



L'écriture inclusive ouvre des liens surprenants. Réflexions en didactique de l'informatique

► Éric BRUILLARD (Laboratoire EDA, Université de Paris)

■ **RÉSUMÉ** • Ce texte présente une réflexion didactique à partir des transformations effectuées automatiquement par des logiciels sur certaines formes d'écriture inclusive. Il explicite les connaissances informatiques nécessaires pour pouvoir comprendre ce qui se passe et présente quelques effets d'apprentissage liés au fonctionnement de certains logiciels de production. Il montre l'importance du détournement pour comprendre les fonctionnements non visibles et développer un esprit critique.

■ **MOTS-CLÉS** • écriture inclusive, contenus informatiques, apprentissage de l'informatique, détournement

■ **ABSTRACT** • *This text presents a didactic reflection based on the transformations carried out automatically by software on some forms of inclusive writing. It explains the informatics knowledge needed to understand what is happening and presents some learning effects linked to the operation of some production software. It shows the importance of hijacking in order to understand non-visible functions and develop critical thinking.*

■ **KEYWORDS** • *inclusive writing, computer literacy, informatics learning, détournement*

Remarque introductive : l'objectif de ce texte est d'essayer de montrer les contenus informatiques qui devraient être pris en compte si on souhaite que les utilisateurs puissent comprendre certains dysfonctionnements (ici une forme simplifiée d'écriture inclusive) liés à des traitements automatiques faits hors de leur contrôle et qu'ils peuvent avoir des difficultés à comprendre et à expliquer. Une tendance courante est souvent de prêter des intentions aux logiciels ou aux machines, alors que les traitements opérés répondent à des logiques d'une toute autre nature. En annexe, j'ai repris un ancien texte qui décrivait un problème analogue, celui d'un correcteur orthographique un peu vite qualifié de raciste.

1. Un étrange phénomène

J'ai constaté début septembre 2020, dans des courriels utilisant l'écriture inclusive destinés à des collègues de l'université, qu'apparaissaient des liens inattendus. Ils étaient portés par des expressions courantes, écrites sous forme inclusive. Je les reprends sans changement : « [Cher.es](#) [ami.es](#)/ nous retrouver [masqué.es](#)/ pour [certain.es](#)/ d'entre vous ». En regardant les liens créés, on constatait une construction directe : insertion de « http:// » suivi de l'expression et ajout final d'un « / ».

Les destinataires des messages, pas les auteurs puisque les transformations s'effectuent à leur insu, avaient vu apparaître ces liens, le plus souvent sans regarder vers quoi ils conduisaient. Or, « [cher.es](#) » conduit au site du même nom (qui est à vendre) affichant une page de vente de matelas, de parfums et un comparateur de prix. Moins direct, « [ami.es](#) » conduit au site « <https://www.boxen1.com/> », page en allemand autour de la boxe. À première vue, on peut être très surpris qu'un message à destination de collègues, pour une réunion censée être pacifique, puisse être transformé et conduire à un site présentant des combats de boxe.

À travers ce petit exemple émergent plusieurs questions, de natures diverses, posées par l'utilisation des technologies informatiques et ayant des implications pour leur enseignement et leur apprentissage : que faut-il savoir pour comprendre ce qui est derrière ce phénomène (qui est loin de se limiter à un apprentissage dit « du code ») et comment peut-on l'enseigner ?

2. Des liens automatiques créés par des applications « intuitives »

Sur un plan strictement informatique, on peut penser que le logiciel de courriel reconnaît ici un motif, ou pattern, du type « XXX.es », c'est-à-dire une suite de caractères se terminant par un point suivi de deux caractères, ce qui est appelé une *expression régulière*. Quand il reconnaît un tel motif, sous certaines conditions (ici, « .es » correspond à un nom de domaine Internet de premier niveau, celui de l'Espagne), il transforme la chaîne de caractères examinée pour qu'elle corresponde à une origine de lien hypertexte.

