WEBAFFIX : une boîte à outils d'acquisition lexicale à partir du Web

Nabil Hathout, Ludovic Tanguy

Webaffix : une boîte à outils d'acquisition lexicale à partir du Web

Nabil Hathout et Ludovic Tanguy
Equipe de Recherche en Syntaxe et Sémantique (ERSS) - UMR 5610, CNRS
Université de Toulouse le Mirail
Prénom.Nom@univ-tlse2.fr

Résumé : Nous présentons ici Webaffix, un outil qui permet de constituer et d'enrichir semi-automatiquement des données lexicales en utilisant le Web comme corpus. Il permet de détecter et d'analyser morphologiquement de unités lexicales nouvelles (c'est-à-dire absentes de listes de références telles que les dictionnaires) construites par suffixation ou préfixation. Nous présentons les techniques utilisées par Webaffix, en déclinant les différents modes d'utilisation que nous avons envisagés et mis en pratique, ainsi que des exemples de résultats produits par diverses campagnes de collecte. Les données ainsi recueillies constituent des ressources lexicales pour différentes applications en traitement automatique des langues, mais également pour l'étude à grande échelle de la morphologie dérivationnelle.

Abstract : This paper deals with the design and use of Webaffix, a tool for semi-automatically detecting new word forms from the World Wide Web. We focus mainly on new derived words, *i.e.* coined from other lexemes through suffixation and/or prefixation processes. We develop the techniques and methods used in Webaffix, along with a sample of results obtained via several studies on French. Resources such as the ones created through the use of Webaffix are useful not only for natural language processing and information retrieval tasks, but also for the linguistic study of word creation.

1. Construire des ressources lexicales grâce au Web

1.1. Problématique

Le Web est devenu un objet d'étude et un domaine d'expérimentation privilégié tant pour le traitement automatique des langues (TAL) que pour la recherche d'information (RI). Cette évolution a créé un besoin nouveau pour des ressources linguistiques et en particulier lexicales qui couvrent correctement la langue utilisée dans les pages Web. Pour répondre à ce besoin, la solution la plus simple consiste à adapter les ressources de « langue générale » habituellement utilisées par les applications de TAL, la première d'entre elles étant le lexique. Il s'agit en effet de la ressource fondamentale pour toute application « linguistique » en traitement automatique, de l'analyse syntaxique jusqu'à la traduction automatique. Or, une source d'enrichissement du lexique très importante est la construction morphologique[1], et ceci est d'autant plus vrai sur le Web où la spontanéité des locuteurs, notamment en matière de création lexicale, est bien souvent débridée... Il n'y a *a priori* pas de source mieux adaptée à la collecte de lexèmes construits que le Web lui-même.

De plus, l'analyse des phénomènes constructionnels, d'un point de vue linguistique cette fois, a beaucoup à gagner à étudier des données en grand nombre, et en particulier celles qui sont produites dans des contextes divers, souvent dans des énoncés de genres marginaux (pages personnelles, récits humoristiques, jargons techniques, etc.). Dès lors se pose la nécessité d'aborder l'exploitation de cette masse de données lexicales qu'est le Web.

[1]Nous utilisons ici la terminologie de Corbin 2001 sans pour autant nous placer dans le cadre du modèle Silex. Nous employons par exemple les termes *construction morphologique* ou *morphologie constructionnelle* à la place de *dérivation morphologique* ou *morphologie dérivationnelle*.

1.2. Aperçu de Webaffix

Dans ce but, nous avons développé Webaffix, une boîte à outils pour l'acquisition lexicale à partir du Web. Webaffix est destiné à la création et à l'extension semi-automatique de lexiques au moyen de formes nouvelles construites morphologiquement. Il propose deux grandes fonctionnalités : la recherche de formes « brutes » sur le Web et leur analyse morphologique. La boîte à outils contient trois composants qui sont décrits dans la suite de cet article :

1. un module de recherche par suffixe qui permet de découvrir sur le Web des formes qui correspondent à un motif décrivant par exemple la présence d'un suffixe graphémique donné (voir section 2.2) ;
2. un composant de prédiction morphologique permettant de calculer les formes des lexèmes bases ou des lexèmes construits (voir section 2.1) ;
3. un méta-moteur, basé sur un moteur de recherche générique sur Internet, auquel il ajoute des fonctionnalités dédiées à l'exploration lexicale du Web. Ce méta-moteur est utilisé par les deux autres modules.

Webaffix permet d'exploiter les éventuelles ressources lexicales dont dispose déjà l'utilisateur. Ce peut être des listes de formes (comme celles qui sont fournies dans les différentes distributions Linux), des lexiques flexionnels (comme les lexiques ABU[2], Multext[3] ou TLFnome[4]) ou des bases de données morphologiques comme la base Verbaction[5]. Mais il est également possible d'utiliser Webaffix sans ressource initiale. De ce point de vue, il s'agit d'un système souple et adaptable.

Webaffix propose deux modes de recherche de formes nouvelles. Le premier exploite un lexique existant ou une base de données morphologiques pour générer une liste de formes candidates au moyen du composant de prédiction morphologique, puis utilise le méta-moteur pour rechercher sur le Web des attestations de ces formes. Le second utilise le module de recherche par suffixe pour repérer sur le Web des formes qui correspondent à un schéma donné (par exemple celles qui finissent par *-able*), sans aucune contrainte sur la base.

Un des problèmes majeurs de toute approche linguistique sur le Web est le manque de fiabilité des sources. Il est donc nécessaire de disposer d'un ensemble de filtres permettant d'éliminer la plus grande partie possible du bruit généré. En plus des problèmes classiques en détection de formes nouvelles (noms propres, fautes d'orthographe, etc.) pour lesquels nous proposons des solutions (voir section 2.2), se pose la question du statut morphologique des résultats. Pour cela, nous utilisons des méthodes d'analyse des formes collectées reposant sur la prédiction des formes de leurs lexèmes bases (c'est-à-dire des mots sur lesquels elles sont construites). Deux modes de sélection des formes collectées sont possibles. Le plus simple est basé sur la recherche sur le Web d'attestations des formes des lexèmes bases. Dans ce cas, le critère pour le filtrage d'une forme comme *copolymérisable* sera l'attestation d'une des formes du verbe *copolymériser*. Un filtrage plus strict est également possible : il s'agit alors de rechercher des pages Web qui contiennent à la fois la forme candidate et l'une des formes de son lexème base.

1.3. Utilisations

Webaffix est un système qui continue à évoluer et à s'adapter aux besoins de ses utilisateurs. Il a été utilisé pour plusieurs études morphologiques menées à l'ERSS (Equipe de Recherche en Syntaxe et Sémantique) : nous en présentons ici quelques-unes qui nous serviront de support à la description des modules de la boîte à outils. Son efficacité a ainsi pu être établie aussi bien pour des suffixes très rares comme *-este* (Plénat et coll. 2002) que pour des suffixes très fréquents et

---

[2] http://abu.cnam.fr

[3] http://www.lpl.univ-aix.fr/projects/multext

[4] TLFnome est un lexique de formes fléchies construit à l'INaLF (CNRS, USR 705, aujourd'hui ATILF, CNRS & U. Nancy 2, UMR 7118) par J. Maucourt et M. Papin, à partir de la nomenclature du *Trésor de la Langue Française* (*T.L.F.*).

