Le TLFi ou Trésor de la Langue Française informatisé
http://www.atilf.fr/tlfi

Jean-Marie Pierrel, Jacques Dendien, Pascale Bernard
ATILF – UMR 7118 CNRS – Université de Nancy
Analyse et Traitement Informatique de la Langue Française
44, avenue de la Libération
BP 30687
F- 54063 NANCY cedex
Jean-Marie.Pierrel@atilf.fr ; Jacques.Dendien@atilf.fr ; Pascale.Bernard@atilf.fr

Résumé :
Le Trésor de la Langue Française (TLF) est un grand dictionnaire de langue française en 16 volumes réalisé par l'Institut National de la Langue Française (INaLF, laboratoire du C.N.R.S) entre le début des années 60 et le milieu des années 90. Le laboratoire ATILF, successeur de l'INaLF depuis le 1er janvier 2001 a mis en oeuvre son informatisation et ouvert sur le web (http://www.atilf.fr/tlfi) la version informatisée de ce dictionnaire – le TLFi – au cours de l'année 2002. L'objectif de cette présentation sera d'expliciter successivement les spécificités de ce dictionnaire informatisé et du balisage XML sous-jacent ayant rendu possible cette réalisation, et de montrer divers exemples d'exploitation et de recherche dans le TLFi.
On s'attachera plus spécifiquement à montrer les fonctionnalités de recherche transversales accessibles depuis l'interface Web et rendues possibles grâce à l'exploitation plein texte de cette version du TLFi qui correspond à un balisage XML fin du contenu du TLF, après rétroconversion de l'ensemble du dictionnaire.

1. Présentation générale

Le TLFi s'appuie sur Le Trésor de la Langue Française (TLF), dictionnaire de langue réalisé entre le début des années 60 et le milieu des années 90 (CNRS 1971-1993), par l'Institut National de la Langue Française dont est issu l'ATILF depuis le 1/1/2001.
Le TLFi (Dendien 1996, Dendien & Pierrel 2003) se présente à la fois comme une base lexicologique et une base de connaissances dont l'accessibilité est immédiate via internet. Il se distingue des autres dictionnaires électroniques par la finesse de la structuration des données en « objets » interrogeables selon divers critères, et par une interface simple et conviviale qui offre trois niveaux de consultation via le logiciel Stella :

- consultation simple du dictionnaire, article par article, avec mise en évidence ou non de tel ou tel type d'information (définition, exemple…) ;
- consultation transversale, par une requête élémentaire utilisant certains critères (définitions dans le domaine de la cuisine…) ;
- consultation plus complexe croisant plusieurs critères. Ces requêtes peuvent être élémentaires ou multi-objets (on peut par exemple extraire tous les mots se terminant par un suffixe et extraire de cette liste ceux qui ont un sens péjoratif). Elles peuvent inclure ou exclure un contenu, etc..

2. Spécificités du contenu

Le TLFi se distingue aussi par la richesse de son matériau et la complexité de sa structure :

- originalité de sa nomenclature (incluant préfixes, suffixes et autres éléments formants) : c'est en tout 100 000 mots avec leur étymologie et leur histoire, et 270 000 définitions ;
- richesse des objets métatextuels (vedettes, codes grammaticaux, indicateurs sémantiques ou stylistiques, indicateurs de domaines, définitions, exemples...) ;
- richesse des 430 000 exemples, tirés de deux siècles de production littéraire française ;
- diversité des rubriques : une rubrique synchronie couvrant la période 1789 à nos 1993, une rubrique étymologie et histoire, et une rubrique bibliographie pour les principaux les articles.

3. Spécificités du balisage

Un des principaux avantages d'un dictionnaire informatisé est de permettre d'effectuer des recherches transversales "plein texte" à travers la totalité de son contenu. Cependant, chaque occurrence du texte cherché a une signification qui dépend essentiellement du type de l'objet textuel dans lequel elle est localisée. Restreindre une recherche "plein texte" à tel ou tel type donné permet donc de diminuer le bruit et d'accroître la précision des recherches. Afin de rendre les interrogations du TLF plus précises et significatives, il a été procédé à un balisage textuel XML de tout le texte du dictionnaire en y injectant des balises repérant le début, la fin et le type de chaque objet textuel rencontré. Une quarantaine de types d'objets différents a ainsi été introduite à l'aide d'automates experts, alors que bien des dictionnaires informatisés s'arrêtent à quelques types essentiels (souvent définitions et/ou citations). Il est ainsi possible de limiter une recherche "plein texte" à l'un de ces types.