La question suivante est de savoir quels logiciels font de telles transformations automatiques. L'application de messagerie *Thunderbird* ne le fait pas. Après quelques essais, on s'aperçoit que la messagerie par défaut des *Mac* fait ces transformations, de même que *Gmail*, la messagerie de *Google* (cf. figure 1).

Message saisi

*Cher.es tous et cher.es toutes
alors.fr comment cela va ?
Bonjour.es et al.es
A vous le bonjour.es et le salut.fr*

Message reçu

*Cher.es tous et [cher.es](#) toutes
[alors.fr](#) comment cela va ?
Bonjour.es et [al.es](#)
A vous le [bonjour.es](#) et le [salut.fr](#)*

Figure 1 • Transformations opérées par une messagerie Gmail (sur PC) : les liens créés sont en bleu et soulignés

La vérification des liens créés dans le message reçu confirme les transformations effectuées; selon les environnements utilisés, les expressions commençant par une majuscule sont – ou ne sont pas – transformées. Il est possible alors de faire des hypothèses et de lancer différentes expérimentations en testant des expressions pour vérifier si des modifications sont effectuées et, si oui, lesquelles.

Pourquoi le logiciel utilisé fait-il une telle transformation sans prévenir le rédacteur du courriel ?¹

¹ A vérifier, mais les logiciels libres éviteront de faire des traitements à l'insu des utilisateurs, ce qui peut être moins le cas pour les logiciels des GAFAM.

Pour répondre à cette question, on sort de la technique proprement dite pour prendre en compte les relations entre les concepteurs d'applications et leurs utilisateurs. En effet, en voulant « faciliter » le travail des utilisateurs, dans certaines configurations, on aboutit à des effets non contrôlés et inattendus. Suite aux injonctions du marketing, beaucoup d'utilisateurs veulent des technologies qu'ils qualifient d'intuitives, c'est-à-dire répondant directement à leurs souhaits, sans mettre en œuvre de procédures jugées complexes. Cette prétendue « intuition » correspond à une stratégie de concepteurs de logiciels, qui depuis longtemps, ont abandonné une logique purement technique pour s'adapter aux pratiques développées par les utilisateurs. Ainsi, Microsoft utilise depuis longtemps des anthropologues afin d'étudier en contexte la manière avec laquelle les personnes utilisent les outils et instruments informatiques (Briard, 2004).

Dans leurs courriels, beaucoup d'utilisateurs écrivent des adresses de pages web et sont très satisfaits quand *la machine s'arrange pour que* ces adresses se transforment en liens activables. Ils n'ont pas besoin de sélectionner une chaîne de caractères, d'ouvrir le menu *Insertion*, de préciser la destination du lien, etc. L'interaction est « fluide » et en général efficace, sauf quand des événements inattendus se produisent.

3. Une mise au point « médian »

La méprise du logiciel vient en partie du fait qu'en français, après un point, on met une espace ou alors un autre signe typographique. Normalement, on ne trouve pas le motif « XXX. » suivi de deux lettres (ou plus). Concevoir des applications qui font automatiquement une transformation (création d'un lien) ne devrait pas poser problème, sauf en cas d'erreur typographique. L'écriture inclusive amène justement ce pattern non standard.

En fait, pas vraiment, puisque l'on est face à une seconde forme de « déviance ». En effet, le point préconisé pour l'écriture inclusive est le *point médian*, afin que le point du féminin ne soit pas « en bas ». Ce caractère n'est pas directement accessible au clavier et nécessite d'effectuer un copier-coller ou une combinaison de touches (Alt 0183 sous Windows² voir [ici](#)) Quand on utilise ce point médian, en acceptant la

² Attention, il faut disposer d'un pavé numérique séparé, sinon, avec un traitement de texte, il faut insérer le caractère.

complexification technique, l'expression n'est plus modifiée. On peut aussi séparer le « e » et le « s » par un point, comme dans l'expression « bienvenu.e.s », qui n'est pas non plus transformée.

Dans ces changements opérés par un logiciel, il faut comprendre que derrière une simplification apparente, se cachent une perte de contrôle de l'utilisateur et une illusion qui rend le fonctionnement des machines (ou des logiciels) opaque. En effet, l'utilisateur peut être conduit à croire que la machine « reconnaît » qu'il veut communiquer une adresse cliquable, alors qu'il ne s'agit que d'une forme (un motif) automatiquement repérée.

