Fouille de données pour associer des noms de sessions aux articles scientifiques

Solen Quiniou, Peggy Cellier, Thierry Charnois

Fouille de données pour associer des noms de sessions aux articles scientifiques

Solen Quiniou[1] Peggy Cellier[2] Thierry Charnois[3]
(1) LINA, Université de Nantes, 2 rue de la Houssinière, BP 92208, 44322 Nantes Cedex 3
(2) INSA de Rennes, IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France
(3) LIPN, CNRS UMR 7030, Université Paris 13 Sorbonne Paris Cité, 93430 Villetaneuse
solen.quiniou@univ-nantes.fr, peggy.cellier@irisa.fr, thierry.charnois@lipn.univ-paris13.fr

Résumé. Nous décrivons dans cet article notre participation à l'édition 2014 de DEFT. Nous nous intéressons à la tâche consistant à associer des noms de session aux articles d'une conférence. Pour ce faire, nous proposons une approche originale, symbolique et non supervisée, de découverte de connaissances. L'approche combine des méthodes de fouille de données séquentielles et de fouille de graphes. La fouille de séquences permet d'extraire des motifs fréquents dans le but de construire des descriptions des articles et des sessions. Ces descriptions sont ensuite représentées par un graphe. Une technique de fouille de graphes appliquée sur ce graphe permet d'obtenir des collections de sous-graphes homogènes, correspondant à des collections d'articles et de noms de sessions.

Abstract. In this paper, we present a proposition based on data mining to tackle the DEFT 2014 challenge. We focus on task 4 which consists of identifying the right conference session for scientific papers. The proposed approach is based on a combination of two data mining techniques. Sequence mining extracts frequent phrases in scientific papers in order to build paper and session descriptions. Then, those descriptions of papers and sessions are used to create a graph which represents shared descriptions. A graph mining technique is applied on the graph in order to extract a collection of homogenous sub-graphs corresponding to sets of papers associated to sessions.

Mots-clés : Fouille de données, fouille de séquences, fouille de graphes, catégorisation d'articles.

Keywords: Data Mining, Sequence Mining, Graph Mining, Paper Categorisation.

1 Introduction

Nous présentons dans cet article notre participation à la tâche 4 de l'édition 2014 de DEFT [1]. Cette tâche consiste à déterminer pour un ensemble d'articles publiés dans des éditions passées de la conférence TALN, dans quelle session ces articles ont été présentés.

L'approche originale que nous proposons pour cette tâche utilise des méthodes de fouille de données. C'est une approche non supervisée qui combine des méthodes de fouille de séquences et de fouille de graphes. Cette approche s'inspire de travaux menés antérieurement sur l'utilisation de la fouille de graphes pour détecter des sous-ensembles de phrases cohérentes dans des textes (Quiniou *et al.*, 2012).

Plus précisément, la fouille de séquences permet d'extraire des expressions fréquentes dans un article qui, combinées aux mots-clefs de l'article, donnent une description de celui-ci. Par apprentissage, on construit aussi des descriptions pour les sessions. Ces descriptions d'articles et de sessions sont ensuite représentées dans un graphe. Ce graphe est exploité via une technique de fouille de graphes qui permet d'obtenir des collections de sous-graphes homogènes. Les collections de sous-graphes homogènes correspondent à des regroupement d'articles et de sessions ayant des descriptions proches.

Dans la suite de l'article chaque étape est détaillée. En section 2, un rappel est fait sur les notions de fouille de données utiles à notre approche. En particulier, deux méthodes de fouille de données sont présentées : la fouille de données séquentielles et la fouille de graphes. En section 3, la chaîne de traitement est détaillée. Enfin, la section 4 présente les différents résultats expérimentaux obtenus en fonction des stratégies choisies.

1. http://deft.limsi.fr/2014/

2 Notions de fouille de données

2.1 Fouille de motifs séquentiels

La fouille de motifs séquentiels (Agrawal & Srikant, 1995) est un champ de la fouille de données ayant pour but la découverte de régularités dans des données se présentant sous forme de séquences. Plusieurs algorithmes (Srikant & Agrawal, 1996; Zaki, 2001; Pei *et al.*, 2001; Yan *et al.*, 2003; Nanni & Rigotti, 2007; Gomariz *et al.*, 2013) ont été proposés pour extraire les motifs séquentiels. Dans cette partie nous introduisons les concepts de base de la fouille de motifs séquentiels utilisés pour le défi.

En fouille de données séquentielles, une séquence S est une liste ordonnée de littéraux appelés *items*, notée $s = \langle i_1 \dots i_m \rangle$. Par exemple, $\langle a\ b\ a\ c \rangle$ est une séquence de quatre items. Une séquence $S_1 = \langle i_1 \dots i_n \rangle$ est dite *incluse* dans une autre séquence $S_2 = \langle i'_1 \dots i'_m \rangle$ s'il existe des entiers $1 \leq j_1 < ... < j_n \leq m$ tels que $i_1 = i'_{j_1}$,..., $i_n = i'_{j_n}$. La séquence S_1 est alors appelée une sous-séquence de S_2 et S_2 est alors appelée une super-séquence de S_1, noté $S_1 \preceq S_2$. Par exemple, $\langle a\ a\ c \rangle$ est incluse dans $\langle a\ a\ b\ a\ c\ d \rangle$.

Une base de séquences, notée SDB, est un ensemble de tuples (sid, S), où sid est un identifiant de séquence, et S est une séquence. Par exemple, la table 1 décrit une base de séquences composée de quatre séquences. Le *support* d'une séquence S_1 dans une base de données de séquences SDB, noté $sup(S_1)$, est le nombre de tuples de la SDB contenant S_1. Par exemple, dans la table 1, $sup(\langle a\ c \rangle) = 3$, car les séquences 1, 2 et 3 contiennent $\langle a\ c \rangle$. Un motif séquentiel *fréquent* est un motif ayant une support supérieur ou égal à un seuil appelé $minsup$.

