Construction et peuplement de structures hiérarchiques de concepts dans le domaine du e-tourisme

Nicolas Béchet, Marie-Aude Aufaure, Yves Lechevallier

Construction et peuplement de structures hiérarchiques de concepts dans le domaine du e-tourisme

Nicolas Béchet[1], Marie-Aude Aufaure[1 2], Yves Lechevallier[1]

[1] INRIA Paris-Rocquencourt
Adresse, France
[2] Ecole Centrale Paris, MAS Laboratory
Adresse, France
nicolas.bechet@inria.fr, Marie-Aude.Aufaure@ecp.fr,
yves.lechevallier@inria.fr

Résumé : Nous proposons dans cet article une méthode de construction et de peuplement automatique de structures hiérarchiques de concepts. Nous nous sommes particulièrement intéressés à la construction d'une structure hiérarchique de services proposés dans des hôtels à partir d'un jeu de données d'une application de e-tourisme. L'objectif est d'associer à chaque service un concept permettant une représentation commune de tous les services. Nos expérimentations sont effectuées à partir des ressources issues de partenaires spécialisés dans la réservation d'hôtels en ligne dont dispose la société Addictrip. La mise en place d'une structure conceptuelle est essentielle pour ces partenaires qui utilisent chacun leurs propres terminologies de description de services d'hôtels. En effet cela permet d'obtenir un espace de représentation commun afin de rendre comparable n'importe quel service provenant de ressources différentes. Notre approche se fonde sur la proximité littérale de termes contenus dans les services en présentant une mesure de proximité à base de n-grammes de caractères. Les résultats obtenus lors de nos expérimentations montrent la qualité de l'approche présentée et ses limites.

Mots-clés : Analyse de Document, Apprentissage Textuel, Apprentissage Supervisé, Méthodes Statistiques

1. Introduction

L'approche proposée dans cet article fait suite à une problématique rencontrée dans le cadre de travaux menés dans le domaine du e-commerce. Ces travaux consistent à agréger des informations relatives à des hôtels afin de permettre une classification de ces derniers. Précisons que l'objectif visé par cet article n'est pas de fournir une classification pertinente d'hôtels mais une description

cohérente de ces hôtels. Ces travaux sont menés en collaboration avec la société *Addictrip* [1] qui met à notre disposition les ressources fournies par leurs différents partenaires (*Booking*, *Splendia*, *Expedia*,*Venere*, *Fastbooking*, *Hotels.com*, ...) dont l'activité est la réservation d'hôtels en ligne. Les services associés à des hôtels sont des informations pertinentes nous permettant d'obtenir une classification. Néanmoins, la liste des services proposés par un hôtel diffère en fonction des partenaires de la société *Addictrip*. Un partenaire peut par exemple parler de "climatisation" alors qu'un second emploiera plutôt le terme "air conditionné", ou encore le choix du terme "wifi" pour l'un et "internet sans fil" pour un autre, etc.

Parmi les partenaires, le recueil de l'information se fait à partir de deux approches :
– Par un questionnaire possédant un nombre fini de services et qui est défini par un partenaire. Les hôteliers renseignant leurs hôtels doivent alors choisir dans cette liste les services représentant leur hôtel.
– Par une liste de services définie par l'hôtelier. Dans ce cas, il n'existe pas de liste prédéfinie de services et l'hôtelier définit sa liste de services en employant son propre vocabulaire.
Notre objectif est alors de proposer un espace de représentation commun qui doit permettre d'associer à tous les services de chaque hôtel un concept.
Les concepts devront également permettre de rassembler des services issus d'un même partenaire en généralisant ainsi l'information comme "climatisation individuelle" et "climatisation dans les chambres" rassemblés dans un concept "climatisation". De plus, les concepts proposés sont organisés hiérarchiquement afin de permettre différentes granularité.
Nos contributions présentées dans ce papier sont les suivantes.
– Définition et construction d'une structure hiérarchique de concepts de référence.
– A partir de cette structure, rassemblement par un expert des services provenant de listes prédéfinies de partenaires.
– Proposition d'une méthode de peuplement automatique des concepts de la structure hiérarchique à partir des partenaires, n'ayant pas de liste prédéfinie.
Nous souhaitons par ces points permettre la comparaison d'hôtels décrits avec une terminologie différente afin de rassembler nos ressources. La figure 1 illustre la mise en place d'une représentation commune d'hôtels dans un