Un problème analogue s'observe sur certains traitements de texte, lorsque les utilisateurs n'utilisent pas les styles hiérarchiques (styles Titre 1, Titre 2, etc.) et que le logiciel « devine » le niveau de titre à la taille des caractères choisie. Le processus est assez hasardeux. Si l'utilisateur contrôle les éléments de structure, il peut piloter automatiquement la forme sans erreur, alors que la délégation à la machine peut être source de confusions.

Ce problème donne une idée du fonctionnement des « intelligences artificielles » qui justement font des inductions à partir de la reconnaissance de patterns. Cela peut bien fonctionner pour la reconnaissance des visages, l'analyse des radios en médecine, mais peut générer des erreurs grossières dans d'autres domaines.

4. Des destinations farfelues, sous contraintes économiques

La création automatique d'un lien, en prenant comme adresse l'expression telle qu'elle est écrite, n'est qu'une partie de la question. La page sur laquelle le lien aboutit en est une autre. La réponse n'est pas technologique mais économique. Pour attirer les internautes, les sites agrègent beaucoup d'adresses différentes d'une part et, d'autre part, essayent d'être bien classés sur un grand nombre de requêtes.

Ainsi, le site <http://cher.es/> est en vente le 14 septembre. De même, si l'on est « mort de rire.es » : le domaine « rire.es » est en vente, pour 288 euros ! Le site <https://www.boxen1.com/> est en lien à « ami.es ». L'appel du site <http://certain.es> conduit à un message d'erreur 403, pouvant laisser croire à un endroit privé auquel on n'aurait pas la permission d'accéder (ce qui correspond à ce type d'erreur) : « *Forbidden. You don't have permission to access / on this server. Additionally, a 403 Forbidden error was encountered while trying to use an ErrorDocument to handle the request* ».

Ainsi, même s'il n'y a pas de page correspondant au lien généré, les navigateurs et les moteurs utilisés vont essayer de donner une réponse, afin d'éviter de confronter l'utilisateur à un message d'erreur ou à l'angoisse générée par le silence. Toujours la recherche de la « fluidité » de l'interaction !

En tous cas, la question de la destination du lien généré n'a aucun *lien* avec la création de ce lien !

5. Un contenu d'enseignement à construire

À la suite de cette petite enquête, on peut s'interroger sur ce qu'il faut savoir pour pouvoir la mener. Pour faire simple, il s'agit d'identifier des contenus d'enseignement, au sens défini par Fluckiger (Fluckiger, 2019). La notion générale d'expression régulière ne fait pas partie des programmes de l'école primaire, mais reconnaître un motif particulier dans des suites de caractères n'est pas très difficile. Ayant identifié des logiciels effectuant les transformations étudiées, des activités de découverte et de tests peuvent être menées avec des élèves, même jeunes. Cela peut contribuer à construire une vision non anthropomorphique du fonctionnement des machines numériques.

Ensuite, étudier les caractéristiques de l'écriture inclusive peut conduire à l'énonciation de certains algorithmes pour générer les expressions et réfléchir à différentes formes de « points » permet de travailler sur l'écriture même de la langue, ce qui peut s'avérer très utile.

Enfin, disposer de quelques connaissances sur l'économie du web et le fonctionnement des moteurs de recherche, éléments qui sont au cœur de certaines pratiques personnelles en fin d'école primaire et au collège, est de nature à aider les élèves à prendre de la distance avec leurs pratiques et à leur donner des capacités d'analyse.

Pointer des dysfonctionnements des technologiques numériques, non pour les éradiquer, mais pour déclencher des réflexions et des activités d'enseignement et d'apprentissage, en favorisant le détournement et l'expérimentation, est certainement au cœur de la didactique de l'informatique. Essayons d'approfondir ce dernier point.

