxiques dites induites. Les approches évaluées se fondent sur l'interrogation d'un moteur de recherche sur le Web. Les résultats du moteur de recherche sont alors couplés avec diverses mesures statistiques : l'information mutuelle, l'information mutuelle au cube, le coefficient de Dice et la fréquence, ou popularité.
Le protocole d'évaluation propose d'utiliser deux corpus, le premier de test et le second de validation, appartenant tous deux au même domaine. Le principe est de retrouver dans le second corpus les relations syntaxiques induites, non présentes originalement dans le premier corpus. Il est alors étudié la taille minimale du corpus d'évaluation afin de permettre une évaluation pertinente.

1 Introduction

Le domaine du traitement automatique des langues (TAL) regroupe de nombreux champs de recherche, pouvant rendre difficile la tâche d'évaluation, de par leurs diversités. Les ouvrages de Dale et al. (2000) et Mitkov (2003) établissent un survol de ces différents champs de recherche parmi lesquelles l'acquisition d'informations lexicales, l'analyse syntaxique, l'extraction d'informations, la traduction automatique, le résumé automatique ou encore des systèmes d'aide à la rédaction. Pospescu-Belis (2007) propose de classer ces sous domaines dans quatre catégories : les systèmes d'analyse ou d'annotation (1), les systèmes de génération ou de synthèse (2), les systèmes combinant l'analyse et la génération (3) et enfin des systèmes interactifs (4), faisant intervenir un (ou plusieurs) utilisateur(s) humain(s). Ces catégories se fondent sur les entrées sorties linguistiques des systèmes de TAL ainsi que sur l'interaction humaine (ou non). Notons qu'une majorité des champs de recherche du TAL peut être classée dans la première catégorie impliquant du contenu linguistique en entrée du système. L'évaluation de tels systèmes consiste le plus souvent à se référer à un (ou plusieurs) intervenant(s) humain(s). Celui-ci se voit attribuer la même tâche que le système TAL. Alors sont utilisées des métriques d'évaluation comme le coefficient *kappa* (Cohen (1960)), permettant de confondre les résultats de juges humains avec ceux du modèle. D'autres métriques assez largement utilisées en TAL sont le rappel et la précision (Salton et Mcgill (1986)), ainsi que leur moyenne harmonique, la f-mesure. Ces mesures permettent, parmi un ensemble de candidats, d'identifier les pertinents. Ces mesures ne sont pas toujours adaptées à une tâche comme l'évaluation de la qualité du classement d'un ensemble de candidat, auxquelles on préférera par exemple l'utilisation des courbes ROC.
Cet article présente un protocole d'évaluation, permettant de valider de manière automatique la qualité d'approches, afin de se passer d'une évaluation humaine. Ces approches permettent d'ordonner des relations syntaxiques dites induites. De telles relations ne sont pas présentes initialement dans un corpus et mais sont acquises à partir de celui-ci. Nous proposons dans un premier temps de définir les relations syntaxiques induites en montrant de quelles manières elles sont acquises, et pour quelles applications elles peuvent être utiles (section 2). Après avoir montré pourquoi il était nécessaire d'effectuer une validation qualitative de ces relations, nous décrierons les différents processus de validations utilisés (section 3). Nous obtenons alors en sortie d'une validation, une liste ordonnée de relations syntaxiques par qualité. Une manière de mesurer la qualité du classement proposé serait alors de faire valider ce classement par un expert. Une telle validation serait très longue et fastidieuse. Il est de plus difficile de trouver des experts d'un domaine, acceptant d'effectuer ce type d'évaluation. Il est alors proposé un protocole d'évaluation automatique se fondant sur l'utilisation de deux corpus d'un même domaine (4). Ce protocole va être ensuite discuté en étant notamment comparé à une validation manuelle triviale.

2 Les relations syntaxiques induites

2.1 Acquisition des relations syntaxiques induites

La génération de relations syntaxiques induites à partir d'un corpus requière une extraction des relations syntaxiques classiques. Ainsi, l'analyseur SYGFRAN (Chauché (1984)) est utilisé afin d'extraire les relations classiques d'un corpus (dans cet article, nous nous intéresserons uniquement aux relations syntaxiques Verbe-Objet). La proximité sémantique des verbes extrais est ensuite étudiée avec la mesure d'Asium (Faure (2000)). Cette mesure, dont la formule est donnée ci-dessous, considère comme proche deux verbes possédant un certain nombre d'objets en commun.
Soit p et q, deux verbes avec leurs objets respectifs $p_1, ..., p_n$ et $q_1, ..., q_n$ illustrés sur la figure 1. $NbOccCom_p(q_i)$

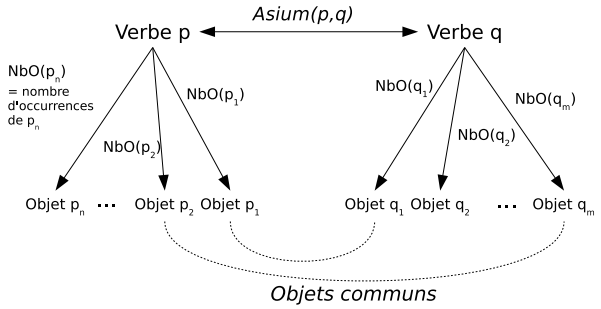

FIG. 1 – *Mesure d'Asium entre les verbes p et q*

représente le nombre d'occurrences des objets q_i en relation avec le verbe q qui sont aussi des objets du verbe p (objets communs). $NbOcc(q_i)$ représente le nombre d'occurrences des objets q_i. La mesure d'Asium est alors définie de la manière suivante :

$$Asium(p,q) = \frac{log_{Asium}(\sum NbOccCom_q(p_i)) + log_{Asium}(\sum NbOccCom_p(q_i))}{log_{Asium}(\sum NbOcc(p_i)) + log_{Asium}(\sum NbOcc(q_i))}$$

Avec $log_{Asium}(x)$ valant :
- pour $x = 0$, $log_{Asium}(x) = 0$
- sinon $log_{Asium}(x) = log(x) + 1$

Un score proche de 1 obtenu avec la mesure d'Asium implique une importante proximité sémantique.
Une fois l'ensemble de la proximité des verbes du corpus mesuré, le rassemblement des objets jugés proches est alors effectué. Dès lors, deux types d'objets peuvent être utilisés afin de définir une relation syntaxique : les objets **communs**, objets étant originalement présents dans le corpus (*argent* et *vêtement* sur la figure 2), et les objets **complémentaires**, objets induits de relations syntaxiques existantes décrits dans Faure et Nédellec (1998); Béchet et al. (2009) (objet *avertissement* pour le verbe *offrir* et *cadeau* pour le verbe *donner* sur la figure 2). La relation syntaxique ainsi formée avec un objet complémentaire (*former bateau* sur la figure 2) est alors appelée une **relation syntaxique induite** par opposition à une **relation syntaxique classique**.

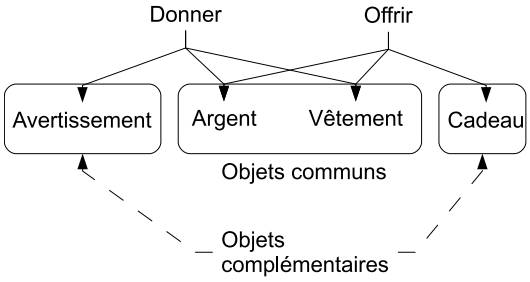

FIG. 2 – *Objets communs et complémentaires des verbes "Donner" et "Offrir".*

2.2 Pourquoi utiliser des relations syntaxiques induites ?

Les relations syntaxiques classiques peuvent être utilisées afin d'enrichir un corpus comme avec l'approche ExpLSA (Béchet et al. (2008)). Celle-ci propose d'enrichir un corpus afin d'améliorer les résultats de classification automatique de textes. Après l'extraction des relations syntaxiques induites et classiques, le principe est de compléter les termes du corpus par les objets provenant des relations syntaxiques (classiques ou induites). En effet, nous pouvons enrichir le corpus avec uniquement les objets communs, ou alors en utilisant les communs et les complémentaires. L'intérêt des relations syntaxiques induites, réside dans le fait qu'elles ne sont pas originalement présentes dans le corpus. Ainsi, le corpus enrichi avec ExpLSA se voit apporter des informations nouvelles pouvant permettre de faciliter une tâche de classification ou de clustering.

Outre l'expansion de corpus, les relations syntaxiques induites peuvent également contribuer à la représentation d'un ensemble de connaissances d'un domaine donné. Montrons par exemple comment des relations syntaxiques induites peuvent permettre de contribuer à l'acquisition ou l'enrichissement d'ontologies qui peuvent être définies comme la représentation de connaissances d'un domaine sous forme de concepts hiérarchisés. Un réseau de dépendance se fondant sur des ressources syntaxiques partagées, notamment des relations de dépendance syntaxique autours des verbes (dans notre cas les relations Verbe-Objet) peuvent servir de support dans un processus d'acquisition ontologique comme dans Bourigault (2002). Ainsi, un concept peut être constitué d'objets provenant de relations syntaxiques Verbe-Objet. Les relations induites peuvent alors apporter une information supplémentaire, impossible à acquérir avec des relations syntaxiques classiques. En considérant les objets de la figure 2, nous pourrions constituer le concept comme suit :
- Avec les objets communs (*argent* et *vêtement*)
- Avec les objets communs et l'objet complémentaire du premier verbe (*argent*, *vêtement* et *cadeau*)
- Avec les objets communs et l'objet complémentaire du second verbe (*argent*, *vêtement* et *avertissement*)
- Avec les objets communs et les objets complémentaires (*argent*, *vêtement* et *avertissement*, *cadeau*)

3 Validation des relations syntaxiques induites par le Web

La qualité des relations syntaxiques induites extraient à partir d'un corpus tel que présenté dans la section précédente est discutable. En effet, si deux verbes partagent un même objet et que l'on considère notre corpus comme bien écrit, il va de soit que les relations syntaxiques produites avec chacun des verbes sont cohérentes, car elles proviennent directement du corpus. Néanmoins, l'idée d'une relation syntaxique induite est qu'elle n'existe pas originalement dans le corpus. Alors, rien ne garantit la cohérence d'une telle relation syntaxique, même si les deux verbes partagent par ailleurs un nombre important d'objets communs. Prenons par exemple la relation *offrir avertissement* issue de la figure 2. Celle-ci n'est manifestent pas cohérente contrairement à l'autre relations induite issue de cette même figure (*donner cadeau*).
Il apparaît alors nécessaire de mesurer la cohérence des relations syntaxiques induites. On pourra par exemple utiliser les relations syntaxiques validées afin de créer des classes conceptuelles, ou ontologies en ne sélectionnant que les premières relations une fois ordonnées, ou utiliser les meilleures relations induites afin d'enrichir un corpus en évitant ainsi l'ajout important de bruit.
Ainsi, il est présenté dans cette section une manière de valider la cohérence des relations syntaxiques induites en se fondant sur le Web. La validation automatique des relations syntaxiques propose de mesurer la dépendance entre verbe et objet d'une relation induite. Il est employé pour cela un moteur de recherche en utilisant une API (*http ://api.search.yahoo.com*). Une requête est ainsi soumise au moteur de recherche. Des mesures statistiques sont finalement employées afin de proposer un classement des relations syntaxiques. Diverses définitions préalables sont nécessaires afin de permettre d'adapter les mesures statistiques à l'étude de la cohérence des relations syntaxiques.