1. www.addictrip.com

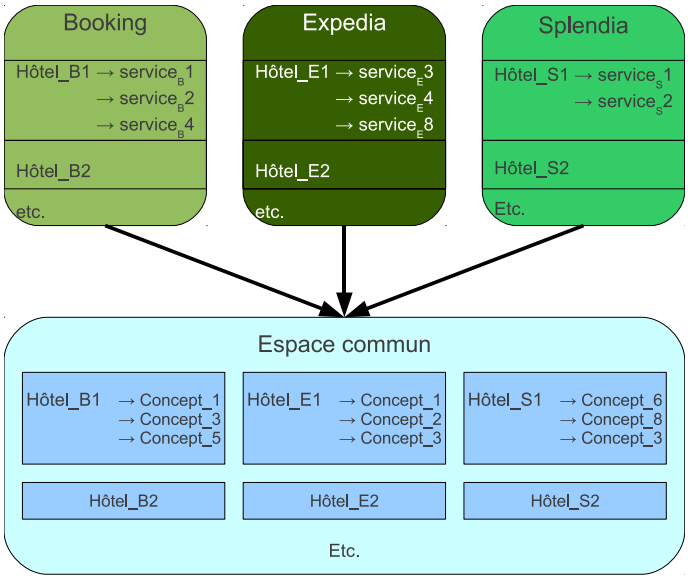

FIGURE 1: Espace de représentation commun d'hôtels provenant de sources différentes

espace conceptuel.

Nous présentons tout d'abord dans la section suivante un survol des méthodes permettant de construire et de peupler ce type de structures ainsi que des mesures de similarités littérales. Nous décrierons ensuite notre méthode d'apprentissage (section 3.) et présenterons les différents résultats expérimentaux obtenus. Nous discuterons pour finir de plusieurs propositions d'amélioration de la méthode avant de conclure en section 4.

2. Les structures conceptuelles et la notion de similarité

Pour ces travaux, nous allons utiliser des structures conceptuelles et la notion de similarité entre deux entités textuelles.

La construction et le peuplement de structures hiérarchiques ont fait l'objet de nombreux travaux dans la littérature, notamment dans le cadre d'ontologies. Ces dernières se fondent dans un premier temps sur le rassemblement de documents permettant la construction d'un corpus du

domaine dont plusieurs approches sont présentées dans Lame (2002). Dès lors, intervient une étape d'extraction de termes candidats issus du corpus précédemment constitué afin de peupler les concepts. Citons par exemple des approches statistiques dont l'objectif consiste à supprimer du corpus les "mots vides" ou ne contenant aucune information utile. Citons par exemple l'utilisation de mesures statistiques tel que le *tf* ou le *tf-idf* Salton *et al.* (1975) expérimentées notamment dans Koo *et al.* (2003) et Maedche & Staab (2004) ou encore l'entropie utilisée dans Brini *et al.* (2005). Il existe également des méthodes syntaxiques reposant dans un premier temps sur l'extraction de syntagmes avec des outils tel que SYNTEX Bourigault (2007).

Une de nos tâches revient à mesurer la proximité entre deux entités textuelles. Cette tâche se fonde sur une proximité dite littérale (sur laquelle nous reviendrons en section 3.2.) des différentes instances de nos concepts et du nouveau terme à catégoriser.
Il existe un nombre important de *mesures de similarité* de ce type visant à estimer la proximité de deux termes en fonction des lettres qu'ils ont en communs. La distance de Levenshtein (1966) fut l'une des premières à prendre en compte cette notion. Elle mesure la proximité de termes en fonction du nombre d'opérations élémentaires qu'il est nécessaire de réaliser afin de passer d'un terme à l'autre. Ces opérations peuvent être des substitutions, suppressions ou insertions. Cette distance, dite *distance d'édition*, est principalement utilisée dans le cadre de corrections orthographiques afin d'estimer le terme le plus probable qu'un utilisateur a souhaité utiliser. Plusieurs mesures se fondes sur celle de Levenshtein comme la métrique de Smith & Waterman (1981) qui fut appliquée dans le domaine biomédicale afin de découvrir des régions similaires dans des brins d'ADN. Citons également Gotoh (1982) qui propose une extension des travaux de Smith & Waterman (1981) ou encore Monge & Elkan (1996) qui segmentent les termes en sous chaînes de caractères. Citons pour finir la distance de Jaro (1989) et celle de Winkler (1999) qui est une extension de la précédente. Ces deux distances prennent en compte, lors de la comparaison de deux chaînes de caractères, d'une part le nombre de caractères en commun mais également l'ordre des caractères.