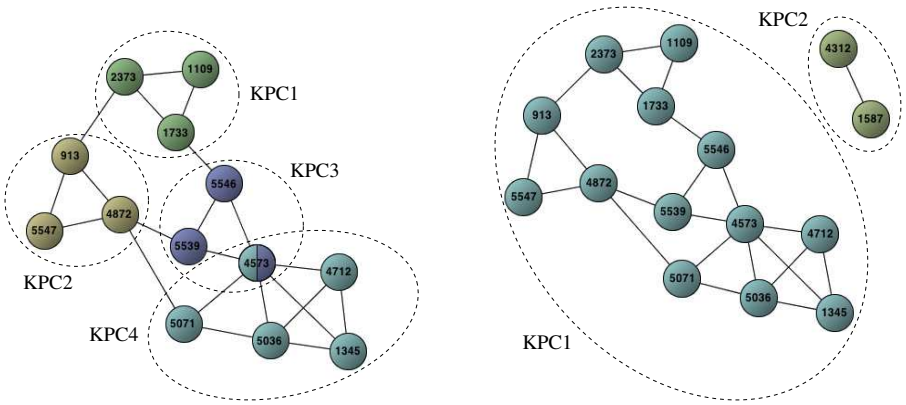
Identifiant	Séquence
1	$\langle a\ b\ c\ d \rangle$
2	$\langle a\ d\ c\ b \rangle$
3	$\langle a\ b\ d\ b\ c \rangle$
4	$\langle a\ d\ b\ b \rangle$

TABLE 1 – Exemple de SDB.

Dans la pratique, le nombre de motifs séquentiels fréquents peut être important. Une façon de réduire leur nombre est l'utilisation de contraintes (Dong & Pei, 2007). Une contrainte permet de focaliser la recherche en fonction des centres d'intérêts de l'utilisateur et limite ainsi le nombre de motifs séquentiels extraits en éliminant les motifs non-pertinents. Un exemple très classique de contrainte, que nous avons déjà introduite, est celle de support minimum qui permet de restreindre l'ensemble des motifs extraits aux motifs fréquents selon un seuil. Dans cette tâche nous utilisons une autre contrainte le *gap* qui permet de maîtriser l'intervalle de temps entre deux items d'un motif. Un motif séquentiel avec contrainte de gap $[M, N]$, noté $P_{[M,N]}$ est un motif tel qu'au minimum M items et au maximum N items sont présents entre chaque item voisin du motif dans les séquences correspondantes. Par exemple, dans la table 1, $P_{[0,2]} = \langle a\ c \rangle$ et $P_{[2,4]} = \langle a\ c \rangle$ sont deux motifs séquentiels avec des contraintes de gap différentes. $P_{[0,2]}$ est contenu dans deux séquences, les séquences 1 et 2, tandis que $P_{[2,4]}$ est contenu dans une seule séquence : la séquence 3.

2.2 Fouille de graphes sous contraintes

La fouille de graphes (Washio & Motoda, 2003) est une technique de fouille utilisée pour extraire des connaissances à partir de données représentées sous forme de graphes. Dans cet article, nous nous intéressons à l'extraction d'un certain type de motifs à partir de graphes attribués (des attributs sont associés aux sommets de ces graphes), à savoir les *collection de k-cliques percolées* (CoHoP par la suite) (Mougel *et al.*, 2012). Notons que nous avons déjà utilisé ce type de motifs pour extraire des sous-parties cohérentes de textes représentés sous forme de graphes (Quiniou *et al.*, 2012). Dans cette section, nous présentons plus formellement les deux principales notions sur lesquelles s'appuie cette technique particulière de fouille de graphe : les k-PC et les CoHoP.



(a) $k = 3$ (b) $k = 2$

FIGURE 1 – Exemple de CoHoP extraite à partir des attributs $\{a_1, a_2\}$, pour deux valeurs de k

2.2.1 k-cliques percolées (k-PC)

Dans un graphe, une *k-clique* est un ensemble de k sommets dans lequel toutes les paires de sommets sont connectées deux à deux par une arête. La notion de *k-clique percolée* (k-PC) peut être vue comme une version relâchée du concept de clique. Une k-PC a été définie par (Derenyi *et al.*, 2005) comme étant l'union de toutes les k-cliques connectées par des chevauchements de $k-1$ sommets. Ainsi, dans une k-PC, chaque sommet d'une k-PC peut être atteint par n'importe quel autre sommet de cette k-PC par un chemin de sous-ensembles de sommets bien connectés (les k-cliques).

Dans la figure 1a, il y a quatre k-PC ($k = 3$) : {1109, 1733, 2373}, {913, 4872, 5547}, {4573, 5539, 5546} et {1345, 4573, 4712, 5036, 5077}. Les trois premières k-PC contiennent une seule 3-clique alors que la dernière k-PC contient cinq 3-cliques (*e.g.*, {1345, 4712, 5036} et {1345, 4712, 4573}). Revenons sur la création de cette dernière k-PC. Nous pouvons tout d'abord constater que les sommets 1345, 4573, 4712 et 5036 sont directement connectés les uns aux autres : ils appartiennent ainsi à la même k-PC. Le sommet 5071 appartient également à cette k-PC puisqu'il est accessible à partir de chacun des quatre sommets précédents, par une série de k-cliques se chevauchant (le paramètre k a un impact sur le nombre de sommets à considérer dans les k-cliques ; dans cet exemple, sa valeur est fixée à 3) : par exemple, pour aller du sommet 5071 au sommet 4712, un chemin de 3-cliques se chevauchant peut être {4712, 4573, 5036} suivi de {4573, 5036, 5071} (avec $k = 3$, les chevauchements de 3-cliques contiennent deux sommets). En revanche, le sommet 4872 n'appartient pas à cette k-PC. En effet, pour cela il faudrait qu'il y ait une 3-clique entre les sommets 4573, 5071 et 4872, ce qui n'est pas le cas.