3.1 Définitions

Nous définissons tout d'abord, la fonction $nb(X)$, comme étant le nombre de pages retournées par le moteur de recherche en réponse à la requête X. Définissons également o et v comme étant respectivement l'objet et le verbe évalué. Ainsi, $nb(o)$ va retourner le nombre de pages trouvées pour l'objet o, ceci reflétant la popularité de l'objet o sur le Web. Définissons enfin $nb_{max}(v, o)$ et $nb_{sum}(v, o)$ comme suit :

$nb_{max}(v, o) = max(nb(v$ **un** $o), nb(v$ **une** $o), nb(v$ **le** $o), nb(v$ **la** $o), nb(v$ **l'** $o))$
et
$nb_{sum}(v, o) = nb(v$ **un** $o) + nb(v$ **une** $o) + nb(v$ **le** $o) + nb(v$ **la** $o) + nb(v$ **l'** $o)$

avec $nb(v$ **un** $o)$ qui est le nombre de pages retournées par le moteur de recherche Yahoo pour la relation syntaxique 'v un o". Les différentes mesures statistiques utilisées pour ordonner les relations syntaxiques sont présentées ci-dessous.

3.2 L'information mutuelle

Une des mesures les plus couramment utilisées en recherche d'information afin d'établir un classement est l'Information Mutuelle (IM) (Church et Hanks (1990)) définie comme suit :

$$IM(x,y) \quad = \quad \log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1}$$

$P(x,y)$ peut alors être vu comme la probabilité des réponses retournées par le moteur de recherche Yahoo pour la relation syntaxique v, o. Cette mesure vise à faire ressortir les co-occurrences les plus rares et les plus spécifiques (Daille (1996); Thanopoulos et al. (2002)). Appliquée au contexte de la validation des relations syntaxiques induites, la formule 1 va devenir [1] :

$$IM(v,o) \quad = \quad \frac{nb(v,o)}{nb(v)nb(o)} \tag{2}$$

Avec $nb(v,o)$, étant soit $nb_{max}(v,o)$ ou bien $nb_{sum}(v,o)$ suivant que l'on utilise le max ou la somme. Par ailleurs, le log_2 a été supprimé, ne modifiant pas le rang obtenu.

3.3 L'information mutuelle au cube

L'information mutuelle au cube est une information empirique fondée sur l'information mutuelle, qui accentue l'impacte des co-occurrences fréquentes, ce qui n'est pas le cas avec l'information mutuelle originale Daille (1994). Cette mesure est définie ainsi :

$$IM^3(x,y) \quad = \quad \log_2 \frac{P(x,y)^3}{P(x)P(y)} \tag{3}$$

Qui adaptée à la mesure de la cohérence des relations syntaxiques devient :

$$IM^3(v,o) \quad = \quad \frac{nb(v,o)^3}{nb(v)nb(o)} \tag{4}$$

3.4 Le coefficient de Dice

Une mesure également intéressante en terme d'évaluation de qualité est le coefficient de Dice (Smadja et al. (1996)) défini comme suit :

$$Dice(x,y) \quad = \quad \frac{2 \times P(x,y)}{P(x) + P(y)} \tag{5}$$

Qui devient :

$$Dice(x,y) \quad = \quad \frac{2 \times nb(x,y)}{nb(x) + nb(y)} \tag{6}$$

Les différentes mesures statistiques vont permettre d'obtenir un classement des relations syntaxiques en fonction de leur pertinence sur le Web. Huit listes de relations syntaxiques triées vont alors être retournées en utilisant les approches suivantes :
- La fréquence d'utilisation de la relation syntaxique sur le Web
- L'information mutuelle
- L'information mutuelle au cube
- Le coefficient de Dice

Ces quartes mesures peuvent être obtenues en utilisant le maximum $nb_{max}(v,o)$ ou la somme $nb_{sum}(v,o)$ nous donnant ainsi les huit listes triées. Il est maintenant nécessaire d'établir un protocole d'évaluation afin de déterminer la qualité des fonctions de rangs proposées, et par la même de la validation Web.

[1] en écrivant $P(x) = \frac{nb(x)}{nb\_total}$, $P(y) = \frac{nb(y)}{nb\_total}$, $P(x,y) = \frac{nb(x,y)}{nb\_total}$ avec $x = v$ et $y = o$

4 Un protocole d'évaluation automatique

4.1 Définition du protocole

La validation manuelle semble être intuitivement la solution la plus adaptée afin de mesurer la qualité de la validation des relations syntaxiques induites par le Web. Néanmoins il est très difficile de trouver des experts afin d'effectuer une telle tâche. En effet, cela nécessiterait un travail fastidieux. Les experts se verraient proposer une quantité non négligeable de relations syntaxiques à expertiser. Il est par conséquent défini dans cet article un protocole expérimental automatique.

Le principe de ce protocole est d'utiliser deux corpus. Un premier, de test (appelé par la suite *corpus T*) duquel ont été extraites les relations syntaxiques qui vont ensuite être ordonnées par les différentes approches présentées section 3. Ce corpus contient 8 948 articles (16,5 Mo) en français et il est extrait du site Web d'informations de Yahoo (*http ://fr.news.yahoo.com/*). Il a été obtenu 60 460 relations syntaxiques induites. Un second corpus, également en français, est alors utilisé comme référence afin de valider les approches (appelé corpus V). Ce second corpus a la particularité d'être beaucoup plus important que le corpus T, contenant plus de 60 000 articles (125 Mo) issus du corpus du quotidien *Le Monde*. Il est de plus du même domaine, actualité avec un style journalistique.
Il est alors proposé de juger une relation syntaxique induite, créé à partir du corpus T, comme pertinente si celle ci est

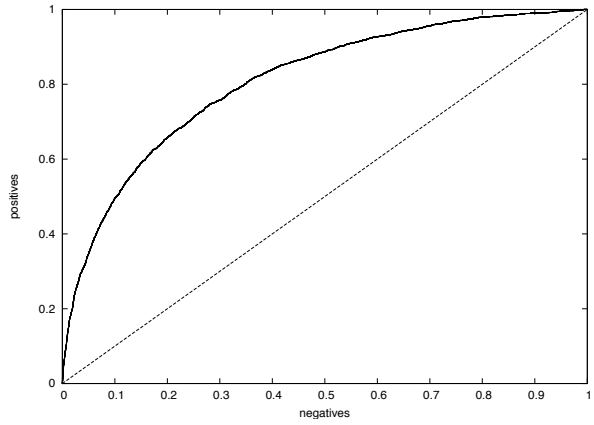

FIG. 3 – *Exemple de courbe ROC*

retrouvée dans le corpus V (comme relation syntaxique dite *classique* par opposition à *induite*). Concrètement, si une relation induite est retrouvée dans le *corpus V*, on la qualifiera de **positive**. Dans le cas contraire, elle sera jugée non pertinente et sera donc qualifiée de **négative**. L'intérêt de cette validation réside dans le fait qu'elle permette de mesurer de manière automatique la qualité des approches proposées, et ceci pour un très grand nombre de relations syntaxiques. Notons que les relations jugées négatives peuvent être de faux négatifs. En effet, une relation qui n'a pas été retrouvée dans le *corpus V* n'est pas pour autant non pertinente. Une fois la notion de positif et négatif définie, nous pouvons utiliser une approche couramment utilisée dans la littérature afin de mesurer la qualité d'une fonction de rang : les courbes ROC.

La méthode des courbes ROC (Receiver Operating Characteristic), détaillée par Ferri et al. (2002), fut utilisée à l'origine dans le domaine du traitement du signal. Cette méthode est fréquemment employée en médecine afin d'évaluer automatiquement la validité d'un diagnostique de tests. On trouve en abscisse des axes représentant une courbe ROC le taux de faux positifs (dans notre cas, le taux de relations syntaxiques induites non pertinentes, soit les relations non retrouvées dans le *corpus V*) et l'on trouve en ordonnée le taux de vrais positifs (dans notre cas les relations pertinentes, soit celles existantes dans le *corpus V*). La surface sous la courbe ROC ainsi créée est appelée AUC (*Area Under the Curve*). Un des avantages de l'utilisation des courbes ROC réside dans leur résistance à la non parité de la répartition du nombre d'exemples positifs et négatifs.
Une courbe ROC représentée par une diagonale correspond à un système où les relations syntaxiques ont une distribution aléatoire, la progression du taux de vrais positifs est accompagnée par la dégradation du taux de faux positifs. La courbe 3 est un exemple de courbe ROC, avec en diagonale, une distribution aléatoire. Considérons le cas d'une validation de relations syntaxiques induites. Si toutes les relations sont positives (ou pertinentes), l'AUC vaudrait 1, ce qui signifie avoir toutes les relations pertinentes en début de liste, donc ordonnées de manière optimale.

4.2 Résultats et Discussions

	AUC
maximum	0,813
somme	0,814
IM max	0,760
IM somme	0,763

	AUC
IM3 max	0,810
IM3 somme	0,812
Dice max	0,800
Dice Somme	0,802

FIG. 4 – *Résultats obtenus pour le corpus de Validation*

La figure 4 présente les AUC obtenues pour les différentes approches définies section 3 en utilisant le corpus de validation dans sa totalité. Il en ressort que l'utilisation de la *somme* plutôt que le *maximum* pour la fonction $nb(v, o)$ donne de meilleurs résultats, mais assez peu significatifs. Par ailleurs la *somme* obtient de meilleurs scores que toutes les autres approches confondues, elle semble donc être l'approche la plus adaptée pour ordonner de manière automatique les relations syntaxiques induites. Toutefois, ces résultats devront être confirmés avec d'autres expérimentations, permettant alors de conclure sur la meilleure approche à utiliser.

