Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia

Cong-Phap Huynh

UNIVERSITÉ DE GRENOBLE

N° attribué par la bibliothèque
/__/__/__/__/__/__/__/__/__/__/

THESE

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : INFORMATIQUE

préparée au laboratoire LIG-GETALP (CNRS-INPG-UJF-UPMF)

dans le cadre de **l'École Doctorale "Mathématiques, Sciences et Technologies de l'Information"**
en cotutelle avec **l'Université de Danang (Vietnam)**

présentée et soutenue publiquement

par

Cong-Phap HUYNH

le 17 juin 2010

Des suites de test pour la TA à un système d'exploitation de corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia

Thèse encadrée par M. Hervé Blanchon
et codirigée par M. Georges Fafiotte et M. Khanh Phan-Huy (cotutelle)

JURY

M. Christian BOITET	Président
M. Christian FLUHR	Rapporteur
M. José ROUILLARD	Rapporteur
Mme. Françoise LÉTOUBLON	Examinateur
M. Bruno POULIQUEN	Examinateur
M. Hervé BLANCHON	Examinateur
M. Georges FAFIOTTE	Examinateur

Remerciements

Ce travail a été réalisé au sein de l'équipe GETALP (Groupe d'Étude pour la Traduction Automatique et le Traitement Automatisé des Langues et de la Parole) du Laboratoire d'Informatique de Grenoble (LIG) - France, et au sein de l'Institut National d'Informatique (NII) - Japon.

Je n'ai pas assez de place ici pour exprimer ma reconnaissance et ma gratitude à toutes les personnes qui m'ont aidé durant cette thèse. J'aimerais adresser mes plus sincères remerciements à :

Monsieur Christian Boitet, Professeur à l'Université Joseph Fourier, pour ses nombreux conseils prodigués, sa disponibilité, sa patience, sa confiance et son soutien de jour en jour pendant la thèse, et qui m'a fait l'honneur de présider le jury.

Monsieur Hervé Blanchon, Directeur du GETALP au Laboratoire d'Informatique de Grenoble, et Maître de Conférences HDR en Informatique à l'UPMF, pour la confiance qu'il m'a accordée pour mener à bien le sujet de recherche proposé, et pour m'avoir si bien encadré.

Monsieur Georges Fafiotte, Maître de Conférences en Informatique à l'UPMF, pour son accueil chaleureux qui m'a fait venir au GETALP, pour ses conseils, ses suggestions, sa disponibilité et son soutien durant ces trois années et six mois.

Messieurs Christian Fluhr, Professeur à l'Institut Supérieur Arabe de Traduction (Alger), et Directeur scientifique de Cadege/GEOLSemantics, et José Rouillard, chercheur au Laboratoire d'Informatique Fondamentale de Lille, et Maître de Conférences en Informatique HDR à l'Université des Sciences et Technologies de Lille, qui ont accepté avec bienveillance d'être rapporteurs et m'ont prodigué beaucoup de remarques pertinentes qui m'ont permis d'améliorer ce document.

Madame Françoise Létoublon, professeur à l'Université Stendhal (Grenoble 3) et IUF, Directrice du Centre d'Etudes Homériques, et Monsieur Bruno Pouliquen, chargé de recherche à WIPO (World Intellectual Property Organization (Genève)), et auparavant chercheur au JRC, qui ont accepté de participer à mon jury de thèse.

Monsieur Khanh Phan-Huy, Professeur Associé à l'Université de Danang-Vietnam, qui a encouragé mes efforts depuis mon DEA, et m'a encadré depuis Danang en tant que co-directeur de thèse en cotutelle.

Monsieur Asanobu Kitamoto, Professeur Associé à la Graduate University for Advanced Studies (Sokendai), et chercheur au NII, pour m'avoir accepté et encadré tout au long des 5 mois de mon stage de recherche au NII en 2009.

Monsieur Jean-Claude Durand, ingénieur de recherche au CNRS, qui a encouragé mes efforts, et a partagé mes difficultés personnelles durant mon séjour en France.