[5] Ce lexique a été réalisé à l'INaLF par A. Berche, F. Mougin, N. Hathout et J. Lecomte (*cf.* section 3.2.1).

productifs comme *-able* (Hathout et Plénat 2002) ou *-tion*, *-ment*, *-age*… (Tanguy et Hathout 2002). Webaffix est par ailleurs indépendant vis-à-vis des langues particulières. Il a par exemple été utilisé sans modification (ou presque[6]) pour constituer un lexique de formes de l'italien ayant les finales *-istica* et *-istico*. Les modules de recherche lexicale de Webaffix sont disponibles et utilisables librement (voir section 4).

1.4. Le Web comme corpus

Le corpus sur lequel opère Webaffix est le Web, et non un corpus classique. Comme le note G. Grefenstette, 1999, nombre de linguistes peuvent, à juste titre, se montrer réticents dans l'emploi du Web comme source d'attestations, étant donné l'impossibilité technique de caractériser les pages sur le plan du domaine, du genre, du statut de l'auteur, de la validité du contenu, etc. Il n'en reste pas moins que le Web constitue la masse textuelle la plus importante accessible pour une recherche linguistique.

Des projets comme WebCorp (`http://www.webcorp.org.uk`) mettent la technologie de base d'un concordancier à l'échelle du Web, en se fiant aux moteurs de recherche générique. Ces moteurs restent de toute façon le seul véritable moyen d'accès aux pages, à moins de développer un système de parcours et d'indexation spécifique, qui ne pourra de toute façon pas prétendre à l'exhaustivité d'un Google™ et autre AltaVista™. Webaffix n'échappe pas à cette règle, et opère comme tous les autres systèmes sur la partie du Web (dont la proportion est d'ailleurs inconnue) accessible par ces moteurs, et donc sur un sous-ensemble variable avec le temps, et sans critère de sélection identifiable.

La question se pose alors du type d'études sur corpus pour lesquelles le recours au Web est justifié. Les études actuelles vont du repérage d'entités nommées (Jacquemin et Bush 2000) à l'étude de textes parallèles bilingues (Resnik 1999), et en règle générale se limitent à l'étude d'unités lexicales en utilisant des extracteurs de contextes et des mesures de cooccurrence. Il paraît en effet plus délicat de mener des études relevant de la syntaxe ou de la sémantique sur un corpus aussi « incontrôlable », et pour lequel on dispose de si peu d'informations.

Dans notre cas, c'est dans le cadre de la morphologie que nous nous situons, avec comme but affiché de constituer des ressources lexicales génériques. Là encore il existe un biais, dont nous devons prendre conscience : la nature illusoire du Web comme un corpus de « langue générale ». Si la variété des domaines abordés, et donc des sous-langages de spécialité représentés peut paraître suffisante, nous n'avons pas d'idée claire de la représentation de chacun de ces domaines.

1.5. Travaux connexes

Les systèmes comparables à Webaffix sont peu nombreux. Les plus proches sont NeoloSearch d'une part et Walim de l'autre. NeoloSearch (Janicijevic et Walker 1997) est destiné à la recherche de néologismes sur Internet. C'est un système totalement automatique dont la mise en œuvre comporte quatre étapes. (1) La première consiste à constituer un corpus en aspirant un ensemble de pages Web. (2) NeoloSearch sélectionne ensuite les formes susceptibles d'être des néologismes en réalisant une analyse statistique des occurrences du corpus et en ne retenant que les hapax. (3) Les néologismes candidats sont filtrés pour exclure les noms propres et les erreurs typographiques. (4) Les formes des néologismes et leurs contextes sont analysés pour les typer et identifier les éventuels suffixes qui y apparaissent. Les principaux points communs entre NeoloSearch et Webaffix concernent les traitements effectués sur les pages récupérées : filtrage des erreurs typographiques et analyse graphémique des néologismes pour déterminer leurs suffixes. En revanche, NeoloSearch ne tire pas réellement profit du Web puisqu'il ne fait appel à un moteur de recherche pour sélectionner les pages qu'il traite. Webaffix, par son utilisation directe d'un moteur de recherche permet de traiter des données bien plus nombreuses, et cela dynamiquement.

Webaffix est également très proche du système Walim (Namer 2003) et de son prédécesseur GéDériF (Dal et Namer 2000). Ces systèmes permettent de rechercher sur Internet des formes nouvelles, construites morphologiquement à partir de bases connues. Les principales différences qui existent entre ces systèmes et une des utilisations possibles de Webaffix (celle qui est

[6] Seule change le paramétrage du filtrage de la langue (voir section 2.2, § 6).

présentée en section 2.1) résident dans le mode de génération des formes. Ces systèmes utilisent un générateur morphologique à base de règles alors que nous calculons les formes des lexèmes suffixés « par analogie », par une méthode d'apprentissage non supervisé qui s'appuie sur un lexique flexionnel. Ainsi, il n'est pas nécessaire de prévoir de traitements particuliers pour les nombreuses allomorphies et allographies qui apparaissent dans les formes construites morphologiquement. Par ailleurs, Walim comme GéDériF n'effectuent pas sur les pages récupérées de filtrages similaires à ceux du méta-moteur de Webaffix (correction orthographique, vérification de la langue…). Enfin, les approches inductives présentées en section 2.2 permettent à Webaffix de repérer une plus grande variété de formes.

1.6. Plan de l'article

Dans la section 2 de cet article, nous décrivons la procédure de détection de nouvelles formes lexicales suffixées, ainsi que les composantes de Webaffix qui sont impliquées dans cette tâche. Dans la section 3, nous détaillons plus précisément les procédés d'analyse morphologique et de vérification permettant d'affiner les résultats de la phase précédente.

2. Recherche de formes nouvelles

La fonction première de Webaffix est d'explorer le Web, via un moteur de recherche générique, pour repérer de formes lexicales nouvelles. Il existe deux façons complémentaires de procéder à ce balayage. La première, de type hypothético-déductif, consiste à créer de formes nouvelles plausibles à partir de bases connues, et à vérifier leur présence sur le Web. La seconde, de type inductif, effectue un balayage plus large, en recueillant toutes les formes se terminant par un suffixe donné. Nous présentons successivement ces deux modes d'utilisation de Webaffix, ainsi que les modules qu'ils mettent en jeu.

2.1. Recherche de formes générées à partir de bases connues

La première méthode consiste à envisager des nouvelles formes construites à partir de bases lexicales connues, puis à vérifier leur présence sur le Web via un moteur de recherche. Ce mode d'utilisation de Webaffix implique donc le module de prédiction morphologique (en mode génération) et le méta-moteur.

La prédiction morphologique est réalisée par un système « réversible » qui peut fonctionner en analyse ou en génération. Dans les deux cas, il utilise des relations d'affixation graphique apprises automatiquement à partir d'un lexique de référence. Ces relations sont représentées sous la forme de « schémas d'affixation » comme `er/able`. Ce schéma peut être utilisé aussi bien pour générer une forme en *-able* comme *adhérable* à partir de *adhérer* que pour analyser une forme comme *sédimentable* en lui associant la base *sédimenter*.