2.2.2 Collections de k-PC homogènes (CoHoP)

Une *collection de k-PC homogènes* (CoHoP) a été définie par (Mougel *et al.*, 2012) comme étant un ensemble de sommets tels que, étant donnés k, α et γ des entiers positifs définis par des utilisateurs :

— tous les sommets sont *homogènes*, c'est-à-dire qu'ils partagent au moins α attributs ;
— la CoHoP contient au moins γ k-PC ;
— toutes les k-PC ayant les mêmes attributs sont présentes dans la CoHoP (contrainte de *maximalité*).

La figure 1a représente ainsi une CoHoP extraite à partir de l'ensemble d'attributs $\{a_1, a_2\}$; comme vu dans la section 2.2.1, elle contient quatre k-PC ($\alpha = 2, k = 3, \gamma = 4$). Il est à noter que, contrairement au calcul des k-PC, l'extraction des CoHoP dépend fortement de l'ensemble d'attributs associés aux sommets du graphe. Sur la figure 1a, les ensembles d'attributs des sommets ne sont pas illustrés (pour ne pas surcharger la figure) mais chaque sommet est en fait étiqueté par un ensemble d'attributs qui contient au moins les attributs a_1 et a_2. En effet, cette CoHoP a été extraite à partir de ces deux attributs.

Les trois paramètres - k, α et γ - ont un impact important sur la structure des CoHoP extraites. Comme précisé précédemment, le paramètre α fixe le nombre minimal d'attributs communs aux sommets des CoHoP extraites et le paramètre γ fixe le nombre minimal de k-PC présentes dans les CoHoP. Le paramètre k a également un impact important sur la structure des CoHoP extraites. En effet, augmenter sa valeur a pour conséquence d'augmenter le degré de cohésion entre les sommets appartenant à une même k-PC. La figure 1b représente la CoHoP extraite à partir du même ensemble d'attributs

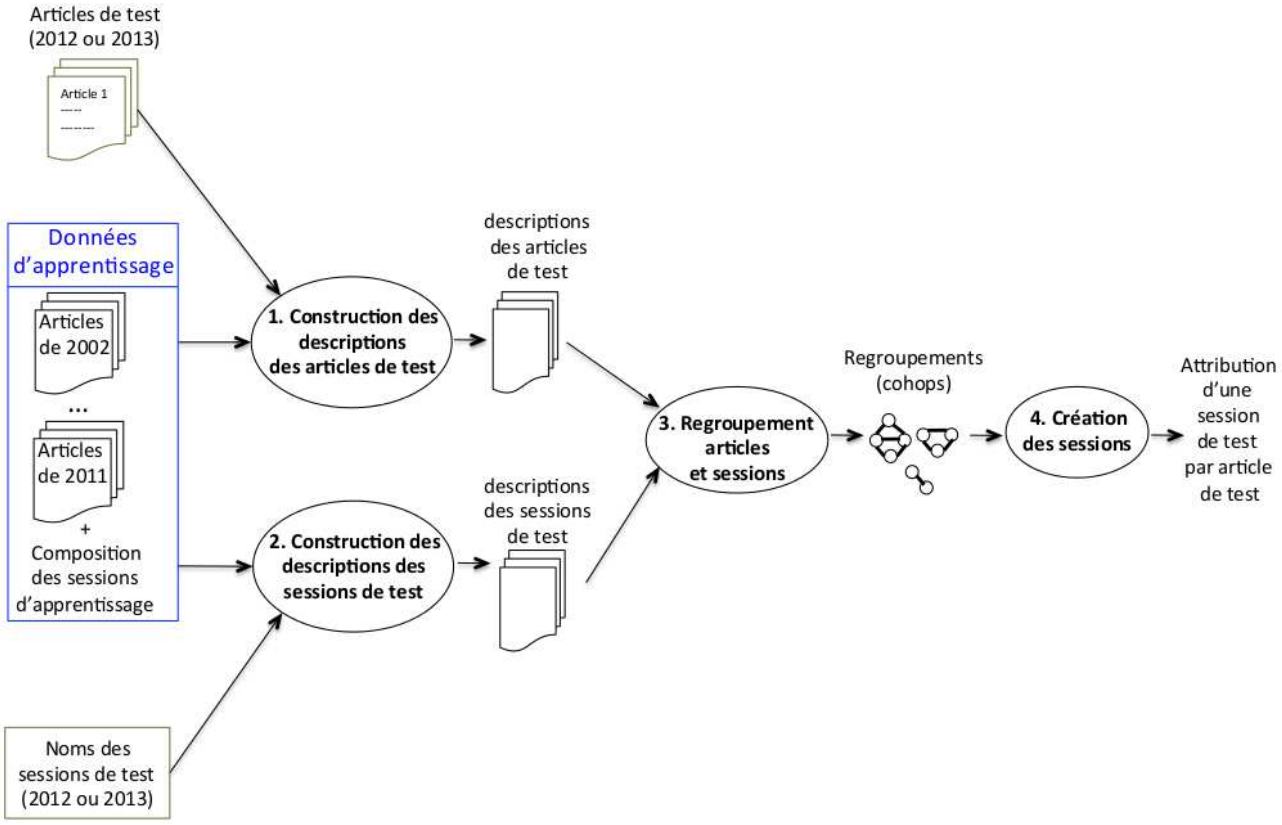


FIGURE 2 – Processus global pour associer des noms de sessions à des articles scientifiques d'une même année.

que celle illustrée par la figure 1a mais en fixant cette fois $k = 2$. Cette CoHoP comporte maintenant 15 sommets répartis en seulement deux k-PC, la plus grosse k-PC (KPC_1) correspondant en fait aux quatre k-PC de la figure 1a. Ainsi, le choix de la valeur de k permet de choisir le degré de cohésion souhaité entre les sommets de chaque k-PC. En effet, un plus grand nombre de sommets doit être directement relié les uns aux autres lorsque la valeur de k augmente (la valeur de k représente ce nombre minimal de sommets).