5.1. Des logiciels de production qui sont en partie leur propre EIAH

Un autre exemple va nous permettre de mettre en perspective le fonctionnement de certains logiciels généraux (comme les traitements de

texte) et les apprentissages plus ou moins implicites que les utilisateurs peuvent réaliser.

Lorsque l'on rédige un paragraphe avec une vérification en français, l'écriture avec des chiffres de « premier » ou « première », « deuxième » ou « seconde », etc., peut conduire à un passage automatique en exposant. Ainsi, avec le traitement de texte Word, on écrit « 1 » puis « er » collé, puis une espace, et on obtient « 1^{er} ». Si on écrit « 1 » suivi de « st », rien ne se passe : « 1st ». Quand on écrit en anglais, avec une langue anglaise comme langue de vérification du texte, on obtient un fonctionnement différent : « 1st ».

Ainsi, le mode de fonctionnement semble refléter les règles typographiques et même en quelque sorte « implémenter » certaines de ces règles. Cependant, ces dernières ne sont pas complètement liées aux langues, mais avant tout aux règles édictées dans les différents pays et qui sont évolutives. En France, on devrait écrire « 2^e », mais on voit souvent « 2^{ème} » et le logiciel de traitement de texte de Microsoft fait automatiquement le passage en exposant (mais il y a peut-être un réglage de paramètre à faire). Avec LibreOffice ou OpenOffice, la transformation se fait avec le « e » mais pas avec le « ème ». Pire, « 2^{ième} », qui est vraiment une forme *fautive* en France, est aussi obtenue automatiquement par Word.

Le logiciel a certainement des effets d'apprentissage fort. Une transformation faite automatiquement est comme une rétroaction positive, montrant ce qui est correct. La répétition des gestes (mettre « ième » après un chiffre et appuyer sur la barre d'espace) agit comme un renforcement positif : l'écriture est forcément correcte, puisqu'elle est automatiquement effectuée par le logiciel lui-même (par la mise en exposant). En outre, saisir une quelconque suite de caractères précédée par un chiffre et suivie par une espace ne conduit pas à une mise en exposant. Cela renforce l'idée que le logiciel utilise des règles, censées être correctes, quand il effectue la transformation (en partie) souhaitée : comme un enseignant qui vous récompense quand vous écrivez quelque chose de correct.

Il y a ainsi un apprentissage par expérience, par l'action. On peut faire des essais et erreurs et observer quand on est récompensé par une transformation : « 1nd » pas de passage en exposant, mais avec « 2nd », cela fonctionne. On apprend ainsi les formes typographiques correctes... en fait, pas vraiment, comme on vient de le voir, avec « 2^{ième} » !

D'ailleurs un enseignement explicite des règles typographiques pourrait se heurter au fonctionnement du logiciel et aux comportements renforcés et « appris » via l'utilisation du logiciel. On n'est pas si loin des technologies dites persuasives (Fogg, 2009), conçues « pour changer les attitudes ou les comportements des utilisateurs par la persuasion et l'influence sociale, mais non par la coercition » - selon Fogg, traduction de (Zouinar, 2019). Pour des raisons notamment éthiques, il est souhaitable de procéder autrement, en changeant la manière d'utiliser les logiciels.

5.2. Détourner les logiciels de production pour en faire des environnements d'apprentissage

Même s'ils ne sont pas à proprement parler des EIAH, dans le sens d'environnements conçus pour l'apprentissage humain, des logiciels peuvent le devenir par détournement. On observe un dysfonctionnement dans un logiciel de production ou dans une fonctionnalité d'un logiciel et on tire parti du caractère réactif de ce logiciel pour le transformer en environnement d'apprentissage : processus d'essais-erreurs, induction, etc. On détourne le logiciel de ses finalités en essayant de prévoir les transformations opérées par le logiciel : faire sciemment des erreurs orthographiques, écrire des formes un peu particulières, etc.