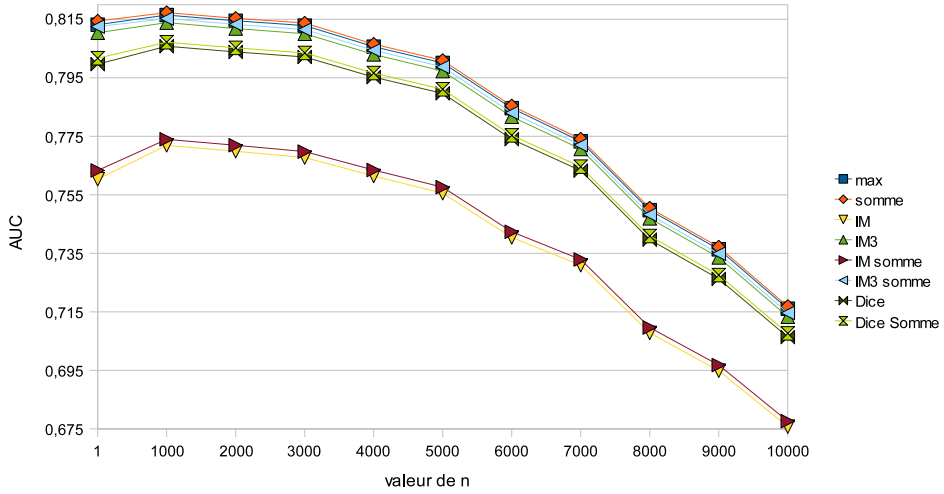

FIG. 5 – *AUC obtenues pour différentes valeurs de* n

Valeur de n	***Nb positifs***
Entier (n=1)	9080
1/1000	23,5
1/2000	11,8
1/3000	7,9
1/4000	5,9
1/5000	4,7

Valeur de n	***Nb positifs***
1/6000	3,9
1/7000	3,3
1/8000	2,9
1/9000	2,6
1/10000	2,3

FIG. 6 – *Nombre de relations couvertes en fonction de la proportion de corpus considérée*

Il est maintenant proposé d'étudier quelle doit être la taille minimum du corpus de test, afin d'évaluer correctement les approches présentées. Ainsi, le corpus de test original va être divisé en n parties. Chaque section de corpus va ensuite servir à valider les approches utilisant la validation Web, et ce en effectuant une validation croisée. Ainsi, pour un $n = 1000$, 1000 expérimentations vont être effectuées afin de calculer une AUC moyenne correspondant à un corpus d'évaluation d'une taille d'environ $60000/1000$ soit 60 articles en moyenne. Notons qu'un $n = 1$ revient à considérer la totalité du corpus de validation.

La figure 5 présente les différentes AUC obtenues pour des valeurs de n variants de 1 à 10 000, soit un corpus original divisé par 10 000. Ces résultats nous montrent que la taille du corpus peut être réduite de 5000, tout en conservant des

résultats équivalents à ceux obtenus avec le corpus de validation dans son ensemble. Pour un n variant de 1 à 5 000, nous observons en effet une variation des AUC de l'ordre de plus ou moins $0,01$ autour de l'AUC obtenue pour le corpus entier ($n = 1$), ce qui reste du même ordre. Pour une valeur de n supérieure à 5 000, les résultats se dégradent, quelque soit la mesure statistique utilisée, pour atteindre des AUC inférieures de $0,1$ points par rapport au corpus entier. Par exemple avec la *somme*, l'AUC passe de $0,81$ pour le corpus entier à $0,72$ pour un $n = 10000$. Cette baisse des AUC peut s'expliquer par une limite théorique du à un trop faible nombre de relations couvertes (nombre de relations syntaxiques induites retrouvées dans le corpus) pour des valeurs de n trop grandes. En effet, un nombre trop faible de relations couvertes reflète un manque de finesse dans les AUC résultantes.

Le tableau 6 présente le nombre de relations couvertes en fonction de la taille du corpus considéré (la taille du corpus correspond à celle du corpus de validation divisée par n). Les AUC se dégradent pour une valeur de n supérieure à 5 000. Il est constaté qu'avec moins de 4 relations syntaxiques couvertes par un corpus, les résultats donnés par le protocole utilisé sont biaisés. Néanmoins, cela signifie également qu'avec seulement 5 relations syntaxiques retrouvées dans un corpus, les AUC obtenues avec le protocole sont équivalents à ceux obtenus avec la totalité du corpus. La figure 7 confirme que

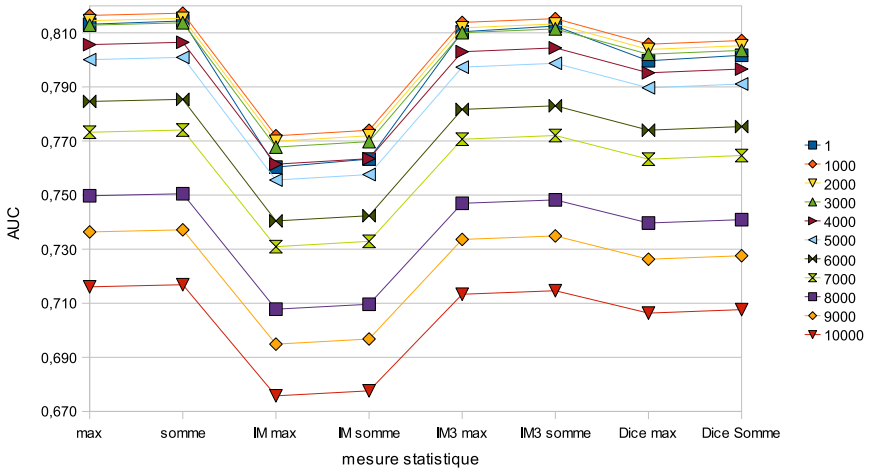

FIG. 7 – *AUC obtenues en fonction des mesures statistiques*

les AUC obtenues, pour un n compris entre 1 et 5 000 sont du même ordre. Cette figure présente les AUC en fonction de la mesure statistique utilisée. Elle permet de constater que l'allure des courbes, suivant la valeur de n, reste la même. Cela signifie que quelque soit la valeur de n (entre 1 et 10 000 ici) on retrouve le même le classement de qualité des mesures statistiques. Par exemple, la mesure *somme* reste la meilleure, quelque soit n.

Le protocole d'évaluation proposé permet donc d'utiliser un petit corpus comme corpus de validation. Il reste cependant important de mesurer la robustesse du protocole vis à vis des relations syntaxiques jugées non pertinentes. Cela correspond à des relations syntaxiques qui n'ont pas été retrouvées dans le corpus de validation. Certaines de ces relations peuvent être des faux positifs, un corpus d'un même domaine ne contenant pas nécessairement toutes les relations induites provenant d'un autre corpus. Une évaluation humaine triviale a donc été effectuée afin de mesurer la quantité de faux négatifs, permettant également de confronter une évaluation automatique à une évaluation humaine. Le protocole suivant est décrit ci-dessous.
Cent relations syntaxiques induites parmi les 60 460 issus du corpus de test préalablement triées par la validation Web avec la *somme*, sont extraites de manière homogène. Ces 100 relations ont alors été évaluées manuellement par un expert, suivant le critère : *La relation syntaxique est elle sémantiquement cohérente ?* avec seulement deux réponses possibles, oui ou non. En d'autre terme, le verbe v peut-il être rencontré avec l'objet o. Avec cette réponse binaire, et l'homogénéité de l'échantillon de relations syntaxiques testées, nous pouvons calculer une AUC moyenne correspondant à une extrapolation des résultats que l'on obtiendrait avec la totalité des relations syntaxiques induites.
Le tableau 8 présente les AUC obtenues avec cette évaluation manuelle ainsi qu'avec l'évaluation automatique. Ils sont également comparés à l'AUC obtenue avec l'ensemble des relations syntaxiques. Les AUC obtenues pour l'approche automatique avec les 100 relations et avec l'ensemble des relations syntaxiques sont très proches. Cela permet de confirmer l'homogénéité de l'échantillon sélectionné. La comparaison avec l'approche manuelle reste assez proche également avec une variation de $0,06$ points. Les faux positifs sont donc existants avec le protocole d'évaluation automatique mais restent

tolérables. Les résultats obtenus avec la validation manuelle confirme également la qualité de la validation Web avec la *somme*.

Nb de relations	100		60460
Type d'évaluation	manuelle	automatique	automatique
AUC	0,88	0,82	0,81

FIG. 8 – *Comparaison de l'évaluation manuelle et automatique*

5 Conclusion

Cet article a présenté un protocole d'évaluation automatique, permettant de se passer d'une évaluation couteuse et fastidieuse que serait une évaluation manuelle. Ce protocole propose de mesurer la qualité d'approches visant à mesurer la cohérence et à ordonner des relations syntaxiques dites induite. Les relations syntaxiques induites sont des relations qui ne sont pas originalement présentes dans un corpus, nécessitants une évaluation de leur cohérence. Le protocole d'évaluation automatique présenté se fonde sur l'utilisation d'un second corpus d'une thématique proche du corpus étudié. Une relation syntaxique induite est alors jugée positive si celle-ci est retrouvée dans le second corpus. Il est alors discuté la taille minimale ainsi que le nombre minimal de relations syntaxiques qui doivent être retrouvées dans le second corpus.
Les expérimentations menées ont permis de montrer que l'on pouvait réduire de manière considérable la taille du corpus utilisé afin de valider les relations syntaxiques (jusqu'à 5 000 fois). La validation humaine proposée a permis de constater le biais occasionné par les faux négatifs, mais ce biais reste faible. Une évaluation manuelle, plus complète, avec notamment plusieurs experts, devra néanmoins être effectuée pour confirmer ces résultats. Des corpus d'autres domaines devrons être également être traités, afin de tester le protocole et la validation de relations syntaxiques induites. Enfin, il sera proposé d'utiliser d'autres relations syntaxiques que les Verbe-Objet, dont notamment les relations Sujet-Verbe.

Références

Béchet, N., M. Roche, et J. Chauché (2008). How the ExpLSA approach impacts the document classification tasks. In *Proceedings of the International Conference on Digital Information Management, ICDIM'08*, University of East London, London, United Kingdom,.

Béchet, N., M. Roche, et J. Chauché (2009). Comment valider automatiquement des relations syntaxiques induites. In *EGC'09, à paraître*.

Bourigault, D. (2002). UPERY : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus. In *Actes de TALN, Nancy*, pp. 75–84.

Chauché, J. (1984). Un outil multidimensionnel de l'analyse du discours. In *Proceedings of Coling, Standford University, California*, pp. 11–15.

Church, K. W. et P. Hanks (1990). Word association norms, mutual information, and lexicography. In *Computational Linguistics*, Volume 16, pp. 22–29.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement 20*(1), 37.

Daille, B. (1994). *Approche mixte pour l'extraction automatique de terminologie : statistiques lexicales et filtres linguistiques*. Ph. D. thesis, Université Paris 7.

Daille, B. (1996). Study and Implementation of Combined Techniques for Automatic Extraction of Terminology. In *P. Resnik and J. Klavans (eds). The Balancing Act : Combining Symbolic and Statistical Approaches to Language, MIT Press*, pp. 49–66.

Dale, R., H. L. Somers, et H. Moisl (Eds.) (2000). *Handbook of Natural Language Processing*. New York, NY, USA : Marcel Dekker, Inc.

Faure, D. (2000). *Conception de méthode d'apprentissage symbolique et automatique pour l'acquisition de cadres de sous-catégorisation de verbes et de connaissances sémantiques à partir de textes : le système ASIUM*. Ph. D. thesis, Université Paris-Sud.