J'exprime aussi toute ma reconnaissance aux personnes qui m'ont aidé, à un moment ou à un autre de cette thèse, et en particulier à Hong-Thai Nguyen, Valérie Bellynck, Laurent Besacier, Jean-Philippe Guilbaud, Mathieu Mangeot, Hung Vo-Trung, Viet-Bac Le, Sereysethy Touch, David Rouquet, Achille Falaise, Mohammad Daoud, et Didier Schwab.

J'exprime également ma gratitude à la Faculté d''Informatique, à l'Ecole Polytechnique, et à l'Université de Danang pour leur soutien.

Je n'oublie pas mes amis en France, au Vietnam et au Japon pour leur présence dans les bons moments comme dans les moments difficultés.

Je tiens à remercier du fond du cœur ma famille, mes grands parents, mes parents, mes beaux parents, mes frères qui m’ont toujours soutenu dans la vie ainsi que dans mes études.

Finalement, je remercie très profondément ma femme, Nguyen Thi Bich, pour son amour, son soutien et son sacrifice durant une longue période d’éloignement lors de mon séjour en France et au Japon pour la réalisation de cette thèse.

Table des matières

Liste des abréviations

Ariane-G5	Environnement de développement et d'exécution de systèmes (linguiciels) de traduction automatique
AT	Aide au Traducteur
BDLM	Base de Données Lexicales Multilingues
DSR	Digital Silk Road
EDL	Environnement de Développement Linguiciel
ERIM	Environnement Réseau pour l'Interprétariat Multimodal
EOLSS	Encyclopedia of Life Support Systems
LSPL	Langage Spécialisé pour la Programmation Linguistique
K	x 1 000
M	x 1 000 000
LN	Langue Naturelle
IL	Interlingua
PIVAX	Base lexicale Web générique pour les systèmes de TA hétérogènes par Pivot d'Acceptions
iMAG	Passerelle Interactive d'Accès Multilingue
SI	Structure Interne
SA	Structure Abstraite
sectra	**s**ystème d'**e**xploitation de **c**orpus de **tra**ductions
SECTra_w	Le sectra développé au cours de la thèse
TRADOH	Un méta-système de traduction automatique multiple pour améliorer la compréhension
TA	Traduction Automatique
TAO	Traduction Automatisée par Ordinateur
TALN	Traitement Automatique des Langues Naturelles
TH	Traduction Humaine
THAM	Traduction Humaine Assistée par la Machine
TMX	Translation Memory eXchange
UNL	Universal Networking Language

Conventions typographiques

Style	**Texte cité**
Citation	Citation
`Code source ou exemples`	Code source ou exemples
`Algorithme`	Algorithme
termes	Terme utilisé dans un contexte ou un système

Liste des figures

Liste des tables

Introduction

En Traduction Automatique (TA), il existe plusieurs types de corpus : les suites de test, les corpus parallèles, les corpus enrichis par des annotations linguistiques et sémantiques, les corpus contenant des données multimédia, etc. Pour chaque type de corpus, il faut des outils spécifiques permettant de l'exploiter.

Cependant, les corpus ainsi que les outils associés ne sont souvent utilisés que par tel ou tel groupe de personnes dans tel ou tel laboratoire ou entreprise, mais ils ne sont pas disponibles pour l'usage par une plus grande communauté, et quelques fois, ils ne sont pas réutilisables. Or, nous savons que la construction de corpus et des outils spécifiques coûte très cher et est très longue.

Cet état de fait nous a conduit à nous poser la question suivant : pourquoi et comment rendre les corpus de traductions et les outils associés exploitables par des communautés intéressées plus larges ?

Pour répondre à cette question, nous nous proposons de construire un système unifié permettant d'exploiter les corpus de traductions, fournissant diverses fonctionnalités de traitement en fonction de chaque type de corpus, et favorisant la coopération de l'humain et de la machine.

Cette réponse est inspirée du principe général des systèmes d'exploitation. En effet, comme Windows ou Linux, un système d'exploitation de corpus de traductions, que nous appellerons un « sectra », exécute des tâches de base (telles que l'import et l'export de corpus), offre les services les plus importants en TA (tels que l'évaluation de TA, la post-édition, etc.), et fournit une plate-forme facilitant l'intégration, la délégation de services via des interfaces de programmation (API (Application Programming Interface)) et des méthodes simples de paramétrage et de programmation directe.