Nous nous servons pour le français du lexique flexionnel TLFnome+index[7] complété par la nomenclature du *Robert Électronique*. L'ensemble contient 750 000 entrées et 125 000 lemmes. Ce lexique de référence est utilisé à la fois pour l'apprentissage de schémas de suffixation et comme collection de bases connues pour la génération des formes construites plausibles.

Schémas de suffixation. Pour une suffixation donnée, par exemple la suffixation en *–able* sur base verbale, l'apprentissage des schémas à partir d'un lexique L est réalisé en deux étapes :

1. On extrait de L deux listes de formes graphémiques B et D telles que B est la liste des lemmes des bases sélectionnables pour la suffixation (par exemple, les lemmes des verbes) et D est la liste des lemmes des lexèmes construits (par exemple, les lemmes des adjectifs en *-able*). Les éléments de B et de D sont des chaînes de caractères formées à partir d'un alphabet A. Nous noterons A^* l'ensemble de toutes les chaînes de caractères que l'on peut former à partir de A.

2. On forme l'ensemble des couples $B \times D$. Pour chaque couple $(b,d) \in B \times D$, on calcule l'ensemble des couples de suffixes $S(b,d)=\{(t_1,t_2) \in A^* \times A^* / \exists r \in A^*, b=r \cdot t_1 \text{ et } d=r \cdot t_2\}$. Par exemple, S(`laver`,`lavable`)={(`laver`,`lavable`), (`aver`,`avable`), (`ver`,`vable`), (`er`,`able`)}, r

[7] Ce lexique est composé de TLFnome et de TLFindex, lexique réalisé à l'INaLF à partir de l'index du *T.L.F.*

valant respectivement la chaîne vide, `l`, `la` et `lav`. On calcule ensuite la signature suffixale de (*b*,*d*), c'est-à-dire l'élément (s_1,s_2)∈ *S*(*b*,*d*) pour lequel *r* est de longueur maximale. Par exemple, la signature de (`laver`,`lavable`) est (`er`,`able`). La signature d'un couple caractérise la série proportionnelle morphographique à laquelle il appartient. Cette série est composée des verbes en *-er* et des adjectifs en *-able* tels que la graphie de l'adjectif peut être calculée à partir de celle du verbe en enlevant à cette dernière le suffixe graphémique *-er* et en lui ajoutant ensuite le suffixe *-able*. Toutes les séries ne sont naturellement pas intéressantes du point de vue morphologique. Un petit nombre de critères permet de caractériser celles qui sont les plus utiles pour la prédiction morphologique et dont les signatures peuvent être retenues comme schéma de suffixation. Ces critères sont : la taille minimale du radical *r* des couples qui forment la série ; la taille maximale des suffixes s_1 et s_2 ; le nombre minimal de couples qui composent la série. Les valeurs habituellement utilisées pour ces paramètres sont respectivement 3, 8 et 10. La figure 1 présente quelques-uns des schémas appris pour la suffixation en *-able* à partir du lexique de référence.

Schéma	Exemple	
`er/able/690`	*multipli**er***	*multipli**able***
`r/ssable/31`	*enfoui**r***	*enfoui**ssable***
`r/able/29`	*échange**r***	*échange**able***

Figure 1. — *Exemples de schémas de suffixation en* -able.

Les signatures des schémas de la figure 1 sont complétées par une valeur qui indique le nombre de couples du lexique d'apprentissage qu'ils permettent de connecter (cette valeur sera appelée dans ce qui suit « fréquence du schéma »). Par exemple, le premier schéma permet de connecter 690 verbes en *-er* à des adjectifs en *-able*.

Génération des formes. Les schémas de suffixation sont utilisés pour générer les formes des lexèmes construits à partir des bases présentent dans le lexique de référence. On peut par exemple créer tous les lexèmes construits en *-able* possibles à partir de l'ensemble des lemmes des verbes du lexique. Souvent, plusieurs des schémas peuvent être utilisés pour un même verbe. C'est le cas par exemple pour le verbe *expurger* auquel les trois schémas de la figure 1 peuvent s'appliquer. Ils produiraient respectivement *expurgable*, *expurgessable* et *expurgeable*. Différentes stratégies peuvent alors être mises en œuvre pour produire la bonne forme du lexème construit. La première consiste à ne pas choisir pour ne pas éliminer la bonne forme ; en contre partie, on augmente fortement le nombre des candidats générés. La deuxième choisit systématiquement le schéma applicable ayant la fréquence la plus élevée. On produirait alors *expurgable* à partir de *expurger*. Une troisième solution consiste à choisir parmi l'ensemble des schémas applicables celui dont la partie gauche est la plus longue. Cette stratégie est efficace si l'on ne limite pas l'apprentissage aux seules signatures, mais que l'on conserve l'ensemble des couples de suffixes qui vérifient les critères de sélection des schémas. On obtient ainsi des schémas « redondants » comme ceux qui sont présentés en figure 2. On pourra ainsi retenir le schéma `ger/geable/26` parmi ceux qui sont compatibles avec le verbe *expurger* car sa partie gauche comprend 3 caractères.

```
er/able/690        liser/lisable/34   ger/geable/26
ter/table/119      r/ssable/31        ir/able/19
ser/sable/126      r/able/29          quer/cable/18
rer/rable/82       ir/issable/29
iser/isable/74     er/eable/28
```

Figure 2. — *Exemple de schémas redondants de suffixation en* -able.

Requêtes au méta-moteur. Il ne reste alors plus qu'à rechercher sur le Web des attestations des formes générées. On produit donc à partir de chacune des formes une requête booléenne que l'on soumet au méta-moteur. Ce dernier fait exécuter la recherche par un moteur de recherche générique du Web (AltaVista) puis en vérifie les résultats. Le détail des vérifications effectuées est présenté dans la section 2.2. Les requêtes peuvent être réduites et n'inclure que la forme

générée. Elles peuvent aussi être complexe et contenir par exemple d'autres formes du lexème construit. Dans le cas de la suffixation en *-able*, les requêtes soumises au méta-moteur comprenaient les adjectifs au singulier, au pluriel, avec ou sans préfixation par *in-* ou par l'un de ses allomorphes. En effet, un certain nombre de ces adjectifs n'apparaissent que sous la forme préfixée comme *inadmettable*, *incrachable*, *inruinable*, *inshootable*... Ces requêtes sont illustrées en (1).

(1) a. `(continuable OR continuables OR incontinuable OR incontinuables)`
b. `(régissable OR régissables OR inrégissable OR inrégissables OR irrégissable OR irrégissables)`

Les quatre formes de la requête (1a) sont attestées, mais seul *régissable* l'est pour (1b).

Résultats. Pour la suffixation en *-able*, 8 313 nouveaux lemmes plausibles ont été générées à partir des 10 434 verbes du lexique de référence. (Par « nouveaux », nous entendons qu'ils n'appartiennent pas au lexique de référence.) Nous avons ainsi pu collecter 1 117 adjectifs, ce qui représente une augmentation de 68 % du nombre d'adjectifs en *-able* du lexique de référence.

