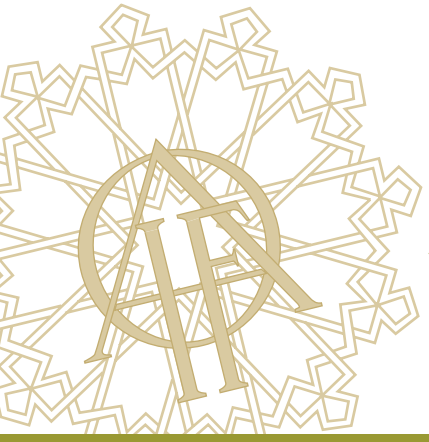
MINISTÈRE DE L'ÉDUCATION NATIONALE, DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE



ANNALES ISLAMOLOGIQUES

en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne en ligne

AnIsl 44 (2010), p. 53-60

Christian Gaubert

Kawâkib, une application Web pour le traitement automatique de textes arabes.

Conditions d'utilisation

L'utilisation du contenu de ce site est limitée à un usage personnel et non commercial. Toute autre utilisation du site et de son contenu est soumise à une autorisation préalable de l'éditeur (contact AT ifao.egnet.net). Le copyright est conservé par l'éditeur (Ifao).

Conditions of Use

You may use content in this website only for your personal, noncommercial use. Any further use of this website and its content is forbidden, unless you have obtained prior permission from the publisher (contact AT ifao.egnet.net). The copyright is retained by the publisher (Ifao).

Dernières publications

9782724711264	*La chapelle d'Osiris Ounnefer Neb-djefaou*	Laurent Coulon
9782724711752	*Guide du Caire Islamique*	Hadrien Collet
9782724711660	*Mélanges de l'Institut dominicain d'études orientales 41*	Adrien Candiard (éd.)
9782724711745	*Bulletin archéologique des Écoles françaises à l'étranger (BAEFE)*	
9782724711271	*Du côté de chez Zeus*	Emmanuel Botte (éd.), Hélène Cuvigny (éd.), Michel Reddé (éd.)
9782724711677	*La Description de l'Égypte*	Paul-Marie Grinevald
9782724711714	*La pensée et la pratique pharmacologiques d'Avicenne*	Sylvie Ayari
9782724711899	*BCAI 40*	

© Institut français d'archéologie orientale - Le Caire

CHRISTIAN GAUBERT

Kawâkib, une application Web pour le traitement automatique de textes arabes

Nous avons réalisé, dans le cadre des travaux de notre équipe de recherche, le logiciel expérimental *Sarfiyya*, qui permet la mise au point de grammaires interactives dans le domaine du traitement automatique de l'arabe (Tala), avec la particularité de recourir le moins possible à un lexique. Afin de diffuser davantage le fruit de ces recherches, nous souhaitions développer un outil de Tala illustrant les axes de recherche de notre équipe et offrant notamment des services dans les domaines de la récupération d'information (*Information Retrieval*, IR), de la classification ou caractérisation de textes et du filtrage sémantique. Ce logiciel s'adressera à un large public.

Une application Web, pourquoi et comment?

Plutôt que de diffuser un logiciel classique, nous avons choisi de réaliser une application Web, dont le concept a émergé depuis la fin des années 2000. Il permet en effet, au prix d'un effort de conception et de développement, de déployer instantanément un logiciel sans installation particulière et cela quel que soit le poste de l'utilisateur (système ou processeur) : tous les ordinateurs modernes comportent en effet un navigateur Web sachant interpréter, entre autres, le langage Javascript, avec des performances accrues depuis 2009. La principale contrainte est que l'ordinateur client doit être connecté au réseau Internet pour utiliser l'application.