D'ailleurs, il y a quelques années, c'était une bonne manière pour essayer de comprendre le fonctionnement des moteurs de recherche : lancer la même requête pendant un certain laps de temps pour étudier l'évolution des réponses proposées. Mais si on le fait de manière très instrumentée, en lançant beaucoup de requêtes, le test peut être repéré par le moteur de recherche (enfin par celui de Google) conduisant à d'autres processus dans la gestion des propositions. On est dans un cas où le test change le phénomène à observer. Le moteur Google intègre également des jeux (la requête « jouer avec Google » fournit des exemples) et des jeux peuvent aider à mieux comprendre son fonctionnement (Simonnot, 2008). Ainsi il est intéressant d'essayer de trouver deux ou trois mots qui, utilisés séparément dans une requête, donnent beaucoup de résultats et plus aucun quand ils sont associés.

Via des consignes particulières qui orientent des détournements, les environnements deviennent des environnements pour l'apprentissage humain, juste en changeant la manière de les utiliser. Bien évidemment, en termes de recherche, il conviendrait de tester ces activités avec un public et d'essayer de décrire, voire de caractériser avec quelques mesures

ad hoc, ce que les utilisateurs arrivent à mieux comprendre. On pourrait réfléchir à intégrer des fonctionnalités aux logiciels généraux pour qu'ils puissent plus aisément être détournés afin de faciliter leur propre compréhension.

En outre, derrière des apprentissages plutôt techniques, on peut également développer des « dispositions » (*esprit critique*) vis-à-vis des dispositifs informatiques en appliquant ce principe d'apprentissage essentiel : le détournement. Demeure une tension également essentielle entre utilisation et compréhension, s'agissant des technologies informatiques : une relative maîtrise de leur fonctionnement ne devrait pas être l'unique objectif, il faudrait toujours y associer des éléments de compréhension.

RÉFÉRENCES

Briard, C. (2004, 27 janvier). Quand Microsoft se branche sur l'anthropologie. *Les Echos*. <https://www.lesechos.fr/2004/01/quand-microsoft-se-branche-sur-lanthropologie-628862>

Fluckiger, C. (2019). *Une approche didactique de l'informatique scolaire*. Rennes : Presses Universitaires de Rennes.

Fogg, B. J. (2009). A behavior model for persuasive design. *Proceedings of the 4th International Conference on Persuasive Technology*, 40, 1-7. <https://dl.acm.org/doi/10.1145/1541948.1541999>

Simonnot, B. (2008). Quand les moteurs de recherche appellent au jeu : usages ou détournements ? *Questions de communication*, 14, 95-114. <https://journals.openedition.org/questionsdecommunication/752>

Zouinar, M. (2019). Théories et principes de conception des systèmes d'automatisation numériques. De la quantification à la régulation distribuée de soi. *Réseaux*, 4(216), 83-117. <https://www.cairn.info/revue-reseaux-2019-4-page-83.htm>

Annexe

Il y a près de 20 ans, j'ai écrit une tribune sur le fonctionnement du correcteur orthographique du traitement de texte de Microsoft Word suite à un dysfonctionnement qualifié de raciste. Cherchant à retrouver le contexte, j'ai trouvé deux entrefilets dans des journaux informatiques.

« Le correcteur d'orthographe de Word 2000 suggère de remplacer le mot "Anti-stress" par "Anti-arabes". Microsoft, profondément gêné par cette affaire, a préféré interrompre momentanément la production de son logiciel ». Zdnetⁱ

« Pour se justifier, Microsoft a fourni l'explication "technique" suivante : proche d'anti-stress, avec 8 lettres communes sur 11, la chaîne de caractères "anti-arabes", qui figure également dans le dictionnaire en temps [sic] que mot "d'usage courant", est automatiquement proposée par l'algorithme.

La direction de Microsoft réfute donc toute intention raciste et a d'ores et déjà édité un correctif, qui sera téléchargeable sur son site d'ici à quelques jours et intégré dans la version de Word 2000 commercialisée à partir du mois prochain. » 01.netⁱⁱ

Mais le texte lui-même, publié en juin 2001 dans une revue du CNDP, longtemps accessible en ligne, ne semble plus l'être, interrogeant sur l'intérêt et la capacité des institutions nationales liées au ministère de l'éducation à conserver les réflexions et publications sur le numérique éducatif.

