Traduction automatisée fondée sur le dialogue et documents auto-explicatifs : bilan du projet LIDIA

Hervé Blanchon[*], Christian Boitet[*], Ali Choumane[**]

[*] *Laboratoire LIG, Équipe GETALP, BP 53, 38041 Grenoble Cedex 9*
[**] *Laboratoire IRISA, Projet Cordial, ENSSAT, BP 80518, 22305 Lannion Cedex*
herve.blanchon@imag.fr, christian.boitet@imag.fr, ali.choumane@irisa.fr

RÉSUMÉ. *Nous dressons un bilan des travaux que nous avons conduits dans le cadre du projet LIDIA de traduction automatisée fondée sur le dialogue pour auteur monolingue. En mettant en œuvre une architecture linguistique à transfert multiniveau, nous avons proposé et évalué une méthodologie de production de questions de désambiguïsation interactive. Les modules mis en œuvre coopèrent au sein d'une architecture distribuée en utilisant un environnement de rédaction « léger ». Nous avons aussi travaillé sur le concept de Document Auto-Explicatif. Un DAE est un document enrichi des réponses fournies par l'auteur lors de l'étape de désambiguïsation interactive, et qui donne aux lecteurs, sur demande, des explications sur la façon de le comprendre de façon à éviter des incompréhensions dues aux ambiguïtés.*

ABSTRACT. *We present the work we carried out on the LIDIA project in the framework of dialogue-based machine translation for the monolingual author. Using a multilevel transfer linguistic approach, we have proposed and evaluated a technique to produce interactive disambiguation questions. The components involved in the translation process cooperate within a distributed architecture through a "light" document processing environment. We have also proposed the idea of Self-Explaining Documents. A SED is a document enriched with the answers provided by the author during interactive disambiguation. It gives readers, on demand, explanations about its intended meaning, in order to avoid any misunderstanding due to ambiguities.*

MOTS-CLÉS : *traduction automatique, traduction interactive fondée sur le dialogue, désambiguïsation interactive, document auto-explicatif.*

KEYWORDS : *Machine Translation, Dialogue-Based Machine Translation, Interactive Disambiguation, Self-Explaining Document.*

TAL. Volume 47 – n° 3/2006, pages 175 à 204

1. Introduction

La Traduction Automatisée Fondée sur le Dialogue (TAFD) vise à permettre, à l'auteur monolingue d'un document, la production d'une traduction de haute qualité à partir de son environnement de rédaction. Dans ce cadre, l'auteur aide le système à traduire vers une ou plusieurs langues cibles, *via* une standardisation (normalisation) et une clarification (désambiguïsation) interactives effectuées en langue source, et en une seule fois, quel que soit l'ensemble des langues cibles visées. Le processus de désambiguïsation interactive joue un rôle crucial. Il s'agit de détecter les ambiguïtés que le module d'analyse du système de TA n'a pas pu résoudre automatiquement afin de produire des questions n'exigeant aucune compétence particulière.

Au cours du développement de notre première maquette, nous avons été conduits à l'idée que les informations obtenues par le système lors de la phase de désambiguïsation interactive pourraient être conservées afin d'enrichir le document avec le sens qu'il véhicule. Un Document Auto-Explicatif (DAE) contient le texte du document ainsi qu'une mémoire des ambiguïtés qu'il contient et une trace du processus de désambiguïsation. Un lecteur de DAE peut « cliquer » sur un segment textuel marqué comme ambigu et obtenir une présentation des différentes interprétations possibles avec un marquage de celle qui doit être retenue.

Nous exposons d'abord notre vision de la TAFD pour auteur monolingue, puis nous montrons comment cette vision a été mise en œuvre, au niveau de l'architecture linguistique, puis de l'architecture informatique. Nous présentons ensuite en détail nos idées sur la désambiguïsation interactive, leur implémentation et le résultat d'une expérience visant à évaluer la compréhensibilité des questions que nous savons produire. Finalement, nous proposons le concept de DAE et détaillons sa mise en œuvre avec la maquette LIDIA-3.

2. Le projet LIDIA

Nous avons proposé le concept de TAFD pour auteur monolingue en 1990 (Boitet, 1990), après avoir travaillé longtemps avec B. Vauquois sur l'approche par « transfert multiniveau » optimisée pour des sous-langages, qui a produit des résultats de très haute qualité. Le défi était de garder ou d'augmenter la qualité tout en « ouvrant » à tous les types de textes, et le pari était qu'on pouvait y arriver en modernisant l'idée déjà ancienne de TA interactive, qui avait buté sur des problèmes d'ergonomie et de coût des experts humains alors nécessaires.

2.1. *Un contexte favorable*

Depuis 1990, le contexte incite de plus en plus à faire progresser la TAFD, pour au moins trois raisons : la limitation des paradigmes actuels de la TA, l'importance croissante des langues nationales dans le contexte de l'internationalisation, et les récents progrès méthodologiques et technologiques.

2.1.1. Limitation des paradigmes actuels

La TA Fondée sur les connaissances Linguistiques (TAFL) est très bien adaptée à la TAO du veilleur. En revanche, elle est loin de pouvoir répondre à tous les besoins en TAO du réviseur. Outre le fait qu'elle demande évidemment autant de révisions que de langues cibles, elle reste trop chère pour des usages légers. D'autre part, une condition essentielle de succès de la TAO du réviseur est de constituer une équipe de développement et de maintenance des linguiciels (dictionnaires, grammaires) qui soit en liaison constante avec l'équipe de révision, et si possible avec les auteurs des documents à traduire.

En ce qui concerne la TA Fondée sur la Connaissance (TAFC) (Nirenburg, 1989), elle est totalement inapplicable en TAO du veilleur, et elle est moins applicable que la TAFL en TAO du réviseur, car tant qu'il faudra construire les « ontologies » spécialement pour la traduction, elle restera beaucoup plus onéreuse, et ce pour un résultat guère meilleur.

Enfin, la TA Fondée sur les Statistiques (TAFS) exige pour l'instant des corpus parallèles gigantesques (50 M mots, ou 200 K pages d'après K. Knight à CICLING-05) pour obtenir une qualité « utilisable, très inférieure à celle de la TAFL spécialisée ».

2.1.2. Importance croissante des langues nationales et de l'internationalisation

Contrairement à ce que d'aucuns prédisaient il y a une cinquantaine d'années, l'internationalisation croissante ne s'est pas accompagnée d'une uniformisation linguistique vers l'anglais, mais au contraire d'un renforcement considérable ou de l'émergence de langues comme le japonais, le chinois, le français, l'espagnol, le malais-indonésien ou l'arabe, etc. La plupart des langues écrites sont « dans Unicode » et « sur le Web ». Les logiciels commerciaux et leur documentation sont localisés en vingt-cinq à quarante langues, et les logiciels libres en plus de soixante-dix langues. Wikipedia existe en plus de cent langues, et cette tendance croît.

Il ne s'agit pas seulement de politique et de culture, mais d'efficacité. Dans les projets coopératifs européens, par exemple, la communication est gênée par la nécessité de lire et d'écrire en anglais. Pour la grande majorité des participants, lire en anglais pose des problèmes de compréhension et prend beaucoup de temps. Quant à écrire, c'est encore plus ardu, voire impossible pour certains, et le résultat est souvent difficile à comprendre, voire illisible. En 1990, de grandes sociétés comme Thomson chiffraient entre 4 et 5 M€ par an les pertes dues aux problèmes de communication multilingue.

Les trois types de TAO « classique » ne peuvent évidemment répondre à ce nouveau besoin. En effet, la TAO du veilleur, sans préédition ni postédition, ne peut donner une qualité suffisante, et la TAO du réviseur comme la TAO du traducteur s'adressent par définition à des spécialistes au moins bilingues, et non à des rédacteurs supposés ne connaître aucune des langues cibles, ou au plus une, et ce imparfaitement.

2.1.3. Progrès méthodologiques et technologiques

L'idée de la TAFD date des années 60 (Kay, 1973), et a été incorporée à plusieurs maquettes ou prototypes dans les années 70 et 80 à l'université Brigham Young (Provo, Utha) (Melby, 1982 ; Weaver, 1988), à l'université Carnegie Mellon (Pittsburgh, Pensylvanie) (Tomita, 1986), et dans les Universités de Manchester et Sheffield (Angleterre) (Wood, 1989 ; Wood *et al.*, 1988). Si ces travaux n'ont pas donné lieu à des systèmes utilisables en pratique, c'est à notre avis que les dialogues devaient être conduits par des spécialistes, que la couverture linguistique était trop limitée, et que l'on ne disposait pas encore d'environnements interactifs conviviaux.

La méthodologie s'est ensuite affinée. Tout d'abord, l'utilisateur envisagé n'était plus un spécialiste, mais un rédacteur, ou plutôt un *auteur* (Huang, 1990 ; Maruyama *et al.*, 1990 ; Somers *et al.*, 1992 ; Wehrli, 1993 ; Witkam, 1983). Nous préférons ce dernier terme, car « auteur » désigne quelqu'un qui désire créer un produit final « propre »[1], alors que les autres termes (« rédacteur », « locuteur », ou « commentateur ») peuvent renvoyer à des personnes désirant seulement produire un message écrit ou parlé de façon « spontanée », en vue d'une communication immédiate, et non disposées à participer à un dialogue avec la machine, éventuellement lourd, pour rendre leur message « propre ».

D'autre part, l'informatique personnelle a fait des progrès gigantesques. On dispose maintenant d'ordinateurs personnels très puissants et bon marché, d'environnements conviviaux, de l'intégration du multimédia, et d'outils de télécommunication permettant le recours à des serveurs. Enfin, les techniques et outils de génie logiciel modernes permettent de construire des systèmes complexes et interactifs bien plus rapidement et sûrement que par le passé.

2.2. *Critères de choix de la TAFD*

Nous proposons quatre critères pour choisir la TAFD. (1) Il faut d'abord que la qualité visée pour le document cible soit élevée, et que la révision soit impossible ou très coûteuse. (2) Le contexte doit être fortement multilingue (1→n, comme pour la dissémination de documentation technique, ou n↔n, comme dans des projets internationaux). (3) La forme du texte source ou son domaine ne doivent pas être trop contraints ou contrôlés (sinon, mieux vaut utiliser la TAFL ou la TAFC). (4) Les utilisateurs doivent être prêts à participer à des dialogues de normalisation et de désambiguïsation.

Il faut de plus pouvoir rendre acceptables les dialogues de standardisation et de désambiguïsation, en les laissant à l'initiative de l'utilisateur, en lui fournissant des moyens de les contrôler et de les réduire (en jouant sur des paramètres, en insérant

1. « Propre » signifie ici conforme à une certaine grammaire (permettant éventuellement des constructions incorrectes, pourvu qu'elles soient attestées) *et* ne comportant pas de parties « réflexives », ou « automodificatrices », si fréquentes dans la parole et même dans l'écriture spontanée (au moins manuscrite), comme des hésitations, des faux départs, des reprises, des répétitions, des corrections, des abréviations arbitraires (apocopes), etc.

directement des marques de désambiguïsation, etc.), et en lui laissant si possible le choix entre plusieurs médias.