3 Chaîne de traitement mise en place

La tâche consiste à déterminer la session scientifique dans laquelle un article de conférence a été présenté. Les articles à catégoriser, que nous appelons articles de test, proviennent des éditions 2012 et 2013 de la conférence TALN. Les noms de sessions, que nous appelons sessions de test, ainsi que le nombre d'articles de test qu'elles contiennent sont fournis. De plus, nous disposons de données d'apprentissage, à savoir des articles et leur sessions associées lors d'éditions passées de la conférence (2002, 2005, 2007, 2008, 2009, 2010 et 2011).

La figure 2 décrit la chaîne de traitement mise en place pour réaliser la tâche. Cette chaîne se découpe en 4 grandes étapes détaillées dans cette section :

1. la construction des descriptions des articles ;
2. la construction des descriptions des sessions ;
3. le regroupements des articles et des sessions dans des clusters appelés CoHoP ;
4. l'association d'un nom de session à chaque article de test.

Notons que les articles sont traités par année. Ainsi l'affectation des sessions aux articles de 2012 se fait séparément de l'affectation des sessions aux articles de 2013.

3.1 Construction de la description d'un article

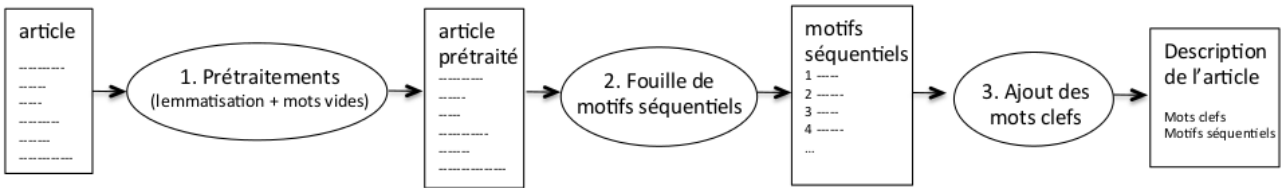


FIGURE 3 – Traitement pour associer une description à un article.

Pour chaque article on crée une description comme défini à la figure 3. Cette description, qui représente donc un article, comporte la liste des mots clefs associés à l'article ainsi que les motifs séquentiels fréquents extraits à partir de l'article. Nous détaillons ci-dessous les 3 traitements qui permettent d'obtenir cette représentation.

3.1.1 Prétraitements des articles

L'étape de prétraitement consiste, dans un premier temps, à normaliser chaque article. Les fichiers de type *txt* fournis par les organisateurs étant issus de l'océrisation de fichiers au format pdf, il subsiste des césures de mots et des fins de ligne ne correspondant pas à de réelles fins de paragraphe. Ces différentes marques sont supprimées afin de reformer les paragraphes originaux. En outre, il n'est retenu de l'article que la partie précédant les références bibliographiques (identifiées par une ligne débutant par un nombre, optionnel, suivi du mot *références*, ou *references*, ou *bibliographie*).

Après ce premier prétraitement, l'outil *TreeTagger* (Schmid, 1995) est utilisé pour segmenter les mots de l'article et les remplacer par leur lemme.

Les « mots vides » sont ensuite éliminés en utilisant la liste de Jean Véronis [2]. L'objectif étant de ne conserver que des mots spécifiques à l'article, nous avons ajouté à cette liste un ensemble de mots a priori non caractéristiques comme *réaliser, résultat, référence, tableau, etc.*

Le dernier prétraitement consiste essentiellement à découper l'article en un ensemble de séquences qui seront ensuite « fouillées ». Pour ce faire, la segmentation est réalisée selon tout type de ponctuation (point, point virgule, point d'interrogation, etc., mais aussi les ponctuations faibles : virgule, parenthèse, guillemet, accolade, etc.). Une séquence représente ainsi une partie de la phrase, qui peut être un constituant, un virgulo ou encore un terme complexe. Afin d'homogénéiser l'ensemble des articles pour la fouille, il est nécessaire d'obtenir une même représentation pour un mot donné. Le problème se pose par exemple pour les mots composés qu'on trouve parfois avec un trait d'union et parfois sans. Un autre problème concerne les erreurs de lemmatisation, et les ambiguïtés issus de *TreeTagger* (*foi|fois* est le lemme produit pour le mot fois). Pour ce faire, les traits d'union, les apostrophes et les symboles 'l' sont remplacés par un espace. Enfin, les caractères spéciaux (par exemple, *æ, œ* peuvent être codés avec un ou deux caractères) sont homogénéisés et les mots comportant plus de la moitié de caractères qui ne sont pas des lettres sont éliminés (cas typique des identifiants).

3.1.2 Fouille de motifs séquentiels

Pour effectuer la fouille de motifs sur chaque article prétraité, nous utilisons DMT4sp (Nanni & Rigotti, 2007) développé par Christophe Rigotti. Cet outil nous permet de définir plusieurs contraintes sur les motifs séquentiels extraits : leur longueur, leur fréquence (en fixant le seuil $minsup$) et la contrainte gap (en choisissant les valeurs $[M, N]$). Nous avons fixé la longueur des motifs à 2 mots minimum et 5 mots maximum. Nous avons choisi une valeur de 2 pour le minsup (c'est-à-dire que le motif doit apparaître dans au moins 2 phrases pour être considéré comme fréquent), choix empirique mais justifié par la petite taille de chaque article. En ce qui concerne la contrainte de gap, nous avons fixé la valeur à $[0, 0]$ (c'est-à-dire des mots contigus), l'idée étant que les motifs intéressants pour la tâche du défi sont des termes complexes (*e. g. analyse syntaxique, apprentissage supervisé*). Ainsi, le nombre moyen de motifs extraits avec ces contraintes à partir du corpus de test (articles TALN de l'année 2012 et 2013) varie entre 172 pour le plus petit ensemble de motifs à 504 pour le plus grand, la moyenne se situant à 281 motifs par article. Pour l'année 2013, les nombres sont similaires : 243 motifs sont extraits en moyenne, le plus petit ensemble comportant 109 motifs et le plus grand 486 motifs.

2. http ://torvald.aksis.uib.no/corpora/1999-1/0042.html

3.1.3 Ajout des mots-clefs

La description d'un article se compose de ces motifs, auxquels sont ajoutés les mots-clefs de l'article, ainsi que le nom de la session. Mots-clefs et noms de session sont prétraités de la même manière qu'un article (voir section 3.1).

