LEXIS
Un logiciel d'indexation et de lexicalisation automatiques pour Macintosh™ (Apple)

Richard GOULET

Il est sans doute présomptueux de vouloir un logiciel de lexicalisation conçu pour les utilisateurs de micro-ordinateurs dans une revue aussi savante qu'*Informatique et Statistique*, surtout quand il s'agit d'un travail d'amateur. Mais puisque le directeur de la revue a eu la magnanimité de m'inviter à présenter mon logiciel, j'essaierai d'en donner une description aussi détaillée que possible. J'avais écrit une première version de cet article il y a environ un an, mais j'en ai constamment retardé la publication, parce que depuis l'été 1990, j'ai entrepris de réécrire entièrement mon programme dans le cadre de la librairie MacApp (Apple). C'est donc la version 2.0 de *Lexis* que je décrirai dans les pages qui suivent.

Lexis se charge de la plus grande partie des tâches liées à la création et à l'exploitation de lexiques exhaustifs de textes grecs et latins : normalisation typographique et morphologique du texte à indexer, filtrage (optionnel) des mots-outils, affectation des références (tome, page, ligne), lemmatisation automatique d'une large proportion du vocabulaire à partir d'un dictionnaire des formes lexicales et de listes de désinences, assistance pour la lemmatisation manuelle du reste du vocabulaire, enregistrement progressif des nouveaux lemmes dans un dictionnaire, génération d'un lexique immédiatement imprimable (format *RTF* : "Rich Text Format") et respectant diverses options fixées par l'utilisateur, création de fichiers de fréquences, de concordances (y compris pour les termes en cooccurrence), recherche de chaînes de caractères, etc. La version 2.0 du logiciel sait également transformer automatiquement les textes du *Thesaurus Linguae Graecae* (Université de Californie à Irvine) en fichiers immédiatement indexables par *Lexis*. On peut donc s'épargner la pénible saisie du texte grec. Des tentatives de récupération de textes, d'abord latins, puis grecs, par reconnaissance optique des caractères, seront bientôt menées pour simplifier encore le travail de l'utilisateur.

Je me suis attaqué à ce projet en octobre 1987, sans avoir jamais suivi la moindre heure de cours en informatique, afin de résoudre au départ un problème personnel : l'indexation des 51 000 mots du *Monogénès* de Macarios de Magnésie, dont je prépare une édition critique et une traduction commentée pour la Collection des Sources Chrétiennes. J'ai cependant travaillé de façon assez irrégulière à ce projet, pendant les temps libres que me laissait la préparation du premier tome du *Dictionnaire des philosophes antiques*, paru aux Editions du C.N.R.S à la fin de 1989.

Lexis respecte intégralement l'interface utilisateur du Macintosh. C'est dire qu'en tant que logiciel, il est relativement facile à utiliser, même si la qualité finale du lexique produit dépend de l'érudition du philologue qui le mettra à contribution. Le recours massif aux procédures de la ToolBox du Macintosh rendrait pour le moment très difficile la transposition du programme dans un univers informatique différent. Une grande partie de mes efforts a été consacrée à surmonter les difficultés soulevées par la programmation de cet ordinateur si particulier qu'est le Macintosh et la mise au point finale de l'interface aurait été compromise sans l'aide obtenue des ingénieurs d'Apple (France) dans le cadre d'un contrat de développement. Je ne m'attarderai pas cependant sur cet aspect du logiciel. Rappelons simplement que la convivialité du programme, qui repose sur des menus déroulants dont on sélectionne les items avec la souris (ou au moyen de raccourcis-clavier rappelés dans les menus), des fenêtres de dialogues ou d'alertes contenant des boutons de divers types, des barres de défilement et des zones d'édition, n'est assurée que par une lourde prise en charge, par le programmeur, de tous les événements susceptibles d'être déclenchés par les choix de l'utilisateur. Heureusement, sur ce point, une simplification considérable du travail de programmation a été obtenue grâce à MacApp, une librairie d'objets fonctionnant aussi bien dans le cadre du *Macintosh Programming Workshop* (MPW) que dans l'environnement de programmation du *Think LightSpeed Pascal*. MacApp est capable de soulager le programmeur des aspects génériques d'une application Macintosh et de lui permettre de se concentrer sur les routines propres à son application. J'aurais sauvé des mois de travail dans la préparation de la version antérieure, s'il ne m'avait fallu reconstruire et parfois redécouvrir par moi-même les moindres éléments de l'interface Macintosh. La venue de MacApp a donc représenté une véritable libération et d'innombrables améliorations par rapport à la version précédente de *Lexis* sont dues à MacApp. Apple incite fortement les programmeurs à travailler dorénavant dans l'environnement MacApp afin d'assurer la compatibilité de leur logiciel avec les versions futures du System. Bien que l'apprentissage soit un peu pénible et que la documentation soit pour l'instant insuffisante, on voit mal, lorsqu'on a déjà mis au point une ou deux applications dans le cadre de MacApp, comment on pourrait

s'en passer.

Bien que la version actuelle de *Lexis* ait été compilée sous LightSpeed Pascal (version 3.02), l'essentiel du programme avait été écrit préalablement en Turbo Pascal. Là encore, ce choix a représenté bien des efforts inutiles et la découverte en cours de route du *LSP* a été une véritable libération, ne serait-ce que par la possibilité par lui offerte de suivre l'évolution des variables du programme dans le cadre du déboggeur symbolique *Lightsbug*. Dans sont état présent, le logiciel représente environ 411 K, dont environ 112 K sont occupés par les diverses ressources (autres que le code) : modèles de fenêtres et de contrôles, menus, dialogues, chaînes de caractères pour l'affichage des messages, icônes, dessins, etc. La nette distinction entre le code du programme et les ressources auxquelles il fait constamment appel est une des particularités du Macintosh et a permis de préparer facilement une version américaine du logiciel sans modifier une seule ligne de code.