2.3. *Situations traductionnelles adaptées à la TAFD*

2.3.1. Entrée écrite

Parmi les situations favorables avec entrée écrite, on peut en mentionner trois : la traduction de volumes relativement faibles de documentation technique en plusieurs langues, typiquement 5 000 à 8 000 pages à distribuer sur CD/DVD ou téléchargeables, par exemple dans les vingt langues officielles de l'UE (et peut-être dans d'autres aussi, comme le russe, l'arabe, le japonais, ou le chinois), la diffusion d'informations dans plusieurs langues (sur la circulation, sur la météo, ou dans des congrès, des manifestations sportives, des situations d'urgence…), qui demande une sortie orale aussi bien qu'écrite, et l'échange télématique de notes et de documents de travail dans des projets internationaux.

2.3.2. Entrée orale

Comme les reconnaisseurs de parole arrivent maintenant à traiter à la fois un grand vocabulaire, de la parole spontanée, et un locuteur arbitraire, mais avec un taux d'erreur très important (WER=30 %) en contexte réel, la TA généraliste de l'oral ne peut être que la TAFD. C'est l'approche suivie depuis quelques années par Spoken Translation Inc., qui commercialise depuis 2006 Converser for HealthCare, destiné aux professionnels de la santé anglophones devant dialoguer avec des patients hispanophones aux USA.

2.3.3. Création dans une langue d'un message dans une autre

Il y a enfin beaucoup de situations intéressantes où le message source n'est créé que pour vérifier le contenu du (ou des) message(s) cible(s). Le message est créé par interaction avec le système. C'est, par exemple, le cas de lettres officielles ou formelles, qui ont des structures très différentes dans différentes cultures. (Somers *et al.*, 1990) proposent le terme de « traduction sans texte source », mais il serait plus exact de parler de génération interactive multilingue que de traduction.

3. Architecture linguistique et voies intermédiaires nouvelles

Le fait que la TAFD suppose un dialogue avec l'auteur permet d'envisager de nouvelles possibilités au niveau des traitements linguistiques. Il ne s'agit pas de solutions radicalement nouvelles, mais comme souvent dans un domaine technique, de nouveaux compromis, de voies intermédiaires nouvelles, avec çà et là des innovations intéressantes.

Dans la majorité des situations adaptées à la TAFD, il faut un système de couverture lexicale et grammaticale très large. D'où des questions théoriques importantes :

- Sachant que l'on n'obtient de bons résultats en « tout automatique » que sur des langages restreints, comment construire une base de connaissances linguistiques utilisable comme une union de sous-langages ? Est-il possible de séparer les aspects grammaticaux et lexicaux ?
- *Comment atteindre la large couverture lexicale nécessaire ?* Un système de TAFL contient typiquement de 3 10^4 à 3 10^5 termes, en deux langues. Mais un système de TAFD visant le grand public et non restreint à un domaine particulier demanderait de 3 10^5 à 3 10^6 termes, et ce en plusieurs langues !
- Dans des situations fortement multilingues, l'approche par interlingua est séduisante. Mais, *comment surmonter les difficultés d'ingénierie rencontrées dans la construction d'un grand lexique interlingue ?* Des études sur l'applicabilité d'UNL à la TA de la parole sont en cours, mais ce problème de taille du lexique n'est pas encore résolu.
- Il est crucial que des non-spécialistes puissent facilement comprendre les questions du système, éventuellement lui demander les raisons de certaines questions, et comprendre ses réponses. Une question importante (et nouvelle) est donc de trouver *comment rendre la base de connaissances linguistiques d'un système de TAFD accessible à un utilisateur naïf.*

Bien que les réponses proposées ici (transfert multiniveau à acceptions, propriétés et relations interlingues ; encodage d'informations linguistiques lors de l'édition) ne soient ni complètes ni définitives, elles sont basées sur une longue pratique de la TAFL et sur une étude approfondie de beaucoup de systèmes de TA passés ou présents, en particulier ceux qui utilisent une étape de désambiguïsation interactive.

3.1. *Transfert multiniveau à acceptions, propriétés et relations interlingues*

Le *transfert multiniveau*, au sens de Vauquois (Vauquois *et al.*, 1985) diffère du transfert sémantique en ce que, outre les attributs et relations interlingues, on garde des attributs et des relations spécifiques à la langue source (classe syntagmatique, genre, nombre, détermination, temps, mode, fonction syntaxique…).

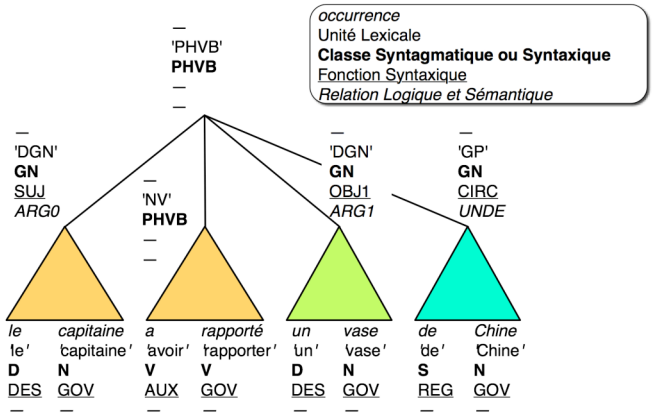

Figure 1. *Analyse multiniveau de «Le capitaine a rapporté un vase de Chine. »*

Cela donne des « filets de sécurité », et permet de faire interagir les niveaux. Une structure multiniveau (figure 1) comporte trois niveaux de représentation linguistique : le niveau des classes syntaxiques et syntagmatiques, le niveau des fonctions syntaxiques, et le niveau des relations logiques et sémantiques.

Nous proposons de rajouter un niveau lexical, celui des *acceptions interlingues* (permettant de désambiguïser les polysémies), sans aller jusqu'à introduire des concepts, puisqu'il faudrait alors construire une ontologie. La base lexicale multilingue sous-jacente contiendra alors un dictionnaire monolingue pour chaque langue traitée par le système, et un dictionnaire interlingue pour les acceptions interlingues. Chaque acception interlingue a une image dans chaque dictionnaire monolingue, avec une définition appropriée dans la langue correspondante, utilisée lors d'une question de désambiguïsation interactive du sens[2].

Interlingue ne signifie pas « indépendant des langues », mais « intermédiaire entre les langues connues du système ». Par exemple, si le système travaille avec le français, l'anglais et le russe, il y aura une seule acception pour « mur » en tant qu'objet concret. Dès qu'on ajoutera l'allemand ou l'italien, il faudra ajouter les raffinements « mur – volume» (Mauer, muro) et « mur – paroi » (Wand, parete).

3.2. *Encodage d'informations linguistiques lors de l'édition*

L'auteur dispose d'un système d'annotations qui lui permet d'encoder des informations linguisitques lors de l'édition. Ce système d'annotations doit concerner tous les niveaux de description linguistique et être incomplet à chaque niveau, car aucune notion non familière ne doit apparaître. Ainsi, « verbe » est une notion familière pour presque tout adulte instruit, mais pas « verbe modal ». Au niveau des fonctions syntaxiques, « sujet », « objet » et « complément » sont familiers, mais sans doute pas « attribut », « épithète », « tête » (ou « gouverneur »). Il en va de même au niveau des cas profonds. Prenons par exemple les trois phrases suivantes.

> Un processus de traduction en ARIANE-G5 se compose d'une suite de trois étapes (analyse, transfert et génération). Chaque étape est constituée d'une suite de différentes phases de traitement. Chaque phase est relative à l'emploi d'un LSPL précis.

Figure 2. *Texte édité*

Voici une vue du texte de la figure 2 si l'on fait une annotation complète.

> { { Un.&art processus.&n,suj { de.&prep traduction.&n,comp { en.&prep ARIANE-G5.&np,comp } } } se.&refl compose.&v,phvb { d'.&prep une.&art suite.&n,obj1 { de.&prep trois.&card étapes.&n,comp { (.&lp analyse.&n,app ,.&ponc transfert.&n,coord et.&cjcoord génération.&n,coord).&rp } } } .&ponc } { { Chaque.&art étape.&n,suj }

2. On peut aussi *expliquer* à l'auteur pourquoi une question de désambiguïsation interactive du sens est posée, et même montrer les mots en question dans les autres langues. L'introduction d'aspects d'auto-apprentissage dans ce genre de systèmes les rendrait plus acceptables par les utilisateurs potentiels.

est.&v,aux constituée.&v,phvb { d'.&prep une.&art suite.&n,comp { de.&prep { différentes.&adj,epit } phases.&n,comp { de.&prep traitement.&n,comp } } } ..&ponc } { { Chaque.&art phase.&n,suj } est.&v,phvb { relative.&adj,atsubj { à.&prep l'.&art emploi.&n,obj1 { d'.&prep un.&art LSPL.&np,comp { précis.&adj,epit } } } } ..&ponc }

Figure 3. *Texte édité annoté*

3.3. *Étapes du processus de traduction*

Une phrase du texte en langue source est analysée pour produire une structure *mmc-source* (multisolution, multiniveau, concrète[3]). Cette structure *mmc* (figure 4) est alors utilisée pour construire un arbre des questions qui seront posées à l'auteur.

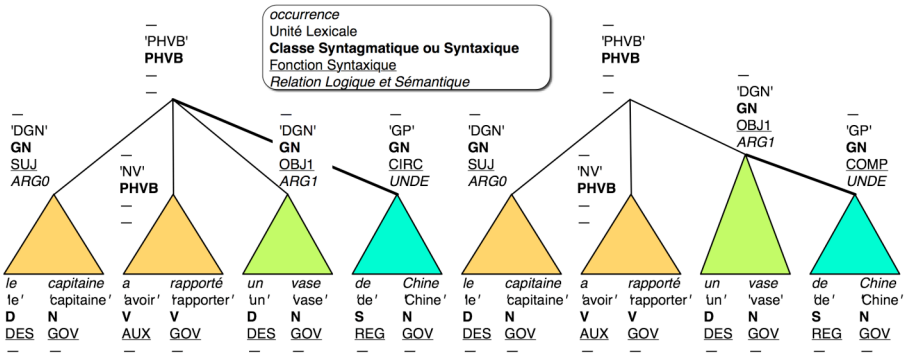

Figure 4. *Structure mmc de la phrase «Le capitaine a rapporté un vase de Chine. »*

À l'issue de l'étape de désambiguïsation interactive, le système obtient la structure *umc-source* (unisolution, multiniveau, concrète) non ambiguë choisie par l'auteur. Cette structure *umc* est ensuite transformée en une structure *uma-source* (unisolution, multiniveau, abstraite[3]).