Sarfiyya étant désormais entièrement conçue en langage Java, il est possible de profiter des possibilités nombreuses d'interfaçage de ce langage avec les serveurs Web. L'une d'elle a retenu notre attention, car elle permet un mode interactif soutenu. Il s'agit de la technique récente dite Ajax de combinaison du protocole HTTP, des langages XML et Javascript pour créer des pages qui reposent sur des programmes résidant sur le serveur et qui permet le rafraîchissement

AnIsl 44 (2010), p. 53-60 Christian Gaubert
Kawâkib, une application Web pour le traitement automatique de textes arabes.
© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

partiel d'une page. L'utilisateur a alors l'impression qu'il travaille de manière connectée au serveur, ce qui n'est pas le cas des applications Web classiques fondées sur l'envoi de données par formulaire et renvoi global du résultat. Certaines parties du logiciel peuvent en outre être traitées directement par la machine connectée et non par le serveur, réduisant ainsi le temps de calcul. Le serveur Tomcat [1] d'applications Java et la bibliothèque DWR [2] sont les deux logiciels *open-source* qui sont employés pour publier ce site selon cette technique.

Le site *Kawâkib* est conçu à partir d'éléments applicatifs produits par le logiciel *Sarfiyya* [3]. Les fonctions et les grammaires sont développées puis compilées pour certaines en mode déterministe [4], puis importées dans *Kawâkib* où elles sont utilisées telles quelles, sans possibilité de variation : elle sont alors figées.

Fonctions

Texte et navigation

Le site présente un corpus de textes pré-installés et gérés par une base de données en ligne dont la gestion est accessible par le lien *Corpus/modifier*. Chaque texte comporte un titre et une référence. Il est possible de saisir ou de copier un nouveau texte, qui sera alors encodé en unicode. Le nombre de mots du texte est calculé dynamiquement, et une zone de saisie permet de rechercher instantanément (voir technique Ajax) un numéro d'occurrence, ainsi que la recherche d'une chaîne de caractères particulière. Dans les deux cas, le contexte des occurrences est rappelé et les éventuels *tokens* de ce contexte sont mis en évidence (voir *infra*, *tokens*). Chacune des fonctions opère soit sur le texte entier, soit sur la sélection courante si elle est non vide. Certaines fonctions (citations, expressions régulières) peuvent s'appliquer, en option, sur le corpus entier.

L'utilisateur a en outre les possibilités d'agrandir ou d'effacer le texte, voire de le masquer s'il souhaite interroger le texte « à l'aveugle », par exemple dans un but pédagogique.

Racines

L'application permet, dans toutes ses versions, l'analyse et le tri des racines les plus fréquentes du texte. Cette détection repose sur une liste assez exhaustive de 6 000 racines trilitères issues principalement du dictionnaire *Al-Ṣiḥāḥ* [5] et sur une sous-liste de 1 200 racines trilitères

1. http://tomcat.apache.org
2. http://directwebremoting.org
3. Gaubert, *Stratégie et règles minimales* ; Audebert *et al.*, Medar 2009.
4. La version déterministe des grammaires ou automates permet de répondre très rapidement à la simple question : ce texte, cette phrase ou mot obéit-il à cette grammaire ? La version non-déterministe donnera une réponse beaucoup plus lente mais avec une analyse détaillée. La combinaison des deux permet de faire chuter le temps de calcul global des grammaires.
5. Gaubert, *Stratégies et règles minimales*, chap. II, module *DataSarf*.

AnIsl 44 (2010), p. 53-60 Christian Gaubert
Kawâkib, une application Web pour le traitement automatique de textes arabes.
© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

« fréquentes », pour laquelle nous donnons des traductions purement indicatives (exemple ذكر : *mémoire*). Ces traductions ne recouvrent que très partiellement les sens multiples qui peuvent être portés par une racine. Il est clair que l'on pourra être amené à traiter des textes comportant des racines n'entrant pas dans cette liste de racines fréquentes, cependant son utilité est avérée pour filtrer les bruits dans les analyses morphologiques.