Cette thèse porte sur trois grands défis posés par la conception et la réalisation d'un sectra, c'est-à-dire d'un système informatique unifié destiné à l'exploitation de corpus de traductions, mise en œuvre à la fois par l'humain et par la machine. Le premier défi consite à offrir un support informatique unifié pour l'évaluation de systèmes de TA. Le deuxième défi concerne le support contributif et collaboratif au travail humain sur des corpus variés en contexte multilingue. Le troisième défi vise à permettre l'exploitation de corpus de traductions de très grandes tailles et de structures variées dans des applications novatrices, comme l'accès multilingue à des sites Web, en particulier culturels, et amélioration collaborative de résultats de TA.

Pour concevoir et réaliser un tel système, il faut traiter plusieurs problèmes difficiles. Au niveau conceptuel, il faut proposer une modélisation, et des définitions génériques (des corpus, des notions associées, des entrées, etc). Au niveau algorithmique, les problèmes posés sont le traitement de grands corpus, l'extension et la délégation de service. Au niveau programmatoire, les problèmes concernent, en particulier, le choix d'une architecture logicielle adaptée permettant à la fois la mutualisation du travail humain et la communication avec d'autres systèmes, la sécurité des données, la gestion des utilisateurs, le développement des API, etc.

L'organisation de cette thèse est la suivante :

Dans le chapitre 1, nous nous attaquons à l'exploitation de corpus pour l'évaluation de systèmes de TA. Il s'agit essentiellement de problèmes de support informatique à la

construction et à l'exploitation de corpus d'évaluation, à l'organisation de campagnes d'évaluation, et à l'évaluation subjective et objective. Pour analyser ces problèmes, nous proposons plusieurs notions, des principes généraux, et des solutions au niveaux conceptuel, algorithmique, et programmatoire. Nous décrivons aussi la conception, le développement et l'expérimentation de la partie de notre sectra, SECTra_w, qui a été utilisée dans la campagne d'évaluation de TA du projet TRANSAT avec FT R&D (voir I.3.3).

Dans le chapitre 2, nous étudions les problèmes liés au défi qui consiste à construire un support contributif et collaboratif au travail humain sur des corpus variés en contexte multilingue. Il s'agit de mettre en place des aides informatiques (par exemple, des moyens de communication entre les contributeurs) et des aides linguistiques (par exemple, l'accès à des informations lexicales) pour l'exploitation de corpus de traductions. Pour mieux analyser ces problèmes, nous proposons plusieurs notions, et des principes généraux comme le principe de délégation de fonctions à des services extérieurs, le principe de proactivité des aides linguistiques, etc. Nous développons ces solutions et ces principes dans SECTra_w et les évaluons dans le contexte de deux vrais projets de post-édition contributive de corpus : EOLSS/UnescoL et DSR (voir II.3.3.1) .

Dans le chapitre 3, nous nous concentrons sur l'exploitation de corpus de traductions dans des applications novatrices. Il s'agit essentiellement de trouver comment permettre à d'autres systèmes d'exploitation de corpus et de ressources traductionnelles d'utiliser un sectra. Pour résoudre ces problèmes, nous proposons des principes et des solutions, en particulier au niveau de la définition et de la gestion d'une vraie mémoire de traductions formée de segments multilingualisés et multicontextualisés, et pour l'usage par des systèmes extérieurs. Nous proposons aussi une architecture logicielle adaptée facilitant l'échange et la communication entre SECTra_w et d'autres systèmes. Nous développons et expérimentons ces solutions et principes dans le cadre de trois projets : OMNIA, iMAG, et MIC-Notepad++ (voir III.3.2.1).

Chapitre I
Support informatique unifié pour l'évaluation de systèmes de TA

Introduction

Depuis 2000, notre équipe a participé à des campagnes d'évaluation (IWSLT-04, IWSLT-06, et TRANSAT). Dans les campagnes de ce genre, les participants doivent effectuer deux tâches principales : la première consiste en la construction de corpus multilingues alignés d'énoncés, et la deuxième consiste en l'organisation de campagnes d'évaluation compétitive de systèmes de traduction automatique (TA) sur les corpus construits.