Une étape de transfert lexical et structural produit maintenant une structure *gma-cible* (génératrice, multiniveau, abstraite). Une structure *gma* est plus générale et génératrice qu'une structure *uma* car les niveaux de surface (fonctions syntaxiques, catégories syntagmatiques…) peuvent ne pas être renseignés. Dans ce cas, ce sont des préférences du transfert qui les instancieront.

L'étape de sélection de paraphrase produit une structure *uma-cible* qui est homogène à la structure qui serait produite en analysant puis en désambiguïsant interactivement le texte cible qui va être généré. Le processus de traduction se termine avec les générations syntaxique et morphologique.

3. Une représentation « concrète » d'un texte est telle qu'on retrouve le texte représenté grâce à un parcours canonique de la structure (mot des feuilles pour un arbre de constituants, parcours infixé pour un arbre de dépendances). Sinon, la structure est dite « abstraite ».

Les parties grisées de la figure 5 montrent le diagramme fonctionnel des processus que nous venons de décrire.

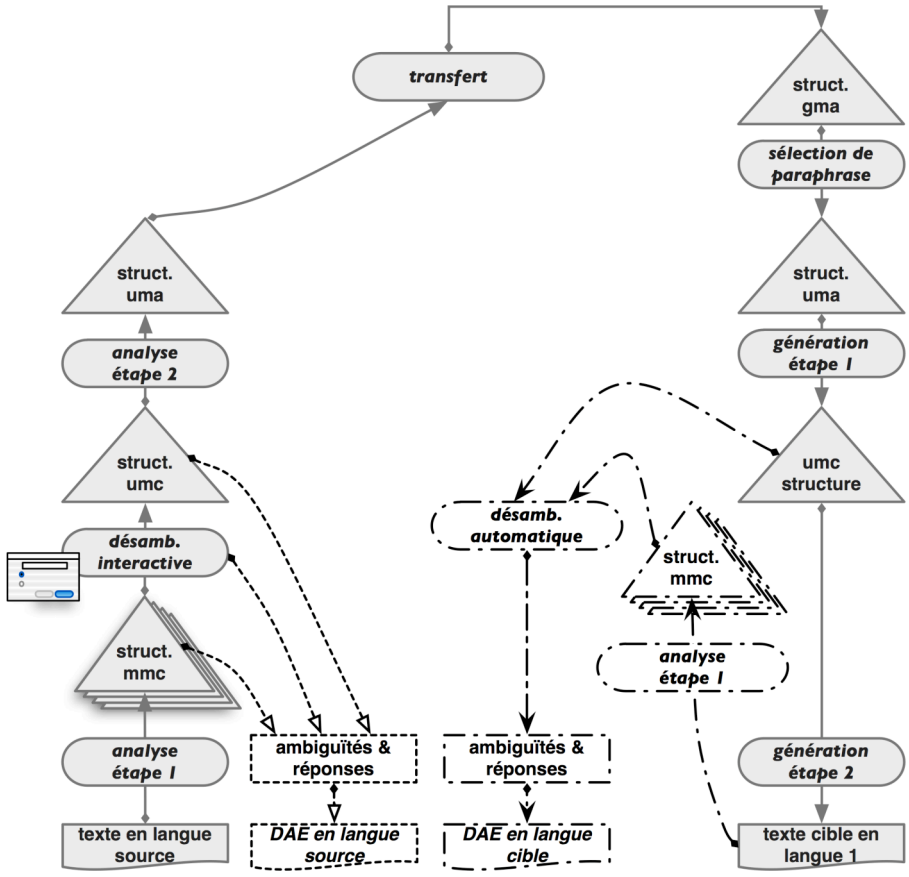

Figure 5. *Architecture linguicielle de LIDIA (en blanc, la construction de DAE)*

4. Architectures informatiques

L'architecture informatique d'un système est intimement liée à des contraintes ergonomiques que l'on se fixe pour celui-ci.

4.1. *Aspects ergonomiques*

En TAFD, l'ergonomie est un aspect crucial, et les choix ergonomiques influencent directement l'architecture de tout le système. Les choix principaux sont les suivants. Le système doit-il tourner sur des ordinateurs personnels bon marché ? Doit-il fonctionner en temps réel, ou l'asynchronisme est-il préférable ? Une architecture avec serveur(s) est-elle possible ? Comment les dialogues doivent-ils

être organisés ? Est-il nécessaire et/ou possible de les conduire dans un environnement multimédia et multimodal ? Par exemple, peut-on améliorer l'efficacité et la convivialité des dialogues de désambiguïsation grâce à l'utilisation de synthèse de la parole et d'interactions graphiques ?

Si l'on vise vraiment la « TAO pour tous », il est impératif de pouvoir utiliser les systèmes de TAFD sur des ordinateurs personnels, ou des stations de travail de bas de gamme. Pour les autres questions, nous répondons : asynchronisme, distribution, interaction non préemptive. Enfin, certains des choix faits en 1990 et décrits ici ne seraient plus justifiés en 2006, ou plus comme en 1990. En particulier, les PDA actuels supportent des systèmes de TA de la parole autonomes, mais les opérateurs de téléphonie mobile favorisent évidemment une approche distribuée.

4.1.1. Asynchronisme

Nous avons choisi une organisation asynchrone pour des raisons ergonomiques et pratiques. D'abord, une organisation en temps réel est nécessaire seulement si nous souhaitons que l'utilisateur réponde immédiatement aux questions posées par le système, ce que nous ne voulons pas. Enfin, si l'on prend en compte le degré de complexité inhérent à n'importe quel système de TAFD général et de bonne qualité, nous ne pouvons espérer ni une exécution en temps réel, ni une implémentation des ressources nécessaires (base lexicale, linguiciel) sur le genre de matériel envisagé.

4.1.2. Architecture distribuée

Une exécution en temps réel n'est pas non plus possible dans le cadre de l'architecture distribuée que nous préconisons, car (a) l'utilisation d'un serveur puissant pour prendre en charge tout ou partie des traitements non interactifs de traduction est préférable au stockage de gros systèmes sur chaque station de travail, même si on envisage une exécution en tâche de fond, et (b) on peut ainsi mieux résoudre les problèmes de maintenance des ressources linguistiques. Les utilisateurs peuvent alors bénéficier de manière transparente de mises à jour des serveurs d'analyse, de désambiguïsation, de transfert et de génération utilisés.

4.1.3. Désambiguïsation interactive multimodale non préemptive

Il nous paraît essentiel de permettre à l'auteur de décider quand il souhaite entamer le dialogue avec le système. En d'autres termes, le système propose et l'utilisateur dispose. En effet, les précédents essais en TAFD nous semblent avoir échoué en partie à cause du caractère modal des dialogues de désambiguïsation.

4.2. *Mises en œuvre*

La situation que nous avions considérée à l'origine du projet LIDIA est la production de documents multilingues sous la forme de documents HyperCard. HyperCard est un environnement de production de documents hypertextes, appelés « piles », dont les pages sont des « cartes ». Les cartes contiennent différents types d'objets, dont des champs textuels. La maquette LIDIA-1 permet de traduire vers

l'anglais, l'allemand et le russe une pile en français qui présente, en contexte, des phrases comportant les ambiguïtés que nous avions choisi de traiter[4].

La maquette LIDIA-1.mail (Boitet *et al.*, 1995a) est composée de trois modules : le module de rédaction basé sur HyperCard, le module de traduction qui fait appel à ARIANE-G5 *via* le réseau en utilisant le protocole SMTP (courriel), et enfin le module de désambiguïsation. Ils communiquent par AppleEvents, à travers des interfaces de communication : client de rédaction, serveur de désambiguïsation, serveur de communication. L'ordonnancement des traitements est assuré par un coordinateur.

Le bilan critique de l'architecture informatique des maquettes LIDIA-1 effectué en 2000 a fait apparaître différents problèmes. (1) L'environnement de rédaction, HyperCard, qui avait connu un grand succès, tombait en désuétude. Cela interdisait donc de diffuser largement notre interface d'accès aux services LIDIA. (2) La communication entre les modules, par AppleEvents, empêchait de faire coopérer des modules développés sur des plates-formes autres que Macintosh, et de déporter le module de désambiguïsation sur une machine située sur un sous-réseau distinct de celui de l'environnement de rédaction. (3) L'utilisation du courriel pour transmettre commandes et données entre le serveur de rédaction et le serveur de traduction était trop lourde. (4) Aucun résultat intermédiaire de traitement n'était conservé.

Dans la nouvelle architecture définie en 2000 (figure 6), tous les modules logiciels qui interviennent dans la chaîne de traitement LIDIA communiquent avec le protocole Telnet à travers un serveur de communication. Il y a :

- un client LIDIA qui sert d'environnement de rédaction et de supervision de la traduction des documents ;
- un serveur de traduction qui assure les traitements d'analyse et de génération ;
- un serveur de désambiguïsation.

La traduction d'un segment de document source se fait comme suit :

1) Le client LIDIA envoie un message de demande d'analyse au serveur de communication qui le transmet au serveur de traduction.
2) Le serveur de traduction traite cette demande, et produit une *structure mmc,* transmise au serveur de désambiguïsation via le serveur de communication.
3) Le serveur de désambiguïsation produit un *arbre de questions* (vide ou non) qui est transmis au client LIDIA via le serveur de communication.
4) S'il y a des questions, le client LIDIA le signale à l'auteur. Lorsque l'auteur a répondu, ou s'il n'y a pas de question, le client LIDIA transmet une demande de traduction au serveur de traduction, *via* le serveur de communication.
5) Le serveur de traduction traite la demande de traduction, et produit un résultat qui est transmis au client LIDIA.