Notons que la mesure proposée dans cet article possède également ces propriétés. Nous nous appuyons en effet sur la notion de n-grammes décrite en section 3.2.2.1. qui permet de prendre en compte les séquences de

caractères. Un certain nombre de mesures ont également été proposées dans la littérature employant des n-grammes de caractères (ou *q-grams*). Ullmann (1977) propose une des premières approches en résolvant automatiquement des opérations comme la suppression, la substitution ou l'inversion de caractères en corrigeant les erreurs contenus dans les termes. Gravano *et al.* (2001) montrent que les n-grammes de caractères peuvent permettre de traiter efficacement des données approximatives sans conséquence sous-jacente pour les bases de données traitées. L'indexation de brins d'ADN est également une des problématiques traitées avec des techniques se fondant sur des n-grammes comme le montre Cao *et al.* (2005). Citons pour finir Salmela & Tarhio (2006) qui proposent plusieurs algorithmes employant des n-grammes dans le but de filtrer les résultats obtenus avec leurs algorithmes sur de la correspondance de patrons.

3. Description de la méthode proposée

Cette section présente notre méthode de construction et de peuplement d'une structure de concepts hiérarchisés.

3.1. Initialisation de la structure hiérarchique

3.1.1. Construction de la structure et définition des concepts

Afin de construire notre structure, nous utilisons des ressources fondées sur une terminologie à nombre fini de services (dans notre cas précis des services d'hôtels). La première étape consiste dès lors à définir le nombre de niveaux hiérarchiques. Nous avons considéré, avec l'expert, que deux niveaux étaient suffisant pour décrire avec précision le domaine de l'hôtellerie par le biais de services d'hôtels. Le plus haut niveau comporte 11 concepts. Citons par exemple des concepts liés aux "enfants", aux "professionnels" ou encore à la "restauration". Dès lors, ces concepts sont décrits par d'autres concepts de second niveau définis également avec l'expert. Un extrait de la hiérarchie obtenue est proposé dans la figure 2. Notons que le service de premier niveau noté "Services Hôtels" corespond à un ensemble de services proposés à l'interieur des hôtels, et non pas la racine de la structure hiérarchique. La section suivante décrit le peuplement des concepts consistant à instancier les concepts de niveau 2 par les services propres aux hôtels.

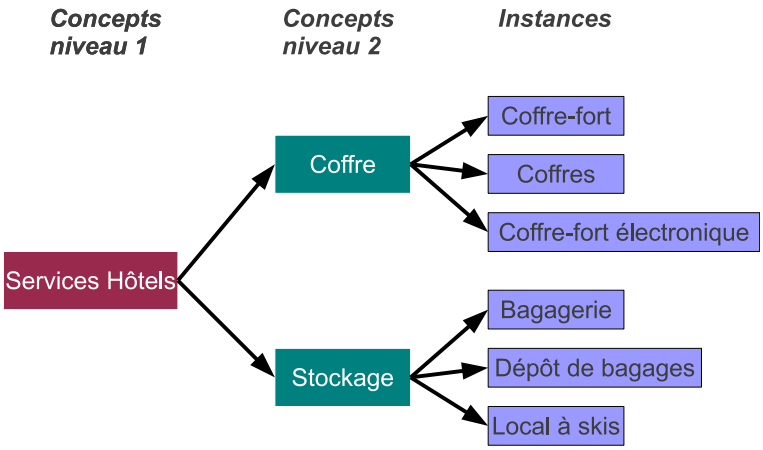

FIGURE 2: Extrait de la structure hiérarchique de concepts obtenu avec l'expert

3.1.2. Peuplement des concepts

Le peuplement initial des concepts est réalisé par l'expert. Les instances de ces concepts sont extraits par l'expert, des listes de services fournis par nos partenaires.
Appuyons nous sur un exemple afin d'expliciter la démarche de l'expert. Considérons le concept de niveau 1 "Santé" et un de ses concepts de niveau 2 "remise en forme". La tâche de l'expert est d'identifier parmi les listes de services fournis, l'ensemble des services se rapportant au concept de la remise en forme. Par exemple, l'expert a sélectionné à partir de la première liste les services "salle de gym", "centre de santé" et "centre fitness", "club de sante" à partir d'une autre liste. Le peuplement est réalisé à partir de services existants dans ces listes. Il est possible de demander à un expert d'instancier les concepts à partir de ses connaissances sans faire référence à des listes prédéfinies.

