Techniques d'enquête, décembre 1988
Vol. 14, n° 2, pp. 307–316
Statistique Canada

QUID, une méthode générale de chiffrement automatique

JACQUES LORIGNY[1]

RÉSUMÉ

Le système QUID, concu et développé par l'INSEE (Paris) est un système de chiffrement automatique de données d'enquête recueillies sous forme d'intitulés littéraux exprimés dans la terminologie du répondant. Le système repose sur l'utilisation d'une très vaste base d'apprentissage composée de phrases réelles codifiées par des experts. L'article présente d'abord le traitement automatique de normalisation préalable des phrases, puis l'algorithme organisant la base de phrases en une arborescence optimisée. Un exemple de classement est donné en illustration. Le traitement des variables annexes de codification, venant compléter l'information contenue dans les phrases, présente actuellement des difficultés qui sont examinées en détail. Le projet QUID 2, version renovée du système, est évoqué succinctement.

MOTS CLÉS: Codification automatique; variables en langue naturelle, appariement de phrases; N-grammes.

1. INTRODUCTION

Le système QUID (abrégé de QUestionnaires d'IDentification) est un système de chiffrement automatique concu et développé par l'Institut National de la Statistique et des Études Économiques (INSEE) depuis les années 1979-1980.

Rappel du problème

Le problème consiste à classer automatiquement un individu enquêté dans un poste défini d'une nomenclature existante (par exemple, la nomenclature des Professions). Pour cela, le système utilise principalement la réponse en clair à la question posée directement (par exemple «Quelle profession ou quel métier exercez-vous actuellement ?»), et accessoirement d'autres informations figurant dans le formulaire d'enquête et supposées préalablement codifiées (par exemple, le code Activité Économique de l'entreprise employant l'individu).

Dans notre terminologie, la réponse directe en clair est appelée «intitulé littéral», ou en abrégé «intitulé». Les informations codifiées complémentaires sont désignées sous le terme générique de «variables annexes».

Nous présentons dans la prochaine section l'approche de base du système QUID, et donnons des résultats de son application à l'INSEE. Dans la section 3, nous décrivons le système dans sa version actuelle. Enfin, nous examinons le problème du traitement des variables annexes dans la section 4. La nouvelle version, QUID 2, présentée dans cette même section, devrait aider au traitement des difficultés rencontrées.

2. LE PRINCIPE DE LA MÉTHODE

2.1 L'approche de base

L'approche de base du système QUID consiste à élaborer une base de données très importante constituée d'intitulés typiques des répondants, accompagnées du code correspondant

[1] Jacques Lorigny, Administrateur à l'Institut National de la Statistique et des Études Économiques 18, Bld Adolphe Pinard 75675 PARIS CEDEX 14 (France).

attribué par un expert. La base de données est la plus étendue possible en vue de permettre l'atteinte d'un taux élevé d'appariement, et l'on ajoute à la base de nouveaux intitulés à mesure que ceux-ci apparaissent.

Dans notre terminologie, la base de données s'appelle «base d'apprentissage», ou «fichier d'apprentissage» (FA) parce qu'elle présente à l'état brut la structure ordinaire d'un fichier plat. Pour constituer le fichier d'apprentissage, nous partons le plus souvent de l'enquête d'une année antérieure, déjà chiffrée manuellement ou par une méthode interactive. Chaque intitulé de la base est accompagné de son code (supposé *a priori* exact), et de sa «fréquence d'occurrence» observée dans le FA, c'est-à-dire du nombre d'individus ayant répondu par cet intitulé.

La tâche de gestion de la base d'apprentissage (apurement, extension) est complètement déconnectée de l'exploitation de chiffrement de l'enquête en cours. Elle est confiée à un atelier centralisé composé de codeurs spécialisés, tandis que l'exploitation de chiffrement proprement dite est le plus souvent décentralisée régionalement.