3.2 Construction de la description d'une session

L'objectif de cette étape est d'obtenir pour chacune des sessions de test une description qui sera ensuite utilisée pour calculer les regroupements des sessions avec les articles. La description d'une session est un ensemble de mots ou d'expressions qui sont des synonymes ou des mots/expressions souvent associés au nom de la session. Le processus qui associe une description à une session de test est un processus en deux étapes détaillées dans cette partie.

3.2.1 Construction des descriptions des sessions d'apprentissage

Avant d'associer une description à une session **de test**, nous commençons par associer à chaque nom de session du corpus **d'apprentissage** une description qui est composée : des mots-clefs des articles qui ont été présentés dans cette session et du nom de la session lui-même. Cette étape est décrite à la figure 4. Dans un souci d'homogénéité les mots-clefs ainsi que les noms de session sont prétraités comme décrit à la section 3.1 (cf Prétraitements (lemmatisation + mots vides)). Une fois ce prétraitement fait, la description de chaque session est créée en prenant les mots-clefs prétraités des articles de la session (cf Regroupement des mots-clefs par session). La table 2 montre deux exemples de descriptions de sessions d'apprentissage : "alignement" et "recherche d'information". Notons que dans un souci de lisibilité, dans cet exemple ces deux descriptions sont données sans prétraitement.

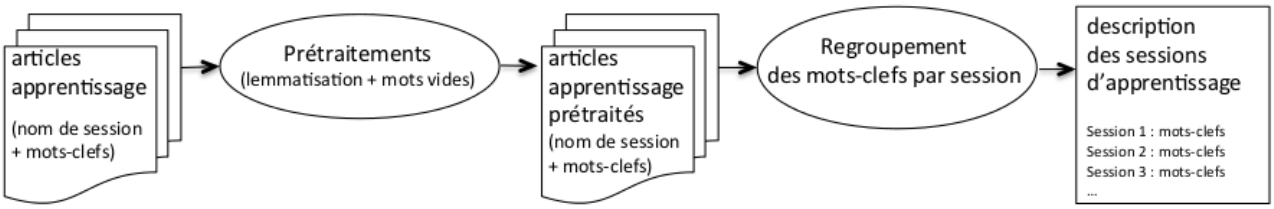


FIGURE 4 – Calcul des descriptions des sessions des données d'apprentissage.

3.2.2 Construction des descriptions des sessions de test

Une fois les descriptions des sessions d'apprentissage calculées, nous cherchons à construire une description pour chaque nom de session de test.

On initialise la description de chaque session de test avec la description de la session d'apprentissage correspondante si elle existe dans les données d'apprentissage.

On enrichit ensuite les descriptions des sessions de test en ajoutant les mots-clefs contenant les mots dits "significatifs" des noms de session. Ces mots-clefs sont issus des descriptions des sessions d'apprentissage. La liste des mots "significatifs" des noms de session a été produite à la main et est donnée à la table 3. On notera que la plupart des mots dits "significatifs" sont le nom de la session lui-même. Toutefois, cette étape est indispensable pour certains noms de session comme "banques d'arbres" qui n'apparait dans aucun texte sous la forme française mais uniquement sous la forme anglaise "treebank".

Notons que dans le cas d'un nom de session regroupant deux sessions, par exemple "traduction|alignement" ou encore "morphologie|segmentation", une description est calculée pour chacune des sessions puis l'union des descriptions est associée à la session composée. Ainsi, dans le cas de la session portant le nom "traduction|alignement", on obtient : $description(\text{"}traduction|alignement\text{"}) = description(\text{"}traduction\text{"}) \cup description(\text{"}alignement\text{"})$.

Nom de la session d'apprentissage	Description
alignement	alignement de phrases, corpus parallèle, recherche cross-lingue d'information, alignement, traduction de collocations, extraction de collocations, parsing, alignement de textes, alignement au niveau des mots, concordancier bilingue, traduction automatique, traduction probabiliste, corpus bilingue, alignement de documents, table de traduction, traduction automatique statistique, contexte source, dépendances syntaxiques, corpus comparable, extraction de lexiques bilingues, points d'ancrage, unités lexicales complexes, désambiguïsation lexicale, world wide web, terminologie, multilinguisme, paraphrase sous-phrastique, corpus parallèle monolingue, hybridation, traduction statistique, modèles de traduction à base de segments, modèles d'alignement mot-à-mot, alignement sous-phrastique, traduction automatique par fragments
recherche d'information	recherche d'information, système de question-réponse, focus, patron d'extraction, grammaire formelle, grammaire catégorielle, description d'arbres, traduction automatique français-anglais, base d'exemples, partage de révision, représentation interlingue, coédition de texte et de graphe unl, communication multilingue, indexation sémantique recherche documentaire, redondance minimale, ontologie, système de questions-réponses évaluation des systèmes de questions-réponses, extraction de réponse, recherche sur le web qristal, systèmes de questions-réponses, repérage d'énoncés définitoires, médecine, patrons lexico-syntaxiques
...	...

TABLE 2 – Table d'association : nom de session et "mots significatifs".

3.3 Regroupement des articles et des sessions

Pour créer des regroupements d'articles et de sessions, nous utiliserons la fouille de CoHoP, telle que présentée dans la section 2.2. Pour cela, il est tout d'abord nécessaire de construire un graphe regroupant les articles et les sessions pour pouvoir ensuite en extraire les CoHoP à partir desquelles les sessions seront créées (voir section suivante). Nous présentons ces deux étapes dans la suite des paragraphes.