3.2. Méthode de peuplement automatique de concepts

Bien que le peuplement expert des concepts soit pertinent, il n'est pas envisageable avec des listes ouvertes, c'est-à-dire avec des services non prédéfinis. Nous présentons alors une approche de peuplement automatique de concepts applicable à grande échelle.

3.2.1. Principe et motivations

La méthode de peuplement automatique s'appuie sur deux hypothèses :
– Les concepts initiaux doivent contenir des instances. Cette méthode ne peut en effet pas s'appliquer pour peupler des concepts initialement dépourvus d'instances. Par exemple, nous seront pas en mesure de rapprocher des termes comme "wifi" et "internet sans fil" sans avoir au préalable dans les concepts initiaux des termes qui leurs sont littéralement proches.
– Les nouveaux services visant à peupler les concepts sont supposés littéralement proches des instances initiales de ces concepts.

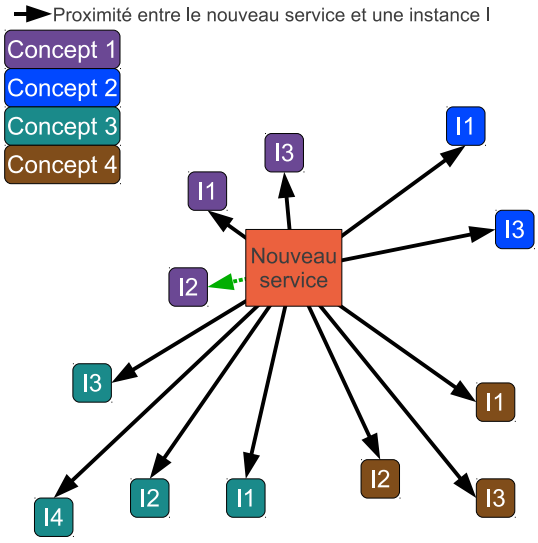

FIGURE 3: Attribution d'un concept à un nouveau service

Le principe de cette approche est de mesurer la proximité d'un nouveau service S par rapport aux instances des concepts initiaux. Dès lors, le concept de l'instance la plus proche de S lui est attribué comme illustré sur la figure 3. Cette figure montre en effet que le nouveau service est très proche des instances du concept 1 et plus particulièrement de l'instance I2. Il sera donc affecté à ce service le concept 1.
La proximité entre un nouveau service et l'ensemble des instances des concepts initiaux est obtenue par une mesure de similarité se fondant sur la notion de n-grammes de caractères et de mots. Notre choix a été motivé par les points suivants.

– La tolérance au données bruitées ou mal orthographiées. La technique des n-grammes de caractères est en effet reconnue dans la littérature pour cette particularité. Elle est notamment très efficace pour traiter des données issues de numérisations OCR (Optical Character Recognition) comme dans Junker & Hoch (1997).
– Le traitement de services possédant divers qualificatifs. Cette méthode va permettre par exemple de rapprocher des services comme “connexion internet adsl”, “connexion internet wi-fi” et “prise internet” en indiquant qu’un hôtel possède un accès à internet.
– L’absence de ressources sémantiques dédiées. Il est difficile d’avoir recours à des ressources comme Wordnet afin d’interpréter le sens d’un service.
Nous détaillons dans la section suivante la mesure de proximité proposée.

3.2.2. La mesure de proximité

3.2.2.1. Proposition de deux indices

La mesure de similarité entre le nouveau service et une instance repose sur deux indices basés sur un comptage d’entités pouvant être des mots ou un ensemble de caractères. En effet, le premier indice permet de valoriser le nombre de mots en commun.
Le second indice s’appuie sur la notion de n-grammes de caractères. Nous pouvons définir un n-gramme de X comme une séquence de n X consécutifs. X peut alors être un caractère ou bien un mot. Dès lors, le second indice peut s’apparenter à l’utilisation de 1-gramme de mots. Le second quant à lui utilise les n-grammes de caractères comme le montre l’exemple suivant.
Ainsi, le couple (*climatisation dans les chambres , chambres climatisées*) obtient un score de 1 avec le premier indice. En fixant le n des n-grammes à 3, nous obtenons comme 3-grammes avec la première instance du couple : “cli, lim, ima, mat, ati, tis, isa, sat, ati, tio, ion, on_, n_d, _da, dan, ...”. Notons que le caractère “_” représente ici un espace. Nous avons cependant fait le choix de ne pas prendre en compte les n-grammes contenant des espaces afin de ne pas introduire de notion de séquence. En effet, il se peut dans un service que deux termes apparaissent dans un ordre inversé comme c’est le cas dans l’exemple du couple (*climatisation dans les chambres , chambres climatisées*). Notons également que les termes contenant moins de n lettres ne seront pas pris en compte car aucun n-gramme ne peut en être extrait.
Dès lors, le résultat de l’indice n’est autre que le nombre de n-grammes communs aux deux membres du couple. Pour notre exemple, nous obtenons un

score de 12.