Afin d'introduire encore plus de précision dans les requêtes, il convenait d'ouvrir une nouvelle dimension : celle de la structure hiérarchique de chaque article. En effet un article de dictionnaire (à l'exception des articles les plus élémentaires) introduit une structure explicitée dans le TLF (comme dans bien d'autres dictionnaires) par des sigles de structure hiérarchisés (I, II, ...; A, B,...; 1, 2, ..., a, b,...). Une indication de domaine technique "mécanique" apparaissant au niveau B, par exemple, signifie clairement que les éventuelles subdivisions hiérarchiques du B traitent de la mécanique. Et une définition trouvée dans le paragraphe b) appartenant au paragraphe 2) qui appartient au B introduit nécessairement un sens usité dans le domaine de la mécanique. De manière générale, il est donc possible d'introduire systématiquement une relation entre deux objets X et Y : X étant hiérarchiquement inférieur, égal ou supérieur à Y.

Il est possible d'effectuer des requêtes comportant N objets, chaque objet ayant un contenu textuel imposé, et les liens relationnels entre objets étant imposés. Pour que la requête ait un sens, il suffit que le graphe dont les sommets sont les objets, et dont les arcs sont les relations hiérarchiques imposées soit connexe. Par exemple, soit une requête spécifiant qu'un objet A de type "catégorie grammaticale" contienne le mot "verbe", qu'un objet B de type "indication de domaine technique" contienne le mot marine, qu'un objet C de type "définition" contienne le mot "voile" ou "voiles", que A soit hiérarchiquement supérieur à B (ce qui signifie que l'indication "marine" est afférente à un verbe), et enfin que B soit hiérarchiquement supérieur à C (ce qui signifie que la définition est trouvée dans une section

d'article traitant de marine) : une telle requête revient de toute évidence à rechercher tous les verbes utilisés dans la marine pour la manœuvre des voiles.
La richesse du balisage du TLFi (plus de 36 millions de balises XML) permet l'identification de nombreux types d'objets et leur mise en relation hiérarchique, et contribue à obtenir des résultats d'un degré de pertinence très élevé, chaque contrainte hiérarchique contribuant à filtrer le bruit.
Pour avoir une idée plus précise de la finesse du balisage, il convient de noter qu'au total, on peut faire le dénombrement suivant, après validation de l'ensemble du dictionnaire :

nombre de balises typographiques : 17 364 854
nombre de balises décrivant la hiérarchie : 1 070 224
nombre de balises repérant les objets textuels : 18 178 634, dont 92997 entrées et 64346 locutions correspondant à 271166 définitions illustrées par 427493 exemples
nombre total de balises XML : 36 613 712
niveau de profondeur hiérarchique maximal : 23.

4. Exemples d'exploitation et de recherche dans le TLFi

S'appuyant sur le logiciel Stella développé au laboratoire et doté d'un compilateur de langage de requête très avancé, le TLFi présente un certain nombre de spécificités qui nous ont conduits à définir un langage de requête spécifique au TLFi et à réaliser son compilateur. Le langage de requête définit un vocabulaire associant un mot-clé à chaque élément XML. Par exemple, le mot-clé définition est associé à l'élément DEF de la DTD. Cette association est réalisée à l'aide d'un simple fichier de correspondance, très facile à mettre à jour avec un simple éditeur de texte. Il est ainsi très aisé, par exemple, de réaliser un jeu de mnémoniques adaptés à des utilisateurs anglophones ou hispanophones, et aussi de permettre ou d'interdire la visibilité de tel ou tel élément XML.

4.1. Principes du langage de requête

Le principe du langage de requête du TLF est extrêmement simple. En voici un exemple.
La requête : " X:domaine(médecine);Y:définition(instrument);
Y i (X);Z:source(Académie);Z i Y; "
s'interprète de la manière suivante :

- soit X un indicateur de domaine technique contenant le mot médecine,
- soit Y une définition contenant le mot instrument,
- Y est inclus dans la portée de X (on remarquera la notation " Y i (X) " qui signifie que l'élément X est inclus dans la portée de l'élément Y, alors que " Y i X " signifie que l'élément X est inclus dans l'élément Y). Cette clause (cf. la notion de portée des objets) implique que la définition est valable dans le domaine de la médecine,
- soit Z une source (bibliographique) contenant le mot Académie, inclue dans Y (cette clause implique que la définition est empruntée au dictionnaire de l'Académie française).

Cette requête va déclencher la recherche de tous les triplets (X, Y Z) respectant le système de contraintes énoncé. Elle peut se paraphraser ainsi : " Chercher, dans le domaine de la médecine, les définitions relatives à un instrument et empruntées au dictionnaire de l'Académie ".