Faure, D. et C. Nédellec (1998). A corpus-based conceptual clustering method for verb frames and ontology acquisition. In P. Velardi (Ed.), *LREC workshop on Adapting lexical and corpus ressources to sublanguages and applications*, Granada Espagne, pp. 5–12.

Ferri, C., P. Flach, et J. Hernandez-Orallo (2002). Learning decision trees using the area under the ROC curve. In *Proceedings of ICML'02*, pp. 139–146.

Mitkov, R. (2003). *The Oxford Handbook of Computational Linguistics (Oxford Handbooks in Linguistics S.)*. Oxford University Press.

Pospescu-Belis, A. (2007). Le rôle des métriques d'évaluation dans le processus de recherche en tal. *TAL (Traitement Automatique des Langues) 47*(2).

Salton, G. et M. J. Mcgill (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA : McGraw-Hill, Inc.

Smadja, F., K. R. McKeown, et V. Hatzivassiloglou (1996). Translating collocations for bilingual lexicons : A statistical approach. *Computational Linguistics 22*(1), 1–38.

Thanopoulos, A., N. Fakotakis, et G. Kokkianakis (2002). Comparative Evaluation of Collocation Extraction Metrics. In *Proceedings of LREC'02*, Volume 2, pp. 620–625.

Summary

This paper presents an evaluation protocol in order to validate the quality of approaches. These approaches allow to rank automatically syntactic relations Verb-Object called induced, by querying a Web search engine. Then, some statistic measures are applied: The mutual information, the cubic mutual information, the Dice's coefficient and the frequency. Two corpora are needed to apply the evaluation protocol. The first one is a test corpus, and the second one is a validation corpus. The idea is to recover induced syntactic relations, which are not originally present in the first corpus, in the second corpus. Finally, the minimum size of the validation corpus is discussed.

S^2MP : une mesure de similarité pour les motifs séquentiels

Hassan Saneifar, Sandra Bringay, Anne Laurent, Maguelonne Teisseire

Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier (LIRMM)
161 rue Ada 34392 Montpellier Cedex 5
Université montpellier 2
{saneifar, bringay, laurent, teisseire}@lirmm.fr

Résumé. Dans le domaine de l'extraction de connaissances, comparer la similarité des objets est une tâche essentielle, par exemple pour identifier des régularités, pour construire des classes d'objets homogènes ou bien pour évaluer la proximité des motifs extraits. Ce problème est très important pour les données séquentielles présentes dans divers domaines d'application (e.g. séries d'achats de clients, navigations d'internautes). Il existe des mesures de similarité comme Edit distance et LCS adaptées aux séquences simples. Cependant elles ne sont pas pertinentes dans le cas des séquences complexes composées de séries d'ensembles, comme les motifs séquentiels. Dans cet article, nous proposons une nouvelle mesure de similarité (S^2MP - Similarity Measure for Sequential Patterns) prenant en compte les caractéristiques des motifs séquentiels. S^2MP est une mesure paramétrable en fonction de l'importance accordée à chaque caractéristique des motifs séquentiels selon le contexte d'application, ce qui n'est pas le cas des mesures existantes. La qualité sémantique de notre mesure ainsi que son efficacité a été validée grâce à des expérimentations sur différents jeux de données. Les expérimentations montrent que les clusters obtenus en utilisant S^2MP sont plus homogènes, plus précis et plus complets que ceux obtenus avec Edit distance.

1 Introduction

Dans de nombreux domaines d'applications comme la bio-informatique ou l'analyse des achats des clients, les données sont stockées sous forme séquentielle. On distingue différents types de **séquences** : (1) **les séquences d'items** composés d'éléments atomiques comme des séquences d'entiers et (2) **les séquences d'itemsets** constituées d'objets plus complexes eux-mêmes composés de plusieurs items.
On distingue également deux types de **séquences d'itemsets** : (1) **les séquences de données**, (2) **les motifs séquentiels fréquents**. Par exemple, dans une base d'achats de supermarché, une séquence de données correspond à la série de paniers d'un client (*les itemsets*) ordonnés chronologiquement. Chaque panier est composé des articles achetés (*les items*).
Les motifs séquentiels présentés par Agrawal et Srikant (1995) sont des schémas fréquents extraits à partir des séquences de données. Un motif séquentiel met en évidence des associations inter-transactions. Par exemple, un motif séquentiel extrait d'une base d'achats de supermarché met en évidence des comportements fréquents dans les achats des clients.
Par exemple, $\langle(Chocolat,\ Soda)(gâteaux,\ chips)(biscuit\ dietétique)\rangle$ se traduit par : souvent les clients achètent du chocolat et du soda, puis au cours d'une prochaine visite au magasin, ils achètent des gâteaux et des chips et ensuite plus tard ils achètent des biscuits diététiques".
Les séquences de données et les motifs séquentiels partagent deux caractéristiques : (1) *chaque itemset est un ensemble non ordonné d'items*, (2) *les itemsets sont ordonnés dans la séquence*.

Dans plusieurs domaines d'applications, des motifs séquentiels sont extraits à partir des données afin d'identifier des corrélations. L'évaluation des motifs séquentiels extraits permet de mesurer la proximité de ces motifs et par conséquence d'identifier la cohérence ou incohérence dans les comportements découverts par les motifs extraits. Or, l'évaluation des motifs séquentiels extraits est une tâche difficile à cause des caractéristiques particulières des motifs séquentiels.
Pour de nombreuses applications, il est nécessaire d'évaluer la proximité des motifs séquentiels. Par exemple, pour réaliser un clustering de motifs séquentiels afin d'identifier des comportements similaires, pour extraire des motifs séquentiels sous contraintes ou pour la visualisation des motifs séquentiels. Ces comparaisons se basent sur une mesure de similarité qui est l'un des concepts centraux de la fouille de données (Moen, 2000). Pour une comparaison pertinente, la mesure de similarité doit être adaptée aux caractéristiques de données. Elle doit également passer à l'échelle pour supporter les gros volumes de données.

Pour une mesure de similarité dédiée aux motifs séquentiels, de nombreuses applications sont envisageables comme le clustering des motifs séquentiels pour identifier des comportements similaires, l'extraction de motifs séquentiels sous contraintes de similarité ou la visualisation de motifs séquentiels. Dans l'objectif de clarifier l'intérêt d'une mesure dédiée aux motifs séquentiels, nous détaillons certaines de ces applications où la comparaison de motif séquentiel est essentielle.

Une telle mesure peut être utilisée pour le regroupement (clustering) des motifs séquentiels en plusieurs groupes (clusters) correspondant à un type homogène de corrélations. Ces groupes sont utilisés pour créer des profils de comportements. Par exemple, dans le contexte de la détection d'anomalies, le clustering de motifs séquentiels extraits à partir des logs de connexion normaux peut être utilisé afin de créer des profils de comportements généraux correspondant à une représentation plus abstraite de ces comportements. Cette modélisation de comportements passe à l'échelle (Sequeira et Zaki, 2002). En outre, le clustering nous permet de détecter des déviations éventuelles dans les données. Dans une base d'achats de supermarché, le regroupement des motifs séquentiels peut aider à la segmentation de la clientèle ou à la prédiction en fonction des comportements d'achat.

L'extraction de motifs séquentiels sous contraintes de similarité est une autre application de la mesure de similarité. Des motifs séquentiels extraits par des techniques apriori (Agrawal et Srikant, 1995) sont très volumineux. Des techniques ont été développées pour extraire des motifs séquentiels jugés intéressants car respectant des contraintes comme des contraintes de similarité. Par exemple, Capelle et al. (2002) extrait uniquement des motifs similaires à un motif de référence.

L'interrogation d'un ensemble de motifs séquentiels est également un contexte où la mesure de similarité est utile. Étant donné un motif séquentiel considéré comme une requête, nous cherchons des motifs similaires. Ces interrogations sont très pertinentes en bio-informatique et plus généralement dans la visualisation des motifs séquentiels.

La définition de la similarité peut varier selon le type de similitudes que l'on cherche à identifier entre deux objets. Autrement dit, on pourrait obtenir différents résultats en comparant des objets selon des points de vue différents. Des mesures de similarité différentes peuvent refléter les différents visages des données et leur contexte. Deux objets peuvent être considérés comme très similaires par une mesure et très différents par une autre mesure (Moen, 2000). En outre, nous affirmons que la similitude ne doit pas être toujours symétrique, comme Tversky (1977) l'a souligné. Par exemple, nous disons "les Turcs se battent comme des Tigres" et non pas "les Tigres se battent comme des Turcs". Dans certaines applications telles que l'extraction de motifs séquentiels sous contraintes de similarité ou l'interrogation d'une base de motifs séquentiels, on compare des motifs à un motif de référence. Il s'agit d'une comparaison directionnelle où une mesure de similarité non-symétrique est applicable.

Plusieurs approches ont été développées pour comparer la similarité entre deux séquences notamment dans le domaine de la bio-informatique. Il existe un grand nombre de travaux portant sur les séquences d'items mais peu de travaux portent sur les séquences d'itemsets. Les mesures dédiées aux séquences d'items ne sont pas adaptées aux séquences d'itemsets surtout aux caractéristiques particulières de motifs séquentiels.
Afin de comparer la similarité des motifs séquentiels, nous définissons ici une mesure de similarité (**S^2MP** : **S**imilarity **M**easure for **S**equential **P**atterns) qui prend en compte les caractéristiques et la sémantique des motifs séquentiels. Cette mesure compare deux motifs au niveau des itemsets et de leurs positions dans la séquence ainsi qu'au niveau de la ressemblance des items dans les itemsets. Notre mesure résulte de la combinaison de deux scores : (1) le score relatif à la similarité des itemsets selon les items qui les composent ; (2) le score relatif à la similarité des itemsets selon leur position et leur ordre dans les séquences.
Par ailleurs, S^2MP s'adapte à différents contextes car il est possible de modifier ou d'adapter chaque score. De plus, S^2MP est une mesure paramétrable en fonction de l'importance accordée à chaque caractéristique des motifs séquentiels (*l'ordre ou la similarité des itemsets*) selon le contexte d'application. Cela rend notre mesure flexible et sensible aux caractéristiques du domaine, ce qui n'est pas le cas des mesures existantes.

La section 2 présente les travaux existant sur les mesures de similarité pour les motifs séquentiels fréquents. Notre mesure de similarité (S^2MP) est présentée dans la section 3. Les résultats obtenus lors des expérimentations sur la mesure de similarité sont détaillés dans la section 4.

2 Inadéquation des mesures avec les motifs séquentiels

Les deux mesures de similarité principalement utilisées pour les séquences d'itemsets sont : **Edit distance** et **LCS**. Dans cette section, nous listons les inconvénients de ces deux mesures pour les motifs séquentiels. Ensuite, nous citons une troisième approche qui a été appliquée aux données multidimensionnelles.

La mesure *Edit distance* a été utilisée dans Capelle et al. (2002) pour extraire des motifs séquentiels sous contraintes de similarité. Les auteurs définissent un motif séquentiel comme une liste ordonnée de symboles appartenant à $\sum$ où $\sum$ est un alphabet comprenant un ensemble fini de symboles.