La difficulté propre à une approche de ce type provient de l'accroissement rapide du temps de recherche dans la base au fur et à mesure que sa taille augmente. Pour y remédier, le système QUID utilise des résultats mathématiques de la Théorie de l'Information (Shannon 1948; Picard 1972; Bouchon-Meunier 1978; M. Terrenoire 1970; Tounissoux 1980), grâce auxquels le temps de recherche est minimisé en organisant la base sous forme d'une structure arborescente optimisée.

L'approche de base du système QUID permet aussi d'opter pour un ensemble de logiciels généraux, c'est-à-dire s'appliquant à tous les champs sémantiques, comme par exemple des intitulés de profession, des intitulés de produits alimentaires, ou des intitulés de communes.

2.2 Les résultats obtenus

Le système est expérimenté sur différents travaux de l'INSEE et fonctionne en exploitation courante pour le chiffrement du code CS (catégorie socio-professionnelle) dans le traitement des DADS (Déclarations annuelles de données sociales) fournies par toutes les entreprises employant des salariés. Indiquons quelques chiffres pour situer les ordres de grandeur.

Dans l'application aux DADS, le fichier d'apprentissage comprend à ce jour 122 000 intitulés (représentant une population d'apprentissage de 650 000 salariés). Son organisation optimisée est une arborescence d'environ 100 000 sommets (dont 86 000 sommets de décision *cf.* 3.2). Il a été utilisé pour chiffrer une population de 570 000 salariés avec une efficacité moyenne de 90%, variant entre 85% et 95% selon les régions. Nous entendons par «efficacité» le pourcentage de cas où le système fournit une réponse unique, que nous acceptons par principe dans les conditions de cette application. Ne disposant pas actuellement de mesure précise de la validité de ces réponses uniques, nous estimons vraisemblable un taux d'erreur de l'ordre de 5% à 10%.Toutefois, la base d'apprentissage est en cours d'apurement à l'atelier d'expertise de Dijon, après quoi le taux d'erreur devrait normalement diminuer dans une proportion notable. Nous aurons des chiffres plus précis à communiquer à ce moment-là.

Du point de vue des contraintes informatiques, l'arborescence optimisée est chargée dans 3 300 kilooctets de mémoire centrale (virtuelle) et le temps de chiffrement automatique d'un cas individuel est de l'ordre de 40 ms d'unité centrale IBM 4341.

Nous possédons depuis quelques mois une variante du logiciel de chiffrement proprement dit destinée aux mini-ordinateurs et qui charge l'arborescence par parties, en fonction de l'espace mémoire autorisé.

Dans d'autres applications que celle des DADS, l'efficacité est moindre et ne dépasse pas 75%. Tout dépend de la qualité et de l'exhaustivité de la base d'apprentissage.

3. LE SYSTÈME QUID DANS SA VERSION ACTUELLE (ou QUID 1)

3.1 Normalisation préalable des intitulés

Avant de construire l'arborescence optimisée, les intitulés bruts subissent d'abord un traitement automatique de normalisation préalable, commandé par un jeu de paramètres externes choisis par l'utilisateur pour son application.

Les mots sont séparés et cadrés dans des zones fixes dont la longueur (unique pour tous les mots) et le nombre maximum (unique pour tous les intitulés) sont paramétrés. Il est conseillé de choisir par ces deux paramètres une valeur plutôt large et de laisser l'algorithme d'optimisation sélectionner lui-même les parties significatives de l'intitulé (*cf.* 3.2). Par exemple, l'application des DADS (*cf.* 2.2) a choisi 4 zones de 12 caractères chacune.

Les «mots vides» sont éliminés. La liste des mots vides est un paramètre externe fourni par l'utilisateur pour son application. Elle comprend le plus souvent les articles, prépositions, *etc* ... et dépend beaucoup de l'application.

Les sigles sont normalisés (I.N.S.E.E. devient INSEE, S N C F devient SNCF).

Enfin l'utilisateur peut effectuer sur la table des mots séparés tout traitement particulier de son choix (sous forme d'un sous-programme en langage PL/1). En fait, cette possibilité apparaît rarement nécessaire et est très peu utilisée (sauf pour le chiffrement des codes de commune à partir des intitulés de commune).