Le logiciel fonctionne en principe sur toute la gamme Macintosh, du Mac Plus au Mac II CI. L'utilisation d'un disque dur reste fort recommandée pour traiter un dossier important et les procédures de tris peuvent être considérablement accélérées sur un matériel de haut de gamme. D'autre part, *Lexis* essaie dans la mesure du possible d'effectuer toutes les procédures importantes en mémoire interne et ne travaille sur disque que lorsqu'il ne peut faire autrement : l'accès à une mémoire vive de 2, 4 ou 8 Mo est donc un facteur de réduction des temps de traitement assurés par le logiciel. Signalons enfin que *Lexis* est parfaitement compatible avec *MultiFinder* et avec le nouveau Système 7.

Avant de décrire plus en détail le logiciel, je dois signaler que pour le grec, *Lexis* utilise une police de caractères conçue spécialement pour le travail de lexicalisation. En préparant la police Corinthe (une police de format PostScript), j'ai renoncé à la correspondance naïve entre caractère grec et caractère romain équivalent que maintiennent habituellement les polices grecques à l'intérieur du code ASCII. Le défaut de cette correspondance est de classer le *gamma* comme septième lettre de l'alphabet ! Corinthe range au contraire les différents signes grecs dans l'ordre alphabétique du grec afin d'accélérer considérablement les tris et les recherches. Il appartient à une configuration spéciale du clavier de gérer la correspondance entre les touches familières du clavier matériel et le code interne de la police. Le passage du clavier romain au clavier grec se fait instantanément sans quitter le clavier au moyen d'une touche de fonction (FKEY), ou mieux au moyen d'une macro-instruction qui modifie d'un seul coup la police de caractères et le clavier. Corinthe peut d'ailleurs être utilisée à partir de configurations personnalisées du clavier (j'en ai défini une qui reprend la disposition familière aux utilisateurs de Sparta par exemple) et une "cdev" intitulée *MultiClaviers*, fonctionnant dans le cadre

du Tableau de bord, permet de sélectionner le clavier spécial que la touche de fonction substituera automatiquement au clavier standard. Corinthe sera bientôt remplacée par Éleusis, une police PostScript du type 1 d'Adobe qui, tout en reprenant le même codage alphabétique, se rapprochera encore plus du graphisme de la police Times.

L'utilisateur n'est cependant pas contraint de travailler avec cette police, car *Lexis* sait importer des textes saisis avec d'autres polices grecques (comme Sparta), exporter ses fichiers dans une police différente et laisse à l'utilisateur le choix du clavier grec (on peut ainsi utiliser la police Corinthe avec le clavier propre à Sparta...).

Lexis traite aussi bien les textes grecs que les textes latins. Il suffirait de supprimer quelques règles de normalisation morphologique pour lui permettre de travailler également sur un texte anglais ou français, mais il est possible que la structuration des données et les procédures conçues pour des langues à flexion ne soient pas les plus appropriées pour traiter ces langues modernes. En revanche, certains collègues arabisants souhaiteraient que l'on ajoute l'arabe aux langues pratiquées par ce logiciel. Il est certain que le Script Manager du Macintosh laisse la porte ouverte à une entreprise de ce type. Si les capacités de *Lexis* à traiter un texte grec ont été longuement éprouvées par moi-même et par plusieurs collègues, au point que nous disposons maintenant d'un dictionnaire de plus de 25 000 formes lexicales grecques, ses performances devant un texte latin n'ont pas été examinées d'aussi près et nous ne disposons pas pour le latin d'un dictionnaire aussi important. Mais le latin pose beaucoup moins de problèmes que le grec et je parlerai dans le reste de cet article essentiellement du travail sur des lexiques grecs.

Pour délimiter de façon encore plus commode notre sujet, éliminons un autre aspect du logiciel où *Lexis* n'a guère de prétentions : celui du traitement de texte. Il m'a semblé inutile de tenter de rivaliser avec les excellents traitements de texte qui ont largement contribué à la renommée de MacIntosh. *Lexis* comporte un traitement de texte interne, mais celui-ci repose essentiellement sur les routines d'édition de texte intégrées à la ToolBox du MacIntosh et il en supporte toutes les limites. Il contraint donc l'utilisateur à travailler sur des sections limitées à 32 K des fichiers-texte qu'il utilise, mais la gêne est limitée par la possibilité offerte de passer immédiatement à n'importe quelle section du fichier. J'ai renoncé également à intégrer à ce traitement de texte les variations de polices de caractères, de tailles et de styles qu'autorise la dernière version du manager TextEdit. Outre le fait que l'enregistrement comme fichiers des textes ainsi enrichis aurait été beaucoup plus difficile à assurer, le travail d'indexation aurait exigé la suppression de toutes ces données adventices. Ces limites une fois reconnues, le traitement de texte interne reste pratique à utiliser et dispense de quitter le logiciel pour aller lire les fichiers-texte. Il comporte toutes les fonctions d'édition classiques du Macintosh (Copier,

Couper, Coller) et assure même toutes les procédures d'annulation habituelles (Annuler Couper, Rétablir Couper, etc.). On peut également imprimer le document.

Lexis fait appel à des fichiers de divers types pour créer le lexique complet du texte à indexer. Pour nous en tenir aux fichiers indispensables, il faut énumérer (1) un fichier-texte contenant les lignes du texte à indexer, (2) un fichier principal contenant d'une part ces lignes sous formes d'une liste d'enregistrements comprenant une version normalisée des lignes du texte et leur référence, et d'autre part une liste des formes et des lemmes qui leur correspondent, (3) un fichier-texte contenant le lexique finalement généré selon le format choisi par l'utilisateur. Il faut ajouter (4) un dictionnaire qui assure la lemmatisation automatique et enregistre les résultats de la lemmatisation manuelle.

