

Deux premières étapes vers les documents auto-explicatifs

Hervé Blanchon, Christian Boitet

GETA, CLIPS-IMAG

BP 53

38041 Grenoble Cedex 9

{Hervé.Blanchon, Christian.Boitet}@imag.fr

Résumé - Abstract

Dans le cadre du projet LIDIA, nous avons montré que dans de nombreuses situations, la TA Fondée sur le Dialogue (TAFD) pour auteur monolingue peut offrir une meilleure solution en traduction multible que les aides aux traducteurs, ou la traduction avec révision, même si des langages contrôlés sont utilisés. Nos premières expériences ont mis en évidence le besoin de conserver les « intentions de l’auteur » au moyen « d’annotations de désambiguïsation ». Ces annotations permettent de transformer le document source en un Document Auto-Explicatif (DAE). Nous présentons ici une solution pour intégrer ces annotations dans un document XML et les rendre visibles et utilisables par un lecteur pour une meilleure compréhension du « vrai contenu » du document. Le concept de Document Auto-Explicatif pourrait changer profondément notre façon de comprendre des documents importants ou écrits dans un style complexe. Nous montrerons aussi qu’un DAE, traduit dans une langue cible L, pourrait aussi être transformé, sans interaction humaine, en un DAE en langue L si un analyseur et un désambiguïseur sont disponibles pour cette langue L. Ainsi, un DAE pourrait être utilisé dans un contexte monolingue, mais aussi dans un contexte multilingue sans travail humain additionnel.

In the LIDIA project, we have demonstrated that, in many situations, Dialogue-Based MT (DBMT) for monolingual author is likely to offer better solutions to multitarget translation needs than machine aids to translators or batch MT, even if controlled languages are used. First experiments have shown the need to keep a memory of the “author’s intention” by means of “disambiguating annotations” transforming the source document into a “self-explaining document” (SED). We present ways to integrate these annotations into an XML document (SED-XML), and to make them visible and usable by readers for better understanding of the “true content” of a document. The very concept of SED might deeply change our way of understanding important or difficult written material. We also show that a SED, once translated into a target language L, might be transformed into an SED in L with no human interaction, if an analyzer and a disambiguator are available for L. Hence, the SED structure might be used in multilingual as well as in monolingual contexts, without addition of human work.

Mots Clés - Keywords

Document Auto-Explicatifs (DAE), désambiguïsation interactive, documents actifs
Self-Explaining Document (SED), interactive disambiguation, active documents

Introduction

Dans beaucoup de situations, des notes de travail, des résumés scientifiques, des documentations techniques, etc., devraient être traduits dans plusieurs langues. Ils ne le sont pas pour trois raisons : (1) ils sont prêts à la dernière minute et les traducteurs disponibles n'ont pas le temps de faire le travail, (2) il n'y a pas de traducteur pour faire le travail, (3) et bien sûr, dans tous les cas, aucune solution automatisée n'est satisfaisante.

La Traduction Automatique interactive Fondée sur le Dialogue (TAFD), telle que nous l'avons prototypée dans le cadre du projet LIDIA [Boitet C. and Blanchon H., 1995], constitue une meilleure piste pour résoudre ce problème que les aides au traducteur et les systèmes de TA de type « boîte noire », même si des langages contrôlés sont utilisés pour la rédaction.

La TAFD offre aussi la possibilité de produire toutes les versions d'un document, soit le document source et tous les documents cibles, sous forme de Document Auto-explicatif (DAE) [Boitet C., 1994]. Un DAE contient le texte du document ainsi que sa représentation linguistique, profonde ou multiniveau, désambiguïsée et augmentée d'une mémoire des ambiguïtés qu'il contient et d'une trace du processus de désambiguïsation.

Avec le concept de DAE, nous proposons une nouvelle manière, totalement novatrice et potentiellement très utile, d'accéder au contenu d'un document et de l'utiliser. Un lecteur de DAE pourrait « 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. Ainsi, un DAE rédigé dans un langage non contrôlé serait moins ambigu qu'un texte rédigé dans un langage contrôlé, qui n'est certes pas ambigu pour la machine, mais le reste très souvent pour un humain. Nous proposons aussi une méthode permettant de traduire un DAE source en un DAE cible, sans effort humain supplémentaire, sans traitement contrastif (et très difficile) des ambiguïtés, et avec capture de toutes les ambiguïtés en cible, rendant ainsi l'intention exacte de l'auteur accessible dans d'autres langues.