Lorsque le traitement des mots est terminé, ceux-ci sont découpés en bigrammes (tranches de deux lettres consécutives) ou trigrammes (tranches de trois lettres consécutives), ou *etc* ... Le choix de ce mode de découpage est paramétré (mais unique pour toute l'application traitée). En pratique, le découpage en bigrammes est le seul a avoir été utilisé jusqu'à présent mais l'idée d'un découpage en trigrammes mériterait d'être expérimentée. Pour la suite de l'exposé, nous considérerons uniquement le découpage en bigrammes.

3.2 L'algorithme de construction de l'arborescence optimisée

Notons $T = (t_1, t_2, \ldots, t_j, \ldots, t_n)$ le code à chiffrer, par exemple l'ensemble des modalités du code Profession.

$Q = (q_1, q_2, \ldots, q_i, \ldots, q_m)$ l'ensemble des bigrammes résultant de la normalisation des intitulés (par exemple m = 24 si l'on a choisi le nombre 4 comme paramètre «nombre de mots» et 12 caractères comme paramètre «longueur de mot»).

X = l'arborescence à construire, que nous appelons un «quid» (questionnaire d'identification).

L'algorithme construit X en descendant du sommet-racine x_0 (placé par convention au «niveau 0») jusqu'aux sommets de niveaux 1, 2, *etc* ...

Au sommet-racine x_0 il associe le FA tout entier et cherche le meilleur bigramme à interroger en premier, c'est-à-dire celui qui, dans le FA tout entier est le plus discriminant pour le code cherché T.

Notons $N(x_0)$ la fréquence d'occurrence totale associée au FA entier, c'est-à-dire la somme des fréquences accompagnant les intitulés de la base,

$N(x_0, j)$ la fréquence d'occurrence du code t_j dans le FA tout entier.

Nous supposons la population d'apprentissage statistiquement représentative de la population à chiffrer (rappelons que le FA est très souvent, en pratique, le fichier d'enquête d'une année antérieure).

On peut donc estimer la probabilité de trouver le code t_j dans la population à chiffrer, par la formule:

$$\Pr(t_j \mid x_0) = N(x_0, j)/N(x_0).$$

L'incertitude a priori sur T est mesurée par l'entropie de Shannon:

$$H(T \mid x_0) = \sum_j \Pr(t_j \mid x_0) \log 1/\Pr(t_j \mid x_0).$$

Supposons qu'un bigramme (quelconque) q_i soit affecté au sommet x_0. À chacune des modalités qu'il prend dans le FA nous associons la sous-base constituée des intitulés possédant cette modalité.

Notons $(a_i^1, a_i^2, \ldots a_i^k, \ldots)$ les modalités prises par le bigramme q_i dans le FA. Pour chacune de ces modalités, donc pour chacune des sous-bases engendrées, nous créerons un sommet y, successeur immédiat de x et placé au niveau 1 de l'arborescence.

L'information apportée par le bigramme q_i (supposé affecté au sommet-racine x_0) est mesurée par la réduction moyenne de l'incertitude sur T en passant de x_0 à l'un des sommets y.

Soit:

$$I(x_0, T, q_i) = H(T \mid x_0) - \sum_{y \in \Gamma(x_0)} \Pr(y)\, H(T \mid y),$$

où l'on note

$\Gamma(x_0)$ l'ensemble des sommets y successeurs au niveau 1 du sommet x_0

$H(T \mid y)$ l'entropie conditionnelle de T au sommet y.
(même formule que ci-dessus en remplaçant x_0 par y).

$\Pr(y) = N(x_0, a_i^k)/N(x_0)$ si a_i^k est la modalité du bigramme q_i qui engendre le sommet y et $N(x_0, a_i^k)$ la fréquence d'occurrence de la modalité a_i^k du bigramme q_i dans le FA tout entier.

L'algorithme effectue ce calcul d'information pour tous les bigrammes $q_1, q_2, \ldots, q_m$, puisqu'au sommet racine x_0 ils sont tous candidats possibles à la sélection comme premier bigramme interrogé.