Le fichier principal fait place sous forme de "ressource" à un *memo* où sont enregistrés divers renseignements concernant les étapes du travail de lexicalisation : nombre de lignes, de formes lexicales, de mots éliminés, date des principales opérations, choix de l'utilisateur concernant le type de lexique à générer, etc.). La structure où sont enregistrés lemmes et formes lexicales peut être définie en Pascal de la façon suivante :

```
Type
    donnees_lexicographiques = record
        lemme : string[17] ;
        forme : string[17] ;
        renvoi_texte : integer ; {pointeur vers une ligne du fichier principal}
        observations : byte ; {qualificateur d'occurrence}
        end ;
```

On aura remarqué qu'un maximum de 32 767 lignes est prévu. La structure enregistrant les lignes normalisées du texte peut de son côté être décrite comme suit :

```
Type
    donnees_texte : record
        ref_tome : integer ;
        ref_page : integer ;
        ref_ligne : integer ;
        ligne_texte : string[128] ;
        end ;
```

Pour simplifier les procédures de tri, le Dictionnaire reproduit la structure définie pour les lemmes et les formes, mais le champ renvoi_texte est pour l'instant inutilisé et le champ réservé pour les qualificateurs d'occurrence est utilisé comme indicateur d'homonymie.

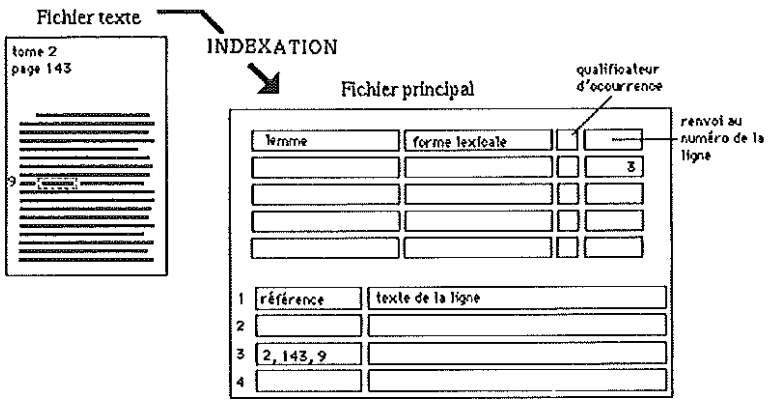

Fig. 1 – La structure du fichier principal.

Lexis opère sur un fichier-texte contenant les lignes d'une édition de référence (séparées par un simple retour de ligne) et une référence sur trois niveaux (tome, page, ligne par exemple), au moins au début des lignes qui rompent la succession numérique de la linéation (nouvelle page, nouveau tome). L'incrémentation du numéro des lignes est faite automatiquement dans les autres cas. Des symboles spéciaux permettent de qualifier une section du texte (titre, citation, conjecture d'un éditeur) ou d'introduire des commentaires qui ne seront pas indexés. Ce fichier-texte peut être saisi directement au clavier, emprunté à un corpus informatisé comme le *Thesaurus Linguae Graecae* (grâce à un transcodeur intégré à *Lexis* et moyennant éventuellement une autorisation d'exploitation) ou (dans un avenir que l'on espère prochain) récupéré au moyen d'un scanner associé à un logiciel de reconnaissance des caractères. Voici un exemple de texte, tiré d'Eunape de Sardes, prêt pour l'indexation:

3␣5␣18␣Πλωτῖνος ἦν ἐξ Αἰγύπτου φιλόσοφος. τὸ
ἐξ Αἰγύπτου νῦν γράφων, καὶ τὴν πατρίδα προσθήσω.
Λυκὼ ταύτην ὀνομάζουσιν· καίτοι γε ὁ θεσπέσιος φιλόσοφος
Πορφύριος τοῦτο οὐκ ἀνέγραψε, μαθητής τε αὐτοῦ γεγενῆσθαι
λέγων, καὶ συνεσχολακέναι τὸν βίον ἅπαντα ἢ τὸν
πλεῖστον. ᵀ9τούτου Πλωτίνου θερμοὶ βωμοὶ νῦν, ᵀ0καὶ τὰ
␣6␣1␣βιβλία οὐ μόνον τοῖς πεπαιδευμένοις διὰ χειρὸς ὑπὲρ τοὺς
Πλατωνικοὺς λόγους, ἀλλὰ καὶ τὸ πολὺ πλῆθος, ἐάν τι παρακούσῃ
δογμάτων, ἐς αὐτὰ κάμπτεται. τὸν βίον αὐτοῦ πάντα
Πορφύριος ἐξήνεγκεν, ὡς οὐδένα οἷόν τε ἦν πλέον εἰσφέρειν·
ἀλλὰ καὶ πολλὰ τῶν βιβλίων ἑρμηνεύσας αὐτοῦ φαίνεται.

L'indexation proprement dite consiste à lire le fichier-texte original et à constituer à partir de là un fichier des formes lexicales, initialement dépourvu des lemmes correspondants. La procédure d'indexation transforme les lignes du texte initial (séparées par un retour de ligne) en chaînes d'un maximum de 128 caractères auxquelles est associée une référence. Une normalisation typographique est également effectuée : suppression des doubles-blancs, des commentaires introduits par l'utilisateur, dégagement des formes élidées par l'insertion d'un espace, élimination des accents d'enclise, remplacement des marques de tabulation par des espaces blancs, etc. La ligne est ensuite analysée. Pour chaque forme lexicale repérée, la procédure d'indexation élimine la ponctuation et les chiffres, transforme les barytons en oxytons, uniformise les formes doubles (*u* et *v* en latin, β et 6 en grec), enregistre un qualificateur d'occurrence si le mot fait partie d'une section ainsi qualifiée et incrémente le nombre de mots retenus ou éliminés. Si l'utilisateur souhaite un lexique sélectif (éliminant par exemple les diverses formes de l'article, καί, μέν et δέ), la forme lexicale est recherchée dans la liste des formes à éliminer et n'est enregistrée que si elle n'y figure pas. Cette liste peut être définie de deux manières : ou bien comme ressource intégrée au logiciel et modifiable au gré de l'utilisateur, ou bien comme fichier-texte externe. L'ensemble de ce travail apparemment très compliqué s'effectue assez rapidement. Sur un SE 30, l'indexation des 6136 lignes du *Monogénès* (51 000 mots environ) a demandé moins de 10 minutes. Comme pour l'ensemble des opérations qui seront décrites ci-dessous, la procédure peut être annulée à tout moment, la progression du travail s'affiche au fur et à mesure dans une fenêtre de dialogue sous forme de "thermomètre" et un message final, parfois accompagné d'un accord musical, signale l'achèvement du processus, tout en fournissant quelques statistiques utiles.