3.2.2.2. Normalisation des indices

Bien que ces indices soient de bons indicateurs de proximité entre un nouveau service et les instances des concepts, ils doivent être normalisés. Notre objectif est alors de normaliser ces indices de telle sorte que deux services proches obtiennent un score voisin de 1 et à l'inverse deux services éloignés obtiennent un score proche de 0.

Nous proposons alors une normalisation adaptée à chaque indice. La norma-

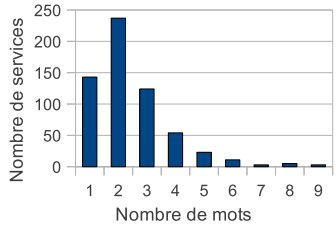

FIGURE 4: Nombre services en fonction du nombre de mots

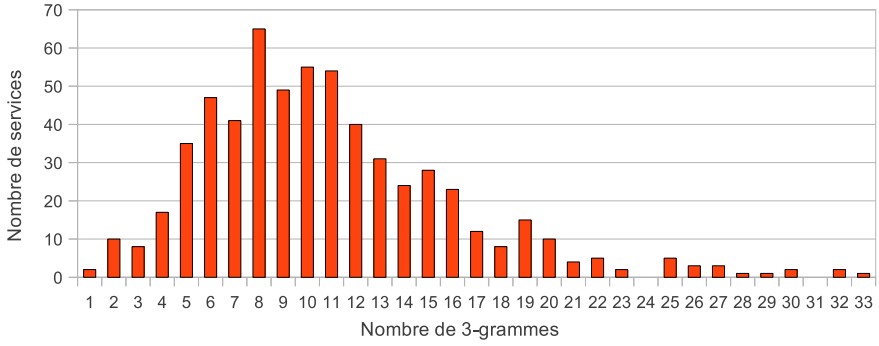

FIGURE 5: Nombre services en fonction du nombre de 3-grammes

lisation des indices doit dépendre des données que nous souhaitons exploiter. Nous nous appuyons alors sur les 603 services constituant les instances initiales de notre structure hiérarchique (section 3.1.2.). Ces instances ont été affectées par expert aux 46 concepts de la structure hiérarchique.

Les services d'hôtels comportent en général peu de mots alors que les 3-grammes de caractères sont plus nombreux. Les figures 4 et 5 illustrent ces propos en indiquant le nombre de services en fonction du nombre de mots contenus dans ces services et respectivement du nombre de 3-grammes de caractères. En conséquence nous proposons de diviser le premier indice $c1$ par le nombre maximal de mots entre les deux services évalués et le second $c2$ par le nombre moyen de n-grammes.

Soient s un nouveau service, i une instance. Alors les indices $c1$ et $c2$ normalisés s'écrivent respectivement :

$$I1(s,i) = \frac{c1(s,i)}{max(nb(s), nb(i))}, I2(s,i) = \frac{c2(s,i)}{avg(nbg(s), nbg(i))}$$

avec $nb(s)$ le nombre de mots que contient le service s, $nbg(s)$ le nombre de n-grammes extraits à partir du service s, en supprimant les n-grammes contenant des caractères "espace".

3.2.2.3. Combinaison des indices

Nous proposons dans cette section une méthode permettant de combiner les deux indices normalisés afin d'en améliorer les performances. Le second indice (noté $I2$) donne une bonne indication de proximité en ce sens qu'un score de $0,8$ indique que deux services sont proches et inversement pour un score de 0.2. Néanmoins, cette affirmation est inexacte pour le premier indice (noté $I1$). En effet, affirmer que deux services sont éloignés s'ils ne partagent pas de mots en communs n'est pas acquis, en considérant notamment les mots mal orthographiés. De plus, si 50% des mots sont les mêmes pour deux services (soit un score proche de $0,5$), ils doivent être considérés comme assez proches.
Ainsi, l'objectif de la combinaison des deux approches nous impose de faire une transformation non linéaire de $I1$ en $I'1$ vérifiant les contraintes suivantes : $I'1 = 0$ si $I1 = 0$ et $I'1 = \alpha$ si $I1 = 1$. En prenant $I'1 = \beta - \frac{\beta}{exp(I1)}$, ces contraintes ne se vérifient que pour $\beta = \frac{\alpha \times e}{e-1}$ avec $e = exp(1)$.
La combinaison des deux approches s'écrit alors :