Comment savoir si mon correcteur orthographique est raciste ?

Éric Bruillard IUFM de Caen

L'histoire remonte à l'an dernier. Au cours d'une émission diffusée sur une chaîne cryptée, un animateur se sert à l'écran du correcteur orthographique d'une version récente d'un traitement de texte célèbre. Il tape le mot composé *Anti-stress* (avec une majuscule). Ne reconnaissant pas ce vocable, le logiciel le souligne. L'appel au correcteur fournit une seule proposition : « Anti-Arabes ». L'association du mot « stress » à « Arabes » est pour le moins troublante. Sur le plateau de l'émission, la

cause est jugée : voilà, à n'en pas douter, un flagrant délit de racisme par machine interposée... Les coupables sont vite désignés : l'entreprise responsable du programme ou ses sous-traitants.

Avant de souscrire à ce verdict sans appel, en se glorifiant même de ne rien comprendre au programme, donnons un moment la parole au correcteur orthographique afin qu'il puisse « expliquer » son comportement.

Deux questions préalables

Nous nous faisons toujours une représentation – même grossière, même en partie inconsciente – des systèmes techniques que nous utilisons. Ainsi, nous supposons que la machine, lorsqu'elle ne reconnaît pas un mot, cherche à en donner un autre, proche. Mais que signifie proche, dans ce cas ? J'ai travaillé la question avec plusieurs groupes d'adultes en formation. L'idée qu'ils s'en faisaient est que la machine consulte une sorte de dictionnaire aux entrées classées alphabétiquement ; la proximité serait alors l'ordre classique du dictionnaire. Or, on s'en rend compte, à la suite de la racine *anti*, il y a loin du « s » de *stress* au « A » de *Arabes*...

L'autre question est celle de l'origine du corpus. D'où sort le mot « Anti-Arabes » ? Il n'est certainement pas dans le dictionnaire de l'Académie. De façon plus générale, d'où viennent les mots reconnus par le dictionnaire orthographique d'un logiciel ? En fait, conçu pour aider le scripteur, ici et maintenant, il doit forcément intégrer les mots courants – et pas uniquement ceux qui sont passés à la postérité. Des sociétés établissent donc des corpus sur la base, notamment, des articles de journaux. On comprend alors qu'un mot comme « Anti-Arabes », malheureusement d'actualité, ait été retenu. On ne peut pas bannir certains mots de la langue sous prétexte que le concept qu'ils véhiculent ne nous plaît pas.

Une démarche expérimentale

Mais comment trouver le fonctionnement d'un correcteur informatique, et, plus exactement, du programme qui, à partir d'une graphie incorrecte dans une langue, propose des mots de graphie proches et acceptables ? Nous choisissons la démarche expérimentale : fournir à la machine des mots mal orthographiés et essayer de déduire de ses propositions des règles de fonctionnement (au passage, il est intéressant de constater que les utilisateurs répugnent à commettre sciemment des

erreurs...). Cette activité est en elle-même formatrice : étudier le phénomène, faire des hypothèses, les mettre à l'épreuve, essayer de prédire sur de nouveaux cas le comportement du programme...

On tombe rapidement d'accord sur la nécessité de travailler de manière méthodique et de commencer avec des mots courts. Ainsi, à partir de *apie*, on obtient dans l'ordre *prie, ripe, pie, pies, rie*. Le mot *rabre* conduit aux propositions *arbre* et *rare*. Dans ces deux exemples, la première proposition s'est faite à partir de l'inversion des deux premières lettres du mot. Le logiciel ne s'attache donc pas à l'ordre alphabétique.

En fait, la logique de conception du programme étant d'aider un scripteur en chair et en os, elle tient compte des sources d'erreur : une inversion de lettres est beaucoup plus courante avec un clavier qu'avec un stylo. La réflexion à mener est donc de nature ergonomique.