Avant de procéder à la lemmatisation du fichier des formes lexicales enregistrées, il est conseillé de le classer alphabétiquement. Le tri des formes grecques demande une méthodologie très complexe. Les procédures ont été généralisées de manière à pouvoir servir au classement du fichier des formes en prenant comme clé de tri les formes et/ou les lemmes, mais aussi au classement des dictionnaires.

Mais, dans le cas du grec, il n'est pas question de classer directement le fichier des formes. Il faut en effet éviter que les accents viennent séparer des mots dont les lettres comme telles sont identiques. Un classement primaire donnerait par exemple la suite des formes suivantes : αὐταί, αὐταῖς, αὐτά, αὖται, etc., parce que tous les α passent devant les ά, tous les ὐ devant les ὖ. Il faut donc un procédé de tri assez complexe à deux niveaux : ne prendre en considération les accents que lorsque les mots ont été classés indépendamment de leur accentuation. Dans sa version la plus compliquée, le tri s'effectue donc sur cinq clés : lemme désaccentué, lemme accentué,

forme désaccentuée, forme accentuée, numéro de la ligne. Le premier classement des 39 000 formes de Macarios demande un peu plus d'une minute sur un SE 30. Cette rapidité a été obtenue par la réécriture en langage assembleur de l'algorithme élaboré.

L'étape suivante est un essai de lemmatisation automatique du fichier des formes. La démarche de *Lexis* est double. Il recherche (par recherche dichotomique) la forme enregistrée dans le dictionnaire et, s'il la retrouve, copie dans le fichier des formes le lemme connu par le dictionnaire pour cette forme. Si la forme est connue comme pouvant se rapporter à plusieurs lemmes (grâce à un indicateur d'homonymie dont nous reparlerons), la recherche échoue et une † est enregistrée comme lemme. Si la forme est inconnue, une seconde tentative de lemmatisation est lancée qui fait appel à une table de désinences remarquables (logée parmi les ressources du logiciel). Les différentes désinences répertoriées sont comparées à la fin de la forme à lemmatiser. Si la désinence est la même, elle est remplacée dans la forme à indexer par la finale (nominatif ou infinitif) correspondante dans la table des désinences. Ce lemme est suffixé d'un astérisque qu'il appartiendra à l'utilisateur de supprimer s'il croit devoir confirmer la lemmatisation. La liste des désinences peut être modifiée par l'utilisateur. Il lui appartient en effet de choisir s'il préfère une liste très sûre, mais limitée, ou une liste plus large faisant place éventuellement à des erreurs faciles à corriger. Si la forme suivante est identique à la précédente, le lemme déjà trouvé est simplement reporté (d'où l'intérêt d'un classement alphabétique préalable des formes), sinon une nouvelle recherche est lancée.

La démarche de *Lexis* est donc fort empirique. Les premiers utilisateurs ne pouvaient compter que sur un dictionnaire de quelques centaines de mots et devaient assurer eux-mêmes la lemmatisation de la plus grande partie du fichier. Mais, comme nous le verrons, ce dictionnaire s'enrichit automatiquement des résultats de la lemmatisation manuelle et les fichiers de plusieurs utilisateurs peuvent être fusionnés en un seul. Dès la fin d'un premier travail en grandeur réelle, nous avions en main un fichier de près de 13 000 formes greques. Celui que je propose aux futurs utilisateurs dépasse les 25 000 formes et devrait pouvoir lemmatiser de 75 à 80 % d'un texte de prose.

La suite du travail requiert un peu plus de collaboration de la part de l'utilisateur, car il lui faudra parcourir le fichier des formes, vérifier la valeur des lemmes proposés par la lemmatisation automatique et trouver lui-même les lemmes sous lesquels ranger les formes non lemmatisées. Cette lemmatisation manuelle s'effectue dans une fenêtre de travail dont la hauteur et le nombre de lignes s'adaptent automatiquement au type d'écran utilisé. Les structures offertes par MacApp permettent de traiter simultanément jusqu'à 32 000 enregistrements; un dialogue permet de sélectionner

facilement la section à traiter si le fichier est de taille supérieure. Chaque ligne range dans différentes cellules le lemme, la forme et le qualificateur d'occurrence. La fenêtre indique également la référence correspondant au couple lemme-forme sélectionné et fournit le contexte, sur plusieurs lignes, de l'occurrence présentement sélectionnée. Une zone d'édition distincte du tableau correspond au contenu de la cellule sélectionnée, comme dans les tableurs du type *Excel*. Les fonctions d'édition standard du Macintosh sont évidemment accessibles (Annuler, Copier, Couper, Coller, Effacer), mais il est également possible de Récupérer comme lemme la forme lorsqu'elle correspond au moins partiellement au lemme que l'on veut introduire, de Copier vers le bas le contenu d'une cellule (lorsque par exemple le lemme vaut pour toute une série de formes successives). Le passage d'une cellule à une autre (par les flèches de défilement, la souris, la touche Retour de ligne ou la touche Enter) provoque la validation immédiate du contenu de la cellule. Tous les éléments de ce tableau sont éditables, y compris les références et le contenu des différentes lignes du texte indexé.