L'algorithme choisit le bigramme qui maximize $I(x_0, T, q_i)$, notons le q_{i0}, et partage alors effectivement la base en autant de sous-bases que de modalités du bigramme q_{i0} rencontrées dans la base. Les sommets y, successeurs de x_0 au niveau 1 sont alors effectivement créés. La construction du niveau 1 de X est terminée.

Pour chaque sous-base obtenue (donc pour chaque sommet y) l'algorithme recommence exactement le même traitement que celui que nous venons de décrire à propos du sommet-racine x_0 *etc* ... *etc* ...

Le processus s'arrête pour un sommet déterminé:

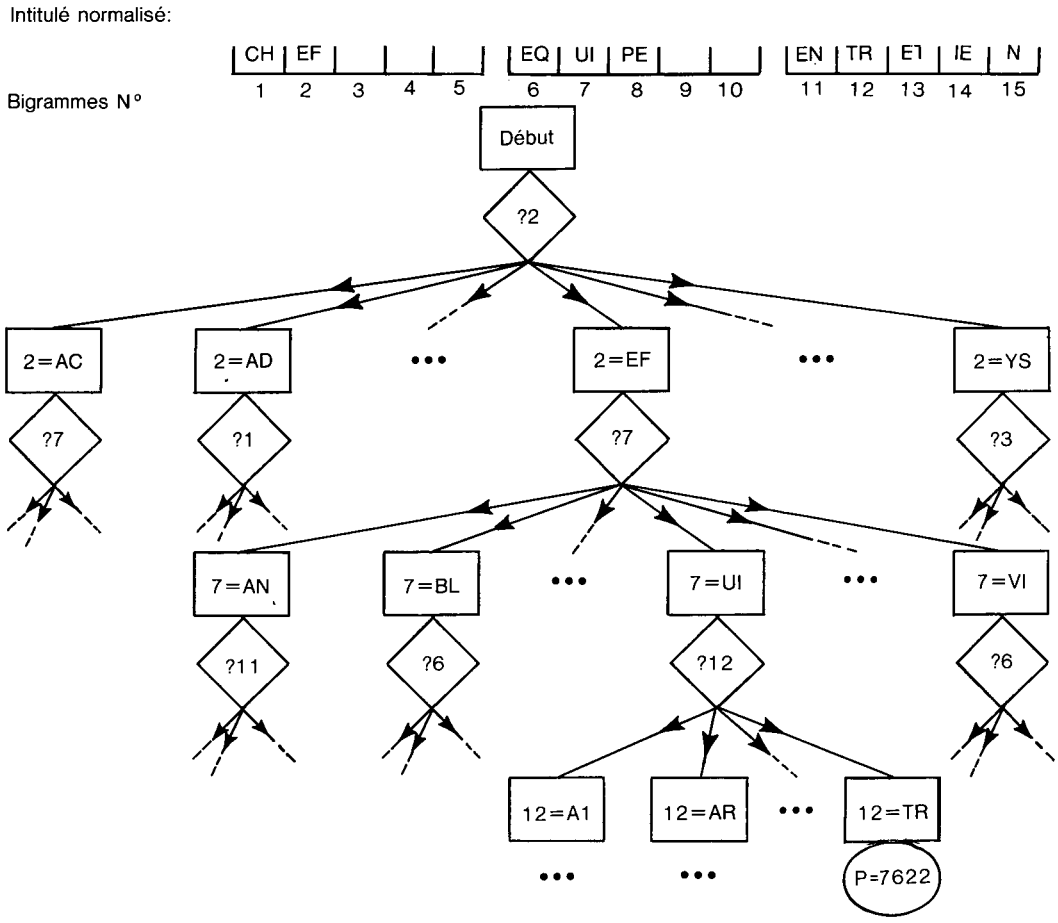
(1) lorsqu'il n'y a plus qu'un intitulé au sommet, et dans ce cas l'entropie conditionnelle équivaut à zéro, ou

(2) lorsqu'il n'existe qu'un nombre restreint d'intitulés qui diffèrent en ce qui a trait aux bigrammes restants, mais qui possèdent tous le même code, ou

(3) lorsqu'il existe deux intitulés ou plus mais qui possèdent des codes différents et non distinguables.