---

4. Les exemples choisis sont « construits » pour faciliter la compréhension des ambiguïtés. Il ne faut cependant pas s'arrêter aux mots, ce sont bien les structures *mmc* sous-jacentes qui sont pertinentes pour le module de désambiguïsation interactive. De plus, quand on extrait d'un texte quelconque une phrase ambiguë, elle paraît toujours « construite ». Par exemple : « je ne vous ai pas écrit exprès. »

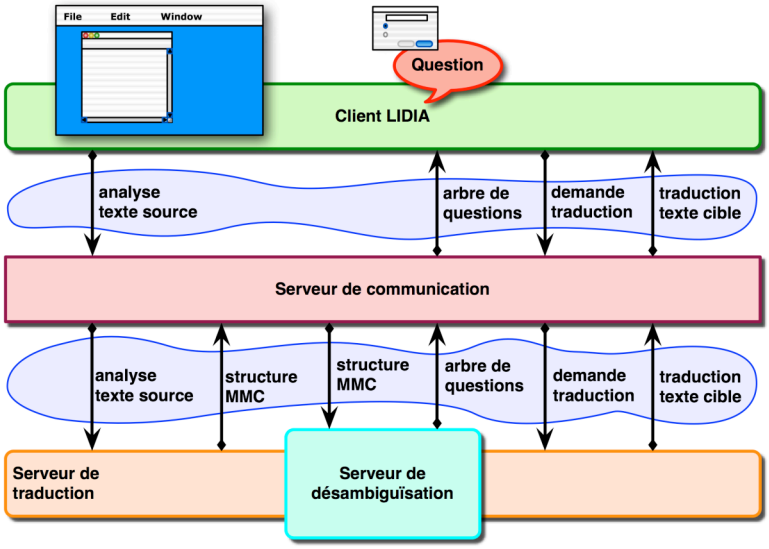

Figure 6. *Architecture informatique mise en œuvre depuis LIDIA-2*

LIDIA-2 (Blanchon *et al.*, 2004) est une version en Java de l'environnement d'accès aux services de TAFD. L'environnement d'accès aux services LIDIA est un microéditeur de documents XML très limité. Le document produit contient le texte rédigé par l'auteur et l'historique de la désambiguïsation interactive. Dans cette version, nous avons utilisé le désambiguïseur de l'anglais (Blanchon, 1995) directement utilisable dans la nouvelle architecture (décrite ci-dessus).

5. Désambiguïsation interactive

5.1. *Proposition*

Dans l'architecture que nous avons proposée pour LIDIA-2[5] (Blanchon, 1995), un module de désambiguïsation est constitué de deux parties : un moteur de désambiguïsation et un linguiciel de désambiguïsation. Le moteur est le noyau du module, il est indépendant de la langue et du corpus à traiter, il est utilisé pour tous les modules de désambiguïsation à développer. En revanche, le linguiciel dépend de la langue et des ambiguïtés présentes dans le corpus à traiter.

Le moteur est associé à différents linguiciels pour composer des modules de désambiguïsation particuliers. Une instance de module de désambiguïsation propre à une application donnée est l'association d'un linguiciel particulier et du moteur.

5. Dans le cadre de la réalisation de la maquette LIDIA-1, la première version du module de désambiguïsation pour le français (Blanchon, 1992) était monolithique.

Idéalement, nous devons fournir au concepteur d'un module de désambiguïsation un ensemble d'outils lui permettant de décrire :

- les types d'ambiguïtés à résoudre sur la base de la représentation *mmc* ;
- la formation des items de la question qui permettront leur résolution ;
- la métaprésentation des questions (invite, mise en relief du segment ambigu, en-tête de la liste des items) ;
- la stratégie de découverte des ambiguïtés, si plusieurs sont présentes ;
- la modalité (écrit ou oral) à utiliser pour résoudre les ambiguïtés ;
- le mode de préparation des questions (toutes à la fois, une par une).

Ces descriptions forment le linguiciel.

Le moteur utilise le linguiciel pour réaliser le processus de désambiguïsation interactive. Il doit donc fournir les services suivants :

- une méthode de reconnaissance des ambiguïtés décrites dans le linguiciel ;
- un ensemble d'opérateurs pour décrire la formation des items de question ;
- un langage de description d'automates pour décrire l'ordre dans lequel les ambiguïtés seront résolues ;
- des classes de dialogue correspondant aux différentes modalités proposées ;
- des méthodes de présentation des questions ;
- différentes stratégies de préparation et de présentation des questions.

5.2. *Mise en œuvre*

Dans le cadre du projet LIDIA, l'analyseur produit une structure d'arbre décoré (figure 4 supra). Les ambiguïtés sont décrites en termes de propriétés de cette structure en utilisant des patrons d'arbres. Ces descriptions ne sont donc pas dépendantes d'un domaine ou d'un corpus d'application particulier.

5.2.1. Définition d'un type d'ambiguïté

Si une phrase est ambiguë, l'analyseur produit une solution pour chacune des interprétations possibles. Il est heureusement possible de distinguer des familles de solutions qui, pour un même segment du texte, auront des représentations différentes.

Pour « Le capitaine a rapporté un vase de Chine. » (figure 4 supra), l'analyseur propose deux solutions. Cette ambiguïté est décrite et généralisée par le couple de patrons suivant :

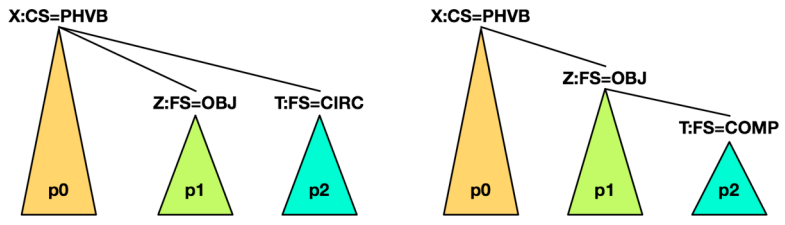

Figure 7. *Faisceau de patrons définissant le type d'ambiguïté (structure argumentaire du verbe) présent dans la phrase « Le capitaine a rapporté un vase de Chine. »*

Un patron décrit une famille d'arbres avec des variables de nœud et des variables d'arbre. Les variables de nœud sont ici **X**, **Z** et **T** ; les variables d'arbre sont **p0**, **p1** et **p2**. Les variables de nœud sont contraintes : la classe syntagmatique de **X** doit être **PHVB**, la fonction syntaxique de **Z** doit être **OBJ**, et la fonction syntaxique de **T** doit être **CIRC** pour le patron de gauche et **COMP** pour le patron de droite. La valeur (le contenu) des variables d'arbre n'est pas contrainte dans la définition du patron.

Un tel ensemble de patrons est appelé **faisceau** dans notre terminologie. Les patrons d'un faisceau partagent un ensemble de variables de nœud et de variables d'arbre. Un type d'ambiguïté d'une classe d'ambiguïtés particulière est défini par la cooccurrence des patrons d'un certain faisceau parmi les différentes solutions produites par l'analyseur. Formellement, une phrase **S** pour laquelle l'analyseur produit **s** solutions $\mathbf{Sol}_i$ présente l'ambiguïté décrite par le faisceau **B** constitué de **b** patrons $\mathbf{P}_j$ si et seulement si :

- le nombre de solutions (**s**) est supérieur ou égal au nombre de patrons (**b**) ;
- chaque solution $\mathbf{Sol}_i$, s'apparie avec un unique patron $\mathbf{P}_j$;
- chaque patron $\mathbf{P}_j$ s'apparie avec au moins une solution $\mathbf{Sol}_i$;
- la distance **fd** entre les valeurs de chacune des variables d'arbre pour les différents appariements est nulle.

La distance **fd** entre deux valeurs de variables d'arbre est nulle, si et seulement si : (1) pour toute variable d'arbre, exceptée la dernière, la projection des occurrences couvertes par la variable est la même dans les différents appariements, et (2) pour la dernière variable d'arbre, la projection des occurrences du texte analysé couvertes par la variable est la même dans les différents appariements, ou sinon, il existe un préfixe commun entre les différentes projections.

5.2.2. Production d'une question relative à un type d'ambiguïté

Une méthode de rephrasage est associée à chaque patron pour produire un item de question. Un rephrasage est obtenu par des manipulations de surface sur les variables d'arbre du patron instanciées lors de l'appariement. Ces manipulations sont décrites au moyen d'un ensemble d'opérateurs qui permettent, par exemple, de projeter le segment de texte couvert par une une variable d'arbre (`Text(px)`), de projeter la conjonction de coordination d'une variable d'arbre (`Coord(py)`), de projeter le texte d'une variable d'arbre privé de la conjonction de coordination (`But_coord(py)`), de projeter l'article d'une variable d'arbre (`Determiner(py)`), de remplacer une préposition ambiguë par une préposition non ambiguë dans le contexte (`Substitue("de", #Objet_1)` → `"à propos de"`), …

Les méthodes de rephrasage pour les patrons de l'ambiguïté définie ci-dessus sont les suivantes :

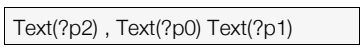

Figure 8. *Méthode de rephrasage associée au patron de gauche de la figure 7*

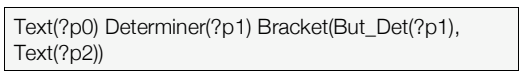

Figure 9. *Méthode de rephrasage associée au patron de droite de la figure 7*

Pour la phrase « Le capitaine a rapporté un vase de Chine. », les items suivants sont produits :

– « de Chine, le capitaine a rapporté un vase. » pour le patron et la méthode associée (figure 8).
– « Le capitaine a rapporté un (vase de Chine). » pour le patron et la méthode associée (figure 8).

Pour produire la boîte de dialogue suivante :

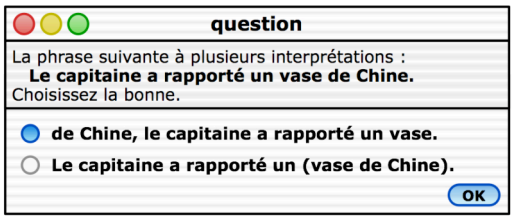

Figure 10. *Boîte de dialogue de désambiguïsation*

5.2.3. Résolution des différentes ambiguïtés présentes dans une phrase

Un module de désambiguïsation reçoit en entrée une structure *mmc* et produit en sortie un arbre de questions. Par itérations successives, les ambiguïtés présentes dans la structure *mmc* sont découvertes en créant des partitions de l'ensemble des solutions. Lorsque chaque partie est réduite à une solution, la production de l'arbre de questions est terminée. Le schéma de la figure 11 illustre ce processus.

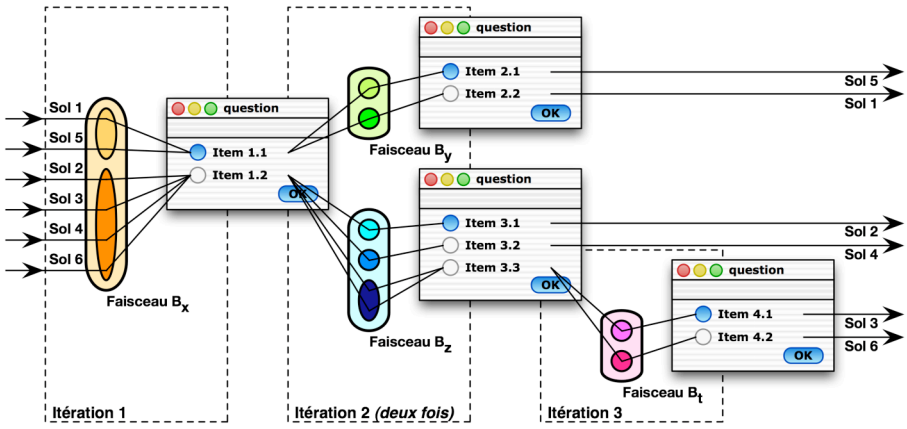

Figure 11. *Construction d'un arbre de questions (vue externe)*

La figure 12 illustre ce processus pour la phrase « Pierre voit le capitaine dans le parc avec un télescope. » Si on tient compte uniquement du sens relatif à la vision effective (« stimuli sur la rétine »), et non au rêve (« imaginer ») pour l'occurrence « voit », l'analyseur produit cinq solutions.