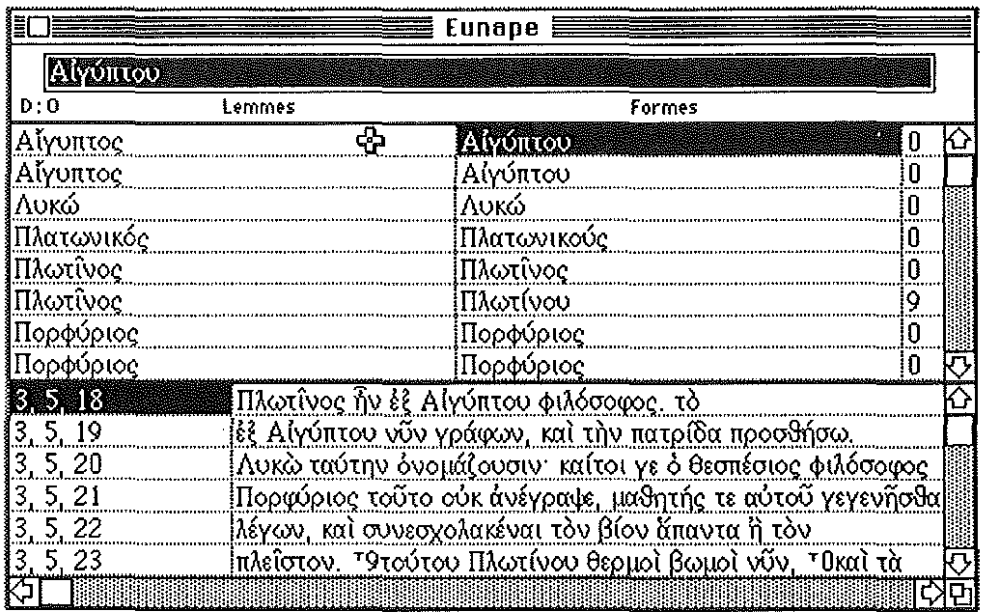

Fig. 2 – La fenêtre de travail où s'effectue la lemmatisation manuelle.

Trois autres fonctions, accessibles elles aussi par les menus ou par des combinaisons de touches au clavier, établissent le contact entre la fenêtre de travail et les autres fichiers. On peut tout d'abord Insérer dans le dictionnaire un ou plusieurs enregistrements lemme-forme. Ces couples sont mis en réserve dans une liste que l'on peut afficher et dont on peut supprimer des éléments, puis, lors de la sauvegarde du fichier (ou lorsque le tableau est plein), ils sont classés alphabétiquement et insérés à leur place dans le dictionnaire. Si le dictionnaire connaît déjà cette forme et ce lemme, il n'y a pas d'enregistrement. Si la forme était déjà connue, mais rattachée à un lemme différent, un indicateur d'homonymie passe de 0 à 1 : le lemme ne sera plus proposé lors de la lemmatisation automatique pour éviter des erreurs liées à l'homonymie et une † signalera cette possible homonymie.

En cours de travail, l'utilisateur peut également décider d'ajouter telle ou telle forme au fichier des formes à éliminer. Les mots ainsi sélectionnés seront introduits automatiquement dans le fichier, en respectant l'ordre alphabétique.

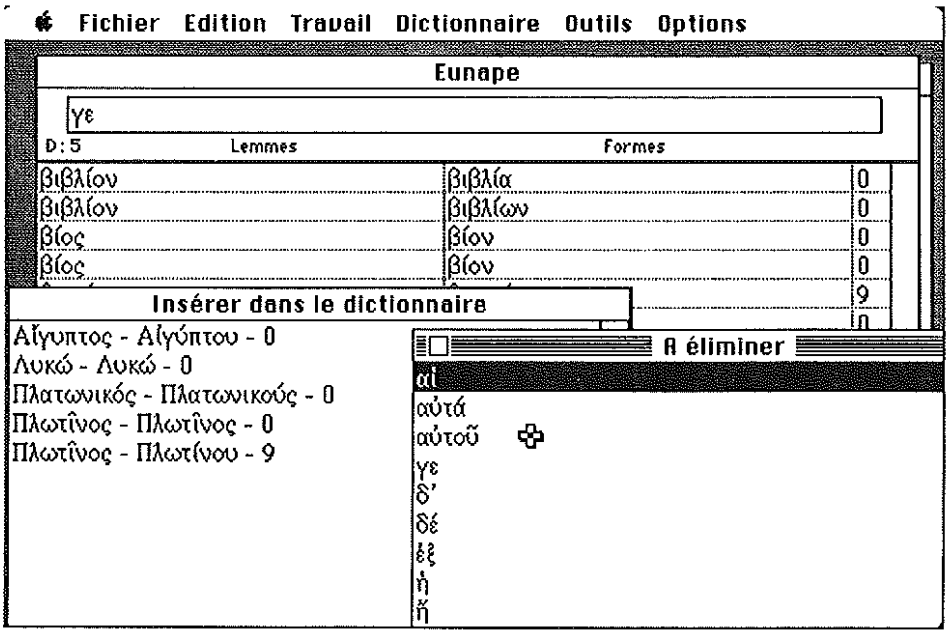

Fig. 3 – La fenêtre de travail et les listes temporaires d'insertion.

Tout en travaillant dans cette fenêtre d'édition, il est possible d'ouvrir également le dictionnaire courant et de le corriger, tout comme il est possible d'ouvrir tous les documents-textes que l'on souhaite traiter.

Le classement alphabétique final du fichier des lemmes et des formes peut prendre deux formes, en fonction du choix de l'utilisateur. Si ce dernier désire obtenir un lexique des lemmes et des formes sous lesquelles les lemmes se rencontrent concrètement dans le texte, le tri s'effectuera selon l'ordre : lemmes, formes, références. En revanche, si l'utilisateur ne souhaite qu'un lexique des lemmes, sans que les formes soient détaillées, alors le tri classera les diverses occurrences d'un même lemme selon l'ordre de leur apparition dans l'ensemble du texte.

La génération finale du lexique laisse à nouveau une large place aux choix de l'utilisateur. Ce dernier peut déterminer lui-même : le titre qui sera donné au lexique, le format du lexique (tableau ou enchaînement en continu des références sur deux colonnes à la page), le séparateur de références (point-virgule et/ou marque de tabulation), le type de lexique (lemmes et formes lexicales ou lemmes seulement), le format du fichier généré (texte seulement ou format *RTF*). Il lui appartient également de décider s'il désire que soit ajouté un indice de fréquence après le nombre d'occurrences pour chaque lemme, ainsi qu'une liste de tous les mots composés où figure également le lemme. Enfin, *Lexis* peut se charger d'élucider la signification des symboles retenus comme qualificateurs d'occurrence en précisant "Homère", "A.T.", "conjecture", etc., à condition que la correspondance soit définie dans un fichier-texte accessible. Au terme de ce processus, on retrouve par exemple sur le bureau électronique un fichier-texte de format *RTF* qu'un logiciel comme Word pourra convertir immédiatement en un document standard parfaitement mis en forme, facile à intégrer à la fin d'une édition et immédiatement imprimable.