Les cas (1) et (2) sont dits «sommets de décision», le cas (3), «sommet d'indécision». Ils constituent ensemble les «sommets terminaux».

La progression de la construction de l'arborescence X se poursuit de niveau en niveau jusqu'à épuisement du FA. En fait, nous n'avons jamais dépassé le niveau 15 mais aucune limite n'est fixée par le système lui-même. Un exemple de classement dans l'arborescence est donné en figure 1.



?2 En ce sommet de l'arborescence interroger le contenu du bigramme n° 2.

2=EF Le contenu du bigramme n° 2 est EF.

P=7622 En ce sommet de l'arborescence, on peut déterminer le code Profession: sa valeur est 7622 (Nomenclature des métiers de 1975).

Dans l'exemple ci-dessus, l'intitulé brut est celui de la profession déclarée par l'individu recensé. L'objectif du système est de déterminer le Code Profession correspondant, dans la Nomenclature des Métiers de 1975.

Dans une première étape, on extrait les dix premiers caractères des trois mois les plus significatifs. On obtient ainsi l'initulé normalisé qui est alors découpé en couples de lettres (appelés bigrammes) numérotés de 1 à 15. Ensuite commence l'interogation proprement dite. Elle s'opère selon un enchaînement de questions-réponses optimisé par un algorithme mathématique basé sur la théorie de l'information. Ce calcul a lieu au cours d'une phase préliminaire qui détermine, en fonction du fichier d'apprentissage donné, le premier bigramme à interroger, puis la séquence des questions suivantes selon la réponse obtenue chaque fois. Ici l'ordinateur interroge d'abord le bigramme n° 2, qui contient EF, puis le bigramme n° 7, qui contient UI, et enfin le bigramme n° 12, qui contient TR. A ce stade il constate qu'il peut sans ambiguité déterminer qu'il s'agit du code Profession 7622 (Agents techniques et techniciens s.a.i.). La durée totale du traitement est en moyenne de 41 millisecondes d'ordinateur IBM 370/148 et la mémoire centrale utillisée, de 380 K octets.

Intitulé brut: chef d'équipe d'entretien.

Figure 1. Exemple de classement d'un intitulé dans l'arborescence.

3.3 L'exploitation de chiffrement proprement dit

Pour chiffrer un intitulé de l'enquête en cours, on commence par le normaliser selon 3.1. Puis, les bigrammes obtenus sont appariés avec ceux du quid chargé dans l'ordinateur. L'exploration conduit à trois issues possibles.

3.3.1 Sommet de décision

Le système fournit un code unique mais qui peut très bien être erroné si la base d'apprentissage se trouve être trop pauvre. Par exemple, dans un de nos premiers essais en 1979, nous obtenions un sommet de décision de niveau 1, par bigramme 2 = *CC*, au vu du seul intitulé appris VACCINEUR VOLAILLES.

Lorsqu'est apparu ensuite l'intitulé à chiffrer RACCOMMODEUR VÊTEMENTS, le code unique obtenu était celui des professions de service à l'agriculture et l'erreur était manifeste.

Le système a donc été complété par une procédure de contrôle des échos uniques, dite «contrôle par la redondance» et consistant à vérifier après la détection d'un écho unique le contenu des trois premiers bigrammes de chaque mot. Un écho unique (issu du cheminement aboutissant à un sommet de décision) est déclaré non douteux quand il existe dans la grappe des intitulés du sommet de décision au moins un intitulé possédant les mêmes bigrammes de redondance que ceux de l'intitulé à chiffrer. Il est déclaré écho douteux dans le cas contraire et par conséquent traité comme anomalie du système automatique. L'expérience a montré que cet aménagement consolidait beaucoup la fiabilité du système sans alourdir notablement les tables en mémoire ni les temps de traitement (même dans les grosses applications, le nombre de formules de redondance par sommet de décision est en moyenne de l'ordre de l'unité et dépasse rarement la dizaine).