$$Comb(I1, I2) = I'1 + I2 = \beta - \frac{\beta}{exp(I1)} + I2$$

Une normalisation de cette combinaison est nécessaire afin que l'intervalle de

définition soit de $[0, 1]$ ce qui s'obtient en divisant le score par le numérateur précédent en fixant $I1$ et $I2$ à 1. Ainsi, la combinaison devient :

$$Comb(I1, I2) = \frac{I'1 + I2}{\beta - \frac{\beta}{e} + 1}$$

Il est alors possible de pondérer le scalaire α. Par exemple, $\alpha = 1$ signifie que l'on accorde autant d'importance aux résultats des deux mesures car $I'1 \in [0, \alpha]$ soit $[0, 1]$ et $I2 \in [0, 1]$.
Ainsi, les deux indices proposés dans cette section ainsi que leur combinaison vont permettre de peupler automatiquement les concepts de notre structure hiérarchique. Rappelons que cette dernière permet une description commune des hôtels provenant de divers sites Web de réservation d'hôtels en ligne.

3.3. Expérimentations

3.3.1. Mesure de la robustesse sémantique de l'approche

3.3.1.1. Protocole expérimental

Afin de mesurer la robustesse sémantique de la méthode de peuplement automatique proposée dans cet article, nous nous focalisons dans un premier temps sur l'évaluation de la qualité de la mesure de proximité définie dans la section 3.2.2. Ainsi, nos expérimentations porteront uniquement sur la structure hiérarchique construite avec l'aide d'un expert. Nous proposons ici de vérifier si en extrayant des instances de ces concepts, nous sommes en mesure avec notre approche de réattribuer correctement le concept d'où a été extrait ce service.
Dans un premier temps, nous appliquons un certain nombre de pré-traitements à nos données :
– Remplacement des caractères accentués.
– Remplacement des caractères en majuscules par des minuscules.
– Formatage des caractères numériques (par un terme générique NUM).
– Suppression des concepts contenant moins de 8 instances. Cette suppression se justifie afin d'éviter, après segmentation des données, que des concepts vides n'apparaissent suite à l'application d'une validation croisée telle que décrite ci-après.
– Utilisation d'une "stop-list" contenant des termes génériques. Ces derniers sont alors supprimés des services d'hôtels.
Dès lors, nous proposons d'effectuer une validation croisée avec nos données

pré-traitées en alternant les données de test et d'apprentissage. Nous avons segmenté notre population qui comprend 603 services en 5 sections de 120 services environ. Dès lors, sont considérés alternativement 80% des données comme apprentissage et 20% comme test.

3.3.1.2. Résultats

Le tableau 1 présente les taux d'erreurs obtenus en faisant varier le paramètre *n* des n-grammes de caractères avec le paramètre $alpha$ fixé à 0.5. Le taux d'inconnus correspond à la proportion de services qui n'ont pu être affectés à un concept. Ce taux d'inconnus croit naturellement avec l'incrémentation du paramètre *n*. Comme les mots de moins de *n* caractères ne sont pas pris en compte, la probabilité qu'un nouveau service soit affecté à un concept diminue en fonction du nombre de mots supprimés dans ce dernier (plus *n* est important). Nos expérimentations montrent que le plus faible taux d'erreur est

alpha	*n*	*Taux d'erreur*	*Taux d'inconnus*
0,5	2	22,86%	0,75%
	3	22,77%	1,12%
	4	24,33%	4,33%

TABLE 1: Taux d'erreur obtenu avec la structure initiale en faisant varier *n*

obtenu avec $n = 3$ avec un écart peu significatif pour $n = 2$. Notre seconde expérimentation consiste à faire varier le paramètre α. Avec $\alpha = 1$ les poids du premier et du second indice sont identiques ; par contre un $\alpha = 0,5$ donnera plus d'importance au second indice. Les résultats obtenus sont donnés

alpha	*n*	*Taux d'erreur*	*Taux d'inconnus*
0	3	21,69%	1,50%
0,25	3	21,59%	1,12%
0,50	3	22,77%	1,12%
0,75	3	23,16%	1,12%
1	3	23,14%	1,12%