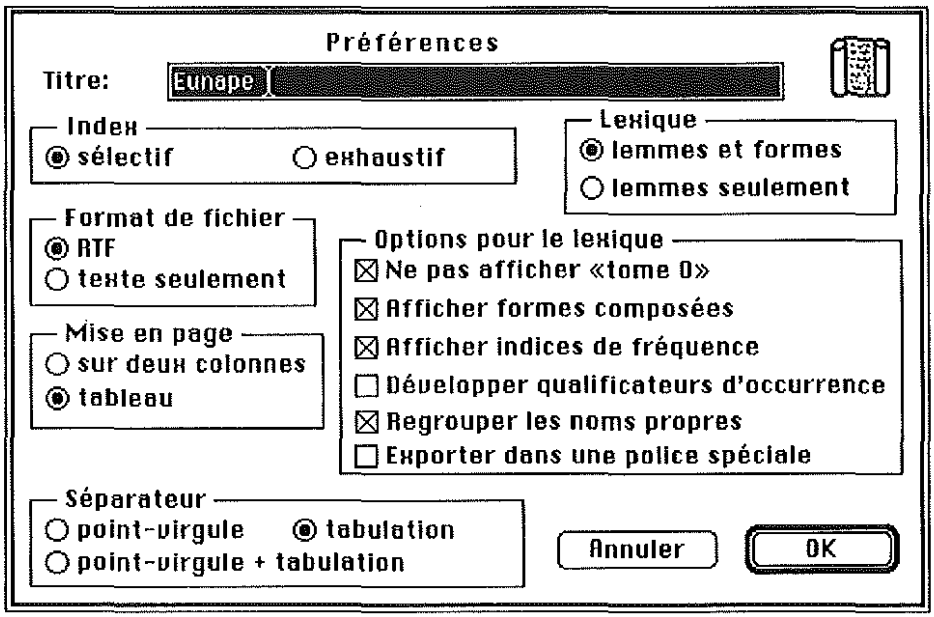

Fig. 4 – Le dialogue enregistrant les préférences de l'utilisateur.

Exemples de lexiques créées au format *RTF* :

EUNAPE, *VIE DE PLOTIN*

Αἴγυπτος				
Αἰγύπτου	3,5,18;	3,5,19	[2 :0.0222].	
Λυκώ	3,5,20	[1 :0.0111].		
Πλατωνικός				
Πλατωνικούς		3,6,2	[1 :0.0111].	
Πλωτῖνος	3,5,18			
Πλωτίνου	3,5,23 (Citation)	[2 :0.0222].		
Πορφύριος		3,5,21;	3,6,4	[2 :0.0222].
ἀλλά	3,6,2	3,6,5	[2 :0.0222].	
ἀναγράφειν				
ἀνέγραψε	3,5,21	[1 :0.0111].		
ἅπας				
ἅπαντα	3,5,22	[1 :0.0111].		
αὐτός				
αὐτά	3,6,3			
αὐτοῦ	3,5,21;	3,6,3;	3,6,5	[4 :0.0444].

βιβλίον				
βιβλία	3,6,1			
βιβλίων	3,6,5	[2 :0.0222].		
βίος				
βίον	3,5,22	3,6,3	[2 :0.0222].	
βωμός				
βωμοί	3,5,23 (Citation)	[1 :0.0111].		
γε	3,5,20	[1 :0.0111].		
γίγνεσθαι				
γεγενῆσθαι	3,5,21	[1 :0.0111].		
γράφειν				
γράφων	3,5,19	[1 :0.0111]	→ ἀναγράφειν.	
διά	3,6,1	[1 :0.0111].		
δόγμα				
δογμάτων	3,6,3	[1 :0.0111].		
ἐάν	3,6,2	[1 :0.0111].		
εἶναι				
ἦν	3,5,18;	3,6,4	[2 :0.0222].	
εἰσφέρειν	3,6,4	[1 :0.0111].		
ἐκφέρειν				
ἐξήνεγκεν	3,6,4	[1 :0.0111].		
ἐξ	3,5,18;	3,5,19	[2.0.0222].	
ἑρμηνεύειν				
ἑρμηνεύσας		3,6,5	[1 :0.0111].	
ἑς	3,6,3	[1 :0.0111].		
ἤ				
ἤ	3,5,22	[1 :0.0111].		
θερμός				
θερμοί	3,5,23 (Citation)	[1 :0.0111].		
θεσπέσιος	3,5,20	[1 :0.0111].		
καί	3,5,19;	3,5,22;	3,5,23;	3,6,2;
	3,6,5	[5 :0.0556].		
καίτοι	3,5,20	[1 :0.0111].		
κάμπτειν				
κάμπτεται	3,6,3	[1 :0.0111].		
λέγειν				
λέγων	3,5,22	[1 :0.0111].		
λόγος				
λόγους	3,6,2	[1 :0.0111].		
μαθητής	3,5,21	[1 :0.0111].		
μόνον	3,6,1	[1 :0.0111].		
νῦν	3,5,19;	3,5,23 (Citation)	[2 :0.0222].	
ὁ				
ὁ	3,5,20			
τά	3,5,23			
τήν	3,5,19			
τό	3,5,18;	3,6,2		
τοῖς	3,6,1			
τόν	3,5,22;	3,5,22;	3,6,3	
τούς	3,6,1			

(...)