Lors de la première itération, l'ambiguïté détectée concerne l'attachement de « dans le parc », comme circonstant du verbe (en haut) ou comme complément de « le capitaine » (en bas). La deuxième itération concerne deux ensembles de deux et trois analyses pour lesquels il faut résoudre l'ambiguïté d'attachement de « avec un télescope ».

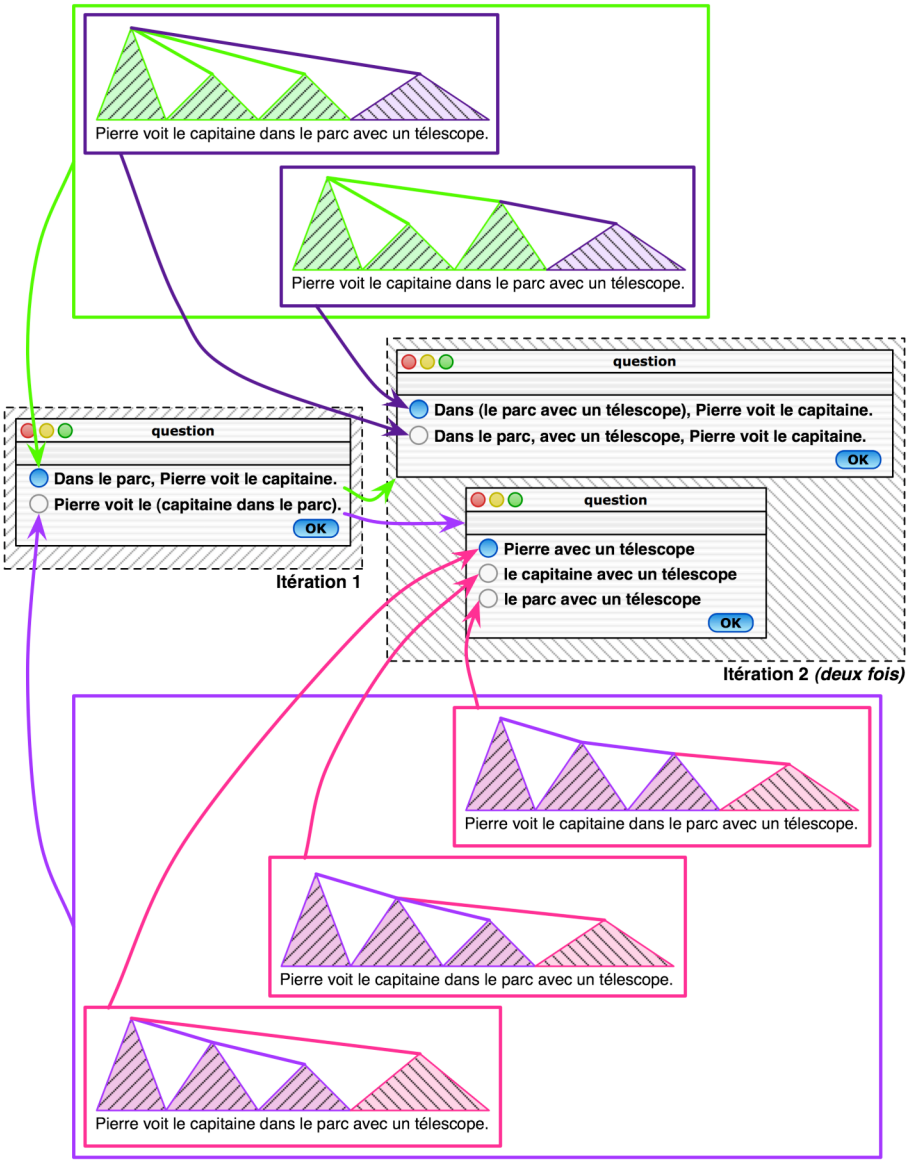

Figure 12. *Arbre de questions pour la phrase « Pierre voit le capitaine dans le parc avec un télescope. »*

5.2.4. Types d'ambiguïtés détectées et résolues

Chaque itération parcourt un automate comportant deux types d'états (figure 13) : des états de reconnaissance de métaclasses d'ambiguïtés et des états de reconnaissance de classes d'ambiguïtés.

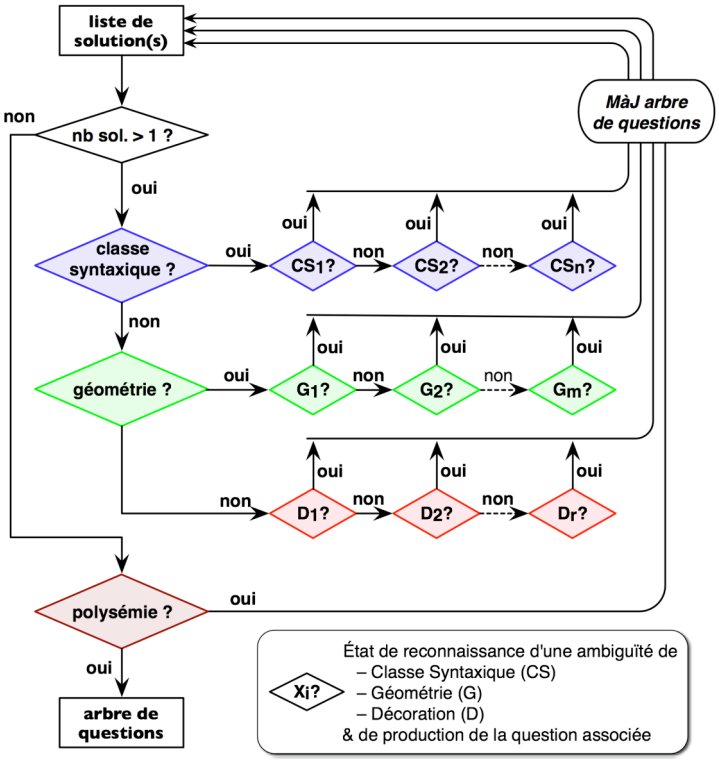

Figure 13. *Automate simplifié de construction d'un arbre de questions*

Nous avons proposé quatre métaclasses d'ambiguïtés, qui sont résolues dans l'ordre suivant[6] :

– les ambiguïtés de classe syntaxique[7] ;
– les ambiguïtés de géométrie sans ambiguité de classe syntaxique[8] ;

6. Dans les exemples des notes 7, 9 et 10, on donne les rephrasages proposés par LIDIA.
7. Dans la phrase « Il observe la photo et la classe », « la classe » peut être analysé comme un article suivi d'un nom (« Il observe la classe ») ou comme un pronom suivi d'un verbe (« Il classe la photo »). Cette ambiguïté provoque la production de deux arbres aux géométries différentes, mais nous considérons que ces ambiguïtés doivent être traitées avant les autres ambiguïtés de géométrie.
8. Il y a ambiguïté de géométrie lorsque les arbres participant à une *structure mmc* ont des géométries différentes sans qu'il n'y ait d'ambiguïté de classe syntaxique, comme dans la

– les ambiguïtés de décoration[9,10] ;
– les ambiguïtés de polysémie[11].

Un état de reconnaissance d'une classe d'ambiguïtés cherche à apparier l'un des faisceaux de description d'un type d'ambiguïté exprimable de différentes façons.

Sans compter la polysémie, le désambiguïseur du français permet actuellement de résoudre vingt classes d'ambiguïtés dues aux interactions multicatégorielles7, à la hiérarchisation des syntagmes8, à la fonction des syntagmes9, et à la restitution des arguments10 telles quelles sont décrites dans (Fuchs, 1996).

5.3. *Évaluation de la compréhensibilité des questions*

Après une expérience pilote (Blanchon *et al.*, 1996b), nous avons conduit une seconde expérience (Blanchon *et al.*, 1996c) afin de vérifier que notre proposition permet de produire des questions de désambiguïsation auxquelles les auteurs peuvent répondre sans faire d'erreur. (Mitamura *et al.*, 1999) font remarquer que les utilisateurs répondent parfois mal. Idéalement, nous aurions dû conduire des expériences permettant à des auteurs de désambiguïser leurs propres documents, ce qui est malheureusement très coûteux. Dans nos expériences, tous les sujets désambiguïsaient le même texte. L'expérience pilote nous a permis de régler les paramètres de la seconde expérience que nous résumons ici[12].

5.3.1. Données et protocole

Pour cette évaluation, nous avons recruté soixante sujets[13] répartis en quatre groupes L'expérimentateur présentait la tâche comme l'évaluation d'un **système de compréhension de la langue naturelle orale** construisant une représentation sémantique des énoncés qui lui sont dictés. Avant l'expérience proprement dite, le texte support de l'expérience était donné aux sujets afin qu'ils le lisent et se l'approprient. Au cours de l'expérience, on pouvait donc considérer que chaque sujet avait une compréhension du texte égale à celle qu'il aurait eue s'il en avait été l'auteur.

---

phrase « Le capitaine a rapporté un vase de Chine. »

9. Dans la phrase « Quel auteur cite ce conférencier ? », il s'agit de reconnaître la fonction des syntagmes. On ne sait pas quelle fonction associer à « quel auteur » et à « ce conférencier » qui peuvent être respectivement sujet et objet (« un auteur cite un conférencier ») ou objet et sujet (« un conférencier cite un auteur »). Pour une même géométrie, il y a une ambiguïté de décoration.

10. Dans la phrase « Pierre a fait porter des chocolats à Lucie », il s'agit de restituer les arguments. Lorsque « à Lucie » est rattaché à « a fait porter », l'analyseur ne peut pas décider si « Pierre a fait porter des chocolats par Lucie » ou si « Pierre a fait porter des chocolats pour Lucie ». Pour une même géométrie, il y a une ambiguïté de décoration.

11. Les ambiguïtés de polysémie ne sont pas reconnues par des faisceaux de patrons. Il suffit de consulter le mot des feuilles d'une solution pour découvrir des occurrences pour lesquelles l'analyseur propose plusieurs numéros de sens.

12. Ces deux expériences sont détaillées dans (Blanchon *et al.*, 1996a).

13. Les sujets étaient de langue maternelle anglaise et non familiers de l'état de l'art en TALN.

Lors de l'expérience, chaque sujet devait lire le texte, qui défilait sur un écran d'ordinateur, à haute et intelligible voix, en marquant une pause à la fin de chaque phrase afin que le **système de compréhension** finisse le calcul de la représentation sémantique, et ait le temps de poser éventuellement une question s'il ne parvenait pas à produire une unique représentation. Le texte comportait trente-cinq phrases ambiguës distribuées également dans sept classes d'ambiguïté.