3.3.1 Construction du graphe regroupant articles et sessions

Le graphe regroupant les articles et les sessions est un graphe non orienté, dans lequel chaque sommet représente un article ou une session et dans lequel une arête est créée entre 2 sommets s'ils partagent au moins deux attributs communs (les attributs des sommets du graphe correspondent, pour les articles : aux motifs et aux mots-clefs de leur description et, pour les sessions : aux mots-clefs de leur description). Nous avons choisi empiriquement ce nombre minimal d'au moins deux attributs communs, en testant les valeurs suivantes sur le corpus d'apprentissage : « au moins un attribut commun », « au moins deux attributs communs » et « au moins trois attributs communs ». La valeur choisie (« au moins 2 attributs communs ») apparaît comme le meilleur compromis entre avoir suffisamment d'arêtes entre les sommets pour extraire des CoHoP intéressante et ne pas créer trop d'arêtes pour limiter le nombre total de CoHoP. De plus, la construction d'une arête entre 2 sommets est autorisée seulement si ces sommets correspondent soit à deux articles, soit à un article et à une session (on interdit la construction d'arêtes entre deux sessions).

On obtient ainsi, pour l'année 2012, un graphe de 29 sommets (22 articles et 7 sessions) et 5 814 arêtes et, pour l'année 2013, un graphe de 41 sommets (32 articles et 9 sessions) et 7 473 arêtes.

3.3.2 Extraction des CoHoP

Pour l'extraction des CoHoP sur le graphe construit précédemment, nous avons utilisé *CoHoP Miner* (Mougel *et al.*, 2012). Comme vu dans la section 2.2, trois paramètres sont alors à fixer : k, α et γ.

Le paramètre k représente l'ordre des k-cliques qui vont former les k-PC. Ce paramètre contraint le degré de cohésion souhaité à l'intérieur de chaque CoHoP mais également le nombre minimal de sommets appartenant à une CoHoP. Nous souhaitons pouvoir extraire aussi bien des grandes cohop que des petites CoHoP contenant un seul nom d'article et un

Nom de la session de test	Mots significatifs associés	Nombre de mots/expressions dans la description
alignement	alignement	37
analyse	analyse	40
apprentissage	apprentissage	26
banques d'arbres	banques d'arbres, treebank, arbre	8
connaissances ǀ discours	connaissances, discours	32
entités nommées	entités nommées	16
exploitation de corpus	exploitation de corpus, corpus	18
extraction d'information ǀ extraction de relations	extraction de relations, relation, extraction d'information	40
fouille de textes ǀ applications	fouille de textes, applications	30
lexique	lexique	49
lexique ǀ corpus	lexique, corpus	77
morphologie ǀ segmentation	morphologie, segmentation	91
réécriture	réécriture	2
sémantique	sémantique	101
syntaxe	syntaxe	93
traduction ǀ alignement	traduction, alignement	82

TABLE 3 – Nom des sessions de test, "mots significatifs" associés et taille de la description.

seul nom de session : c'est pourquoi nous avons choisi $k = 2$.

Le paramètre α représente le nombre minimal d'attributs communs à tous les sommets de chaque CoHoP. Pour pouvoir extraire des CoHoP à partir d'un seul attribut, nous avons choisi $\alpha = 1$. Cela permet, par exemple, d'extraire des CoHoP correspondant à un nom de session telle que *traduction*.

Le paramètre γ représente le nombre minimal de k-PC composant chaque CoHoP. Afin de pouvoir extraire des CoHoP composées d'une seule k-PC, nous avons choisi $\gamma = 1$.

Nous extrayons ainsi 129 CoHoP, pour l'année 2012, et 283 CoHoP, pour l'année 2013.

3.4 Création des sessions

La création des sessions constitue la dernière étape de notre chaîne de traitement. Dans notre approche, il s'agit en fait d'attribuer un nom de session à chacun des articles de l'année considérée. Nous ne cherchons pas à répartir les articles dans les sessions, ce qui a pour conséquences qu'un article peut n'être associé à aucun nom de session et qu'une session peut ne contenir aucun article.

L'attribution des noms de sessions aux articles est réalisée à partir des CoHoP précédemment extraites. L'ensemble des CoHoP peut être décomposé en trois sous-ensembles disjoints :

— les CoHoP ne contenant que des noms d'articles (ensemble noté $\mathcal{C}_A$) ;
— les CoHoP contenant un seul nom de session et un ou plusieurs noms d'articles (ensemble noté $\mathcal{C}_{A+1S}$) ;
— les CoHoP contenant au moins deux noms de sessions et un ou plusieurs noms d'articles (ensemble noté $\mathcal{C}_{A+S}$).

Nous avons proposé 3 stratégies pour attribuer des noms de sessions aux articles (chaque stratégie correspond à un *run* soumis) ; elles se distinguent par les sous-ensembles de CoHoP sur lesquels elles s'appuient.

3.4.1 Stratégie 1 : session la plus représentée dans $\mathcal{C}_{A+1S}$

La première approche consiste à ne considérer que les CoHoP contenant un seul nom de session et un ou plusieurs noms d'articles, c'est-à-dire l'ensemble $\mathcal{C}_{A+1S}$. Pour chaque article de test, a, on choisit comme session celle qui apparaît dans le plus grand nombre de CoHoP de $\mathcal{C}_{A+1S}$:

$$session(a) = arg \max_{s \in S} N_{\mathcal{C}_{A+1S}}(c_{a,s}),$$

avec S l'ensemble des sessions de l'année considérée et $N_{\mathcal{C}(c_{a,s})}$ le nombre de CoHoP de l'ensemble C qui contiennent l'article a et la session s.

Une première remarque concernant cette stratégie est qu'elle n'est pas complète. En effet, si un article n'apparaît dans aucune CoHoP de $\mathcal{C}_{A+1S}$ alors aucune session ne lui est attribuée. Une seconde remarque concerne le cas où, pour un article donné, plusieurs sessions apparaissent avec celui-ci dans le même nombre de CoHoP de $\mathcal{C}_{A+1S}$. Dans ce cas, la première session trouvée sera affectée à l'article.