EUNAPE, *VIE DE PLOTIN*

Αἴγυπτος 3,5,18;3,5,19 [2 :0.0222].
Λυκώ 3,5,20 [1 :0.0111].
Πλατωνικός 3,6,2 [1 :0.0111].
Πλωτῖνος 3,5,18;3,5,23 (Citation) [2 :0.0222].
Πορφύριος 3,5,21;3,6,4 [2 :0.0222].
ἀλλά 3,6,2;3,6,5 [2 :0.0222].
ἀναγράφειν 3,5,21 [1 :0.0111].
ἅπας 3,5,22 [1 :0.0111].
αὐτός 3,5,21;3,6,3;3,6,3; 3,6,5 [4 :0.0444].
βιβλίον 3,6,1;3,6,5 [2 :0;0222].
βίος 3,5,22;3,6,3 [2 :0.0222].
βωμός 3,5,23 (Citation) [1 :0.0111].
γε 3,5,20 [1 :0.0111].
γίγνεσθαι 3,5,21 [1 :0.0111].
γράφειν 3,5,19 [1 :0.0111]. → ἀναγράφειν.
διά 3,6,1 [1 :0.0111].
δόγμα 3,6,3 [1 :0.0111].
ἐάν 3,6,2 [1 :0.0111].
εἶναι 3,5,18;3,6,4 [2 :0.0222].
εἰσφέρειν 3,6,4 [1 :0.0111].
ἐκφέρειν 3,6,4 [1 :0.0111].
ἐξ 3,5,18;3,5,19 [2 :0.0222].
ἑρμηνεύειν 3,6,5 [1 :0.0111].
ἐς 3,6,3 [1 :0.0111].
ἤ 3,5,22 [1 :0.0111].
θερμός 3,5,23 (Citation) [1 :0.0111].
θεσπέσιος 3,5,20 [1 :0.0111].
καί 3,5,19;3,5,22; 3,5,23;3,6,2;3,6,5 [5 :0.0556].
καίτοι 3,5,20 [1 :0.0111].
κάμπτειν 3,6,3 [1 :0.0111].
λέγειν 3,5,22 [1 :0.0111].
λόγος 3,6,2 [1 :0.0111].
μαθητής 3,5,21 [1 :0.0111].
μόνον 3,6,1 [1 :0.0111].
νῦν 3,5,19;3,5,23 (Citation)[2 :0.0222].
ὁ 3,5,18;3,5,19; 3,5,20;3,5,22;3,5,22;3,5,23;3,6,1; 3,6,1;3,6,2;3,6,3;3,6,5 [11 :0.1222].
οἷος 3,6,4 [1 :0.0111].
ὀνομάζειν 3,5,20 [1 :0.0111].
οὐ 3,6,1 [1 :0.0111].
οὐδείς 3,6,4 [1 :0.0111].
οὐκ 3,5,21 [1 :0.0111].
οὗτος 3,5,20;3,5,21; 3,5,23 (Citation)[3 :0.0333].
παιδεύειν 3,6,1 [1 :0.0111].
παρακούειν 3,6,2 [1 :0.0111].
πᾶς 3,6,3 [1 :0.0111].
πατρίς 3,5,19 [1 :0.0111].
πλῆθος 3,6,2 [1 :0.0111].
πολύς 3,5,23;3,6,2;3,6,4; 3,6,5 [4 :0.0444].
προστίθεσθαι 3,5,19 [1 :0.0111].
συσχολάζειν 3,5,22 [1 :0.0111].
τε 3,5,21;3,6,4 [2 :0.0222].
τίς 3,6,2 [1 :0.0111].
ὑπέρ 3,6,1 [1 :0.0111].
φαίνεσθαι 3,6,5 [1 :0.0111].
φιλόσοφος 3,5,18;3,5,20 [2 :0.0222].
χείρ 3,6,1 [1 :0.0111].
ὡς 3,6,4[1 :0.0111].

Avant de quitter le logiciel, il importe d'en signaler quelques fonctionnalités complémentaires. On peut tout d'abord établir la liste de tous les termes employés dans le texte, soit dans l'ordre alphabétique, soit selon leur ordre de fréquence. Le nombre d'occurrences est complété par l'indice de fréquence (par exemple νόμος : 0,0023). Cette liste peut être ensuite enregistrée sous forme d'un fichier-texte.

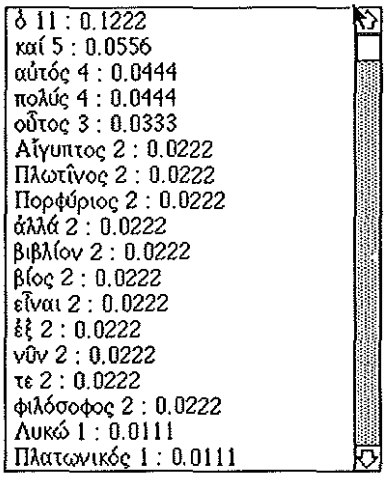

Fig. 5 – Liste de fréquences.

Des recherches de chaînes de caractères peuvent être effectuées dans le fichier des formes ou dans le dictionnaire. Toutes les formes où se retrouve la chaîne proposée sont alors présentées sous forme de liste. Ces recherches peuvent comporter des "filtres" spécifiant comme critères de recherche pour certains caractères des intervalles, des énumérations, des types (lettre, chiffre, signe quelconque), des caractères facultatifs, etc. A nouveau un algorithme écrit en langage assembleur assure des temps de traitement fort intéressants : la recherche et l'affichage de toutes les occurrences d'απο dans un fichier de 39 000 formes demande de 3 à 4 secondes sur un SE 30. La sélection d'un des mots retrouvés dans la liste provoque l'affichage de la première forme correspondante dans le fichier et de la ligne de texte où cette forme apparaît.

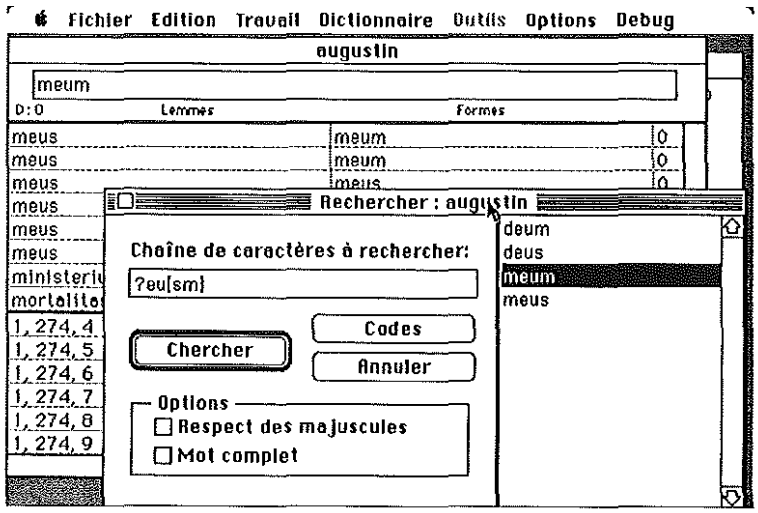

Fig. 6 – Exemple de recherche de chaîne avec filtre.