On remarquera l'aspect non procédural du langage de requête qui permet de décrire un système de contraintes à résoudre, mais pas les opérations nécessaires pour y parvenir. On voit, dans cet exemple que le langage de requête met en jeu à la fois le type des objets textuels, leur contenu textuel éventuel, et les relations entre les objets de la requête. La différence entre les deux types de relation (inclusion et dépendance hiérarchique) est fondamentale et constitue le seul point délicat à appréhender pour une bonne manipulation du langage de requête.

Pour conclure cette présentation rapide du langage de requête du TLFi, il convient de comprendre la notion de connexité d'une requête : toute requête introduit un ensemble d'éléments ({X, Y, Z} dans notre exemple) muni d'une relation binaire, et peut donc être considéré comme un graphe G. Si le graphe G est non connexe, cela signifie que la requête demande de rechercher au moins deux sous-ensembles d'éléments sans aucun lien logique l'un avec l'autre. Une telle requête n'a évidemment aucun sens, et sera rejetée par le compilateur.

4.2. Les différentes possibilités d'exploitation du langage de requête

Comme nous venons de le voir, une requête permet de décrire ce qu'il faut chercher. Il reste encore à exprimer la manière dont les différents résultats trouvés doivent être restitués à l'auteur de la requête. Ce dernier point dépend du contexte dans lequel la requête a été élaborée : exploitation à distance via un serveur de requêtes ou exploitation sous contrôle d'une interface pour le Web.

4.2.1. Exploitation à distance via un serveur de requêtes. Ce mode, réservé au monde de la recherche, consiste à utiliser Internet pour poster une requête à une application résidant sur le serveur gérant le TLFi. La requête est empaquetée dans une coquille XML. Il est nécessaire de compléter la requête proprement dite avec des clauses exprimant quelles informations on veut obtenir.

4.2.2. Exploitation sous contrôle d'une interface pour le Web. Dans ce mode, mis en œuvre dans la version Web, l'utilisateur ne manipule pas directement le langage de requête. En effet, l'interface graphique lui propose différents formulaires de recherche. Lors de la soumission du formulaire, les données de l'utilisateur sont collectées et automatiquement transformées en une expression de requête. On trouvera dans (Bernard et al. 2002) une présentation explicitant les divers usages possibles de cette interface web.

L'interface offre des possibilités graduées :

- recherche simple : elle consiste à rechercher les articles concernant le mot tapé par l'utilisateur ;
- recherche assistée : elle propose un formulaire de recherche permettant à l'utilisateur d'imposer des contenus textuels à des types d'objets donnés, le logiciel prenant en charge de manière transparente les relations hiérarchiques entre objets ;
- recherche complexe : elle propose un formulaire permettant d'exprimer une requête avec une puissance comparable à la manipulation directe du langage de requête. L'utilisateur peut explicitement spécifier les types des objets recherchés, leurs contenus textuels éventuels et leurs relations hiérarchiques.

Dans tous ces cas, lors de la soumission du formulaire, le logiciel collecte les informations, les transforme en une requête qui est soumise au compilateur, puis exécutée.
Les résultats de la recherche peuvent être affichés suivant deux modes différents :

- mode global : supposons qu'une recherche mette en jeu N objets textuels. Chaque résultat est donc constitué d'un N-uplet d'objets. Le mode global consiste à afficher tous les résultats. Pour chaque résultat est affiché le contenu textuel de chaque objet du N-uplet, ou éventuellement, au choix de l'utilisateur, le sous-ensemble du N-uplet constitué par les objets qui l'intéressent le plus ;
- mode "en contexte" : il consiste à afficher les résultats un par un. Le texte de l'article dans lequel le N-uplet a été trouvé s'affiche intégralement. Le début et la fin du texte correspondant à chaque objet du N-uplet sont matérialisés par de petites images colorées et numérotées de 1 à N. Dans le cas où toute la totalité de l'article n'est pas visible à l'écran (ceci arrive fréquemment en raison de la taille importante des articles du TLF), l'utilisateur dispose de boutons de navigation lui permettant de centrer l'affichage immédiatement sur l'objet du N-uplet qui l'intéresse le plus particulièrement.