2.2. Recherche globale par suffixe sur le Web

La méthode présentée au paragraphe précédent ne permet pas de collecter des formes construites sur une base non répertoriée dans le lexique de référence. C'est notamment le cas de lexèmes appartenant à des registres techniques (comme *wapiser*[8]*/wapisable*), ou bien correspondant à des notions récentes (comme *pacser*[9]*/pacsage*). Une fonctionnalité importante de Webaffix est donc la découverte de formes nouvelles au moyen du module de recherche par suffixe. Le principe en est simple : il consiste à interroger automatiquement un moteur de recherche généraliste du Web afin de repérer toutes les formes (graphémiques dans une première approximation) nouvelles se terminant par le suffixe. Les phases de cette recherche sont les suivantes.

1. Déclinaison des patrons en sous-requêtes. Cette étape est nécessitée par une contrainte inhérente au moteur de recherche (AltaVista[10]) que nous utilisons[11] : l'opérateur d'interrogation par troncature impose de préciser au moins trois caractères à l'initiale du mot. Par exemple, pour le suffixe *-esque*, on ne peut soumettre directement une requête "`*esque`", mais on doit la décliner en "`aba*esque`", "`abb*esque`", ..., "`zyt*esque`". Les trigrammes initiaux peuvent être générés en énumérant les 60 000 combinaisons de trois caractères du français (lettres accentuées comprises). Une solution plus économique consiste à se limiter aux seules séquences qui se trouvent à l'initiale des entrées de notre lexique de référence. Le nombre de trigrammes est ainsi réduit à 3 500.

2. Filtrage des formes déjà connues (c'est-à-dire, présentes dans le lexique de référence). Ce filtrage est effectué directement par le moteur de recherche, en utilisant la négation habituelle des requêtes booléennes ("*X* AND NOT *Y*" ou "*X -Y*"). Par exemple, pour "`aba*age`", la requête complète est (2).

(2) `aba*age -abattage`

[8] *Wapiser* signifie modifier un site Web pour le rendre accessible via le Wap, ou protocole d'utilisation du Web sur un téléphone portable.
[9] *Pacser* (*se*) signifie souscrire au contrat d'union libre (PACS) récemment institué en France.
[10] http://www.altavista.com
[11] Il est à noter que le choix du moteur s'est d'ailleurs fait sur cette possibilité, la grande majorité des moteurs de recherche du Web ne proposant aucune possibilité d'interrogation par troncature. Parmi les deux seuls qui le proposent (AltaVista et NorthernLight), le premier impose de préciser les trois premiers caractères des formes tronquées, et le second quatre.

Il faut noter qu'il s'agit ici aussi d'une approximation, puisque cette négation concerne les occurrences dans les pages, et non les formes recherchées. Ainsi, la requête (2) ne permettra pas de retrouver une forme nouvelle correspondant à ce schéma (comme *abacarage*) si elle apparaît exclusivement dans des pages qui contiennent aussi *abattage*. Toutefois, nous avons estimé que cette probabilité est négligeable par rapport aux autres sources de silence inhérentes au corpus mouvant et mal défini qu'est le Web.

Les étapes suivantes ont pour but de « nettoyer » les résultats de cette recherche, en examinant les pages Web renvoyées par AltaVista. Ces pages sont d'abord converties en texte ASCII en supprimant le marquage HTML.

3. Vérification des résultats d'Altavista. Dans le cas d'une recherche par troncature, Altavista ne donne pas directement la forme recherchée, mais uniquement la page, ce qui nécessite une recherche dans celle-ci. Cette analyse peut tout à fait se révéler infructueuse : la page peut avoir disparu, ou son contenu a pu être modifié depuis son indexation par AltaVista… Il se peut également que le processus d'indexation d'AltaVista ait commis des « erreurs ». C'est notamment le cas quand une balise HTML est insérée dans un mot, comme pour la capitalisation ou la mise en gras de la première lettre d'un titre (3).

(3) `<b>D</b>éfragmentation`

La page qui contient cette chaîne a été proposée en réponse à la requête "`éfr*tion`". Dans ce cas, c'est la chaîne "`éfragmentation`" qui est indexée mais elle n'apparaît pas dans la page comme un mot graphémique. Elle sera donc ignorée. Notons également que pour éviter les noms propres, toutes les pages dans lesquelles les formes recherchées comportent une majuscule sont automatiquement rejetées.

4. Vérification orthographique. Cette vérification est effectuée hors contexte, en comparant par quelques heuristiques les formes candidates avec celles du lexique de référence, en recherchant les sources d'erreurs suivantes, présentées par ordre de priorité décroissante :

(a) les fautes d'accents, quelles qu'elles soient, et sans limite de nombre par mot, comme "pr**ê**f**è**rable" pour "préférable" ;

(b) les dédoublements (ou plus) de lettres comme "grot**t**esque" pour "grotesque" ou au contraire : "décolage" pour "décol**l**age" ;

(c) l'inversion de deux lettres consécutives comme "obli**ag**tion" pour "obligation" ;

(d) l'ajout ou la suppression d'une lettre comme "adap**a**table" pour "adaptable" ou bien "abillage" pour "**h**abillage" ;

(e) la modification d'une lettre comme "me**r**tion" pour "mention".

Il est toutefois possible de paramétrer le niveau de correction orthographique en fonction des utilisations de Webaffix. Ainsi, lors d'une première campagne d'acquisition pour un nouveau suffixe, il est recommandé d'être plus strict que dans les cas où des ressources étendues sont disponibles. En effet, il est courant, dans les cas où la correction est trop stricte, de « perdre » des formes correctes, comme "b**ê**lance" abusivement corrigé en "balance"...

5. Vérification de la segmentation des mots. Il s'agit en fait d'une vérification orthographique en contexte destinée au traitement des mots collés ou mal découpés comme dans l'exemple (4) où "avantageusesque" n'est pas un adjectif en *-esque*.

(4) les prestations obtenues sont moins **avantageusesque** celles dont bénéficie un salarié à revenu égal,

Dans ce cas, on rejette le mot s'il existe un découpage qui donne deux mots présents dans le lexique de référence ("*avantageuses* + *que*") et qui a une fréquence sur AltaVista supérieure à celle du mot suspect : "avantageuses que" est présent dans 785 pages alors que "avantageusesque" ne l'est que dans une seule. Cette correction peut cependant être à l'origine de découpages abusifs comme dans le cas de l'adjectif *lestable* découpé en "*les* + *table*" ; or il n'y a que 105 occurrences pour l'adjectif alors qu'il y a 257 occurrences de la séquence erronée "les table". En d'autres termes, les fautes de frappe et d'accord surviennent plus fréquemment que certains mots construits...

Le problème inverse se pose dans le cas de textes gardant des traces d'une mise en page préalable à leur formatage HTML, comme les césures dans l'exemple (5). Dans ce cas, la présence d'un tiret à gauche du candidat "mentation" permet de vérifier la pertinence du recollage sur les mêmes principes que précédemment.