Dans cet article, nous présentons d'abord le projet LIDIA, notre premier prototype (LIDIA-1), ainsi que le concept de DAE. La section 2 est consacrée à la présentation de l'implémentation LIDIA-2. Dans la section 3 nous présentons notre premier visualiseur de DAE et proposons des améliorations à court terme. Nous proposons enfin, dans la section 4, des pistes de recherches pour le futur.

1 LIDIA-1: le berceau du concept de DAE

L'idée des Documents Auto-Explicatifs (DAE) est issue de nos travaux sur la Traduction Automatique Fondée sur le Dialogue (TAFD).

1.1 Le projet LIDIA

Les efforts passés visant à améliorer la qualité des traductions produites par des systèmes de TA ont montré que la TA de haute qualité est possible, mais seulement pour des typologies de textes (domaine, style) très contraintes. On peut citer, par exemple, les bulletins météorologiques (METEO, TAUM, anglais↔français), les brèves boursières du Nikkei (ALT/Flash, NTT, japonais→anglais), ou les documents techniques (BV/aéro/FE pour les manuels de maintenance d'avions, Systran pour des documents XEROX en anglais contrôlé).

La TAFD de haute qualité est un nouveau paradigme pour des situations traductionnelles pour lesquelles les approches — fondées sur la langue, fondées sur la connaissance — ne sont pas appropriées [Boitet C. and Blanchon H., 1995]. En TAFD, bien que les sources de

connaissances linguistiques soient encore cruciales, et que des connaissances extra-linguistiques puissent être utilisées si elles sont disponibles, l'accent est mis sur la pré-édition indirecte au moyen d'un dialogue de désambiguïsation avec l'auteur permettant d'obtenir des traductions de haute qualité sans révision.

La première situation que nous avons considérée est la production de documents techniques multilingues sous la forme de documents HyperCard. HyperCard est un environnement de production de documents hypertextes dont les « pages » sont appelées cartes. Les cartes contiennent différents types d'objets, dont des champs textuels. Du point de vue linguistique, nous utilisons une approche fondée sur un transfert multiniveau avec des acceptions, propriétés, et relations interlingues. Notre première maquette, LIDIA-1, démontre l'idée avec un document HyperCard qui présente, en contexte, des phrases ambiguës en français. Ce document peut être traduit vers l'anglais, l'allemand et le russe. Bien que cette maquette soit réduite du point de vue de sa couverture linguistique, elle montre le potentiel de l'approche.

1.2 Quelques éléments sur l'interface de LIDIA-1

L'utilisateur peut activer les traitements LIDIA les plus fréquents grâce à une palette d'outils. La première ligne d'outils (Figure 1), considérée de gauche à droite, permet de traduire l'objet sélectionné, voir la progression des traitements, voir les annotations et voir la rétrotraduction en français. La seconde ligne permet de se déplacer parmi les cartes du document.

Après l'analyse, un bouton (? !! - Figure 2) apparaît au dessus de l'objet à traduire si son contenu est ambigu et nécessite une désambiguïsation interactive.

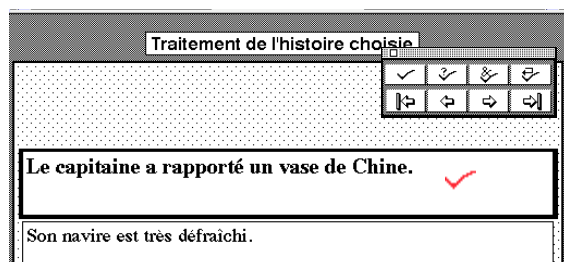


Figure 1 : sélection d'un objet textuel à traduire

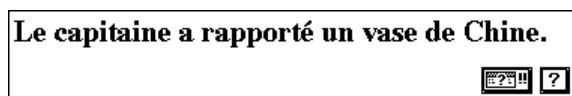


Figure 2 : Questions de désambiguïsation en suspens pour un objet

Lorsqu'il décide de résoudre les ambiguïtés concernant un objet particulier, l'utilisateur clique sur ce bouton et les questions sont proposées comme ci-dessous. Dans le contexte de la phrase, l'auteur doit choisir d'attacher « de Chine » à « vase » (un vase chinois, Figure 3). Un second dialogue apparaît alors pour le sens du mot « capitaine » (Figure 4).