Pour être complet, ajoutons que ce contrôle par la redondance n'est pas figé une fois pour toutes. L'utilisateur dispose de deux paramètres externes: la liste des bigrammes sur lesquels il entend exercer le contrôle, et le nombre (maximum) des bigrammes retenus. Il peut ainsi doser la sévérité du contrôle d'appariement selon ses objectifs respectifs de qualité et d'«efficacité» du chiffrement automatique.

3.3.2 Un sommet d'indécision

Le système fournit plusieurs codes possibles (le plus souvent, deux codes) et affiche leurs fréquences d'occurrence respectives au sommet considéré. C'est un cas de rejet traité manuellement par l'agent disposant du dossier de l'enquête en cours de traitement.

3.3.3 Un cas de réponse inconnue

Lorsque, au cours de l'exploration du quid, la modalité recherchée ne se trouve pas dans les modalités apprises du bigramme interrogé, la recherche échoue. C'est aussi un cas de rejet à traiter manuellement.

Les cas nouveaux rencontrés au cours d'une exploitation sont mémorisés, puis centralisés dans l'atelier d'expertise, contrôlés, enfin incorporés au FA en vue d'une nouvelle version enrichie du quid.

L'itération d'apprentissage se fait actuellement à un rythme annuel pour des raisons de commodité mais rien n'empêche de l'organiser à un rythme plus rapide pour une exploitation plus évolutive, telle que celle d'un Recensement de Population par exemple.

4. LE PROBLÈME DU TRAITEMENT DES VARIABLES ANNEXES

Dans la version actuelle QUID 1, les variables annexes sont simplement structurées en bigrammes et traitées comme des données littérales. Il en résulte des difficultés et des défauts qui nous conduisent à préparer une version QUID 2 fonctionnant à deux étages:

- au premier étage, le QUID 1 réservé au traitement de l'intitulé littéral et produisant soit le code définitif (quand il est complètement déterminé par l'intitulé), soit un code interne désignant une règle ou une table de décision opérant sur les variables annexes pour achever le calcul.
- au second étage, les règles ou tables de décision achevant la détermination du code définitif.

Examen détaillé des difficultés rencontrées

Il se trouve que certaines nomenclatures particulièrement complexes comme le code PCS (Nomenclature des Professions et des Catégories socio-professionnelles) font appel à la combinaison d'un intitulé littéral et de plusieurs variables annexes.

Par exemple, le chiffrement du code PCS utilise la variable annexe Catégorie Professionnelle (en abrégé CPF). Voici la question telle qu'elle figurait dans le bulletin individuel du Recensement de la Population de 1982:

Indiquez la catégorie professionnelle de votre emploi actuel:

	- manoeuvre ou manoeuvre spécialisé	1
- ouvrier	- ouvrier spécialisé (OS, O1, O2, O3 ...)	2
	- ouvrier qualifié (P1, P2, P3, TA, OP, OQ ...)	3
- employé		4
- technicien, dessinateur		5
	- dirigeant des ouvriers ou des techniques	6
- agent de maîtrise	- dirigeant des agents de maîtrise ou des techniciens	7
- ingénieur ou cadre		8

L'adjonction de cette question subsidiaire est rendue nécessaire par le fait que l'intitulé seul ne suffit pas toujours à classer l'individu dans la nomenclature PCS.

Par exemple un AGENT D'EXPLOITATION FORESTIÈRE

- doit être classé en 6916 (ouvriers d'exploitation forestière ou de sylviculture) si sa CPF est 1, 2, 3 ou 4
- et doit être classé en 4801 (Personnel de direction et d'encadrement des exploitations agricoles ou forestières) si sa CPF est 5, 6, 7 ou 8.

Le système actuel considère ces variables annexes comme s'il s'agissait de données littérales. Elles sont placées à la fin de l'intitulé et structurées comme lui en bigrammes (par exemple, la variable CPF complétée par un blanc est placée dans le (m + 1)ème bigramme). Mais la solution n'est pas satisfaisante et plusieurs défauts apparaissent:

Défaut n° 1. L'insuffisance du FA conduit à de nombreux cas de réponse inconnue.