(5) créer une réserve d'eau pour l'ali- mentation en eau potable de la région...

6. Vérification de la langue. AltaVista, comme tous les moteurs de recherche généralistes sur le Web, effectue un diagnostic de langue sur les pages qui ne l'indiquent pas explicitement dans leurs entêtes. Dans un premier temps, les requêtes générées par Webaffix indiquent que l'on limite la recherche aux pages rédigées en français. Cependant, il subsiste plusieurs problèmes que Webaffix doit régler, notamment pour les pages multilingues. AltaVista attribue en effet une seule langue à chaque page Web, *a priori* en fonction du début du document ou de la langue majoritaire. En résultat, la forme candidate peut très bien apparaître dans un segment en anglais ou en espagnol au sein d'une page autrement en français. Webaffix vérifie donc systématiquement dans une fenêtre de 100 caractères autour de chaque occurrence du mot-cible, s'il n'y a pas plus d'un mot-outil des autres langues romanes et germaniques (anglais, allemand, espagnol, italien). Les mots-outils ont été sélectionnés par leurs fréquences, en enlevant les cas de recouvrement avec le français. Par exemple, "*or*" n'appartient pas à notre anti-dictionnaire de l'anglais.

Quelques problèmes résiduels demeurent par exemple pour les segments trop courts comme en (6). C'est aussi le cas des citations de termes anglais dans un texte technique ou des titres originaux d'œuvres…

(6) […] il nous faut aller la trouver dans les pages de Sept jours pour expier (days of **atonement**) de WJ Williams.

La méthode des mots-outils n'est pas non plus bien adaptée à la détection des langues trop proches, notamment l'ancien et le moyen français comme en (7).

(7) tant soit peu, diminue, Ny que ma foy descroisse **aulcunement** . Car ferme amour sans eulx est plus, que nue.

7. Vérification du contexte (URL, code informatique, etc.). Ce filtrage permet d'éliminer certains emplois spécifiques du Web comme les segments de code informatique (8) ou les URL (9), les adresses de courrier électronique… La méthode de filtrage se fait simplement sur la base de certaines combinaisons de marques typographiques (notamment les slashes, les accolades, les soulignés...).

(8) Method Summary (package private) void **actionaffichage** _détaillé() Méthode qui permet un affichage de...

(9) `http://www.abacdepannage.fr/`

Résultats. Le module de recherche par suffixe est destiné à être utilisé soit en première approche, soit en complément du mode de collecte décrit en section 2.1. C'est ce que nous avons réalisé pour l'étude du suffixe *-able*. La campagne de collecte nous a permis de repérer 2 011 nouveaux lexèmes (avant révision), qui s'ajoutent à ceux obtenus par la méthode présentée précédemment, soit une augmentation du corpus d'environ 73%. Toutefois, les formes ainsi relevées doivent ensuite être analysées, ne serait-ce que pour retrouver leur lexème base, ce qui n'est bien entendu pas nécessaire dans le cas de la première méthode où l'on part de bases connues. Nous présentons dans la section suivante la phase d'analyse qui, elle aussi, fait appel au Web comme corpus.

3. Analyse morphologique des candidats

Le but de cette procédure est double : elle permet de filtrer le bruit résiduel dans les formes collectées par la recherche par suffixe, mais aussi, pour celles qui sont correctes, de calculer leur base. Le principe de cette analyse est le suivant : à partir d'une forme nouvelle, nous calculons la ou les formes de leurs bases plausibles (par exemple, la base verbale *géocoder* pour la forme collectée *géocodage*[12]), puis nous vérifions leur existence sur le Web. Si besoin est, nous vérifions également l'existence d'un lien sémantique entre le lexème construit et sa base. Plus précisément, nous avons envisagé deux niveaux de vérification. Le premier, minimal, consiste à rechercher des occurrences de ces bases sur le Web. Le second mode est plus strict. Il impose de trouver sur le Web des pages qui contiennent à la fois les formes collectées et des occurrences de leurs lexèmes bases. Par exemple, on cherchera des pages qui contiennent à la fois *géocodage* (au singulier ou au pluriel) et une des formes fléchies du verbe *géocoder*. Ces deux modes de filtrage sont présentés successivement.

3.1. Recherche du lexème base

Pour déterminer si une forme qui comporte un suffixe graphémique donné est effectivement construite au moyen du suffixe linguistique correspondant, on peut vérifier son appartenance à la série correspondant à ce suffixe. En d'autres termes, on doit pouvoir lui associer un lexème base. Par exemple, pour que *wapisable* soit un adjectif construit en *-able*, il faudrait que son lexème base soit le verbe *wapiser*. Si c'est le cas, il devrait exister des attestations de certaines des formes de ce verbe. Pour le vérifier, on soumet donc au méta-moteur la requête dont un extrait est donné en (10).

(10) `(wapise OR wapises OR wapisé OR …… OR wapisant)`

Plusieurs de ces formes (la plupart impersonnelles, comme *wapiser*, *wapisant*, *wapisés*...) ont ainsi pu être repérées et *wapisable* est donc retenu (ainsi que son verbe base, incidemment).

La requête précédente est construite au moyen du module de prédiction morphologique. Les formes qui la composent sont générées de la même façon que les formes des lexèmes construits en section 2.1 à ceci près que les schémas appris et appliqués sont étiquetés par des catégories morphosyntaxiques. Ainsi, dans le cas de la suffixation en *-able*, une forme est générée pour chaque étiquette verbale. Cependant, certaines formes ont été volontairement omises pour limiter la taille de la requête : celles du subjonctif imparfait, du conditionnel passé et du passé simple. Les formes générées sont d'autre part filtrées en utilisant le lexique de référence pour supprimer celles qui appartiennent aussi à des catégories non verbales. Par exemple, le nom *bridage* est construit au moyen du suffixe *-age* sur le verbe *brider*. Or la forme *bride* (1re et 3e personne du singulier du présent de l'indicatif) est aussi une forme nominale et sera donc exclue de la requête. C'est également le cas de *brides* (2e personne du singulier du présent de l'indicatif).

[12] Le géocodage est le répérage d'un lieu géographique dans un système de coordonnées spatiales à l'échelle de la planète.

Ce type de vérification permet de supprimer un certain nombre d'erreurs qui échappent au module de recherche par suffixe comme "`sonttresponsable`" ou "`unstoppable`" (pour le suffixe *-able*) car *sonttresponser* n'est pas attesté et *unstopper* l'est seulement dans des paragraphes en anglais. Plus généralement, 25 % des formes en *-able* repérées par le module de recherche par suffixe ne passent pas ce filtre. Ce filtrage étant relativement grossier, il n'élimine que très peu de formes correctes (moins de 5%, voir quelques exemples ci-dessous).