Exemple 2.1. Soit $M_1 = \{(ab)(c)\}$ et $M_2 = \{(a)(c)\}$, deux motifs séquentiels. On associe aux itemsets des symboles : $X = (ab)$, $Y = (c)$ et $Z = (a)$
Les opérateurs d'Edit distance étant appliqués sur les éléments de la séquence (*i.e. les itemsets*), la distance est donc le coût de substitution de X par Z. Un itemset est ici réduit à un événement caractérisé par les valeurs de certains attributs (*les items*). Un motif séquentiel est donc considéré comme une séquence d'événements[1] ordonnés en fonction de leurs occurences (Mannila et Ronkainen, 1997; Moen, 2000). Par conséquent, les itemsets (ab) et (a) sont traités comme deux symboles (*événements*) différents. Or, (ab) et (a) peuvent être deux comportements similaires.
ApproxMAP développé par Kum et al. (2003) est un algorithme pour fouiller les séquences consensus[2] qui se déroule en deux phases : (1) le clustering de séquences (2) la fouille de motifs consensus directement à partir de chaque cluster. Dans la phase 1, les auteurs ont utilisé Edit distance comme mesure de similarité mais en remplaçant le coût de l'opérateur substitution par la *différence ensembliste normalisée*. Bien que cette modification surmonte la limitation d'Edit distance argumentée précédemment, les auteurs ont noté que la différence ensembliste donne plus de poids aux éléments communs. Elle est donc appropriée si les points communs sont plus importants que les différences.
En outre, Moen (2000) discute l'influence des coûts des opérations d'Edit distance sur le degré de similarité dans le cas des séquences. Elle indique qu'il est plus naturel de donner plus de poids à l'insertion (*ou suppression*) des ensembles rares qu'aux ensembles fréquents. L'influence du coût des opérations et le type des opérations sont justifiés dans (Moen, 2000).
Par ailleurs, Edit distance ne s'adapte pas à différentes définitions de la similarité. Par exemple, lorsque l'on cherche à comparer des motifs séquentiels extraits à partir de données bio-informatiques (*transcriptomiques*), le contenu des itemsets (items) est plus important que l'ordre des itemsets. Or, Edit distance ne permet pas de prendre en compte cette caractéristique liée aux données. Pour une comparaison pertinente et sémantiquement correcte, il faut comparer deux motifs séquentiels au niveau des itemsets et de leurs positions dans les séquences ainsi qu'au niveau des items dans les itemsets.

La mesure LCS (Longest Common Subsequence) est utilisée pour la comparaison des séquences Sequeira et Zaki (2002). Cette mesure donne la longueur de la sous-séquence commune la plus longue à deux séquences. Nous donnons ci-dessous trois limitations.

Exemple 2.2. Soit trois motifs séquentiels :
$M_1 = \{(\mathbf{bc})(\mathbf{d}f)(e)\}$, $M_2 = \{(a\mathbf{bc})(mn)(\mathbf{d}e)(gh)(fg)\}$ et $M_3 = \{(\mathbf{bc})(\mathbf{d}f)\}$.
La sous-séquence commune la plus longue est $\{(bc)(d)\}$.
$LCS(M_1, M_2) = 2$, $LCS(M_1, M_3) = 2$.
Premièrement, LCS ne prend pas en compte la position des itemsets (*dans l'ordre*) dans les deux séquences. La sous-séquence $\{(bc)(d)\}$ est une sous-séquence avec des itemsets consécutifs dans $M_1 = \{(\underline{bc})(\underline{d}f)(e)\}$ et $M_3 = \{(\underline{bc})(\underline{d}f)\}$ et non consécutifs dans $M_2 = \{(a\underline{bc})\mathbf{(mn)}(\underline{d}e)(gh)(fg)\}$.
Deuxièmement, LCS ne considère pas la longueur de la partie non-commune entre les deux séquences. La partie non commune dans M_2 (*i.e.* $\{(\mathbf{a}bc)(\mathbf{mn})(d\mathbf{e})(\mathbf{egh})(\mathbf{fg})\}$) est plus longue que celle de M_3 (*i.e.* $\{(\mathbf{e})(bc)(d\mathbf{f})\}$). C'est pourquoi la normalisation de LCS par le nombre d'items dans les séquences est nécessaire.
Troisièmement, le nombre d'items non-communs dans les itemsets (*où les items communs apparaissent*) n'influence pas la valeur de LCS. L'itemset (bc) de la sous-séquence est inclus dans (abc) de M_2 mais il est égal à (bc) de M_1 et de M_3. Ce problème n'est pas résolu par la normalisation.

Plantevit et al. (2007) proposent une comparaison de la similarité entre deux motifs séquentiels multidimensionnels $S_1 = \{b_1, b_2, \ldots, b_k\}$ et $S_2 = \{b_1^{'}, b_2^{'}, \ldots, b_k^{'}\}$:
$d(s1, s2) = Op(dist(bj, b'j))\ pour j = 1 \ldots k$ avec $dist$ une mesure de distance et Op un opérateur d'agrégation. La comparaison se fait entre des blocs[3] correspondants. On compare le bloc i de S_1 avec le bloc i de S_2. Ce type de comparaison n'est pas pertinent lorsqu'il existe un décalage dans l'une des séquences.

Les mesures présentes dans littérature présentent certains inconvénients dans le cas des motifs séquentiels. Une mesure de similarité dédiée aux motifs séquentiels doit prendre en compte :

[1] Edit distance est fréquemment utilisée pour les séquences d'événements Moen (2000)
[2] Une séquence consensus est partagée par de nombreuses séquences et couvre de nombreux motifs courts.
[3] Ici, Un bloc de données peut être vu comme un itemset

1. les motifs séquentiels sont des séquences ordonnées d'itemsets (*ensemble d'items*) et non d'items,
2. les **positions** (*distance dans l'ordre*) des itemsets lors du calcul de la similarité,
3. le nombre d'**items communs** et **non communs** au niveau de la séquence et au niveau des itemsets correspondants,
4. idéalement, une mesure de similarité doit être modulaire et paramétrisable à chaque niveau de comparaison afin de pouvoir l'adapter aux contextes différents.

C'est en prenant en compte ces quatre critères que nous avons proposé la mesure S^{2}MP.

3 S^{2}MP : Description

La mesure de similarité (S^{2}MP : Similarity **M**easure for **S**equential **P**atterns) résulte de l'agrégation de deux scores :
– le **score de mapping** qui mesure la ressemblance de deux motifs en fonction des liens qu'il est possible d'établir entre les itemsets,
– le **score d'ordre** qui mesure la ressemblance des deux séquences vis-à-vis de l'ordre et de la position des itemsets.

Le déroulement de l'algorithme se fait en deux phases qui correspondent au calcul de ces deux scores.
Dans la phase 1, nous mettons en correspondance les itemsets des deux séquences en fonction de leur contenu et calculons pour chaque lien un poids (*similarité entre deux itemsets*) selon le nombre d'items communs et non communs. Le score de mapping est alors calculé en considérant l'ensemble des poids.
À la phase 2, l'objectif est de donner un score en fonction de la ressemblance de deux séquences selon l'ordre et la position des itemsets. Parmi tous les liens établis à la phase 1, nous cherchons ceux qui respectent l'ordre des itemsets dans les deux séquences. Cela signifie que nous rejetons les mappings croisés (cf. figure 2). Nous mesurons également la ressemblance des itemsets mis en correspondance au niveau de leurs positions dans les deux séquences. Finalement, le score d'ordre est calculé en fonction du pourcentage de liens respectant l'ordre des itemsets et du score mesurant la ressemblance des itemsets au niveau de leur position dans les séquences.

Phase 1 – Calcul du score de mapping. En entrée de ce calcul, nous avons les deux séquences à comparer. Nous établissons des liens entre chaque itemset i de la séquence 1 $Seq_1(i)$ et l'itemset j le plus ressemblant dans la séquence 2 $Seq_2(j)$. Nous évaluons ces liens par des poids.

Définition 3.1. $poids(i,j)$ entre le $i^{ème}$ itemset de la Seq_1 et le $j^{ème}$ itemset de la Seq_2 :

$$poids(i,j) = \frac{|Seq_1(i) \cap Seq_2(j)|}{(|Seq_1(i)| + |Seq_2(j)|)/2}$$

Nous formalisons le lien (*mapping*) entre deux itemsets ainsi :

$$Mapping(Seq_1(i), Seq_2(j)) \mid poids(i,j) = \max_{x \in [0,|Seq_2|]} \{poids(i,x)\} \wedge poids(i,j) \neq 0$$
$$\textbf{Si } poids(i,j) = poids(i,k) \Rightarrow pos(j) < pos(k)$$

Si plusieurs liens sont possibles avec des poids égaux, nous prenons le lien qui est associé à l'itemset ayant une $position$[4] (pos) inférieure, c'est-à-dire l'itemset qui est situé avant les autres dans l'ordre de la séquence.

Conflit. Un itemset sélectionné de la $2^{ème}$ séquence pour être mis en correspondance avec un itemset de la séquence 1 peut avoir déjà été utilisé dans un autre lien. Nous appelons cette situation "conflit" (cf. figure 1). Pour résoudre ce problème, nous recherchons un nouveau candidat non déjà lié avec la fonction "Résolution de conflit". Pour les deux

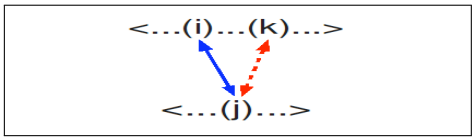

FIG. 1 – *Conflit de mapping.*

itemsets de séquence 1 en conflit ($Seq_1(i)$ et $Seq_1(k)$), nous cherchons quatre nouveaux candidats dans la Seq_2 (*autre que l'itemset candidat actuel :* $Seq_2(j)$) avant et après $Seq_2(j)$:

[4]Ici, la $position$ correspond à la place des itemsets dans l'ordre de la séquence. Par exemple dans la séquence (a)(ab)(c), la position de l'itemset (ab) est égale à 2 (*pos(ab) = 2*).