3.4.2 Stratégie 2 : session la plus représentée dans $\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}$

La seconde approche consiste à considérer, en plus des CoHoP contenant un seul nom de session et un ou plusieurs noms d'articles ($\mathcal{C}_{A+1S}$), les CoHoP contenant au moins deux noms de sessions et un ou plusieurs noms d'articles ($\mathcal{C}_{A+S}$). Pour chaque article de test, a, on choisit alors de lui affecter la session s qui apparaît dans le plus de CoHoP de l'ensemble $\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}$:

$$session(a) = arg \max_{s \in S} N_{\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}}(c_{a,s}).$$

Les deux remarques faites pour la stratégie 1 s'appliquent également pour la stratégie 2.

3.4.3 Stratégie 3 : session la plus représentée et prise en compte du nombre d'articles par session

Dans cette troisième approche, nous prenons en compte le nombre d'articles à retrouver dans chaque session pour interdire l'attribution d'une session à un article si cette session a déjà été attribuée au nombre d'articles recherché. L'attribution des sessions aux articles ne se fait donc plus de manière indépendante ; l'ordre dans lequel les articles sont considérés devient important. Pour ce faire, nous procédons de manière itérative, en 3 étapes, en considérant un sous-ensemble différent de CoHoP lors de chaque étape :

1. on attribue des noms de sessions aux articles, à partir de $\mathcal{C}_{A+1S}$;
2. on attribue des noms de sessions aux articles sans nom de session, à partir de $\mathcal{C}_{A+S}$;
3. on attribue des noms de sessions aux articles sans nom de session, à partir de $\mathcal{C}_A$.

Lors de la première étape, on commence par trier les articles selon le nombre décroissant de CoHoP de $\mathcal{C}_{A+1S}$ dans lesquelles ils apparaissent. Pour chaque article de test, a, pris dans l'ordre établi par le tri, on attribue à l'article la première session s qui n'est pas déjà remplie, s'il en existe une, en ayant au préalable trié les sessions selon le nombre décroissant de CoHoP de $\mathcal{C}_{A+1S}$ dans lesquelles elles apparaissent avec l'article :

$$session(a) = arg \max_{s \in S, |s| < nbArt(s)} N_{\mathcal{C}_{A+1S}}(c_{a,s}),$$

avec $nbArt(s)$ le nombre d'articles à retrouver pour la session s.

Pour la seconde étape, on procède comme pour la première étape mais en prenant cette fois en compte les CoHoP de $\mathcal{C}_{A+S}$. On cherche ainsi à attribuer un nom de session, s, aux articles de test, a, qui n'en ont pas encore, en considérant des CoHoP moins précises puisqu'elles contiennent plusieurs noms de sessions :

$$session(a) = arg \max_{s \in S, |s| < nbArt(s)} N_{\mathcal{C}_{A+S}}(c_{a,s}).$$

À l'issue des deux première étapes, les articles sans session le sont soit parce que les sessions avec lesquelles ils apparaissent dans des CoHoP de $\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}$ sont déjà remplies, soit parce qu'ils n'apparaissent dans aucune CoHoP de $\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}$. Lors de la troisième étape, on cherche alors à associer une session à ces articles, en utilisant les CoHoP de $\mathcal{C}_A$ qui ne contiennent que des noms d'articles. L'idée est de trouver, parmi les noms de sessions restantes, la session s qui est associée au plus grand nombre d'articles apparaissant avec l'article de a considéré, dans des CoHoP de $\mathcal{C}_A$:

$$session(a) = arg \max_{s \in S, |s| < nbArt(s), session(a')=s} N_{\mathcal{C}_A}(c_{a,a'}).$$

Les deux remarques faites pour les stratégies 1 et 2 s'appliquent également pour la stratégie 3.

4 Résultats des expérimentations et discussion

Pour la tâche 4 à laquelle nous avons participée, 5 équipes ont participé, chacune ayant soumis jusqu'à 3 essais. Au total cela représente 13 soumissions. Les scores pour la meilleure soumission de chaque équipe varient de 0,2778 à 1,000 (mesure : précision à 1) et les moyennes sont les suivantes : Moyenne=0,5926 ; Médiane=0,4815 et Ecart-type=0,2860.

La figure 5 (a) montre les résultats obtenus avec la stratégie 1 (session la plus représentée dans $\mathcal{C}_{A+1S}$). La figure 5 (b) montre les résultats obtenus avec la stratégie 2 (session la plus représentée dans $\mathcal{C}_{A+1S} \cup \mathcal{C}_{A+S}$). Enfin la figure 6 montre les résultats obtenus avec la stratégie 3 (prise en compte du nombre d'articles par session). On constate que les trois stratégies donnent des résultats approchant la médiane donnée par les organisateurs. La stratégie 3 est légèrement meilleure.

Les résultats semblent dépendre de la qualité de la description associée à une session. En effet, prenons la session "réécriture" qui n'était pas présente dans les données d'apprentissage et qui a une description très pauvre avec seulement 2 descripteurs : réécriture et réécriture graphe (cf table 3). Cette session obtient le score 0 quelle que soit la stratégie utilisée. Une piste pour améliorer les résultats obtenus est ainsi de travailler sur un enrichissement des descriptions des sessions permettant un meilleur regroupement avec les articles.

Une autre remarque concerne le nombre d'articles pour lesquels nos approches n'ont pas désigné de session. Avec la stratégie 1, 7 articles n'ont pas de session associée (5 en 2012 et 2 en 2013). Avec la stratégie 2, 9 articles n'ont pas de session associée (5 en 2012 et 4 en 2013). Avec la stratégie 3, 7 articles n'ont pas de session associée (3 en 2012 et 4 en 2013). Les stratégies définies lors de la tâche ne cherchent pas à optimiser la distribution des articles dans les sessions (le nombre d'articles par session étant connu). Une stratégie cherchant à optimiser cette distribution est une autre piste d'amélioration de l'approche proposée.