Parmi les principales limites de cette technique signalons le fait que, souvent, les fautes que l'on rencontre pour les lexèmes construits apparaissent aussi dans leurs lexèmes bases. On trouve par exemple à la fois *demmarable* et *demmarer*, traduisant chez certains auteurs une forme de cohérence dans une orthographe erronée. Certaines fautes de frappe interviennent également, comme dans le cas de *superstable* où une occurrence de *superster* (pour *superstar*) a déclaré le couple comme valide... D'autre part, cette technique élimine certains lexèmes construits qui sont à la fois préfixés et suffixés comme *inéffilochable* ou *inlocalisable*. Ces deux formes sont en fait construites par préfixation à partir de *éffilochable* et *localisable*. Si le second est bien attesté, le premier ne l'est pas. En revanche, ni *inéffilocher*, ni *inlocaliser* ne sont attestés. Si l'on s'intéresse seulement aux lexèmes construits dont la dernière étape de construction est la suffixation en *-able*, alors le filtre convient. Nous revenons sur ces problèmes en section 3.3.2 où nous proposons un traitement pour ce type de lexèmes construits. Signalons également que cette technique impose de faire des hypothèses linguistiques fortes en particulier sur les catégories des lexèmes bases. Ainsi, toutes les formes en *-able* ne sont pas construites sur des verbes. Ce filtrage élimine une forme comme *piscinable* construite sur *piscine*[13].

Nous présentons maintenant le second mode de filtrage, plus restrictif mais garantissant une précision bien plus élevée dans les résultats.

3.2. Restriction par les cooccurrences

Comme on vient de le voir, la vérification par recherche des lexèmes bases constitue un complément utile pour le module de recherche par suffixe. Malgré son important bruit résiduel, il bien est adapté aux études linguistiques qui comportent nécessairement un important travail de dépouillement indispensable à la compréhension des phénomènes étudiés. Toutefois, le taux de précision est bien trop faible dans les cas où l'on souhaite seulement constituer des ressources lexicales « de bas niveau », utilisables pour le TAL par exemple, pour lesquelles la phase de révision doit être la plus courte possible. Nous proposons à cet effet une technique qui permet de réduire le nombre des candidats erronés en appliquant un filtre plus sélectif. L'idée est d'ajouter à la vérification précédente une contrainte de cooccurrence. En d'autres termes, un candidat est retenu seulement s'il apparaît avec une forme de sa base dans une même page Web. Le but est dans ce cas de garantir une meilleure précision dans les résultats obtenus quitte à réduire la couverture de la base de données construite *in fine*. Cette technique est inspirée des travaux de Baayen et Neijt 1997 et de Xu et Croft 1998. Nous en rappelerons les principes généraux avant d'aborder ses aspects techniques, et les effets de son adaptation au Web. Cette technique a été employée à grande échelle dans le cadre de l'extension du lexique Verbaction (*cf.* section 3.3.1).

3.2.1. Principes

Baayen et Neijt 1997 ont montré que les contextes des mots dérivés contiennent fréquemment des « ancres » c'est-à-dire des indices qui facilitent leur interprétation. C'est ainsi que les mots dérivés apparaissent régulièrement précédés (et plus rarement suivis) d'une forme du lexème sur lequel ils sont construits.

La présence du lexème base dans le contexte peut relever de la coopération conversationnelle : il s'agit alors de fournir des éléments permettant d'interpréter le dérivé en le mettant en relation avec sa base. Si la dérivation morphologique est un moyen d'exprimer des notions complexes de manière concise, on peut néanmoins supposer qu'en français comme dans d'autres langues, elle permet assurer la continuité thématique et référentielle dans le discours mais aussi d'éviter les répétitions et même de varier la façon dont les idées sont présentées et développées. Dans tous ces cas, on peut faire l'hypothèse que les lexèmes construits peuvent

[13] Dans le contexte des annonces immobilières, un terrain est dit *piscinable* si l'on peut y creuser une piscine.

être utilisés pour « paraphraser » leurs lexèmes bases. La cooccurrence des deux lexèmes est ainsi prévisible sans être systématique. Cette observation a été exploitée en RI par Xu et Croft 1998 pour filtrer des appariements morphologiques produits par un raciniseur. Le filtrage est basé sur une variante de la mesure d'information mutuelle attendue (EMIM ; *Expected Mutual Information Measure*) calculée entre des formes morphologiquement apparentées cooccurrentes dans des fenêtres de 200 mots. En nous appuyant sur cette même observation, nous proposons une technique simple permettant de repérer des couples de lexèmes morphologiquement apparentés. Il s'agit d'explorer le Web pour y chercher des pages qui contiennent des formes des deux lexèmes qui composent chaque couple. Nous présentons ci-dessous les aspects techniques de cette méthode, ainsi que certains exemples typiques de cooccurrence (*cf.* section 3.2.3).

3.2.2. Aspects techniques

À partir d'un couple (lexème base, lexème construit), une requête booléenne est construite et soumise au méta-moteur, indiquant que l'on recherche des pages contenant au moins une forme de la base et une forme du lexème construit. Par exemple, sur l'hypothèse que *piscine* permet de construire *piscinable*, la requête (11) est générée et soumise au méta-moteur :

(11) `(piscinable OR piscinables) AND (piscine OR piscines)`

Cette fonctionnalité du méta-moteur de Webaffix destinée au repérage des cooccurrences nécessite toutefois que des précautions supplémentaires soient prises. En plus des vérifications habituelles du découpage, de la langue du contexte et des diverses formules informatiques, Webaffix élimine certaines cooccurrences qui n'ont pas de justification linguistique et ne correspondent donc pas à notre hypothèse de travail. En effet, certaines pages Web sont des catalogues de mots et non des textes ni même des assemblages de textes. Les types de pages repérés sont :

Les listes de mots ordonnées lexicographiquement. Il peut s'agir de lexiques généraux ou spécialisés établis soit par des linguistes informaticiens pour des applications de TAL, soit par des spécialistes de la sécurité informatique pour ce qui concerne les mots de passe. Dans les deux cas, les mots sont placés les uns après les autres, par ordre alphabétique, et ces types de pages sont donc détectables.

Les tables de fréquences lexicales. Les cooccurrences non linguistiques apparaissent aussi dans des pages décrivant des analyses lexicométriques, mises à disposition sur le Web, et pouvant prendre la forme de tables de fréquence des mots extraits d'un corpus. Dans ce cas, l'ordre n'est plus alphabétique, mais reste encore détectable : la détection se base sur l'alternance systématique mot/nombre.

(12) `le 744 de 699 que 516 je 491 vous 347 à 324 et 312...`

Les leurres pour moteurs de recherche. Un troisième type de pages qui présentent ce problème sont des sites commerciaux proliférant sur le Web (en grande partie des sites pornographiques), qui tentent d'attirer les internautes via les moteurs de recherche en plaçant systématiquement sur leurs pages de très grandes listes de mots, provenant bien souvent de lexiques génériques de la langue pour arriver ainsi en tête des listes des réponses dès qu'une requête comporte beaucoup de mots.

Malheureusement, il apparaît de plus en plus de cas où les mots insérés ne sont pas dans l'ordre alphabétique, ni même placés dans un endroit précis du texte. Une raison à cela est sans doute le développement des modes de détection de telles pages par les moteurs de recherche ou par les logiciels de filtrage. Dans ce cas, il est absolument impossible de détecter de tels non-textes, sans mettre en place des moyens lourds. Pour l'instant, toutefois, ces méthodes semblent apparaître assez rarement, mais risquent de se multiplier rapidement.