En multipliant les expériences, on s'aperçoit que les solutions proposées résultent de trois transformations élémentaires : l'inversion de deux lettres consécutives, la soustraction ou l'ajout d'une lettre. Cela indique bien que les opérateurs sont de nature formelle. Et la combinaison de ces transformations donne l'idée de distance entre deux suites de caractères. La notion de proximité devient alors claire : elle dépend du nombre de transformations élémentaires nécessaires pour passer d'un mot à un autre. Ainsi, des rapprochements apparemment magiques viennent-ils simplement du fait que deux mots sont proches, dans le sens qui vient d'être défini.

Apparaît alors une notion de complexité, qui peut être mesurée par le nombre de combinaisons d'opérations élémentaires. On convoque les mathématiques pour montrer que le nombre de ces combinaisons croît avec la longueur du mot de départ. Ainsi, pour les mots longs, il est difficile d'appliquer exactement les mêmes principes. Si l'orthographe est trop altérée, on a peu de chance de retrouver une graphie correcte par le jeu des transformations élémentaires. Le nombre de propositions faites par l'ordinateur diminue d'ailleurs considérablement.

Sur des mots encore plus longs, il arrive que la seule suggestion de la machine consiste à remplacer un mot par deux (c'est le cas pour les adverbes qu'elle ne reconnaît pas, pour lesquels elle propose de traiter le suffixe *ment* comme un mot séparé). Notons que l'oubli de l'espace entre deux mots est une erreur que personne ne ferait avec un stylo.

Différents essais montrent que les principes mis en évidence sont à peu près respectés, avec cependant, parfois, des propositions plus énigmatiques, sans doute à mettre sur le compte de l'obligation, pour le programme, de répondre quasi instantanément.

Une machine mieux contrôlée

Pour résumer, cette expérience a permis de se convaincre qu'aucun traitement sémantique n'est opéré par la machine. On peut être déçu de ne pas avoir su mettre en lumière le processus qui a conduit à cette proposition du logiciel. L'énigme n'a certes pas été complètement levée mais des pistes ont été données et d'autres restent à explorer. Par exemple : les mots *stress* et *Arabes* ont tous les deux six lettres, dont trois en commun. Existe-t-il beaucoup de mots de six lettres qui acceptent le préfixe *anti*? Et parmi ces mots, y en a-t-il beaucoup à posséder plus de trois lettres **en** commun ?

Reste, bien sûr, la solution de demander des comptes à l'entreprise. Mais outre qu'elle n'acceptera sans doute pas de dévoiler ses secrets de fabrication, il n'est pas sûr que les explications qu'elle fournirait seraient lisibles par des non-spécialistes...

L'intérêt de cette expérimentation, même si elle n'est qu'exploratoire, c'est qu'elle a fait comprendre aux adultes en formation que la machine n'avait pas de comportement magique ou aléatoire et qu'elle n'était habitée par aucun « malin génie » qui aurait truffé de mots pièges sa banque de données... Au passage, on a eu l'occasion de réfléchir au rôle de l'interface d'un logiciel de bureautique. On s'est posé quelques questions : pourquoi les mots mal orthographiés sont-ils soulignés ? De quelles connaissances ou informations dispose la machine et quel processus met-elle en œuvre ? Quelles sont les compétences nécessaires ou utiles à l'utilisateur pour s'approprier un tel programme ? Quels détournements peut-il opérer ? (On sait qu'il est parfois plus rapide de faire sciemment des fautes, sachant que le recours au correcteur permet d'aller plus vite. C'est notamment le cas lorsqu'il faut inclure des caractères qui ne sont pas directement accessibles au clavier - par exemple des majuscules accentuées.)

Avoir une meilleure idée de ce que fait la machine ne rendra pas toujours un usager plus performant, mais cela lui permettra d'exercer un contrôle plus grand sur les instruments technologiques qu'il utilise. Sous des formes diverses, les dispositifs informatiques sont maintenant partout

Éric BRUILLARD

présents dans notre environnement quotidien. Ignorer totalement leur fonctionnement ne peut que renforcer des visions naïves ou erronées ; aussi est-il important d'essayer de le dévoiler en invitant les élèves à une démarche de type expérimental et en aiguisant leur curiosité - par exemple, à propos d'un correcteur politiquement incorrect...