Sans détailler ici la méthode employée pour l'extraction des racines, notons qu'elle repose sur plusieurs étapes : analyse morphologique selon un automate nominal et un automate verbal, filtrage des solutions aberrantes, reconstitution d'une racine potentielle, validation par liste et micro-lexiques. Cette analyse est exploitée par deux fonctions : *la recherche de racines*, et le tri des *racines fréquentes*.

L'utilisateur est aidé, dans la *recherche de racine*, par une interrogation instantanée (voir technique Ajax) des listes de racines. Il peut ainsi savoir, en effaçant par exemple R2, la liste des racines attestées de type ط * ق comme طرق, طبق, etc. Il en ira de même en effaçant R1 ou R3. Si la racine demandée est dans la liste des plus fréquentes, la traduction indicative sera donnée. Ce mode d'interrogation de la liste des racines trouvera son utilité auprès d'utilisateurs désireux, par exemple, de comparer des racines « voisines », d'émettre des hypothèses sur un caractère manquant ou illisible, de rechercher des rimes ou tout jeu de consonances, etc.

Le résultat de la recherche par racine est une liste d'occurrence des mots construits sur cette racine, avec le numéro du mot dans le texte.

La fonction *Racines fréquentes* repose sur l'analyse du texte, puis sur le tri des racines potentielles par fréquence d'apparition. La liste des racines fréquentes peut comporter des bruits (racines aberrantes), mais ils sont généralement aisément repérables car les différentes occurrences de chaque racine sont rappelées, ainsi que les traductions indicatives.

Répétitions

Toutes les séquences de caractères précisément répétées sont repérées par ce programme, puis triées par fréquence décroissante. Aucun traitement linguistique supplémentaire n'est appliqué. Les *tokens* sont toutefois mis en évidence dans les extraits répétés. Noter que cette détection n'est pas spécifique à l'arabe et fonctionne quelle que soit la langue du texte. Le temps de traitement, s'il reste linéaire, peut être long, et nous œuvrons à son amélioration.

Tokens

La détection des *tokens* est effectuée par la méthode décrite dans notre thèse[6]. Elle est fondée sur une liste de plus de 300 mots-outils, conjonctions, adverbes, etc., incluant toutes les informations de concaténations possibles afin de limiter les bruits. Le programme utilisé ici ne tient pas compte du contexte proche des *tokens* pour lever leur éventuelle ambiguïté, car les bonnes performances de la détection suffisent dans un premier temps pour mettre en

6. *Ibid.*, chap. V.

© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

évidence les *tokens* sans chercher à les catégoriser : on ne distinguera par exemple pas ici من *min* préposition de من *man* pronom personnel.

Les éléments proclitiques sont simplement repérés par leur position et non par analyse morphologique, ce qui induit un bruit tolérable qu'il conviendra de juguler dans une prochaine version.

Les *tokens* et mots assimilés repérés sont mis en évidence par un jeu de couleur : rouge pour les *tokens*, orange pour les prépositions (*bi-*, *ka-*, *li-*) et coordonnants proclitiques (*wa-*, *fa-*).

Noter qu'il est possible d'avoir une vision *des seuls tokens du texte* en actionnant le réglage *Phrase vide*, et cela dans tous les résultats comportant une analyse des *tokens*. Vidés de leurs occurrences nominales et verbales, les textes et leurs extraits apparaissent sous un angle nouveau, et se prêtent à une étude centrée uniquement sur le rôle structurant et prédictif des *tokens*.

Suites de tokens

La fonction voisine de *Suites de tokens* liste l'ensemble des suites de deux *tokens* ou plus, afin d'étudier ces appariements. Bien que parfois fortuits, ils sont une source importante d'expressions structurant le discours (أيا كان، ومع ذلك فإن، ...). *Kawâkib* offre la possibilité d'enregistrer ces suites (un lien hypertexte déclenche une insertion dans la base de données) et de les annoter (lien *Modifier*), en vue de les repérer ensuite dans autres textes du corpus (bouton *Expressions*) et d'en établir ainsi le contexte d'emploi. Ces expressions serviront par la suite de matière à la construction d'automates.