Nous avons préparé deux jeux de questions avec des rephrasages de l'ambiguïté différents : rephrasage ***H***umain (produit par un humain cherchant à désambiguïser un énoncé) et rephrasage ***M***achine (produit par notre méthode de désambiguïsation). Nous avons aussi utilisé deux modalités pour proposer les questions : des boîtes de dialogue ***T***extuelles et des questions ***O***rales préenregistrées. Nous avons utilisé la modalité orale, d'une part pour correspondre à la modalité d'entrée, et d'autre part parce que l'oral est une modalité intéressante de désambiguïsation (Lehiste, 1973 ; Lehiste *et al.*, 1976 ; O'Shaughnessy, 1989 ; Streeter, 1978). Les sujets pouvaient demander à réécouter une question orale. En combinant le type de question et la modalité, nous avions donc quatre groupes de sujets (***MT***, ***MO***, ***ST***, ***SO***).

5.3.2. Résultats

Les données collectées nous ont permis de faire trois types d'analyse à partir des réponses aux questions de désambiguïsation, du comportement des sujets, et des réponses à un questionnaire.

5.3.2.1. Analyse statistique

En évaluant si la différence de performance des deux paires de groupes de sujets (***Mx***, ***Hx***) et (***xO***, ***xT***) était ou non statistiquement significative, nous avons obtenu les résultats suivants : (1) nous n'avons pas observé de différence entre les performances des groupes ***Mx*** et ***Hx*** ; (2) les performances des sujets sont différentes entre les questions écrites et les questions orales, mais nous ne pouvons proposer aucune conclusion définitive, puisque la différence entre les groupes ***xO*** et ***xT*** n'est pas significative.

5.3.2.2. Analyse de comportement

La vitesse de lecture ainsi que la longueur des pauses entre les phrases est très différente d'un sujet à l'autre. L'expérimentateur n'a jamais demandé aux sujets rapides de ralentir, il les a plutôt conduits à réaliser que leur lecture était trop rapide en déclenchant une question après que le sujet avait commencé la lecture de la phrase suivante. Les sujets s'adaptaient finalement en réduisant leur vitesse de lecture afin de ne plus être interrompus. Pour les groupes ***HO*** et ***MO***, nous avons remarqué une adaptation marquée. La faible vitesse d'élocution des questions orales a sans doute influencé la vitesse de lecture des sujets.

Les groupes ***HO*** et ***MO*** avaient la possibilité de demander à réécouter la question. Cependant, cette option a été très rarement utilisée (environ 3,5 % des cas). Cela peut signifier que la vitesse d'élocution utilisée pour poser une question permettait aux sujets de bien comprendre et différencier les interprétations proposées.

Le temps mis par les sujets pour répondre aux questions est plus court chez les groupes ***HO*** et ***MO*** qui n'avaient pas à lire la question. En revanche, pour les groupes ***HE*** et ***ME***, les temps de réponse varient de cinq à vingt secondes sans que l'on puisse distinguer une différence significative entre ces deux groupes. Les questions du système ne sont pas plus longues à interpréter.

Enfin, les sujets ont tous été à l'aise avec la tâche et confiants dans leurs réponses. Ils ont tous déclaré que la tâche était simple à accomplir.

5.3.2.3. Analyse du questionnaire post-expérimental

À propos de la difficulté des questions, seulement 15 % des sujets (neuf sur soixante) ont trouvé qu'il était difficile de répondre. Dans les groupes ***HO*** et ***MO***, trois sujets ont dit qu'ils auraient préféré que les questions soient posées sous forme écrite. Ce nombre est certes faible, mais il montre que la modalité pourrait être laissée au choix de l'utilisateur.

Les sujets des groupes ***HO*** et ***MO*** ont suggéré (1) que les métainformations de chaque question soient les plus courtes possible ; (2) que des variations d'intonation soient plus marquées. Les sujets des groupes ***HE*** et ***ME*** ont suggéré l'utilisation d'indications typographiques (parenthèses et gras).

Interrogés sur la stratégie qu'ils ont mise en œuvre pour répondre aux questions, les sujets disent utiliser le plus souvent le contexte et la substitution des propositions dans la phrase complète.

Sans que nous ne le leur ayons demandé, certain sujets nous ont dit qu'au fur et à mesure qu'ils progressaient dans leur lecture et dans la réponse aux questions, les questions devenaient prévisibles en fonction de certains patrons syntaxiques. Même si nous aurions aimé qu'ils soient plus nombreux à faire cette remarque, cela indique quand même que des patrons sont discernables par un non-initié, même après un bref entraînement.

5.3.2.4. Commentaires

Il n'y a donc pas de différence statistiquement significative entre les performances des sujets, que ce soit : (1) en fonction du style des questions (humain, machine), ou (2) en fonction de leur modalité (orale, écrite). Le premier résultat est essentiel au succès d'un module de désambiguïsation interactive. Nous avons aussi constaté que les sujets sont capables d'interpréter les questions sous forme humaine, mais nous ne pensons pas que des questions produites par un système de DI puissent être aussi naturelles. Rappelons que l'une des contraintes que nous nous imposons pour la production automatique de questions est de ne pas produire des rephrasages (interprétations) qui puissent être ambigus. Il est donc tout aussi essentiel que les utilisateurs puissent comprendre les questions produites par le système. Les résultats de notre expérience montrent que c'est le cas.

Nous avons aussi cherché à savoir laquelle des modalités permet d'avoir des questions plus compréhensibles. C'est une question de conception, et cela concerne

seulement les modalités de sortie que le système doit être capable de manipuler. Les résultats obtenus, ainsi que les commentaires de certains des sujets qui auraient préféré des questions écrites plutôt qu'orales, suggèrent qu'un module de désambiguïsation interactive devrait proposer plusieurs modalités en sortie et permettre à l'utilisateur de choisir celle qu'il préfère. Notre étude montre que les deux modalités sont également acceptables et « déchiffrables ».

Bien que l'option « répéter la question » n'ait pas été très utilisée, il est tout de même nécessaire de la prévoir pour le cas où l'utilisateur n'aurait pas bien compris la question après la première écoute. D'autres suggestions proposées par les sujets peuvent facilement être mises en œuvre. Par exemple, un contexte plus grand peut être proposé dans la question. Cela facilite la stratégie de réponse mise le plus souvent en œuvre (remplacement du segment ambigu par ses rephrasages dans le contexte de la phrase). Le contenu des questions orales peut aussi être rendu plus compact et l'énonciation desdites questions plus rapide. Comment mieux utiliser l'intonation et la prosodie reste une question ouverte intéressante.

6. Documents auto-explicatifs (DAE) et maquette LIDIA-3

Les questions de désambiguïsation interactive permettent à l'auteur de choisir, parmi un ensemble de paraphrases exclusives, la paraphrase qui correspond à son intention. Ces choix permettent au système de conserver la bonne interprétation pour produire une traduction de qualité. Pour les besoins de traduction proprement dits, les interprétations concurrentes, les questions et les réponses ne sont pas conservées.

Conserver ces informations ouvre la voie à une possibilité nouvelle : transmettre un document avec son sens. Les paraphrases sélectionnées représentent l'intention de l'auteur pour chacun des mots, groupes syntagmatiques ou phrases ambigus. Nous proposons de les utiliser pour enrichir le document source en produisant un Document Auto-Explicatif (ce concept a été proposé dans (Boitet, 1994)).

Un DAE est un « document actif » (Quint *et al.*, 1994) dans lequel des balises, affichées à la demande, permettent de mettre en relief les segments ambigus. Comme un document en pdf, un DAE est un document non éditable. Le lecteur d'un DAE peut sélectionner un segment pour obtenir l'« explication » associée. L'auteur est donc certain que le contenu de son document sera bien « compris » par le lecteur, ce qui est indispensable pour assurer une bonne communication.

Sur la figure 5, les parties non grisées permettent de voir comment la production de DAE s'intègre dans l'architecture linguicielle LIDIA.

Avec la maquette LIDIA-2, on produit un DAE en filtrant le document produit et en conservant, pour chaque phrase, les réponses aux questions. Un visualiseur de DAE permet au lecteur de cliquer deux fois sur une phrase pour faire apparaître les rephrasages sélectionnés par l'auteur lors de la désambiguïsation interactive.

Nous avons utilisé AMAYA14 pour réaliser la maquette LIDIA-3 (Choumane *et al.*, 2005). AMAYA (Quint *et al.*, 2004) est à la fois un éditeur de documents pour le Web, qui offre des services XML évolués, et un navigateur. Il permet de créer des documents XHTML conformes et inclut une application d'annotation collaborative basée sur RDF, XLink, et XPointer. L'interface d'AMAYA a été enrichie d'un menu permettant l'accès aux services LIDIA.

6.1. *Scénario implémenté*

La figure 14 résume le scénario que nous proposons pour la maquette LIDIA-3. L'auteur rédige un document source (*document édité*) auquel est associé un *document compagnon* enrichi avec les données produites lors de la traduction interactive (arbre de questions, réponses, et traductions). Le *document édité* est aussi annoté afin de permettre l'accès aux arbres de questions ou aux explications. Comme le document édité peut être modifié à tout moment, le *document édité* et le *document compagnon* doivent être aussi synchronisés.

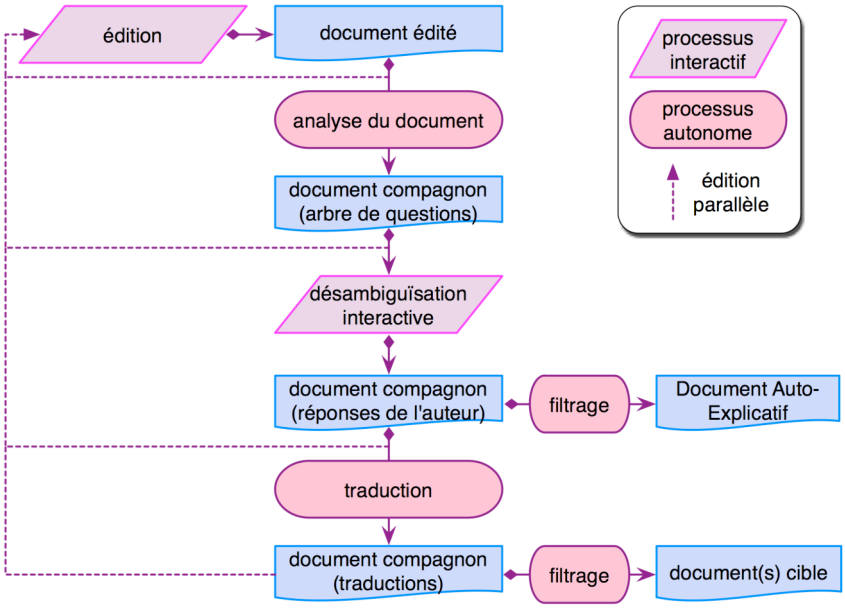

Figure 14. *Scénario de la maquette LIDIA-3*

6.2. *Vue externe de la maquette LIDIA-3*

Lorsque l'auteur demande l'analyse du *document édité* figure 15, un *document compagnon* XML est créé. Pour l'exemple, l'analyseur traite les trois phrases et produit une analyse multiple pour la seconde et la troisième. Ces analyses sont

14 http://www.w3.org/Amaya/

prises en charge par le module de désambiguïsation qui prépare, pour chacune d'entre elles, un arbre de questions.