5. Ressources complémentaires en vue d'améliorer l'interface

L'apport décisif du dictionnaire électronique est de permettre des recherches transversales, grâce à des mécanismes tels que ceux décrits ci-dessus. Aussi passionnante que soit leur étude, il faut constater avec une certaine humilité que la grande masse des utilisateurs n'est guère intéressée par de telles possibilités : l'accès se fait par les mots afin d 'en vérifier l'orthographe, le sens, et éventuellement trouver des exemples d'emploi.
Il est amusant de constater que cet emploi pourtant basique et traditionnel du dictionnaire est trop souvent méconnu, ce qui plonge l'utilisateur dans une situation ubuesque : la plupart des dictionnaires électroniques ne permettant d'accéder aux mots que si l'on en connaît l'orthographe exacte, l'utilisateur qui désire vérifier l'orthographe d'un mot est censé en connaître l'orthographe !
Après quelques mois de mise en ligne du TLF sur Internet, nous avons constaté que si l'ignorance de l'orthographe du mot recherché est la première cause d'échec, il en existe d'autres : l'utilisateur omet les accents (habitude d'ignorer les caractères accentués des claviers, ou claviers sans caractères accentués ?) ; l'utilisateur donne une forme flexionnelle (ex. « fatigante », au lieu de « fatigant »). Dans des cas extrêmes, il peut y avoir superposition de tous ces phénomènes : le fait de frapper « jenero » pour trouver « général » cumule l'ignorance de l'orthographe, des accents et propose une forme flexionnelle.
Devant ce constat, nous sommes attachés à trouver des remèdes en développant des algorithmes permettant un prétraitement morphologique et phonétique de la donnée de l'utilisateur et mettant en jeu des bases de données auxiliaires.
Prenons, à titre d'exemple, le cas de « jenero » : le logiciel envisage les omissions d'accents. À un moment donné du traitement, l'hypothèse « jénéro » sera envisagée. Cette hypothèse sera transformée en chaîne phonétique par un module de lecture. La chaîne phonétique ainsi obtenue sera recherchée dans la base de données phonétiques qui va alors mener à la piste « généraux ». La piste sera prolongée à son tour par un traitement morphologique qui va mener à « général ». Toutes ces phases de transformation sont tentées dans tous les ordres possibles, afin qu'aucune piste ne soit omise.

Dans la cas où ce prétraitement débouche sur plusieurs solutions (ex. « cuisso » donne les solutions « cuisseau » ou « cuissot »), un tableau synoptique des différents résultats trouvés est présenté à l'utilisateur qui pourra y faire son choix.
L'ensemble de ces mesures a conduit à une quasi-disparition des échecs, les seuls cas résiduels provenant de demandes dont les distances orthographique ou phonétique à une éventuelle réponse dépassent des bornes acceptables. Nous ne doutons pas que ce mode d'accès particulièrement tolérant a largement contribué au succès du TLF auprès du grand public.

6. Usage actuel du TLFi

Dans un souci de mise à disposition de la communauté de ce produit et parallèlement à la définition d'une version CDRom qui devrait être disponible courant 2004, le choix a été fait de rendre cette ressource accessible gratuitement sur la toile à l'adresse www.atilf.fr/tlfi depuis le mois de mars 2002. Cette mise à disposition du public a reçu un accueil qui est allé bien au-delà des premiers espoirs des concepteurs de ce dictionnaire informatisé et qui, d'une certaine façon correspond à une réelle validation du TLFi.
Au cours du mois de janvier 2004, nous avons ainsi servi plus de 2 900 000 pages du TLFi et l'évolution des accès indique une progression d'un million de pages servies au cours des douze derniers mois.

7. Remerciements

Nous tenons à remercier l'ensemble des rédacteurs du TLF et des membres de l'ancien INaLF et de l'ATILF d'aujourd'hui sans qui le TLFi n'existerait pas.

8. Références

Bernard, P., Dendien, J., Lecomte, J. et Pierrel, J.M. 2002 « Un ensemble de ressources informatisées et intégrées pour l'étude du français : FRANTEXT, TLFi, dictionnaires de l'Académie et logiciel Stella, présentation et apprentissage de leurs exploitations », *Actes de TALN 2002*, vol 2, p. 3-36, Nancy, 24-27 juin 2002, disponible aussi à l'adresse : http://www.loria.fr/projets/TALN/TALN/index.html.

CNRS 1976-1994 TLF, Dictionnaire de la langue du 19e et 20e siècle, CNRS, Gallimard, Paris.

Dendien, J. 1996 « Le projet d'informatisation du TLF », in *Lexicographie et informatique*, Didier Erudition, Paris, pages 25-34.

Dendien, J. & Pierrel, J.M. 2003 « Le Trésor de la Langue Française Informatisé : un exemple d'informatisation d'un dictionnaire de langue de référence », Revue Traitement Automatique des Langues (TAL) Vol 43 – n°2/2003, p. 11-37, Editions Hermès.