3.2.3. Types de cooccurrences — le cas des déverbaux en *-age*

Comme on le voit au regard de ces cas-limites, la notion de cooccurrence est problématique sur le Web. Cependant, si l'hypothèse initiale de Baayen et Neijt a été appliquée à des textes journalistiques ou techniques, elle n'en reste pas moins valide pour les pages Web. Elle correspond en fait à différents phénomènes, dont nous présentons certains exemples ici. Une étude plus approfondie est en cours, mais nous avons déjà pu dégager une certaine typologie des cas de cooccurrences positifs. Nous envisageons également d'utiliser des procédures automatiques de détection de ces schémas afin de définir un mode de vérification encore plus précis.

Titres. Dans certains cas, assez courants dans les pages en langue de spécialité, le déverbal (resp. le verbe) constitue tout ou partie du titre, et le verbe (resp. le déverbal) est utilisé dans le développement (figures 3 et 4).

> ***2ème étape : L' abacarage***
> Chaque pièce du filet est ensuite abacaré c'est-à-dire que l'on "arrondi" les bords. Pour cela on utilise un outil tranchant appelé abacare qui permet d'enlever un filet de cuir d'une faible largeur.

Figure 3. — *Exemple de cooccurrence dans une configuration titre/développement.*

> Pourquoi baigner ?
> Le document de M.J.Morière m'incite à penser que cette pratique du baignage remonterait au XIX° siècle seulement.

Figure 4. — *Exemple de configuration développement/titre.*

(Inter-)Définitions. Toujours dans le cadre de pages en langue de spécialité, on trouve également des exemples d'interdéfinition entre la base et le déverbal. Celles-ci peuvent prendre la forme d'une entrée de lexicographique, comme en figure 5 ou bien être insérées au fil du discours comme en figure 6.

> affacturer, v.
> *Domaine :* Économie et finances.
> *Définition :* Pratiquer l'affacturage.
> *Anglais :* factor (to).
> (Source : arrêté du 29 novembre 1973)

Figure 5. — *Exemple de cooccurrence dans une définition lexicographique.*

> [...] les béotiens, qui, ignorant tout des subtilités du "cavage" (en Provence le verbe "caver" recouvre l'action de déterrer les truffes), massacrent les truffières [...]

Figure 6. — *Exemple de cooccurrence dans une définition en discours.*

Alternance classique. Dans les autres cas, les formes du verbe et du déverbal s'alternent naturellement, sans explicitation particulière de l'une vers l'autre (figure 7).

> [...] Régine zappait : elle accorda 35 secondes aux aventures de Wallace et Gromit
> [...] Régine coupa le son et augmenta le rythme de zappage.

Figure 7. — *Exemple de cooccurrence dans une alternance classique.*

Autres caractéristiques. En fonction du degré d'innovation du nom déverbal, il arrive que celui-ci soit entouré de précautions (guillemets) ou précédé d'avertissements comme en figure 8.

[…] SYNTEC exprime son désaccord concernant la demande des syndicats de ne pas permettre la pratique de "**l'écrêtage**" du contrôle du temps de travail. […] Comme par hasard, il faisait abstraction d'une des demandes principales concernant le dispositif permettant la mesure du temps de travail effectif : nous avions à l'unanimité insisté pour que ces dispositifs quels qu'il soient, n'aient pas la possibilité d'écrêter le nombre d'heures travaillées quotidiennement, mais l'obligation de représenter la réalité brute des heures de travail [...]

Figure 8. — *Exemple de signalisation du néologisme.*

Les cas où les deux formes sont entre guillemets ou en mention indiquent des emplois typiquement métalinguistiques comme en figure 9.

Later : Façon de faire et manière de vivre gratuitement explosive.
[…] Le terme "latage" est courament employé. Ex : "*Piège de Cristal* est un gros latage".

Figure 9. — *Exemple d'emploi en mention.*

Les fréquences relatives au sein d'un texte de la base et du déverbal ne permettent pas pour l'instant de tirer des conclusions directes. Il apparaît toutefois que le déséquilibre, s'il existe, reste constant d'un texte à l'autre. Par exemple, pour *encapsulage*:*encapsuler*, c'est systématiquement le verbe qui est le plus fréquent (le rapport est de 1 à 3 en moyenne), alors que pour *covoiturage*:*covoiturer* c'est le déverbal qui est à chaque fois le plus fréquemment attesté avec un rapport de 1 à 10. Il est probable qu'au-delà du type de construction morphologique et des caractéristiques internes des lexèmes, le style de discours à une influence déterminante sur les fréquences d'emploi.

3.3. Exemples d'application

Le principe de la recherche de cooccurrences entre un lexème dérivé et son lexème base a été appliqué dans différents contextes. Nous présentons ici deux exemples de telles applications : une première, orientée TAL, ayant pour but d'étendre une ressource morphologique générique pour le français, et une seconde, plus théorique, sur l'ordonnancement des procédés morphologiques.

3.3.1. Extension du lexique Verbaction

L'analyse morphologique et la vérification par les cooccurrences ont été utilisées pour étendre le lexique Verbaction qui contient 6 471 couples verbe:nom, tels que le nom est morphologiquement apparenté au verbe et qu'il dénote l'action ou l'événement correspondant à ce verbe comme *élire*:*élection*, *déménager*:*déménagement*... Webaffix a permis d'augmenter ce lexique, en y ajoutant des couples de lexèmes moins courants, voire des néologismes (Tanguy et Hathout 2002).

Verbaction est une ressource avant tout destinée au traitement des variations morpho-syntaxiques (Jacquemin 2001). Les hypothèses théoriques sous-jacentes à sa réalisation sont minimales. Ainsi, aucune distinction n'est faite entre les conversions et les constructions suffixales dans la mesure où la nature de lexème converti ou suffixé n'est pas prise en compte par les systèmes qui utilisent cette ressource. Les relations sémantiques spécifiques entre noms et verbes ne sont pas non plus explicitées.

L'extension de Verbaction a été réalisée en 3 étapes, à savoir (1) la recherche par suffixe de formes se terminant en *-ade*, *-age*, *-ance*, *-ement*, *-ence*, *-erie*, *-tion* ; (2) la prédiction des formes des lexèmes bases (par exemple, on prédit les formes de *orthorectifier* pour *orthorectification*[14]) ; (3) la vérification par cooccurrence au moyen du méta-moteur. Pour ce qui

[14] L'*orthorectification* est un procédé de traitement d'image numérique.

concerne la prédiction du lexème base, l'apprentissage des schémas se fait directement sur la base Verbaction existante, et permet d'atteindre une bien meilleure précision.

La vérification par les cooccurrences est relativement sévère puisqu'en moyenne seuls 13 % des candidats passent ce filtre. En contrepartie, la précision est relativement élevée : 52 %, ce qui constitue un résultat très honorable pour un filtrage totalement automatique. Ce score global cache en fait une disparité entre les différents suffixes, allant de 20 % pour *-erie* à 85 % pour *-age*.