– $nextMaxBefor_i$ et $nextMaxBefor_k$
– $nextMaxAfter_i$ et $nextMaxAfter_k$

Nous créons ensuite les quatre couples possibles entre ces candidats :

$< Seq_1(i), Seq_2(j) >, < Seq_1(k), nextMaxBefore_k >$ $< Seq_1(k), Seq_2(j) >, < Seq_1(i), nextMaxBefore_i >$
$< Seq_1(i), Seq_2(j) >, < Seq_1(k), nextMaxAfter_k >$ $< Seq_1(k), Seq_2(j) >, < Seq_1(i), nextMaxAftre_i >$

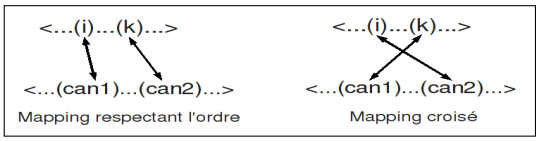

FIG. 2 – *Mapping croisé et l'ordre des itemsets.*

Nous considérons ici deux types de mises en relation : celles respectant l'ordre et les autres (cf. figure 2). Nous appelons ces dernières des mappings croisés : Supposons qu'un itemset à la position i de la première séquence soit mis en relation avec un itemset à la position i' dans la deuxième séquence. Un itemset à la position j (où $i < j$) est mis en relation avec un itemset à la position j' de la deuxième séquence. Ce lien est conforme à l'ordre si $i' < j'$ et non conforme (mapping croisé) si $i' > j'$.
Nous calculons la pertinence des quatre couples identifiés précédemment avec un score dit $coupleScore$ dont le calcul dépend du type de mise en relation :
Si l'ordre est conforme :

$$coupleScore(i, Can_1)(k, Can_2) = \frac{Poids(i,Can_1)+Poids(k,Can_2)}{2}$$

Si l'ordre est non conforme (*mappings croisés*) :

$$coupleScore(k, Can_1)(i, Can_2) = \frac{1}{2} \times \frac{Poids(k,Can_1)+Poids(i,Can_2)}{2}$$

Le couple ayant le $coupleScore$ le plus élevé est sélectionné comme sortie de la fonction de résolution de conflit.

Boucle de conflit. Au cours de la phase 1, une boucle de conflit peut se produire, c'est-à-dire que le candidat renvoyé par la fonction de résolution de conflit est lui-même mis en relation avec un autre itemset. Dans ce cas, nous continuons à faire appel à cette fonction jusqu'à ce que ses sorties ne soient pas déjà mises en relation. Si finalement on ne trouve pas de candidat pour un des itemsets en conflit, nous associons le candidat initial $Seq_2(j)$ avec l'itemsets en conflit le plus similaire et l'autre itemset reste sans correspondance.

Sortie de la phase 1. À la fin de cette phase, nous avons obtenu un ensemble de relations entre les itemsets des deux séquences. Ces liens sont conservés dans une liste nommée $mapOrder$. Nous mettons à la $i^{ème}$ place de cette liste la $position$ de l'itemset de la séquence 2 mis en correspondance avec le $i^{ème}$ itemset de la séquence 1. Nous obtenons :

$$mapOrder = \{t_1, t_2, ..., t_i, ..., t_n\}$$

t_i est la position de l'itemset de la Seq_2 associé au i^{ème} itemset de la Seq_1.

Nous obtenons également le **score de mapping** c.-à-d. la moyenne de tous les poids des liens. Ce score prend en compte les mappings croisés. En effet, pour comparer <(a)(b)> avec <(b)(a)> et <(a)(b)> avec <(b)(d)> si l'on utilisait seulement les mappings respectant l'ordre (*i.e.* (b)→(b)) nous traiterions <(a)(b)> avec <(b)(a)> de la même manière que <(a)(b)> avec <(b)(d)>. Or, en considérant l'ensemble des poids, nous prenons en compte le poids de (a)→(a) (*égal à 1*) dans le premier cas et les itemsets sans mapping *(a),(d)* dans le deuxième cas. Nous considérons donc ici l'influence des liens ne respectant pas l'ordre sur le degré de similarité.

Phase 2 – Calcul du score de l'ordre. En entrée de cette phase, nous avons les deux séquences et également la liste $mapOrder$ sortie de la phase 1.
L'objectif de ce score est :

1. de trouver les liens respectant l'ordre des itemsets dans les deux séquences ;
2. de prendre en compte les positions des itemsets correspondants dans les deux séquences.

Pour cela, nous calculons deux scores : (1) $totalOrder$ et (2) $positionOrder$.

$totalOrder$ mesure le pourcentage de liens établis dans la phase 1 respectant l'ordre des itemsets (*l'exclusion des mappings croisés*). Dans la liste $mapOrder$, tant que les $positions$ augmentent, les liens correspondant respectent l'ordre. Nous cherchons donc toutes les sous-séquences croissantes et optimales de $mapOrder$ pour trouver toutes les séries de liens qui sont conformes à l'ordre.

$$totalOrder = \frac{nbOrderedItemSets}{aveNbItemSets}$$

$nbOrderedItemSets$ = le nombre d'itemsets dans la sous-séquence ordonnée

$aveNbItemSets$ = la moyenne du nombre d'itemsets dans les deux séquences

Deuxièmement, nous mesurons la largeur entre les itemsets mis en relation en fonction de leur position avec le score $positionOrder$. Par exemple, sur la figure 3-(**a**), les itemsets sont mis en relation d'une manière plus proche que les item-

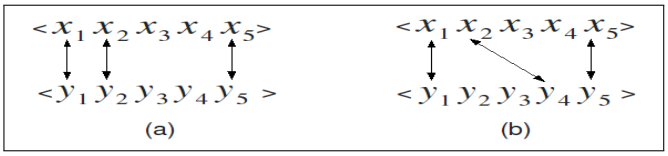

FIG. 3 – *Distance entre mapping en fonction des positions des itemsets.*

sets de la figure 3-(**b**). Nous utilisons de nouveau la $position$ des itemsets pour obtenir cette information. $positionOrder$ indique si la distance entre deux liens successifs dans la séquence 1 est égale à la distance entre ces mêmes liens dans la séquence 2. Sur la figure 3-(**a**), les liens sont à égale distance dans les deux séquences alors qu'ils ne le sont pas sur la figure 3-(**b**). $positionOrder$ se calcule ainsi :

$$positionOrder = \sum_{i=1}^{|sub|} \left(\frac{\mid (sub(i) - sub(i-1)) - (mapOrder^{-1}(sub(i)) - mapOrder^{-1}(sub(i-1))) \mid}{aveNbItemSets} \right)$$

$sub(i)$ = la valeur de $i^{ème}$ position dans la sous-séquence

$mapOrder^{-1}(x)$ = la position de "x" dans $mapOrder$

Sortie de la phase 2 : Finalement, pour chaque sous-séquence croissante de $mapOrder$, nous multiplions $totalOrder$ et $positionOrder$ et nous gardons le score le plus élevé comme *score d'ordre* :

$$\textbf{orderScore} = \max\{totalOrder(sub) \times (1 - positionOrde(sub))\}$$
$$sub \in \{sous\_seqs\ croissantes\ et\ maximales\ du\ mapOrder\}$$

Phase 3 – Calcul du degré de similarité. Nous calculons le degré de similarité S^2MP par une agrégation entre le score d'ordre et le score de mapping avec une moyenne pondérée par des coefficients. Cela nous permet de moduler les scores selon les contextes d'application et la définition de la similitude.

$$S^2MP = \frac{(orderScore \times Co_1) + (mappingScore \times Co_2)}{Co_1 + Co_2}$$

Illustration. Soient les deux motifs $M_1 = \{(bc)(df)(e)\}$ et $M_2 = \{(abc)(mn)(de)(egh)(fg)\}$ que nous avons utilisés pour montrer les inconvénients de LCS.
Phase 1 : Calcul du score de mapping. Pour chaque itemset de $M_1(i)$, on cherche l'itemset le plus ressemblant dans $M_2(j)$ en calculant les poids.

- $Poids\,(M_1(1), M_2(1)) \Longrightarrow Poids\,((bc),(abc)) = \frac{2}{\frac{3+2}{2}} = 0.8$
- $Poids\,(M_1(1), M_2(2)) \Longrightarrow Poids\,((bc),(mn)) = \frac{0}{\frac{2+2}{2}} = 0$
- $Poids\,(M_1(1), M_2(3)) \Longrightarrow Poids((bc),((de)) = \frac{0}{\frac{2+2}{2}} = 0$
- $Poids\,(M_1(1), M_2(4)) \Longrightarrow Poids((bc),((egh)) = \frac{0}{\frac{2+3}{2}} = 0$
- $Poids\,(M_1(1), M_2(5)) \Longrightarrow Poids((bc),((fg)) = \frac{0}{\frac{2+2}{2}} = 0$

On choisit le lien ayant le poids le plus élevé : (bc)→(abc). Nous effectuons les mêmes calculs pour les autres itemsets :

$Poids((df),(abc)) = \frac{0}{\frac{3+2}{2}} = 0$	$Poids((e),(abc)) = \frac{0}{\frac{3+2}{2}} = 0$
$Poids((df),(mn)) = \frac{0}{\frac{2+2}{2}} = 0$	$Poids((e),(mn)) = \frac{0}{\frac{2+2}{2}} = 0$
$Poids((df),((de)) = \frac{1}{\frac{2+2}{2}} = 0.5$	$Poids((e),((de)) = \frac{1}{\frac{1+2}{2}} = 0.6$
$Poids((df),((egh)) = \frac{0}{\frac{2+3}{2}} = 0$	$Poids((e),((egh)) = \frac{1}{\frac{1+3}{2}} = 0.5$
$Poids((df),((fg)) = \frac{1}{\frac{2+2}{2}} = 0.5$	$Poids((e),((fg)) = \frac{1}{\frac{2+2}{2}} = 0$

Dans le cas du $2^{ème}$ itemset de M_1, les $poids((df),(de))$ et $poids((df),(fg))$ sont égaux. On sélectionne alors l'itemset ayant la $position$ le plus petit (*i.e.* (de)) pour l'associer à l'itemset (df). On a donc : (df)→(de).
Pour le $3^{ème}$ itemset, l'itemset sélectionné est (de). Or, cet itemset a déjà été associé à l'itemset (df) de M_1. Par conséquent, nous utilisons la fonction de résolution de conflit. Nous cherchons de nouveaux candidats dans M_2 pour les itemsets en conflit (df) et (e) avant et après l'itemset candidat actuel (de). Nous obtenons les candidats suivants :

pour l'itemset (df) :	**pour l'itemset (e) :**
$nextMaxBefore_{(df)} = \emptyset$	$nextMaxBefore_{(e)} = \emptyset$
$nextMaxAfter_{(df)} = (fg)$	$nextMaxAfter_{(e)} = (egh)$

Les couples de mappings possibles :
$\langle((df),(de)),((e),(egh))\rangle$
$\langle((e),(de)),((df),(fg))\rangle$.

En utilisant les poids et en considérant les liens ne respectant pas l'ordre, nous obtenons :

$coupleScore(((df),(de)),((e),(egh))) = \frac{0.5+0.5}{2} = 0.5$
$coupleScore(((e),(de)),((df),(fg))) = \frac{1}{2} \times \frac{0.6+0.5}{2} = 0.27$

Nous choisissons le couple ayant le $coupleScore$ le plus élevé : $\langle((df),(de)),((e),(egh))\rangle$.
(df)→(de) et (e)→(egh).