Micro-precision : 0.425925925925926
- **Micro-precision for year 2013 : 0.4375**
  - Precision for connaissances|discours : 0
  - Precision for entités nommées : 0
  - Precision for syntaxe : 1
  - Precision for sémantique : 0.5
  - Precision for traduction|alignement : 0.857142857142857
  - Precision for fouille de textes|applications : 0.333333333333333
  - Precision for morphologie|segmentation : 0.5
  - Precision for apprentissage : 0
  - Precision for lexique|corpus : 0.166666666666667
- **Macro-precision (sessions for 2013) : 0.37**3015873015873
- **Micro-precision for year 2012 : 0.40**9090909090909
  - Precision for alignement : 0.666666666666667
  - Precision for réécriture : 0
  - Precision for exploitation de corpus : 0
  - Precision for banques d'arbres : 0.5
  - Precision for extraction d'information|extr. de relations : 0.666..67
  - Precision for analyse : 0.666666666666667
  - Precision for lexique : 0.333333333333333
- **Macro-precision (sessions for 2012) : 0.40**4761904761905

Macro-precision (year) : 0.423295454545455
Macro-precision (session) : 0.386904761904762

(a) Stratégie 1

Micro-precision : 0.425925925925926
- **Micro-precision for year 2013 : 0.4375**
  - Precision for connaissances|discours : 0
  - Precision for entités nommées : 0
  - Precision for syntaxe : 1
  - Precision for sémantique : 0.5
  - Precision for traduction|alignement : 0.857142857142857
  - Precision for fouille de textes|applications : 0
  - Precision for morphologie|segmentation : 0.75
  - Precision for apprentissage : 0
  - Precision for lexique|corpus : 0.166666666666667
- **Macro-precision (sessions for 2013) : 0.36**3756613756614
- **Micro-precision for year 2012 : 0.40**9090909090909
  - Precision for alignement : 0.666666666666667
  - Precision for réécriture : 0
  - Precision for exploitation de corpus : 0
  - Precision for banques d'arbres : 0.5
  - Precision for extraction d'information|extr. de relations : 0.666..67
  - Precision for analyse : 0.666666666666667
  - Precision for lexique : 0.333333333333333
- **Macro-precision (sessions for 2012) : 0.40**4761904761905

Macro-precision (year) : 0.423295454545455
Macro-precision (session) : 0.381696428571429

(b) Stratégie 2

FIGURE 5 – Résultats des deux premières stratégies utilisées.

Références

AGRAWAL R. & SRIKANT R. (1995). Mining sequential patterns. In *Int. Conf. on Data Engineering* : IEEE.

DERENYI I., PALLA G. & VICSEK T. (2005). Clique percolation in random networks. *Physical Review Letters*, **94**, 160–202.

DONG G. & PEI J. (2007). *Sequence Data Mining*. Springer.

GOMARIZ A., CAMPOS M., MARÍN R. & GOETHALS B. (2013). Clasp : An efficient algorithm for mining frequent closed sequences. In J. PEI, V. S. TSENG, L. CAO, H. MOTODA & G. XU, Eds., *Proc. of the Pacific-Asia Conf. on*

Micro-precision : 0.444444444444444
- **Micro-precision for year 2013 : 0.375**
  - Precision for connaissances|discours : 0
  - Precision for entités nommées : 0
  - Precision for syntaxe : 0
  - Precision for sémantique : 0.5
  - Precision for traduction|alignement : 0.857142857142857
  - Precision for fouille de textes|applications : 0.333333333333333
  - Precision for morphologie|segmentation : 0.25
  - Precision for apprentissage : 0.5
  - Precision for lexique|corpus : 0.166666666666667
- **Macro-precision (sessions for 2013) : 0.28**968253968254
- **Micro-precision for year 2012 : 0.54**5454545454545
  - Precision for alignement : 0.666666666666667
  - Precision for réécriture : 0
  - Precision for exploitation de corpus : 0.333333333333333
  - Precision for banques d'arbres : 0.75
  - Precision for extraction d'informations|extraction de relations : 0.666666666666667
  - Precision for analyse : 0.666666666666667
  - Precision for lexique : 0.666666666666667
- **Macro-precision (sessions for 2012) : 0.53**5714285714286

Macro-precision (year) : 0.460227272727273
Macro-precision (session) : 0.397321428571429

FIGURE 6 – Résultats de la stratégie 3.

Advances in Knowledge Discovery and Data Mining, volume 7818 of *Lecture Notes in Computer Science*, p. 50–61 : Springer.

MOUGEL P.-N., RIGOTTI C. & GANDRILLON O. (2012). Finding collections of k-clique percolated components in attributed graphs. In *Proc. of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, p. 181–192.

NANNI M. & RIGOTTI C. (2007). Extracting trees of quantitative serial episodes. In *Knowledge Discovery in Inductive Databases 5th Int. Workshop KDID'06, Revised Selected and Invited Papers*, p. 170–188 : Springer-Verlag LNCS 4747.

PEI J., HAN J., MORTAZAVI-ASL B., PINTO H., CHEN Q., DAYAL U. & HSU M. (2001). Prefixspan : Mining sequential patterns by prefix-projected growth. In *ICDE*, p. 215–224 : IEEE Computer Society.

QUINIOU S., CELLIER P., CHARNOIS T. & LEGALLOIS D. (2012). Fouille de graphes sous contraintes linguistiques pour l'exploration de grands textes. In *Actes de la Conférence sur le Traitement Automatique des Langues Naturelles*, p. 253–266, Grenoble, France.

SCHMID H. (1995). Improvements in part-of-speech tagging with an application to german. In *Proc of the ACL SIGDAT-Workshop*.

SRIKANT R. & AGRAWAL R. (1996). Mining sequential patterns : Generalizations and performance improvements. In P. M. G. APERS, M. BOUZEGHOUB & G. GARDARIN, Eds., *EDBT*, volume 1057 of *LNCS*, p. 3–17 : Springer.

WASHIO T. & MOTODA H. (2003). State of the art of graph-based data mining. *SIGKDD Explorations*, **5**(1), 59–68.

YAN X., HAN J. & AFSHAR R. (2003). Clospan : Mining closed sequential patterns in large databases. In D. BARBARÁ & C. KAMATH, Eds., *SDM* : SIAM.

ZAKI M. J. (2001). SPADE : An efficient algorithm for mining frequent sequences. *Machine Learning Journal*, **42**(1/2), 31–60. special issue on Unsupervised Learning.