Par exemple, si le FA ne comprend qu'un AGENT D'EXPLOITATION FORESTIÈRE de CPF = 2 et un autre de CPF = 7 il ne pourra pas retrouver un AGENT D'EXPLOITATION FORESTIÈRE de CPF différente de 2 ou 7 (c'est-à-dire *a priori* dans 6 cas sur 8). Le défaut est aggravé lorsque la variable annexe est très diluée comme par exemple la variable Activité Économique de l'entreprise (en abrégé variable annexe AE).

Défaut n° 2. L'insuffisance du FA conduit à des cas d'erreur.

Par exemple, si le FA comprend un seul AGENT D'EXPLOITATION FORESTIÈRE, de CPF = 2, le bigramme CPF ne discrimine plus rien et ne figurera pas dans la clé de recherche, de sorte qu'un AGENT D'EXPLOITATION FORESTIÈRE de CPF = 7 sera classé en PCS = 6916 au lieu de 4801. C'est un cas d'erreur.

Pour pallier ce défaut dans le système actuel, on ne peut qu'appliquer le contrôle de redondance aux variables annexes (et obtenir ainsi un cas douteux traité en rejet et correction manuelle au lieu d'un cas d'erreur passant inaperçu). Mais, là encore, ce n'est qu'un pis-aller. En effet, les variables annexes produisent un foisonnement anarchique dans le FA. Chaque référence du FA a sa propre combinaison croisée de modalités des variables annexes et il est peu probable de retrouver la même combinaison pour un nouvel individu à chiffrer. On aboutira donc à de nombreux cas d'échos douteux, donc à des rejets du chiffrement automatique, ce qui amoindrit le bénéfice pratique de l'exploitation de masse.

Les deux défauts n° 1 et n° 2 sont reliés à l'incomplétude relative du FA. Par exemple, il suffirait de placer dans le FA huit intitulés AGENT D'EXPLOITATION FORESTIÈRE complétés chacun par une des modalités possibles de CPF (1 à 8) pour que les deux défauts disparaissent. Mais hélas, dans les applications réelles, il se trouve que l'incomplétude relative du FA ne diminue que lentement au fur et à mesure de sa croissance jusqu'à un régime de croisière. Contrairement à l'espace lexicographique des intitulés littéraux qui, lui, tend à se densifier assez rapidement, l'espace croisé des variables annexes conserve très longtemps une vaste frontière passant lentement de la densité d'occupation 0 à la densité 1 (un individu).

Défaut n° 3. Il existe une troisième catégorie de difficultés qui tient non plus à l'incomplétude du FA mais à la sensibilité excessive de QUID par rapport aux erreurs inévitablement contenues dans le FA (et cela toujours pour ce qui concerne les variables annexes).

Prenons un exemple simplifié. Supposons que l'intitulé SECRÉTAIRE DE DIRECTION doive être chiffré PCS = 4615 (personnel de secrétariat de niveau supérieur) et ceci quelle que soit la valeur de toutes les variables annexes. Considérons le FA suivant dans lequel une erreur s'est glissé (par exemple une faute de saisie du code PCS):

Intitulé	**v.a. CPF**	**v.a. AE**	**code PCS**
Secrétaire de direction	7	49 11 (création de mode, haute couture)	4615
Secrétaire de direction	7	83 43 (coopératives de crédit)	4616 ↑ erreur

Alors que la variable annexe AE ne devrait pas servir au chiffrement du code PCS, l'algorithme QUID va s'en emparer pour séparer les deux sommets de décision.

- L'un en faveur de 4615 au vu du bigramme AE1 = 49.
- L'autre en faveur de 4616 au vu du bigramme AE1 = 83.

Le résultat est qu'au stade du chiffrement proprement dit, toutes les secrétaires de direction appartenant à d'autres branches économiques que celles commençant par 49 ou 83 sortiront en «cas de réponse inconnue''. En outre, celles de toutes les branches commençant par 83 produiront bien entendu des erreurs, mais c'est surtout le premier phénomène qui est gênant et «injuste» puisqu'il affecte un domaine bien plus large que celui de l'erreur initiale.