Les mises en relation finales sont :

$\langle M_1(1) = (abc), M_2(1) = (ab)\rangle$
$\langle M_1(2) = (df), M_2(3) = (a)\rangle$
$\langle M_1(3) = (e), M_2(4) = (ca)\rangle$

Nous créons la liste $mapOrder$. $mapOrder(1) = 1$ car la $position$ de l'itemset $M_2(1)$ mis en relation avec l'itemset $M_1(1)$ de la $1^{ère}$ séquence est égal à 1.
Nous obtenons : $mapOrder = \{1, 3, 4\}$

Finalement, le score de mapping est la moyenne des poids :

mappingScore = $\frac{poids((bc),(abc))+poids((df),(de))+poids((e),(egh))}{3} = 0.6$

Phase 2 – Calcul du score de l'ordre. Nous cherchons toutes les sous-séquences croissantes et optimales de la séquence $mapOrder$:

La seule sous-séquence croissante maximale trouvée : $\{1, 3, 4\}$

D'après les formules $totalOrder$ **et** $positionOrder$**, le score de l'ordre se calcule ainsi :**
totalOrder((1,3,4)) = $\frac{3}{(3+5)/2} = 0.75$
positionOrder((1,3,4)) = $\frac{|(3-1)-(2-1)|}{(3+5)/2} + \frac{|(4-3)-(3-2)|}{(3+5)/2} = 0.25$
orderScore = $0.75 \times (1 - 0.25) = 0.56$

Phase 3 – Calcul de la similarité. Avec une multiplication entre le score de mapping ($= 0.6$) et le score d'ordre ($= 0.56$), nous obtenons le degré de similarité des deux motifs séquentiels. Ici, ayant considéré les deux scores de la même niveau d'importance, nous choisissons l'un comme coefficient des deux scores.

$$\boldsymbol{S^2MP(M_1, M_2) = \frac{(1\times 0.56)+(1\times 0.6)}{2} = 0.58}$$

4 Expérimentations

Une mesure de similarité doit saisir la ressemblance des objets comparés. Une telle mesure est généralement utilisée au sein d'un autre algorithme comme un clustering. Elle doit donc se calculer efficacement et passer à l'échelle. Nous expérimentons S^2MP avec deux objectifs principaux : (1) démontrer la qualité sémantique du degré de similarité obtenue par S^2MP, (2) mesurer l'efficacité de l'algorithme de S^2MP au niveau du temps d'exécution et de la taille mémoire utilisée.

Qualité sémantique de S^2MP. Nous testons S^2MP pour évaluer sa qualité sémantique, c'est-à-dire que l'on cherche à mesurer la valeur de vérité associée à cette mesure, sa signification lors de la comparaison de deux motifs séquentiels. Pour cela, nous testons S^2MP pour évaluer sa qualité sémantique et comparons les résultats obtenus par S^2MP et d'Edit distance.
Nous avons adapté l'algorithme de clustering k-means aux motifs séquentiels et aux deux mesures S^2MP et Edit distance. Nous avons appliqué ces deux versions de l'algorithme sur le même jeu de données synthétiques constitué de 100 motifs séquentiels de tailles différentes. Nous utilisons Edit distance avec la différence ensembliste comme le coût de substitution (*i.e. la version utilisée dans ApproxMAP*). Les clusterings ont été réalisés jusqu'à ce que les clusters n'évoluent plus. Pour des applications comme le clustering où l'on a besoin d'une mesure symétrique, nous prenons la moyenne de S^2MP effectuée dans les deux directions pour rendre la mesure symétrique.
Pour comparer des clusters obtenus avec les deux mesures, nous ne pouvons pas utiliser les techniques de comparaison des clusterings (*mesures inter et intra-clusters*) effectuées à partir de la mesure de similarité (*ce que nous voulons évaluer*). L'algorithme de clustering est constant et ce sont les mesures qui diffèrent. La meilleure solution donc est de mesurer la pureté des clusters obtenus avec les deux mesures ou de comparer les résultats de deux clusterings avec un clustering de référence. Nous avons choisi un point de vue sur les données, nous avons paramétré S2MP en conséquence puis nous avons appliqué S2MP et Edit distance. Nous avons comparé les résultats en fonction du partie pris sur les données pour vérifier que le comportement de S2MP correspondait bien avec notre objectif. Nous créons donc manuellement 10 groupes de motifs séquentiels similaires comme références à partir du jeu de données utilisé pour les clusterings. Parmi ces 10 groupes de référence, 4 groupes contiennent des motifs très différents des motifs séquentiels des autres groupes et 6 groupes contiennent des motifs qui ressemblent aux motifs d'un autre groupe. Cela nous permet d'évaluer la précision de chaque mesure lorsqu'il s'agit de distinguer les clusters avec une distance inter-cluster petite.
Nous calculons la précision et le rappel des clusters de chaque clustering par rapport aux références. Nous mesurons aussi l'entropie des clusters de chaque clustering qui permet de déterminer les groupes les plus homogènes (donc « meilleurs »). L'entropie moyenne de chaque clustering se calcule en prenant la moyenne des entropies de ses clusters.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
S^2MP	0.98	0	0.99	0.86	0.95	0	0.97	0.95	0.65	0
Edit distance	0.97	0	0.99	1.20	0.89	0	0.98	0.98	0.70	0.99

TAB. 1 – *Entropie des clusters obtenus avec S^2MP et Edit distance.*

Ces expérimentations montrent que les clusters obtenus avec S^2MP sont plus homogènes que ceux obtenus avec Edit distance. Le tableau 1 montre l'entropie moyenne obtenue avec chaque mesure. Plus l'entropie est petite plus l'ensemble est homogène. L'entropie moyenne de clustering avec S^2MP est 0.63 et en utilisant Edit distance est égale à 0.77.
Le tableau 2 montre la précision et le rappel des clusters obtenus avec chaque mesure.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Précision S^2MP	0,57	1	**0,53**	0,71	0,65	1	0,5	0,6	0,68	**1**
Rappel S^2MP	0,4	**1**	**0,7**	**1**	**1**	0,5	0,5	0,4	**1**	**0,4**
Précision Edit	0,6	1	**0,53**	0,58	0.68	1	0,4	0,62	0,83	**0,57**
Rappel Edit	0,3	**1**	**0,6**	**1**	0,9	0,8	0,2	0,5	**1**	**0,4**

TAB. 2 – *Précision et rappel des clusters obtenus avec S^2MP et Edit distance.*

Nous expérimentons également S^2MP et Edit distance sur leur capacité à identifier des motifs séquentiels similaires dans des contextes différents. Nous considérons le contexte des données bio-informatiques et les caractéristiques particulières des motifs séquentiels extraits à partir de puces ADN. Dans ce domaine, d'après les experts, les contenus des itemsets

(*i.e. items*) sont plus importants que l'ordre des itemsets lorsqu'il s'agit de comparer deux motifs. Par exemple, les deux motifs $M_1 = \{(a)(b)(c)\}$ et $M_2 = \{(b)(a)(c)\}$ peuvent être considérés comme proches car ils contiennent les mêmes itemsets mais pas dans le même ordre.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
S^2MP	0.99	0,52	0,99	0,99	0,99	0	0	0,99	0	0,98
Edit distance	1,2	1,2	1,3	1,3	1,3	1,2	1,4	0,95	0,4	1,7

TAB. 3 – *Entropie des clusters obtenus avec S^2MP et Edit distance quand le contenu des itemsets est plus important que leur ordre – (e.g. motifs extraits issus de puce ADN).*

Pour comparer S^2MP et Edit distance dans telle situation, nous créons manuellement 10 groupes de 10 motifs similaires au niveau du contenu des itemsets mais ayant des ordres différents. L'idée est d'avoir des groupes de référence dans lesquels les séquences sont régroupées selon la similarité des contenus des itemsets plutôt que selon l'ordre des itemsets. Ces groupes ont une distance inter-groupe petite. Nous effectuons un clustering sur ce dernier jeu de données avec S^2MP en choisissant le deux comme coefficient du score de mapping et un comme coefficient du score d'ordre (*i.e. nous configurons S^2MP pour que les contenus des itemsets soient considérés comme plus importants que l'ordre des itemsets*). L'idée n'est pas de trouver des poids idéaux qui peuvent être déterminés par les expérimentations.
Ensuite, nous réalisons un clustering sur ce jeu de données avec Edit distance. Les résultats montrent que S^2MP permet de mieux identifier les motifs similaires dans un contexte d'application particulier tel que les motifs séquentiels issus de données transcriptomiques. Cela prouve que S^2MP est bien paramétrable est adaptable à différents contextes. Les résultats obtenus avec Edit distance dans ce contexte, ne sont pas satisfaisant.

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
Précision S^2MP	0,55	**0,71**	0,54	0,54	0,55	**1**	**1**	0,53	**1**	**0,57**
Rappel S^2MP	0,5	**1**	0,6	0,6	0,5	**0,6**	**1**	0,7	**1**	**0,4**
Précision Edit	0,52	**0,61**	0,46	0,60	0,50	**0,5**	**0,5**	0,37	**0,9**	**0,16**
Rappel Edit	0,9	**0,6**	0,6	0,3	0,5	**0,6**	**0,7**	0,3	**0,9**	**0,2**

TAB. 4 – *Précision et rappel des clusters obtenus avec S^2MP et Edit distance quand le contenu des itemsets importe plus que leur ordre.*

Le tableau 3 souligne la pertinence de S^2MP dans ce contexte en fonction d'entropie des clusters obtenus. Sur ce jeu de données, l'entropie moyenne de clustering en utilisant S^2MP est 0.64 et en utilisant Edit distance est 1.19. Le tableau 4 montre la précision et le rappel des clusters obtenus avec S^2MP et Edit distance sur ce jeu de données.

Efficacité de S^2MP. Nous montrons ici que notre mesure de similarité est très efficace au niveau du temps d'exécution et de la taille de mémoire utilisée. Nous expérimentons l'efficacité de S^2MP selon trois paramètres : (1) le nombre d'items dans les séquences, (2) le nombre itemsets et (3) le nombre de séquences dans la base.
Nous créons une matrice de similarité Mat_{sim} de dimension $(n \times n)$ où n représente le nombre de motifs séquentiels dans la base. S^2MP n'étant pas symétrique, nous calculons la totalité de $Mat_{sim}(n, n)$. Le temps nécessaire pour remplir $Mat_{sim}(n, n)$ est le temps nécessaire aux $n \times n$ comparaisons de similarité. Les expérimentations sont effectuées sur une machine ayant un CPU Intel 2GHz avec 2Go de mémoire vive. La mesure de similarité est développée en Java 5. Nous avons utilisé ici deux types de jeu de données : (1) des motifs séquentiels fréquents (2) des séquences de données. Les motifs séquentiels fréquents sont extraits à partir de données synthétiques générées par le générateur de données IBM quest[5]. Pour vérifier l'influence des conflits sur le temps d'exécution, nous avons fait un test avec des séquences de données car elles permettent d'obtenir de nombreux conflits lors de la mise en relation des itemsets.