TABLE 2: Taux d'erreur obtenu avec la structure initiale en faisant varier α

dans le tableau 2 et montrent que l'indice 2 à base de n-grammes de carac-

tères permet une meilleure affectation de nos services dans des concepts. Le score optimum est obtenu avec $\alpha = 0,25$ avec un écart faible pour les autres valeurs. Les taux d'erreur obtenus sont de l'ordre de 22%. Ce faible score s'explique par les points suivants.
– Nous appliquons avec cette validation croisée la politique du pire. En effet, lors de l'ajout de services provenant d'un nouveau site Web de réservation en ligne d'hôtels à notre structure conceptuelle, il peut apparaitre des services identiques à ceux déjà présents ce qui est impossible dans la structure conceptuelle. Ainsi, lors de la validation croisée, un service présent chez deux partenaires sera équivalent à un service unique.
– Le manque de services. En effet, 420 services répartis dans 46 concepts constitue un ensemble d'apprentissage assez faible et peut expliquer ces résultats de l'ordre de 22%.
– Nous avons décidé que quelque soit le score obtenu, chaque service devait être affecté à un concept ce qui augmente le taux d'erreur. Nous discuterons de méthodes permettant de faire varier le taux d'inconnu, diminuant ainsi le taux d'erreur dans la section 4.
Compte tenu des justifications précédemment évoquées, notre méthode de peuplement de structure conceptuelle possède ainsi une bonne robustesse sémantique en affectant 4 services sur 5 dans le concept approprié. Nous proposons dans la section suivante de mesurer l'impact d'un jeu d'apprentissage plus important.

3.3.2. Mesure de la qualité d'apprentissage

3.3.2.1. Protocole expérimental

Afin de mesurer la qualité d'apprentissage de notre méthode de peuplement de structures conceptuelles, nous proposons maintenant d'expérimenter notre approche avec un jeu de données plus conséquent et avec notre méthode de peuplement automatique. Les partenaires choisis sont composés de services écrits en langage naturel. En conséquence, nous avons supprimé de ces ressources les descriptions rares de services (les services décrivant au maximum 3 hôtels ne sont pas pris en compte). Ainsi, plus de 2000 services ont été ajoutés à nos concepts, couvrant environ 10 000 hôtels. Nous avons repris les valeurs des paramètres ayant permis une bonne optimisation des résultats dans la section précédente à savoir : $\alpha = 0,25$ et $n = 3$.
L'objectif de nos expérimentations n'est pas ici de juger la qualité de la nouvelle structure obtenue suite à l'ajout de nouveaux services, mais de mesurer

l'impact de la qualité d'apprentissage. En effet, le nombre moyen de services par concepts est ici augmenté ce qui rend l'apprentissage plus efficace.
Nous suivons dès lors le même protocole expérimental que précédemment en faisant une validation croisée avec 5 sous ensembles.

3.3.2.2. Résultats

Les résultats obtenus en fonction de α sont présentés dans le tableau 3. Notre hypothèse concernant le fait qu'une base d'apprentissage plus conséquente permettrait une meilleure attribution des concepts est confirmée. Nous obtenons en effet des taux d'erreurs plus faible, de l'ordre de 18% avec un taux d'inconnus lui aussi logiquement réduit. Par ces résultats, nous confirmons

Alpha	*Taux d'erreur*	*Taux d'inconnus*
0	17,76%	0,38%
0,25	17,41%	0,29%
0,5	18,42%	0,29%
0,75	19,14%	0,29%
1	19,27%	0,29%

TABLE 3: Taux d'erreur obtenu avec un important ensemble d'apprentissage

la qualité de notre approche et l'importance non négligeable du jeu apprentissage. Ainsi, plus le jeu d'apprentissage augmente et plus notre méthode devient efficace.
Nous proposons dans la section suivante une synthèse des travaux présentés dans cet article ainsi qu'une discussion sur les améliorations possibles.