Expressions régulières et grammaires

Au-delà du simple repérage d'expressions, il est indispensable d'étudier leur distribution dans les textes, nombre d'entre elles se répondant par exemple dans une même phrase. De très nombreuses structures peuvent être repérées en employant des *expressions régulières* (ou *regex*), qui sont mathématiquement équivalentes aux automates finis. *Kawâkib* utilise le moteur d'expressions régulières que nous avons développé pour *Sarfiyya*, lequel, outre les caractéristiques classiques des expressions régulières, permet de faire appel aux automates déjà développés en morphologie et en syntaxe.

Principaux symboles employés

lettres arabes (ا ب ت ...) : représentent leurs propres lettres
A : une lettre arabe quelconque
p : espace
+ : alternative ; comme ي + ت : soit ت soit ي
e : *epsilon-transition* (rigoureusement un mot vide), comme لا+e : avec ou sans لا
* : un nombre quelconque de fois (dont zéro) : A* = un mot arabe
x : une suite quelconque de mots précédés d'espaces = (espaces A*)*
y : une suite quelconque de mots suivis d'espaces = (A* espaces)*

AnIsl 44 (2010), p. 53-60 Christian Gaubert
Kawâkib, une application Web pour le traitement automatique de textes arabes.
© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

Nous donnons ici quelques exemples d'expressions régulières. Seules les parties centrales sont à expliciter, l'application se charge d'envelopper l'expression souhaitée des termes nécessaires à son application au texte entier.

1. Expressions alternatives

exemples : فإن ذلك إما أنه مشكوك فيه بقوة، أو أن

فإن x إما x أو

2. Repérage de structures argumentatives comportant une négation

exemples : etc. ،وإنما ... ليس بل، الآن حتى يفعل لم إلا، ذلك يفعل لا

ل (ا+ن+م) p (ي+ت+ن+أ) y (بل+إلا+ولكن)

وإنما x ((ت+e) ليس (ت+نا)+ليس (لس

3. Expression *min el-*, annonciateur de subordonnée conjonctive

exemples : من الممكن أن، من غير المتوقع أن

(ف+و+e) من p (p غير+e) ال p* A ان p

Visigram

L'option *Visigram* offre une représentation graphique sous forme d'automate de l'expression régulière. Cet automate, dessiné en une ligne, permet à l'utilisateur averti de vérifier si l'expression composée correspond à ce qu'il souhaite réellement tester.

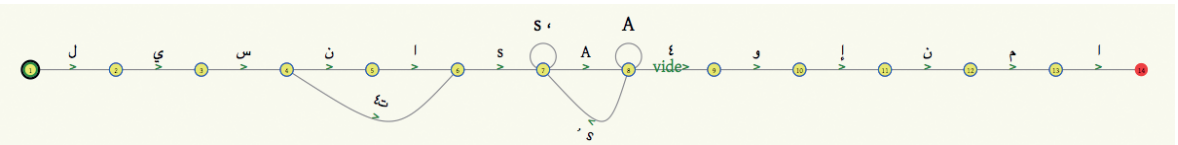


Fig. 1. Partie de l'expression ليس ... وإنما représentée par *Visigram*.

Il est préférable de développer des grammaires plus complexes, comme celles repérant les citations, en dehors du cadre des expressions régulières car elles peuvent devenir difficiles à manipuler et le logiciel *Sarfiyya* est alors plus adapté car il offre non seulement une visualisation mais aussi la possibilité d'éditer graphiquement les grammaires développées.

Résultats de l'analyse par expressions

Les résultats sont présentés de la façon suivante : chaque phrase de chaque texte du corpus contenant l'expression est reproduite et le passage correspondant rigoureusement à l'expression est mis en évidence par un code de couleur et des marqueurs verticaux. Le numéro d'occurrence du premier mot de la phrase est rappelé.

© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

Citations et autres grammaires

Les grammaires ayant atteint un niveau satisfaisant de développement sont installées et placées en accès direct (liste à gauche, sous *Réglages et actions*), une fois leur version déterministe calculée. Ainsi la fonction **Citations** effectue l'analyse du texte entier par l'automate *cit1-6* développé dans *Sarfiyya*[7]. Ce dernier automate modélise un nombre conséquent de verbes de citation suivis de leur régime (etc. ،عبّر عن ،أشار إلى) ; il comporte une version déterministe volumineuse (près de 10 000 états) d'un calcul fastidieux et qui est chargée au démarrage de *Kawâkib*. Les résultats sont présentés de même façon que pour les tests d'expressions régulières.

Performances

Nous donnons ici les temps de calcul constatés en réseau local, pour 1 000 mots, avec *Kawâkib* installé sur un serveur Web (équipé d'un processeur Quad Xeon cadencé à 2 Ghz).

Les temps, établis ici sur la base d'un texte de 20 000 mots, peuvent fluctuer selon la nature des textes (variété des racines, répétitions, nombre de citations, etc.) et sont à moduler, pour une utilisation via le réseau Internet, par la taille des informations à transmettre et donc par la vitesse du réseau employé.

Fonction	Temps (s) / 1 000 mots
racines triées par fréq.	0,6
tokens	0,08
répétitions	2,5
citations	0,26
regex, selon complexité	0,05 à 0,20

Fig. 2. Temps de calcul des principales fonctions de *Kawâkib*.

Versions de *Kawâkib*

Version publique : http://www.ifao.egnet.net/kawakib (fig. 3)

Cette version s'adresse aux étudiants de l'arabe, aux personnes désireuses d'accompagner leur lecture d'un texte de notions linguistiques ; elle n'est cependant ni un dictionnaire ni une aide à la traduction. Le site est disponible avec une interface française et arabe ; une version anglaise est également en préparation.

Les textes analysés par cette version sont tronqués à 5 000 caractères, soit environ 900 mots. Les fonctions de navigation (occurrences et séquence) et d'extraction de citation ne sont pas disponibles, la détection des *tokens* est moins complète (pas de mise en évidence des coordonnants et prépositions proclitiques) et les racines fréquentes sont limitées aux vingt-cinq premières.

7. Audebert *et al.*, Medar 2009, p. 112.

© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net

Version professionnelle : http://www.ifao.egnet.net/kawakibpro (fig. 4)

Cette version comporte toutes les fonctions décrites dans cet article, sans limite de longueur pour le texte à analyser. Elle est pour le moment réservée à notre équipe de recherche (accès par mot de passe) et à ses partenaires, pour évaluation, et a vocation à être rendue publique au terme de cette recherche.

Perspectives

Outil de travail de notre équipe de recherche, cette première version de *Kawâkib Pro* est amenée à s'enrichir d'une gestion centralisée de ressources linguistiques, notamment pour le repérage et la détection des *tokens* de discours (voir article de Claude Audebert dans ce volume). Nous projetons de rendre possible l'enregistrement interactif d'annotations linguistiques, à partir des exemples rencontrés dans les textes. S'appuyant sur les analyses des automates déjà créés, ces remarques collectées par l'équipe serviront de base à la création de nouveaux automates et donc à définir leur interdépendance. Nous espérons disposer d'une description grammaticale (voir article André Jaccarini dans ce volume) à la fois abstraite, opérationnelle et utile à la recherche d'information dans les textes arabes.

AnIsl 44 (2010), p. 53-60 Christian Gaubert
Kawâkib, une application Web pour le traitement automatique de textes arabes.
© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net



Fig. 3. *Kawâkib.*



Fig. 4. *Kawâkib Pro.*

© IFAO 2026 AnIsl en ligne https://www.ifao.egnet.net