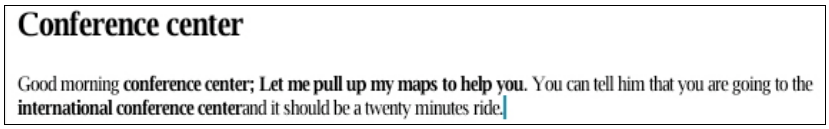

Figure 15. *LIDIA-3, document édité*

Les arbres de questions sont envoyés à AMAYA qui met à jour le *document compagnon* et annote le *document édité* (crayons et support des ambiguïtés en rouge (figure 16).

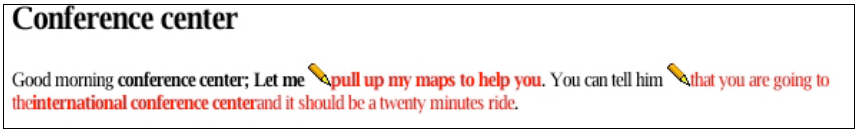

Figure 16. *LIDIA-3, document édité annoté*

Lorsque l'auteur clique sur un crayon qui signale des questions, une première question est posée (figure 17). La sélection de la paraphrase qui représente le sens voulu par l'auteur se fait ici au moyen du crayon qui signale qu'une ou plusieurs autres questions sont en suspens.

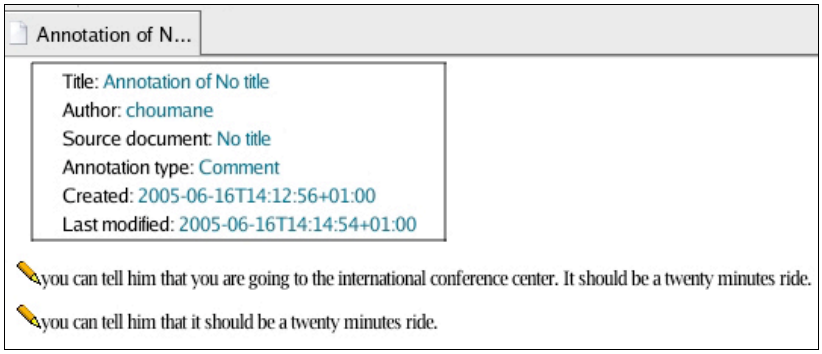

Figure 17. *LIDIA-3, question intermédiaire*

Quand l'auteur choisit l'une des paraphrases, la question suivante est présentée (figure 18). Comme cette question est ici la dernière, le choix se fait *via* un bouton de sélection exclusif.

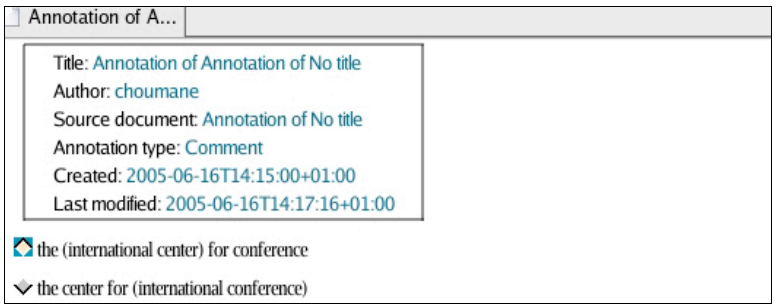

Figure 18. *LIDIA-3, question terminale*

La troisième phrase est maintenant complètement désambiguïsée, et le *document compagnon* est mis à jour pour refléter les choix de l'auteur. Dans le *document édité*, le crayon est simultanément remplacé par une coche verte (figure 19).

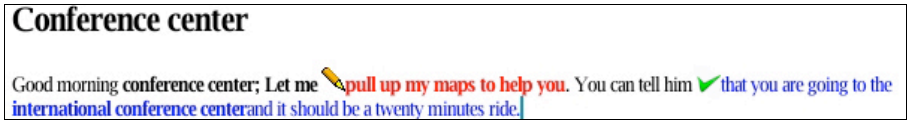

Figure 19. *LIDIA-3, document édité mis à jour*

Les phrases désambiguïsées du *document édité* deviennent « auto-explicatives ». Quand il est complètement désambiguïsé, le document édité est donc un document auto-explicatif. En cliquant sur la coche verte (figure 20), le lecteur fait apparaître les explications en reproduisant les paraphrases sélectionnées par l'auteur lors de l'étape de désambiguïsation interactive.

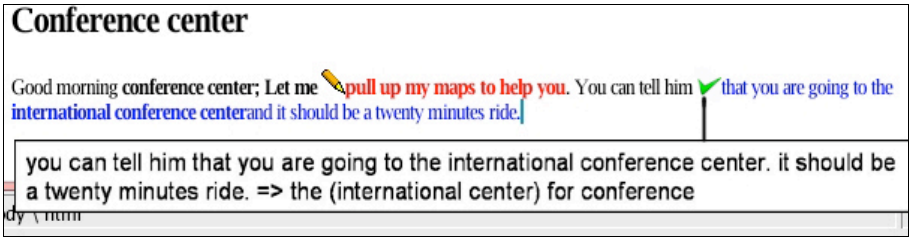

Figure 20. *LIDIA-3, présentation des explications*

6.3. *Vue interne de la maquette LIDIA-3*

Le *document compagnon*, les interactions entre le *document compagnon* et le *document édité*, la présentation des questions, sont les trois éléments nouveaux introduits dans la maquette LIDIA-3 ; nous les présentons dans cette section.

6.3.1. Document compagnon

Le *document compagnon* (figure 21), défini par un schéma XML[15], comporte des `paragraphes` composés de `phrases`.

```
<paragraph id="a1">
  <sentence stamp="1" status="nonDesamb">
     Goog morning conference center.
  </sentence>
  <sentence stamp="2" status="nonDesamb">
    <original sourceLang="En">
      Let me pull up my maps to help you.
    </original>
    <translation/><disambiguation/>
  </sentence>
  …
</paragraph>
```

Figure 21. *Document compagnon avant analyse*

Une `phrase` comporte un élément `désambiguïsation` qui est vide si la phrase n'a pas encore été analysée. Sinon, l'élément `désambiguïsation` est un ensemble de `questions` portant chacune sur un segment de la phrase défini par son caractère de début et son caractère de fin. Une `question` se compose d'au moins deux `reformulations` (paraphrases) associées au(x) numéro(s) d'analyse(s) concernée(s). Si la phrase n'est pas ambiguë, l'élément `désambiguïsation` est constitué du résultat d'analyse sous forme de *structure-umc*. Une reformulation peut contenir une ou plusieurs autres questions. Après analyse, l'élément désambiguïsation est enrichi comme le montre la figure 22.

```
<disambiguation>
  <question chBegin="16" chEnd="35" questionLang="En" questionType="G">
    <reformulation>
      <text> let me pull up (my maps to help you) </text>
      <refAnalyse>2<refAnalyse>
      <disambiguation> <solution id="2"> (umc) </solution> </disambiguation>
    </reformulation>
    <reformulation>
      <text> to help you, let me pull up my maps </text>
      <refAnalyse>1<refAnalyse>
      <disambiguation> <solution id="1"> (umc) </solution> </disambiguation>
    </reformulation>
  </question>
</disambiguation>
```

Figure 22. *Document compagnon après analyse, une question est prête*

15. `http://wam.inrialpes.fr/people/roisin/lidia/CDoc.xsd`

6.3.2. Interactions entre le document édité et le document compagnon

Chaque modification du document édité doit être reproduite dans le document compagnon. L'analyse et la désambiguïsation doivent être exécutées de nouveau, seulement sur les parties du document qui ont été modifiées. Le document compagnon est synchronisé avec le document édité au niveau des éléments XHTML de type `<p>`, `<h1>`, `<h2>`, … en appliquant une transformation (XSLT) sur le document édité.

Lorsqu'un arbre de questions est retourné à AMAYA, le document compagnon est mis à jour. Des balises `<span>` sont ajoutées dans le document édité pour mettre en relief le support des ambiguïtés. Lorsque les questions sont en attente de réponse, le support est en rouge. Il devient bleu dès que l'auteur a répondu. Le document édité est annoté, au moyen de Xpointers, pour permettre de demander l'affichage des questions ou des explications. Ces annotations sont présentées au moyen d'un crayon rouge (figure 16) pour une question en suspens, et d'une coche verte (figure 19) pour une explication.

6.3.3. Préparation des questions et des explications

Pour chaque question, un fichier XHTML est créé. La question peut être une question terminale ou non. Dans le second cas, les questions suivantes sont représentées comme des annotations dans le fichier XHTML qui représente la question courante.

Lorsque l'auteur répond à une question, le document compagnon est mis à jour pour prendre en compte le choix courant. Lorsqu'il a répondu à toutes les questions correspondant à un support d'ambiguïté, l'auteur, et plus tard le lecteur, peut accéder aux explications en cliquant sur la coche verte. À cet instant, un cadre (figure 20) est préparé et affiché à la volée. Il contient les paraphrases sélectionnées par l'auteur lors de l'étape de désambiguïsation interactive.

7. Bilan et perspectives

7.1. Bilan

L'architecture informatique que nous avons proposée pour les maquettes LIDIA-2 et LIDIA-3 semble adaptée aux exigences de la diffusion de la TAFD pour tous. La maquette actuelle réalisée avec AMAYA permet à l'auteur de produire des documents XHTML. De plus, le modèle de *document compagnon*, l'accès aux services d'analyse, de désambiguïsation et de traduction sont complètement indépendants de l'environnement de rédaction.

Des questions intéressantes se posent encore sur l'architecture linguicielle et la désambiguïsation interactive (cf. infra). Nos travaux nous ont permis de proposer le concept de Documents Auto-Explicatifs. Ce concept n'est pas né au sein de la communauté du document numérique, parce qu'elle n'envisage pas, jusqu'à présent,

la possibilité de faire une analyse complète des énoncés, mais s'intéresse principalement à l'édition et au formatage des documents, et, depuis l'avènement de XML, à l'annotation des documents, mais cela sans que leurs auteurs ne soient impliqués.