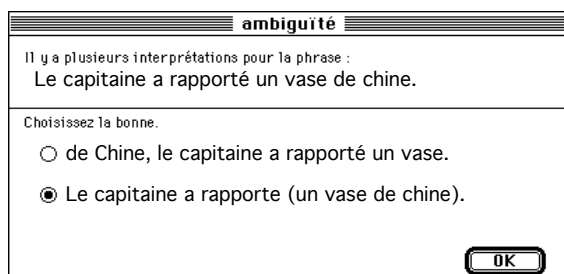


Figure 3 : Désambiguïsation strucurelle

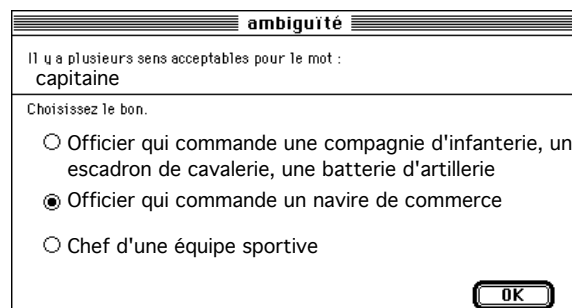


Figure 4 : Désambiguïsation du sens du mot « capitaine »

La Figure 5 montre la traduction de la phrase « le capitaine a rapporté un vase de Chine » dans deux contextes différents. Cela permet d'illustrer l'intérêt de la désambiguïsation interactive.

Erste Geschichte	Zweite Geschichte
Der Hauptmann hat eine Vase aus China mitgebracht. Die Vase ist englisch.	Der Kapitän hat eine chinesische Vase mitgebracht. Sein Boot ist sehr verblasst.
Behandlung	Behandlung

Figure 5 : Traduction d'une même phrase dans deux contextes différents

1.3 Étapes de production d'un DAE

Nous avons proposé et motivé le concept de DAE dans [Boitet C., 1994]. Nous donnons ici un bref aperçu (Figure 6) des étapes de traitement mises en œuvre et des structures de données produites dans le cadre l'architecture linguistique LIDIA. Nous montrons aussi comment la production de DAE en langues source et cible peut s'y intégrer.

Chaque phrase du texte en langue source est d'abord analysée pour produire une structure *mmc-source* (multisolution, multiniveau¹, concrète²). Cette structure *mmc* est alors utilisée pour construire un arbre des questions qui seront posées à l'auteur. 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 abstraite *uma-source* (unisolution, multiniveau, abstraite²).

Un composant de transfert lexical et structural produit ensuite 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. S'ils le sont, ils ne servent que de guide³ à la première étape de la génération, qui "choisit la paraphrase à générer" et produit une structure *uma-cible* qui est homogène à la structure qui serait produite en analysant et en désambiguïsant interactivement le texte cible qui va être généré. Le processus de traduction se termine avec les générations syntaxique et morphologique.

Au cours de la traduction, ou après l'analyse suivie de désambiguïsation interactive uniquement, les informations nécessaires à la construction d'un DAE sont conservées. En effet, la production d'un DAE ne nécessite pas l'exécution de chacune des étapes du processus de traduction. Pour produire un DAE, il suffit d'une étape d'analyse multiple suivie d'une étape de désambiguïsation interactive.

La production d'un DAE en langue cible est détaillée dans la section 4.3.

¹ La structure contient trois niveaux d'interprétation linguistique : le niveau des classes syntaxiques et syntagmatiques, le niveau des fonctions syntaxiques, et le niveau des relations logiques et sémantiques.

² Une représentation d'un texte est dite « concrète » si l'on retrouve directement le texte représenté par un parcours simple de la structure (mot des feuilles pour un structure syntagmatique, parcours infixé pour une structure de dépendances). Sinon, la structure est dite « abstraite ».