Défaut n° 4. Enfin, l'algorithme QUID actuel présente une rigidité excessive dans le choix de la question optimale. Le plus souvent, il en résulte une simple inversion de l'ordre des questions dans le cheminement de recherche, par rapport à l'ordre qu'aurait préféré le concepteur. L'effet est donc secondaire puisque le résultat final est identique. Mais il peut aussi se produire des distorsions plus graves.

Prenons un exemple (en partie fictif). Supposons que, selon la nomenclature, l'intitulé SECRÉTAIRE DE DIRECTION soit à chiffrer PCS = 4615 comme précédemment si la variable annexe CPF = 1 à 7, et PCS = 3726 (cadres de gestion courante des autres services administratifs des entreprises) si la CPF est égale à 8.

Considérons le FA contenant les deux références suivantes:

Intitulé	v.a. AE	v.a. CPF	code PCS
Secrétaire de direction	49 11	8	3726
Secrétaire de direction	83 43	7	4615

Ces deux références sont donc bien correctement chiffrées. L'algorithme QUID arrivant à un sommet où il a tiré tout le parti possible des bigrammes de l'intitulé littéral, doit choisir maintenant un bigramme dans les variables annexes afin de séparer les deux issues finales PCS = 3726 et PCS = 4615. Dans notre exemple simple mais non dénué de réalisme, les trois bigrammes candidats AE1, AE2 et CPF apportent la même quantité d'information (un bit). La convention arbitraire prise dans notre algorithme est qu'en cas d'égalité, il choisit la première question dans l'ordre des déclarations des variables annexes. Ce qui dans notre exemple est fâcheux puisqu'on retombe dans l'aberration vue plus haut (défaut n° 3). Or, on ne peut pas trouver un bon ordre des variables annexes qui éviterait ce défaut dans tous les cas de figure. On ne peut que chercher un ordre de déclaration statistiquement le moins mauvais (en tâtonnant à partir de l'ordre des clivages conceptuels, de la capacité néguentropique de chaque variable annexe, *etc* ...)

5. CONCLUSION

Le système QUID dans sa version QUID 1 actuelle rend de précieux services à l'INSEE mais présente encore des points faibles dans le traitement des variables annexes.

La nouvelle version QUID 2 devrait améliorer ce traitement tout en restant fidèle à notre «approche de base» du problème de la codification automatique, que l'on peut résumer en deux points:

1. Séparation de la base d'apprentissage (ici, base de règles et de tables de décision écrites en clair, indépendantes les unes des autres, apurées et gérées par un atelier d'expertise autonome) et des logiciels de chiffrement automatique (ici, logiciels de chargement et d'exploration des tables).
2. Construction de logiciels généraux, c'est-à-dire indépendants du champ sémantique traité.

C'est du moins l'objectif que nous essaierons de maintenir.

REMERCIEMENTS

Je remercie les arbitres pour leur aide précieuse dans la rédaction de cet article.

BIBLIOGRAPHIE

BOUCHON-MEUNIER, B. (1978). Sur la réalisation de questionnaires. Thèse d'État, Paris.

KNAUS, R. (1987). Methods and problems in coding natural language survey data, *Journal of Official Statistics*, 3, 45-67.

LORIGNY, J. (1982). Mesures d'entropie et d'information pour les systèmes ouverts complexes. Thèse d'État, Paris.

LORIGNY, J. (1985). Manuel d'utilisation du système QUID. Institut National de la Statistique et des Études Économiques, Direction de la production, Paris.

PICARD, C.-F. (1972). *Graphes et Questionnaires.* Paris: Gauthier-Villars.

SHANNON, C.E. (1948). A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27, 379-423, 623-656.

TERRENOIRE, M. (1970). Un modèle mathématique de processus d'interrogation: les pseudo-questionnaires. Thèse d'État, Grenoble.

TOUNISSOUX, D. (1980). Processus séquentiels adaptifs de reconnaissance de formes pour l'aide au diagnostic. Thèse d'État, Lyon.