3.3.2. Étude des lexèmes construits par préfixation et suffixation

Comme on a pu le voir pour certains exemples dans les sections précédentes, les lexèmes construits sont parfois le résultat d'une suffixation et d'une préfixation. Nous présentons ici quelques résultats et pistes d'investigation sur ces phénomènes complexes obtenus par l'utilisation de Webaffix.

Tout d'abord, le module d'analyse morphologique peut être utilisé, avec le même principe que celui présenté en section 2, pour apprendre des schémas de préfixation. Un premier résultat de cet apprentissage sur un ensemble de formes nouvelles a permis la mise au jour de préfixes nouveaux comme *euro-* (*europétition*), *cyber-* (*cyberpublication*), *vapo-* (*vapogazéification*) ou *web-* (*webpromotion*). Ensuite, se pose le problème de l'analyse des formes nouvelles dans lesquelles une opération de préfixation a été repérée, comme par exemple *aquamarquage* ou *antipollution*. L'analyse morphologique doit être capable, à terme, de proposer une base pour ces lexèmes en envisageant les deux opérations (préfixation et suffixation), et doit donc repérer l'ordre d'application de celles-ci.

Les schémas de préfixation et de suffixation sont utilisés pour prédire trois groupes de bases potentielles pour chaque lexème collecté : celles qui sont préfixées seulement, celles qui sont suffixées seulement et celles qui sont à la fois préfixées et suffixées. Par exemple, *défragmentation* sera associé à *fragmentation* (par préfixation), à *défragmenter* (par suffixation) et à *fragmenter* (par préfixation et suffixation). On effectue ensuite séparément le filtrage par les cooccurrences pour chacune de ces bases. L'objectif est de déterminer lesquelles, parmi les cooccurrences suivantes sont présentes (figure 10).

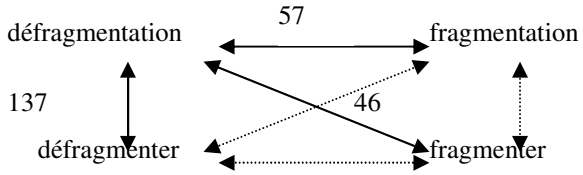

Figure 10. — *Détermination de l'ordre des affixations.*

Les liens en pointillés n'ont pas été testés. On peut ainsi identifier la base la plus probable pour le candidat. Lorsque les trois cooccurrences existent, la décision peut être prise en comparant les nombres de pages de qui contiennent chacune d'elle. Ces valeurs sont données dans la figure 10 pour le quadruplet (*défragmentation*, *défragmenter*, *fragmentation*, *fragmenter*). On constate ainsi que parmi les liens testés, *défragmentation*:*défragmenter* est le plus fort, résultat conforme à l'intuition.

4. Conclusion

Les utilisations que nous venons de présenter de la boîte à outils Webaffix témoignent de sa réelle capacité à répondre aussi bien à des besoins de linguistes en leur permettant de réaliser des études morphologiques « extensives » de procédés constructionnels, mais aussi d'informaticiens en les aidant à créer ou étendre des ressources lexicales pour le TAL.

Les problèmes rencontrés par cette approche du Web comme corpus, et qui ne sont qu'en partie réglés par les procédures que nous avons décrites, sont contrebalancés à notre avis par la grande richesse des résultats, qu'aucune autre source de données ne peut pour l'instant égaler. De tels travaux utilisant le Web ne sont à l'heure actuelle qu'à leurs balbutiements en

linguistique. Comme le notent Kilgarriff et Grefenstette, 2003, les moteurs de recherches actuellement disponibles ne proposent pas les fonctionnalités nécessaires à une utilisation confortable, comme l'utilisation non contrainte des troncatures, ou une meilleure indexation des lexèmes. Webaffix serait un des outils qui bénéficierait d'un moteur généraliste plus sophistiqué.

Il se pourrait également que, à l'inverse, l'étude des créations lexicales constitue un indice important dans la caractérisation des styles de discours sur le Web. Dans une étude en cours, nous nous concentrons en effet sur les différences de fonctionnement et de productivité entre des suffixes supposés équivalents, qui pourraient s'expliquer entre autre par le contexte extra-linguistique des productions.

Le code Perl du module de recherche par suffixe est téléchargeable à l'URL http://www.univ-tlse2.fr/erss/membres/tanguy/webaffix.html

Références

BAAYEN, R. H. & NEIJT, A. 1997 « Productivity in context: a case study of a Dutch suffix » *Linguistics*, 35, 565–587.

CORBIN, D. 2001. « Préfixe et suffixes : du sens aux catégories » *Journal of French Language Studies*, 11(1):41–69.

DAL, G. & NAMER, F. 2000 « Génération et analyse automatiques de ressources lexicales construites utilisables en recherche d'informations » *T.A.L.*, 41(2), 423–446.

GREFENSTETTE, G. 1999. « The World Wide Web as a Resource for Example-based MT Tasks », *Proceedings of the 21st ASLIB International Conference on Translating and the Computer*, London, November 1999.

KILGARIFF, A. & GREFENSTETTE, G. 2003. « Introduction to the special issue on the Web as corpus » *Computational Linguistics*, 29(3):333-347.

HATHOUT, N. & NAMER, F. & DAL, G. 2002. « An Experimental Constructional Database: The MorTAL Project ». In P. Boucher (ed.) *Many Morphologies*, 178-209, Cascadilla Press, Somerville.

HATHOUT, N. & PLENAT, M. 2002. « Quelques considérations sur la suffixation en *–able* ». Communication aux *Journées de Morphologie de l'ERSS*. Toulouse.

JACQUEMIN, C. 2001. *Spotting and Discovering Terms through NLP*. MIT Press, Cambridge MA.

JACQUEMIN, C. & BUSH, C. 2000. « Fouille du Web pour la collecte d'entités nommées » *Actes de la 7ème conférence TALN, Lausanne.*

JANICIJEVIC, T. & WALKER, D. 1997. « Neolosearch: Automatic Detection of Neologisms in French Internet Documents. » *Proceedings of ACH-ALLC'97*, 93-94, Kingston, Canada.

NAMER, F. 2003. « Valider les unités morphologiques par le Web ». In Silexicales 3, *Actes du 3e Forum de morphologie*, 142-150, Lille.

PLENAT, M. 1988. « Morphologie des adjectifs en *-able* » *Cahiers de grammaire*, 13, 101–132.

PLENAT M., TANGUY L., LIGNON S. & SERNA N. 2002. « La conjecture de Pichon » *Corpus*, **1**(1):105-150.

RESNIK, P. 1999. « Mining the Web for bilingual text » *Proceedings of the 37th Meeting of the ACL*, 527-534, College Park, Maryland.

TANGUY, L. & HATHOUT, N. 2002 « Webaffix : un outil d'acquisition morphologique dérivationnelle à partir du Web ». *Actes de la 9ème Conférence Annuelle sur le Traitement Automatique des Langues Naturelles (TALN-2002)*, pp. 245–254, Nancy : ATALA.

XU J. & CROFT W. B. 1998 « Corpus-Based Stemming using Co-occurrence of Word Variants » *ACM Transaction on Information Systems*, 16(1) : 61–81.