³ On utilise un attribut "tactique" ensembliste LOCK à valeurs dans les noms des attributs non interlingues pour dire si tel ou tel attribut comme la classe ou la fonction syntaxique a été calculé par le transfert à titre d'ordre ou de préférence).

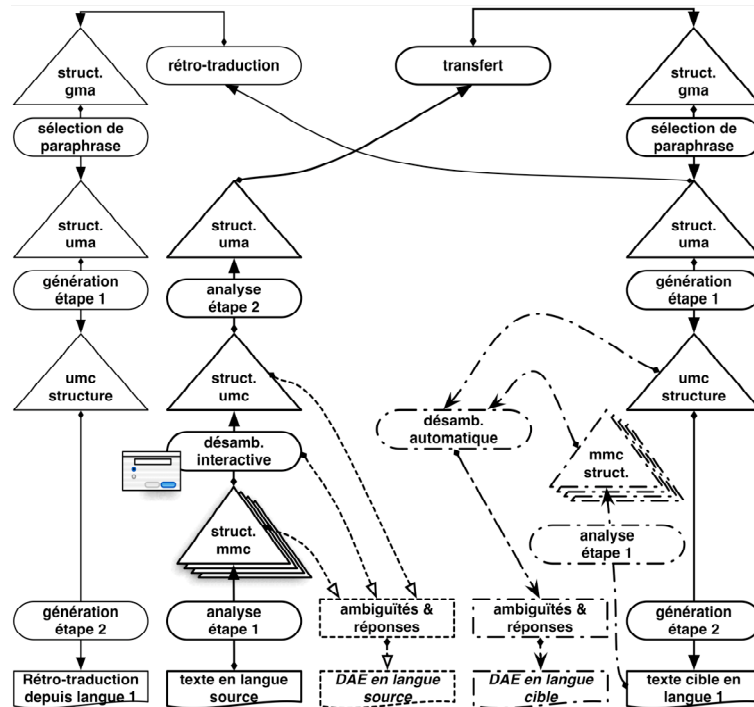


Figure 6 : Organisation linguicielle en TAFD et production de DAE

2 LIDIA-2 : production d'un DAE

Avec LIDIA-2, nous avons changé l'architecture logicielle au niveau de l'accès distribué aux traitements ainsi que le format des données. En particulier, tous les composants utiles à la traduction interactive sont maintenant accédés avec le même protocole (telnet) alors que dans LIDIA-1 nous utilisions le courrier électronique (SMTP) et les AppleEvents pour accéder au linguiciel et au désambigüiseur. Un document LIDIA-2 contient l'historique de la désambigüisation interactive. Il est ensuite filtré pour produire un DAE. Ces deux fichiers sont représentés en XML. L'environnement LIDIA-2 est implémenté en JAVA.

Dans la première version de LIDIA-2, nous avons utilisé un désambigüiseur de l'anglais [Blanchon H., 1995] fondé sur une étude de corpus [Fais L. and Blanchon H., 1996]. Ce module était, en effet, directement intégrable à la nouvelle architecture d'intégration des composants que nous avons proposée.

2.1 Exemple de session

L'utilisateur (l'auteur) personnalise d'abord son environnement. Il peut alors créer un nouveau document ou ouvrir un document existant. La fenêtre du document (Figure 7) est divisée en deux sections : la partie supérieure est la fenêtre d'édition, la partie inférieure affiche des informations relatives à l'état des traitements effectués sur le document.

Après que l'utilisateur ait demandé l'analyse du document, les phrases ambiguës sont colorées en brun et les phrases non ambiguës sont colorées en vert. Dans la Figure 7, le texte contient sept phrases ambiguës et une phrase qui ne l'est pas (la première).

Lorsque l'utilisateur fait un clic double sur une phrase ambiguë, les questions de désambigüisation correspondantes sont affichées. L'ordre d'affichage assure un parcours depuis la racine jusqu'à une feuille dans l'arbre des questions associé à la phrase considérée. Pour la phrase "let me pull up my maps to help you", il n'y a qu'une question pour deux interprétations. Lorsqu'elle est désambigüisée, la phrase est colorée en vert.