Même si nous n'en sommes qu'à nos premiers pas, et si nous nous sommes rendu compte qu'il faudrait revoir le processus de désambiguïsation (cf. infra), nous avons prouvé que notre proposition est implémentable. Une telle aide à l'accès au sens représentera une innovation majeure, impossible, et même inimaginable, avant l'avènement de l'informatique et des traitements puissants et conviviaux qu'elle permet, en particulier, en traitement des langues.

7.2. Perspectives

Dans le contexte de vraies applications, l'analyseur rencontrera un grand nombre d'ambiguïtés. Il est donc possible que, pour certaines phases, l'arbre des questions ait une telle profondeur que l'auteur n'accepte de répondre qu'aux questions cruciales.

Supposons, par exemple, qu'une phrase de longueur **N** possède $\mathbf{k^N}$ interprétations et que les descripteurs d'ambiguïtés sont constitués en moyenne de **p** patrons. **(k/p).N** questions désambiguïseraient complètement cette phrase. Si par exemple **(k/p)=1/2**, il y aurait alors cent vingt questions pour une page de deux cent quarante mots. Bien qu'il ne faille pas plus de dix minutes pour répondre à toutes ces questions[16], si chaque réponse prend cinq secondes, l'auteur peut vouloir consacrer moins de temps à la désambiguïsation interactive.

En d'autres termes, étant donné une structure *multisolution multiniveau concrète*, quelques réponses à des questions de désambiguïsation, et certaines préférences de l'utilisateur, le système devrait être capable de faire des choix et de produire une traduction unique, ou alors une représentation factorisée explicitant les différentes traductions possibles en langue cible. Afin d'implémenter une telle stratégie, il est nécessaire que les modules utilisés puissent mettre en œuvre des techniques heuristiques de désambiguïsation automatique ou soient capables de manipuler des structures ambiguës.

À partir « du degré de complétion de la désambiguïsation » d'une phrase, et en prenant en compte la « crucialité pour la traduction » des ambiguïtés non résolues, il est sans doute possible de calculer un « niveau de certification du sens » associé à la traduction, et de le calculer au niveau des paragraphes, des sections, etc., jusqu'au document lui-même.

Pour améliorer l'interface de présentation d'un DAE, il faudrait que les supports des ambiguïtés soient indiqués, comme c'est le cas avec la maquette LIDIA-3, afin

16. Ce temps doit être comparé aux mesures de temps de travail fournies par les traducteurs professionnels. Pour une page standard (de 250 mots) et pour chaque langue cible, il faut compter 1 heure pour produire une première traduction et 20 minutes de postédition.

que le lecteur puisse identifier précisément les segments qui posent un problème d'interprétation. Mais les patrons que nous utilisons actuellement pour le français n'utilisent pas la notion de support de l'ambiguïté. Ils capturent en effet très souvent un segment plus grand que le support, afin de permettre un rephrasage compréhensible de l'ambiguïté.

Si le processus de désambiguïsation interactive fournissait le support de l'ambiguïté, on pourrait choisir d'attacher la description du support de l'ambiguïté à chaque faisceau. Le support peut, en effet, être défini à partir des variables utilisées dans les patrons. Une seconde approche, proposée dans (Boitet *et al.*, 1995b), oblige à changer la description des ambiguïtés, en utilisant uniquement leur support. Cependant, pour faire des rephrasages tels que ceux que nous produisons actuellement, il faudrait décrire les informations supplémentaires à utiliser lors de la fabrication des items de dialogue.

L'architecture linguistique que nous proposons (figure 5) permet aussi de produire des DAE en langue cible. Comme l'étape de génération produit une structure intermédiaire équivalente à une structure d'analyse désambiguïsée (*unisolution multiniveau concrète*), il suffirait de faire une analyse multiple (*multisolution multiniveau concrète*) des phrases effectivement générées, puis de construire un arbre des questions concernant ces phrases. Sachant que l'on connaît la structure *unisolution multiniveau concrète* à retenir, on pourrait calculer automatiquement les réponses aux questions de désambiguïsation à la place d'un lecteur en langue cible. On pourrait donc produire un DAE en langue cible sans intervention humaine.

Atteindre cet objectif est cependant difficile en pratique puisqu'il faut disposer d'un analyseur multiniveau dans chacune des langues traitées (source et cibles). Nous espérons construire un prototype complet implémentant cette idée grâce à des coopérations internationales, comme le consortium U++C dédié au développement d'outils et de ressources pour UNL.

Bibliographie

Blanchon H. « A Solution to the Problem of Interactive Disambiguation », *Proc. COLING-92*, Nantes, France, July 23-28, 1992, vol. 4/4, p. 1233-1238.

Blanchon H. « An Interactive Disambiguation Module for English Natural Language Utterances », *Proc. NLPRS'95*, Seoul, Korea, Dec 4-7, 1995, vol. 2/2, p. 550-555.

Blanchon H., Boitet C. « Deux premières étapes vers les documents auto-explicatifs », *Proc. TALN 2004*, Fès, Maroc, 19-21 avril 2004, vol. 1/1, p. 61-70.

Blanchon H., Fais L. « How to ask Users About What they Mean: Two Experiments & Results », *Proc. MIDDIM'96*, Le Col de Porte, Isère, France, August 12-14, 1996, vol. 1/1, p. 238-259.

Blanchon H., Fais L. Pilot Experiment on the Understandability of Interactive Disambiguation Dialogues, Technical report, n° TR-IT-0177. July, 1996. ATR-ITL. 18 p.

Blanchon H., Fais L. A Second Experiment on the Understandability of Interactive Disambiguation Dialogues, Technical Report, n° TR-IT-0167. April, 1996. ATR-ITL. 22 p.

Boitet C. « Towards Personal MT: general design, dialogue structure, potential role of speech », *Proc. COLING-90*, Helsinki, Finland, August 20-25, 1990, vol. 3/3, p. 30-35.

Boitet C. « Dialogue-Based MT and self explaining documents as an alternative to MAHT and MT of controlled language », *Proc. Machine Translation Ten Years On*, Cranfield, England, Oct. 12-14, 1994, 7 p.

Boitet C., Blanchon H. « Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup », *Machine Translation*. vol. 9, n° 2, 1995a, p. 99-132.

Boitet C., Tomokio M. « Ambiguities & ambiguity labelling: towards ambiguity databases », *Proc. RANLP'95 (Recent Advances in NLP)*, Tzigov Chark, Bulgaria, 14-16 September, 1995, vol. 1/1, p. 13-26.

Choumane A., Blanchon H., Roisin C. « Integrating translation services within a structured editor », *Proc. DocEng 2005 (ACM Symposium on Document Engineering)*, Bristol, United Kingdom, November 02-04, 2005, vol. 1/1, p. 165-167.

Fuchs C. *Les ambiguïtés du français*. Paris, Ophris, 1996, 184 p.

Huang X. « A Machine Translation System for the Target Language Inexpert », *Proc. COLING-90*, Helsinki, August 20-25, 1990, vol. 3/3, p. 364-367.

Kay M. (1973) *The MIND system*. in Rustin R. (ed.), Courant Computer Science Symposium 8: Natural Language Processing. Algorithmics Press, Inc. New York. pp. 155-188.

Lehiste I. « Phonetic disambiguation of syntactic ambiguity », *Glossa*. vol. 7, n° 2, 1973, p. 107-122.

Lehiste I., Olive J. P., Streeter L. A. « Role of duration in disambiguating syntactically ambiguous sentences », *Journal of the Acoustical Society of America*. vol. 60, n° 5, 1976, p. 1199-1202.

Maruyama H., Watanabe H., Ogino S. « An Interactive Japanese Parser for Machine Translation », *Proc. COLING-90*, Helsinki, August 20-25, 1990, vol. 2/3, p. 257-262.

Melby A. K. « Multi-Level Translation Aids in a Distributed System », *Proc. COLING-82*, Prague, 5-10 juillet 1982, vol. 1/1, p. 215-220.

Mitamura T., Nyberg E., Torrejon E., Igo R. « Multiple Strategies for Automatic Disambiguation in Technical Tranlsation », *Proc. TMI-99*, University College, Chester, England, August 23-25, 1999, vol. 1/1, p. 218-227.

Nirenburg S. « Knowledge-based Machine Translation », *Machine Translation*. vol. 4, 1989, p. 5-24.

O'Shaughnessy D. « Specifying accent marks in French text for teletext and speech synthesis », *International Journal of Man-Machine Studies*. vol. 31, n° 4, 1989, p. 405-414.

Quint V., Vatton I. « Making structured documents active », *Electronic Publishing Origination, Dissemination, and Design*. vol. 7, n° 2, 1994, p. 55-74.

Quint V., Vatton I. « Techniques for Authoring Complex XML Documents », *Proc. DocEng, ACM Symposium on Document Engineering*, Milwaukee, Wisconsin, USA, October 28-30, 2004, vol. 1/1, p. 115-123.

Somers H., Jones D. « La génération de textes multilingues par un utilisateur monolingue », *Meta*. vol. 37, n° 4, 1992, p. 647-656.

Somers H. L., Tsujii J.-I., Jones D. « Machine Translation without a source text », *Proc. COLING-90*, Helsinki, August 20-25, 1990, vol. 3/3, p. 271-276.

Streeter L. A. « Acoustic determinants of phrase boundary perception ». vol. 60, n° 6, 1978, p. 1582-1592.

Tomita M. « Sentence disambiguation by asking », *Computers and Translation*. vol. 1, n° 1, 1986, p. 39-51.

Vauquois B., Boitet C. « Automated Translation at Grenoble University », *Computational Linguistics*. vol. 11, n° 1, 1985, p. 28-36.

Weaver A. (1988) *Two Aspects of Interactive Machine Translation*. in Vasconcellos M. (ed.), Technology as Translation Strategy. State University of New York at Binghamton. Binghamton. p. 116-123.

Wehrli É. (1993) *Vers un système de traduction interactif*. in Bouillon P. and Clas A. (ed.), La traductique. Les Presses de l'Université de Montréal, AUPELF/UREF. p. 423-432.

Witkam A. P. M. *Distributed Language Translation — Feasibility Study of a Multilingual Facility for Videotex Information Networks*. Utrecht, The Netherlands, BSO, 1983, 340 p.

Wood M. M. G. (1989) *Japanese for speakers of English: The UMIST/Sheffield Machine Translation Project*. in Peckham J. (ed.), Recent Developments and Applications of Natural Language Processing. Kogan Page Limited. London. p. 56-64.

Wood M. M. G., Chandler B. « Machine Translation For Monolinguals », *Proc. COLING-88*, Budapest, 22-27 août 1988, vol. 2/2, p. 760-763.