[5] $www.almaden.ibm.com/cs/projects/iis/hdb/Projects/data_mining/datasets/syndata.html$

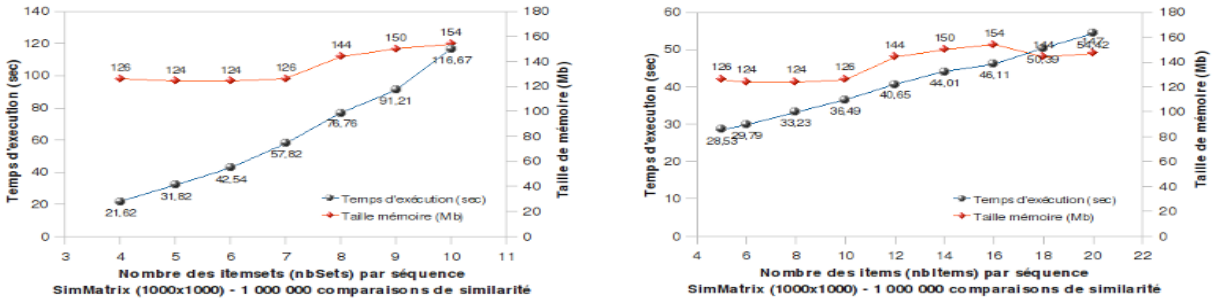

FIG. 4 – *Temps de calcul et taille de mémoire en fonction du nombre d'itemsets (gauche) et d'items (droit).*

Sur les tests en fonction du nombre d'itemsets et d'items, les jeux de données sont constitués de 1 000 motifs séquentiels (*c.-à-d. nous réalisons 1,000,000 de comparaisons de similarité*).

Résultats d'expérimentation sur l'efficacité de S^2MP. La figure 4 représente l'évolution du temps de calcul des 1,000,000 comparaisons et la taille de mémoire utilisée par rapport au nombre d'itemsets (gauche) et d'items (droit) par séquence. Selon les courbes, la taille de mémoire utilisée ne change pas considérablement. Le temps de calcul de $Mat_{sim}(1^M, 1^M)$ quand il y a 10 itemsets par séquence est satisfaisant (116 secs). Notre expérimentation montre que le nombre d'items par séquence n'influence pas le temps de comparaison de la similarité autant que le nombre d'itemsets par séquence. Sur la figure 5, nous montrons le temps de calcul de $Mat_{sim}(n, n)$ (figure de gauche) et la taille mémoire

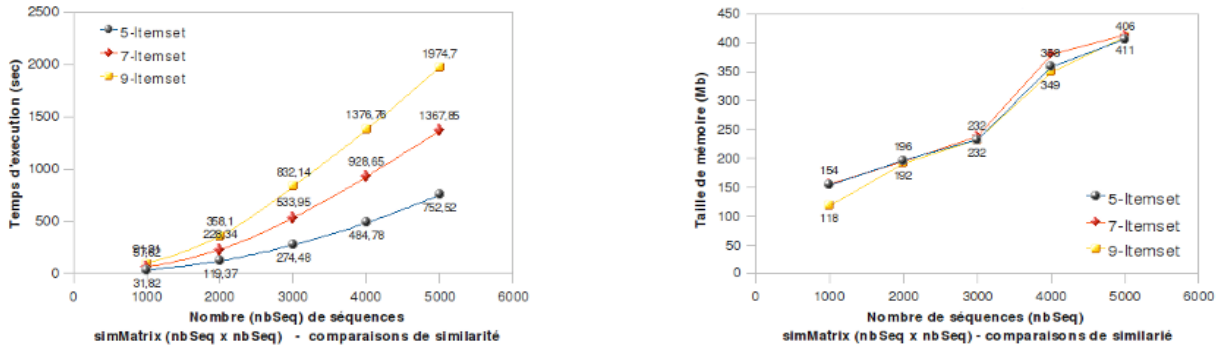

FIG. 5 – *Temps de calcul de $Mat_{sim}(n, n)$ (gauche) et taille de mémoire utilisée (droit) en fonction du nombre de séquences.*

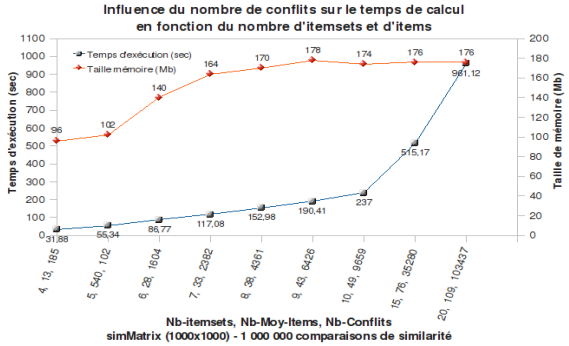

FIG. 6 – *Influence des conflits sur le temps de calcul de la mesure de similarité*

utilisée (figure de droite) quand le nombre de séquences augmente. Dans chaque cas, il y a $n \times n$ comparaisons de similarité où n est le nombre de séquences dans le jeu de données. Nous réalisons ce test sur des séquences de 5, 7 et 9 itemsets. Dans le cas, où il y a 5 000 séquences (*c.-à-d. 25,000,000 de comparaisons de similarité*) et que chaque séquence contient

9 itemsets, le temps d'exécution est seulement de 1974 secs.
Sur la figure 6, nous montrons le résultat de l'expérimentation sur les séquences de données. Pour chaque cas, nous avons noté le nombre de conflits résolus lors du calcul de $Mat_{sim}(1^M, 1^M)$. L'axe X représente les différents jeux de données. Pour chacun, le nombre d'itemsets et le nombre moyen d'items par séquences sont notés. Il existe 1 000 séquences dans chaque jeu de données (1^M *comparaisons de similarité*). Les courbes représentent le temps de remplisage de $Mat_{sim}(1^M, 1^M)$ pour chaque cas et la taille de mémoire utilisé. Dans le cas où il y a 20 itemsets et en moyenne 109 items par séquence, 103 437 conflits sont résolus et le temps d'exécution de 1^M calcul de similarité est égal à 961 secs.

5 Conclusion

Dans cet article, nous avons défini une mesure de similarité (S^2MP) adaptée aux motifs séquentiels prenant en compte toutes leurs caractéristiques et notamment leur sémantique. Le degré de similarité est composé de deux scores. Ces scores mesurent la similarité des motifs séquentiels à la fois au niveau de l'ordre des itemsets et de leurs positions dans les séquences (*score d'ordre*) mais aussi au niveau des items contenus dans les itemsets correspondant (*score de mapping*). La combinaison de deux scores indépendants permet d'avoir une mesure modulaire. Elle est donc adaptable selon le contexte et le sens des itemsets dans le domaine en modifiant le score concerné. S^2MP surmonte les inconvénients des mesures classiques comme LCS et $Edit\ distance$ pour les motifs séquentiels. Les expérimentations montrent que S^2MP se calcule très rapidement même lorsque le nombre de séquences ayant plusieurs itemsets est élevé. Plusieurs domaines et méthodes comme le clustering de motifs séquentiels, la détection d'outliers, l'extraction de motifs séquentiels sous contrainte de similarité, la compression des motifs séquentiels, la visualisation des motifs similaires, etc. sont envisageables comme applications de S^2MP.

Références

Agrawal, R., C. Faloutsos, et A. N. Swami (1993). Efficient similarity search in sequence databases. In D. Lomet (Ed.), *Proceedings of the 4th International Conference of Foundations of Data Organization and Algorithms (FODO)*, Chicago, Illinois, pp. 69–84. Springer Verlag.

Agrawal, R. et R. Srikant (1995). Mining sequential patterns. In P. S. Yu et A. S. P. Chen (Eds.), *Eleventh International Conference on Data Engineering*, Taipei, Taiwan, pp. 3–14. IEEE Computer Society Press.

Bozkaya, T., N. Yazdani, et Z. M. Ozsoyoglu (1997). Matching and indexing sequences of different lengths. In *CIKM*, pp. 128–135.

Capelle, M., C. Masson, et J.-F. Boulicaut (2002). Mining frequent sequential patterns under a similarity constraint. In *IDEAL*, pp. 1–6.

Garofalakis, M. N., R. Rastogi, et K. Shim (1999). SPIRIT : Sequential pattern mining with regular expression constraints. In *The VLDB Journal*, pp. 223–234.

Guralnik, V. et G. Karypis (2001). A scalable algorithm for clustering sequential data. In *ICDM '01 : Proceedings of the 2001 IEEE International Conference on Data Mining*, Washington, DC, USA, pp. 179–186. IEEE Computer Society.

Hartigans, J. (1975). *Clustering Algorithms*. John Wiley and Sons, Inc.

Jianhua Zhu, Z. W. (2005). Fast : A novel protein structure alignment algorithm. Volume 58, Bioinformatics Program, Boston University, Boston, Massachusetts ; Biomedical Engineering Department, Boston University, Boston, Massachusetts.

Kum, H.-C. (2004). *Approximate Mining of Consensus Sequential Patterns*. Ph. D. thesis, University of North Carolina.

Kum, H.-C., J. Pei, W. Wang, et D. Duncan (2003). Approxmap : Approximate mining of consensus sequential patterns. In *SDM*.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. Technical Report 8.

Mannila, H. et P. Ronkainen (1997). Similarity of event sequences. In *TIME '97 : Proceedings of the 4th International Workshop on Temporal Representation and Reasoning (TIME '97)*, Washington, DC, USA, pp. 136. IEEE Computer Society.

Moen, P. (2000). *Attributs, Event Sequence, and Event Type Similarity Notions for Data Mining*. Ph. D. thesis, University of Helsinki, Finland.

Morzy, T., M. Wojciechowski, et M. Zakrzewicz (1999). Pattern-oriented hierachical clustering. In *Advances in Databases and Information Systems*, pp. 179–190.

Pei, J., J. Han, B. Mortazavi-asl, H. Pinto, Q. Chen, U. Dayal, et M. chun Hsu (2001). Prefixspan : Mining sequential patterns efficiently by prefix-projected pattern growth. pp. 215–224.

Plantevit, M., S. Goutier, F. Guisnel, A. Laurent, et M. Teisseire (2007). Mining unexpected multidimensional rules. In *DOLAP '07 : Proceedings of the ACM tenth international workshop on Data warehousing and OLAP*, pp. 89–96.

Sequeira, K. et M. J. Zaki (2002). Admit : anomaly-based data mining for intrusions. In *KDD*, pp. 386–395.

Tversky, A. (1977). *Pschological Review* (84), 327–352.

Summary

In the field of knowledge extraction, comparing the similarity of objects is an essential task, for example to identify regularities or to build homogeneous clusters of objects. In the case of sequential data seen in various fields of application

(e.g. series of customers purchases, Internet navigation) this problem (i.e. comparing the similarity of sequences) is very important. There are already some similarity measures as Edit distance and LCS suited to simple sequences, but these measures are not relevant in the case of complex sequences composed of sets of items, as is the case of sequential patterns. In this paper we propose a new similarity measure taking the characteristics of sequential patterns into account. S^2MP is an adjustable measure depending on the importance given to each characteristic of sequential patterns according to context, which is not the case of existing measures. Our measure has been validated by experiments on different data sets.