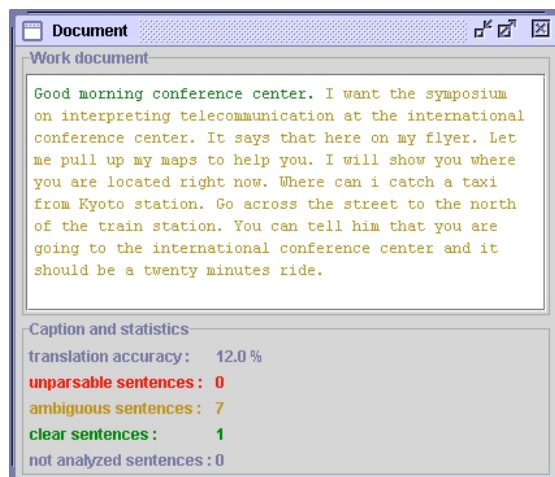


Figure 7 : Fenêtre de document (après analyse)

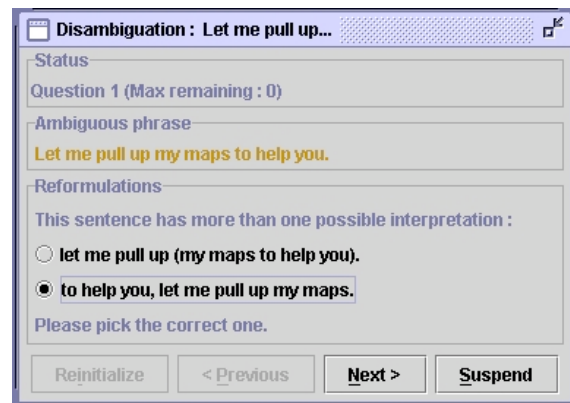


Figure 8 : Une question de désambiguïsation

2.2 Support à la manipulation du document

Un document LIDIA-2 (cf. Annexe) est un document XML. Nous avons choisi l'API DOM pour gérer et manipuler le document LIDIA-2 en cours d'utilisation et l'API SAX pour vérifier la bonne syntaxe des documents lors de leur ouverture. Ce document contient une entête (`<description>`) et un contenu (`<support>`) — le texte — sous forme de paragraphes (`<paragraphe>`) et de phrases (`<phrase>`), saisies par l'utilisateur et enrichies par les informations collectées lors des différentes étapes de traitement.

L'entête comporte le titre du document, la langue source et le nom de l'auteur. Pour chaque phrase du texte, le `<support>` comprend la langue source, le texte de la phrase, l'arbre des questions, ainsi que la ou les traductions obtenues dans les langues cibles choisies. L'arbre des questions est une représentation, *à la lisp*, de celui qui est produit par le module de désambiguïsation. Cet arbre est enrichi d'une trace du chemin suivi par l'utilisateur lors de la désambiguïsation effective.

2.3 Filtrage vers un DAE

Pour produire le DAE associé au document LIDIA-2 en cours, celui-ci est filtré. On conserve dans le DAE l'entête du document LIDIA-2, et l'organisation en paragraphes et phrases du `<support>`. Pour chaque phrase, on retient le texte d'origine et la trace du parcours de l'auteur dans l'arbre des questions. (cf. Annexe).

3 Visualisation d'un DAE (étape 2)

Nous concevons une DAE comme un document autonome et « portable » qui doit pouvoir être diffusé. Il faut donc proposer un environnement de visualisation autonome.

3.1 Objectif et contraintes

Un visualiseur de DAE doit permettre à un lecteur de lire le contenu du document et d'appréhender le « sens exact » de ce que l'auteur a voulu dire. Le visualiseur doit donc pouvoir signaler au lecteur les segments ambigus, et, à la demande, en révéler le « sens » choisi par l'auteur lors de la phase de désambiguïsation interactive.

Le premier visualiseur que nous présentons ici est assez simple et l'interaction avec le lecteur assez pauvre. Il est écrit en JAVA pour être portable et téléchargeable sur le web. L'API DOM est utilisée pour gérer le document.

3.2 Interface du premier visualiseur

Quand le lecteur ouvre un DAE avec le visualiseur, le contenu textuel du DAE est affiché (Figure 9).