4. Conclusion

Cet article a présenté une méthode de construction et peuplement automatique de structures hiérarchiques de concepts. Cette dernière fut ici appliquée au domaine de la catégorisation de services d'hôtels. Le principe est de construire à l'aide d'un expert la structure hiérarchique et les concepts puis d'effectuer un peuplement initial et manuel des concepts. Cette base d'apprentissage va alors permettre une affectation automatique de nouveaux services avec notre approche fondée notamment sur la notion de n-grammes de caractères. Notons que notre approche, bien que fortement dépendante du domaine, peut

également être généralisée en adaptant les normalisations présentées en section 3.2.2.2.
Nos expérimentations ont montré le potentiel de cette approche qui peut cependant être discutée. Nous favorisons avec notre approche *le rappel* en cherchant à attribuer à chaque service un concept. Afin de favoriser la *précision*, nous pourrions introduire un seuil à notre mesure de proximité décrite section 3.2.2. Ainsi, si aucun service de la base d'apprentissage n'est au dessus de ce seuil lors de la comparaison avec un nouveau service, ce dernier sera considéré comme indéfini. En d'autre termes, nous considérons que ce service ne peut être affecté à un concept. En fixant ce seuil à 0.5, $\alpha = 0.25$ et $n = 3$, nous obtenons, en respectant le protocole expérimental défini en section 3.3.1.1., un taux d'erreur de $14,37\%$ et un taux d'inconnus de $23,23\%$. Il en ressort une meilleure attribution des concepts avec une augmentation du nombre d'inconnus.
L'introduction d'un tel seuil pose cependant le problème du traitement des inconnus. Ces derniers pourraient alors être réévalués avec une autre méthode fondée sur des techniques de Web mining consistant notamment à interroger un moteur de recherche Web à la manière de Turney (2001).
Nous envisageons par ailleurs de comparer la mesure proposée dans cet article à d'autres mesures de la littérature afin d'en montrer la valeur ajouté dans notre contexte. Nous souhaitons pour finir mettre en œuvre l'approche de classification permettant de construire des classes d'hôtels, quelque soit la terminologie utilisée pour les décrire, afin d'avoir un retour plus précis de la société Addictrip sur la qualité de notre approche.

Références

BOURIGAULT D. (2007). *Un analyseur syntaxique opérationnel : SYNTEX*. Mémoire d'hdr en sciences du langage, université de toulouse 2, france.

BRINI A., BOUGHANEM M. & DUBOIS D. (2005). A model for information retrieval based on possibilistic networks. In *SPIRE*, p. 271–282.

CAO X., LI S. C. & TUNG A. K. H. (2005). Indexing dna sequences using q-grams. In *In Proceedings of the 10th International Conference on Database Systems for Advanced Applications*, p. 4–16.

GOTOH O. (1982). An improved algorithm for matching biological sequences. *Journal of molecular biology*, **162**(3), 705–708.

GRAVANO L., IPEIROTIS P. G., JAGADISH H. V., KOUDAS N., MUTHUKRISHNAN S., PIETARINEN L. & SRIVASTAVA D. (2001). Using q-grams

in a dbms for approximate string processing. *IEEE Data Eng. Bull.*, **24**(4), 28–34.

JARO M. A. (1989). Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. *Journal of the American Statistical Association*, **84**(406), 414–420.

JUNKER M. & HOCH R. (1997). Evaluating ocr and non-ocr text representations for learning document classifiers. In *ICDAR '97 : Proceedings of the 4th International Conference on Document Analysis and Recognition*, p. 1060–1066, Washington, DC, USA : IEEE Computer Society.

KOO S. O., LIM S. Y. & LEE S. J. (2003). Building an ontology based on hub words for information retrieval. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, p. 466– : IEEE Computer Society.

LAME G. (2002). *Evaluating OCR and Non-OCR Text Representations for Learning Document Classifiers*. PhD thesis.

LEVENSHTEIN V. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, **10**, 707.

MAEDCHE A. & STAAB S. (2004). Ontology learning. In *Handbook on Ontologies*, p. 173–190.

MONGE A. & ELKAN C. (1996). The field matching problem : Algorithms and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, p. 267–270.

SALMELA L. & TARHIO J. (2006). Multi-pattern string matching with q-grams. *ACM Journal of Experimental Algorithmics*, **11**, 1–19.

SALTON G., YANG C. S. & YU C. T. (1975). A theory of term importance in automatic text analysis. *Journal of the American Society for Information Science*, **26**(1), 33–44.

SMITH T. & WATERMAN M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195–197.

TURNEY P. (2001). Mining the Web for synonyms : PMI–IR versus LSA on TOEFL. In *Proceedings of ECML'01, Lecture Notes in Computer Science*, p. 491–502.

ULLMANN J. R. (1977). A binary n-gram technique for automatic correction of substitution, deletion, insertion and reversal errors in words. *Comput. J.*, **20**(2), 141–147.

WINKLER W. E. (1999). *The State of Record Linkage and Current Research Problems*. Rapport interne, Statistical Research Division, U.S. Census Bureau.