Enfin, *Lexis* sait créer une véritable concordance affichant toutes les occurrences d'un lemme (ou de deux lemmes en cooccurrence) dans leur contexte sur trois lignes. Si moins de 80 passages sont ainsi repérés, ils peuvent être alors immédiatement consultés à l'écran dans le cadre du traitement de texte.

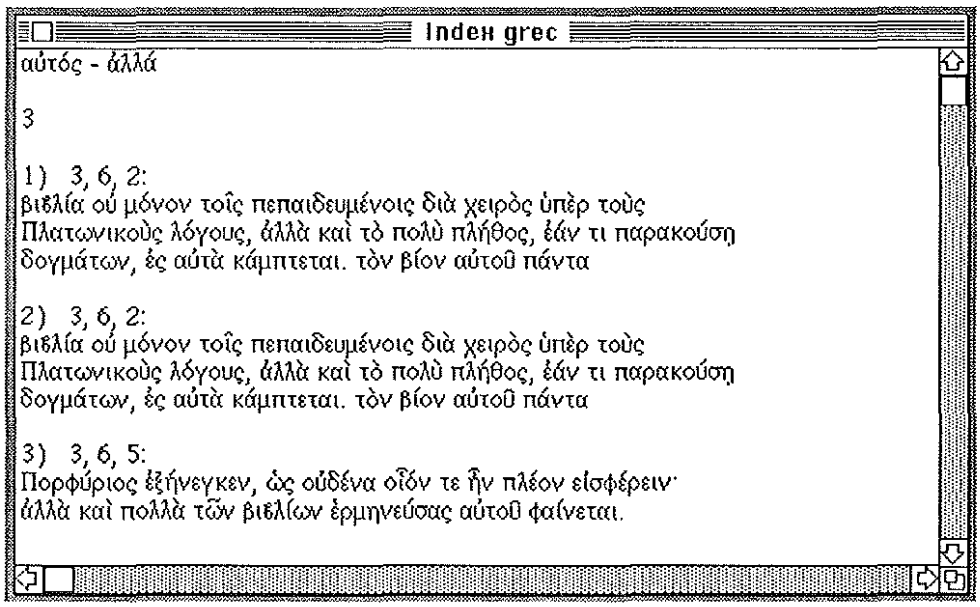

Fig. 7 – Exemple de concordance.

Je ne m'attarderai pas sur d'autres fonctions secondaires du logiciel. On peut par exemple fusionner deux dictionnaires tout en filtrant les doublets et en repérant les formes homonymes. On peut de même fusionner deux dossiers (fichiers des lemmes/formes), ce qui permet de travailler progressivement sur les différents livres ou les différents chapitres d'un texte et de rassembler tout le travail au dernier moment. Pour faciliter l'archivage, il existe une procédure de compactage des fichiers (fichiers des formes et dictionnaires), à laquelle correspond évidemment une procédure de décompactage. Les fichiers ainsi compactés sont réduits jusqu'au tiers de leur dimension normale. On peut également détruire un fichier en restant dans le logiciel, modifier le contenu du dictionnaire, le classer alphabétiquement, "éditer" le contenu des listes contenant les mots à éliminer et les désinences remarquables, etc.

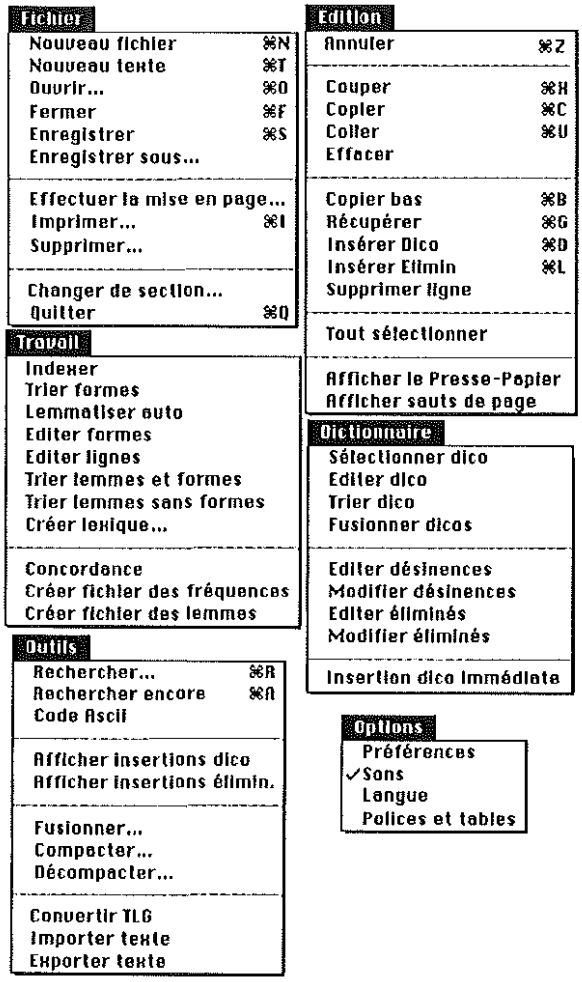

Fig. 8 – Les menus de Lexis.

Un manuel de 150 pages, comprenant une section Apprentissage, une description précise de toutes les étapes du travail d'indexation et une section Références (fichiers, menus, etc.), est fourni avec le logiciel. D'autre part, une aide sous forme de "bulles" est constamment disponible à l'écran (uniquement sous Système 7 pour l'instant) en cours de travail.