Dans cette implémentation, les segments ambigus ne sont pas surlignés. Pour obtenir les informations relatives aux différentes lectures possibles d'une phrase, l'utilisateur doit faire un double clic sur son texte. Une boîte de dialogue apparaît alors. Elle permet au lecteur de naviguer dans l'arbre des questions en voyant les rephrasages sélectionnés par l'auteur (=> ... <=) lors de la désambiguïsation, comme le montre la Figure 10.

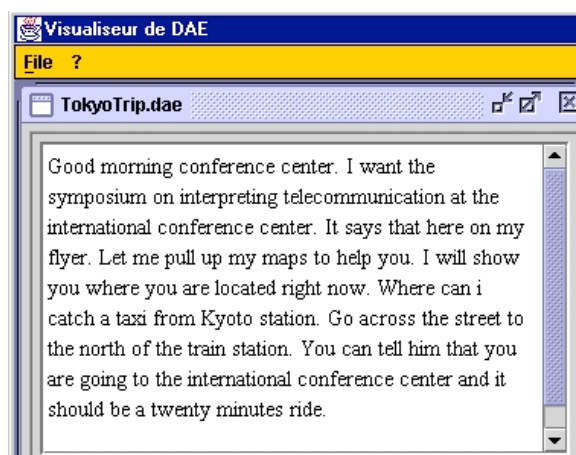


Figure 9 : Environnement de lecture d'un DAE

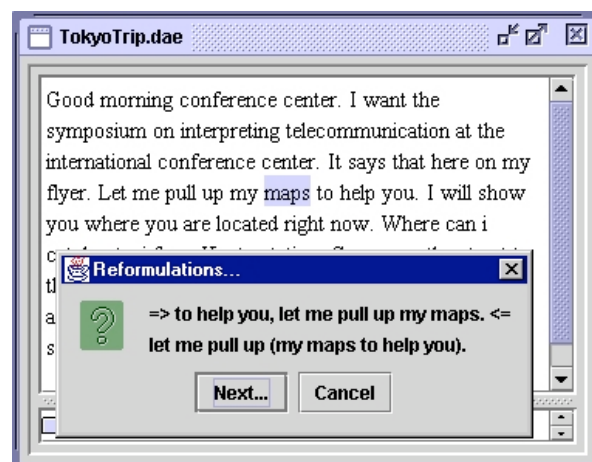


Figure 10 : Affichage des rephrasages sélectionnés lors de la désambiguïsation

3.3 Améliorations à court terme

Afin d'améliorer l'implémentation de LIDIA-2, nous avons plusieurs objectifs à court terme. Les perspectives à long terme sont détaillées dans la section 4.

3.3.1 Intégrer les linguiciels LIDIA-1 dans la nouvelle architecture

Notre premier objectif à court terme est d'intégrer dans l'architecture LIDIA-2 les modules d'analyse, de désambiguïsation interactive, de transfert et de génération développés pour la maquette LIDIA-1 [Boitet C. and Blanchon H., 1995]. Cela nous permettrait d'avoir une plateforme d'expérimentation plus riche. Nous verrons dans la section 4.1 qu'aller plus loin dans la localisation des ambiguïtés implique des changements importants pour le module de désambiguïsation interactive lui-même.

3.3.2 Permettre des changements de réponse aux questions de désambiguïsation

Dans certains cas, il peut être intéressant de refaire la désambiguïsation interactive, soit pour corriger un résultat de traduction (la désambiguïsation interactive aurait dans ce cas été mal faite), soit pour produire une nouvelle traduction pour montrer l'intérêt de la désambiguïsation.

Toutes les informations nécessaires sont déjà disponibles dans un document LIDIA-2. Ainsi une nouvelle désambiguïsation peut être effectuée de manière autonome (hors ligne). Si le

nouveau parcours de désambiguïsation est le même que le précédent, les bonnes traductions vers les différentes langues cibles auront déjà été calculées. Si le parcours est différent, de nouvelles traductions devront être produites (en ligne).

3.3.3 Créer automatiquement des corpus multilingues auto-explicatifs alignés

Nous avons dit que LIDIA-2 peut accepter des demandes de traduction vers plusieurs langues cibles. Les traductions sont conservées dans le document LIDIA-2.

Il pourrait donc être intéressant d'exporter un document multilingue déjà aligné au niveau de sa structure. On pourrait même envisager de conserver dans le document LIDIA-2 toutes les structures intermédiaires produites lors du processus de traduction afin de calculer automatiquement différents alignements pour chaque phrase (au niveau des mots, des segments, des syntagmes).

Cela pourrait être utile, par exemple, en apprentissage des langues, et aussi pour l'étude contrastive des ambiguïtés. On sait, également, que les besoins de corpus alignés croissent avec le développement de moteurs statistiques pour le traitement de la langue naturelle, notamment en traduction.

4 Perspectives à plus long terme

Nos objectifs à long terme ont un impact sur les modules qui participent à la chaîne de traduction (analyse, transfert, génération), et sur le module de désambiguïsation interactive.

4.1 Prendre en compte du support de l'ambiguïté

Afin d'améliorer l'utilisabilité d'un DAE, l'aspect le plus important nous semble la localisation précise des ambiguïtés. Pour être capable de faire cela, le moteur de préparation de l'arbre des questions doit être capable de localiser précisément le support [Boitet C. and Tomokio M., 1995] de chaque ambiguïté. Cet objectif peut être atteint soit en calculant le support de l'ambiguïté, a posteriori, à partir des descripteurs d'ambiguïtés actuels, soit en modifiant les descripteurs d'ambiguïtés afin qu'ils soient définis directement sur les supports de l'ambiguïté. Cette dernière solution, qui a notre préférence, implique plus de modifications du module de désambiguïsation que la première.

4.2 Autoriser une désambiguïsation interactive incomplète

Dans le contexte de vraies applications, l'analyseur rencontrera un grand nombre d'ambiguïtés. Pour chaque phase, il est donc possible que l'arbre des questions ait une telle profondeur que l'auteur n'accepte de répondre qu'aux questions cruciales. En d'autres termes, étant donnée une structure *mmc*, quelques réponses à des questions de désambiguïsation et, éventuellement, des préférences utilisateur, le système doit être capable de faire des choix et de produire ou bien une traduction unique, ou bien 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 et/ou soient capables de manipuler des structures ambiguës.

4.3 Créer des DAEs en langues cibles

Comme l'étape de génération produit une structure intermédiaire équivalente à une structure d'analyse désambiguïsée (*umc*), il suffit pour créer un DAE en langue cible de faire une analyse multiple (*mmc*) des phrases effectivement générées, puis de construire un arbre des questions concernant ces phrases. Sachant que l'on connaît la structure *umc* à retenir, on peut ensuite calculer automatiquement les réponses aux questions de désambiguïsation. On dispose

alors des informations nécessaires pour produire un DAE en langue cible, et ce sans aucune intervention humaine : le système aura ainsi effectué « la désambiguïsation que l'auteur aurait pu faire lui-même en langue cible » (en supposant qu'il la connaisse et ait la patience de la faire une seconde fois)

Atteindre cet objectif suppose qu'on dispose d'un analyseur adéquat (à sorties multiples) dans chacune des langues visées. C'est le cas de quelques enconvertisseurs vers UNL, ainsi que de certains analyseurs de systèmes de TA opérationnels (ALT/JE de NTT, PT de Linguatex, notamment). Nous espérons pouvoir bientôt expérimenter cette idée grâce à des coopérations avec certains d'entre eux.

Conclusion

Nous avons montré une première implémentation du concept de document auto-explicatif. Cette idée se situe dans le champ de recherche sur les documents actifs [Quint V. and Vatton I., 1994]. Nous travaillons sur un environnement LIDIA intégré à un éditeur de documents XML à la Thot (<http://opera.inrialpes.fr/Thot.en.html>).

Notre structure de document XML est assez simple, et toute l'information qu'il contient n'est pas au format XML. Par exemple, la structure *mmc* et l'arbre des questions sont représentés dans un formalisme à la lisp, ce qui nécessite des modules de gestion spécifiques, alors qu'un traitement avec l'API DOM serait plus efficace et robuste.

Cependant, ces deux premières étapes (le nouvel environnement LIDIA-2 et le visualiseur de DAE) représentent des résultats originaux, et les perspectives de ce travail sont variées. Nous espérons pourvoir exposer des résultats plus pratiques dans le futur.

Remerciements

Ghislain Gressard a développé la première implémentation de l'interface java de LIDIA. Eugénie Schonek a réalisé le premier visualiseur de DAE.

References

Notre proposition de construire des document auto-explicatifs semble tout-à-fait original, qu'il s'agisse du concept ou de réalisation comme celle montrée ici. En effet, nous n'avons rien trouvé à ce sujet dans la littérature. Cela explique que la bibliographie qui suit contienne très peu de références externes.

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

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

Boitet C. & Blanchon H. (1995). *Multilingual Dialogue-Based MT for monolingual authors: the LIDIA project and a first mockup*. in Machine Translation. vol. 9(2): pp. 99-132.

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

Fais L. & Blanchon H. (1996). *Ambiguities in Task-oriented Dialogues*. Proc. MIDDIM'96. Le col de porte, Isère, France, 12-14 Août 1996. vol. 1/1: pp. 263-275.

Quint V. & Vatton I. (1994). *Making structured documents active*. vol. 7(2): pp. 55-74.

Annexe : documents XML produits

Nous donnons simultanément les fichiers .ldi et .dae correspondant à l'exemple donné dans l'article. La structure complète des fichiers .ldi et .dae est donnée, mais nous avons supprimé l'analyse (liste d'arbres), trop volumineuse. Le contenu propre au fichier .ldi est en italiques. Il s'agit de la source, du timbre à date, de l'analyse et de la traduction.

```
<?xml version="1.0" ?>
<work>
  <description>
    <title><![CDATA[A trip to Tokyo]]></title>
    <language><![CDATA[ENG]]></language>
    <auteur>  <firstname><![CDATA[herve]]></firstname>
              <lastname><![CDATA[blanchon]]></lastname> </auteur>
  </description>
  <support>
    <paragraphe>
      <phrase source="ENG" stamp="11054803544635">
        <original><![CDATA[Good morning conference center.]]></original>
        <analyse><![CDATA[...]]></analyse>
        <traduction cible="FRA"><![CDATA[Bonjour ici le centre de conférences international.]]>
        </traduction>
      </phrase>
      <phrase source="ENG" stamp="21054803544655">
        <original><![CDATA[ I want the symposium on interpreting telecommunication at the
international conference center.]]></original>
        <question>
          <reformulation choix="NON"><![CDATA[the symposium (on interpreting telecommunications at
the international conference center)]]>
            <question>
              <reformulation><![CDATA[the (international center) for conference]]>
                <analyse><![CDATA[...]]></analyse>
              </reformulation>
              <reformulation><![CDATA[the center for (international conference)]]>
                <analyse><![CDATA[...]]></analyse>
              </reformulation>
            </question>
          </reformulation>
          <reformulation choix="OUI"><![CDATA[at the international conference center, the symposium
on interpreting telecommunications]]>
            <question>
              <reformulation choix="OUI"><![CDATA[the (international center) for conference]]>
                <analyse><![CDATA[...]]></analyse>
              </reformulation>
              <reformulation choix="NON"><![CDATA[the center for (international conference)]]>
                <analyse><![CDATA[...]]></analyse>
              </reformulation>
            </question>
          </reformulation>
        </question>
        <traduction cible="FRA"><![CDATA[Je veux le symposium sur la communication interprétée qui
se déroule au centre de conférences international.]]></traduction>
      </phrase>
      ...
      <phrase source="ENG" stamp="51054803544695">
        <original><![CDATA[ I will show you where you are located right now.]]></original>
        <question>
          <reformulation choix="NON"><![CDATA[I will show you (where you are located right now).]]>
            <analyse><![CDATA[...]]></analyse>
          </reformulation>
          <reformulation choix="OUI"><![CDATA[right now, I will show you where you are located.]]>
            <analyse><![CDATA[...]]></analyse>
          </reformulation>
        </question>
        <traduction cible="FRA"><![CDATA[Je vais tout de suite vous montrer où vous
êtes.]]></traduction>
      </phrase>
      ...
    </paragraphe>
  </support>
</work>
```