Pour l'organisation d'évaluations compétitives, les participants utilisent souvent des outils fournis par les organisateurs (par exemple le NIST). Ces outils fournissent seulement des interfaces pour faire l'évaluation subjective, et les participants ne peuvent jamais voir les phrases *réelles* avec les résultats des méthodes d'évaluation objective. Cependant, les interfaces d'évaluation subjective sont figées, avec des critères prédéfinis par les organisateurs, et normalement on ne peut plus y accéder après la fermeture de ces campagnes. Par conséquent, les participants ne peuvent pas utiliser ces outils pour organiser leurs propres campagnes d'évaluation pour des projets internes dans un but de recherche.

Pour construire les corpus multilingues alignés d'énoncés utilisés dans ces campagnes, on procéde souvent par extraction à partir de corpus existants (des livres de phrases, ou le corpus BTEC [Kikui et al., 2003 ; Takezawa et al., 2002]), en les soumettant aux systèmes de TA à évaluer, et en les post-éditant. Cependant, ces campagnes ne fournissent jamais de système facilitant cette tâche. On a souvent effectué cette tâche en utilisant des outils qui n'ont pas été conçus pour elle, tels qu'Excel, MS Word, etc. car on n'a pas non plus trouvé ailleurs de système de gestion de corpus facilitant cette tâche.

D'autre part, nous souhaitons depuis longtemps disposer d'un système puissant qui permette de gérer les corpus utilisables pour la TA. Ces corpus comprennent des suites de test pour la TA, des corpus multilingues bruités, des corpus enrichis d'annotations linguistiques ou sémantiques, ou des corpus alignés de documents et métadocuments multilingues, multiannotés et multimédia.

Il nous est donc apparu intéressant de trouver comment construire un système unifié qui offre un support informatique adapté aux campagnes d'évaluation, et à l'exploitation de corpus. On rencontre plusieurs problèmes difficiles quand on veut relever ce défi, ce qui explique sans doute que, malgré les besoins importants, il n'existe pas encore de tel système.

Pour pouvoir construire un tel système, nous étudions d'abord la gestion de corpus dans des Environnements de Développement Linguiciel (EDL) de TA, le support informatique de campagnes d'évaluation compétitives récentes, et le support informatique pour l'évaluation de systèmes « en opération ». Nous analysons ensuite les objectifs et les problèmes émergents avant de proposer plusieurs notions nouvelles, des principes généraux, et des solutions à ces problèmes. Avant de passer à l'implémentation, nous spécifions les fonctionnalités et les caractéristiques nécessaires de ce système. Enfin, nous présentons notre expérience avec ce système dans une campagne d'évaluation réelle effectuée dans le cadre du projet TRANSAT en 2007. Nous terminons ce chapitre avec une évaluation montrant que le défi a été relevé.

I.1 État de l'art et problèmes émergents

I.1.1 État de l'art

I.1.1.1 Gestion de corpus dans les EDL de TA

I.1.1.1.1 Situation générale

Les corpus utilisés en TA ont évolué, depuis les suites de test et les corpus d'essai des débuts, vers des corpus parallèles bilingues ou multilingues, bruts ou enrichis par des métadonnées et une grande variété d'annotations linguistiques [Boitet, 2007]. Nous étudions donc la gestion des corpus selon l'évolution des corpus utilisés dans les systèmes de traduction automatique et leurs environnements de développement de linguiciels (EDL).

a. Suites de test depuis toujours

Les suites de test sont utilisées depuis les débuts de la TA. Un texte comprend une ou plusieurs phrase(s) de test, non reliées, correspondant à des phénomènes linguistiques précis. La gestion des suites de test est assez simple dans les EDL de TA. Nous illustrons cela avec Ariane-G5 et METAL.

Ariane-G5. Ariane-G5 [Boitet, 1993b] est un EDL pour la TAO. Cet environnement est construit autour de cinq *langages spécialisés pour la programmation linguistique* (LSPL).

Dans Ariane-G5 (Figure 1), on peut développer de façon modulaire non régressive toutes les "phases", car les LSPL utilisés ont des niveaux de modularité. Par exemple, un format morphologique correspond à une sous-grammaire en ATEF, et un système transformationnel écrit en ROBRA contient un graphe de contrôle et des grammaires transformationnelles ne contenant pas plus d'une vingtaine de règles en moyenne. Cependant, le logiciel n'impose pas tel ou tel type de développement.

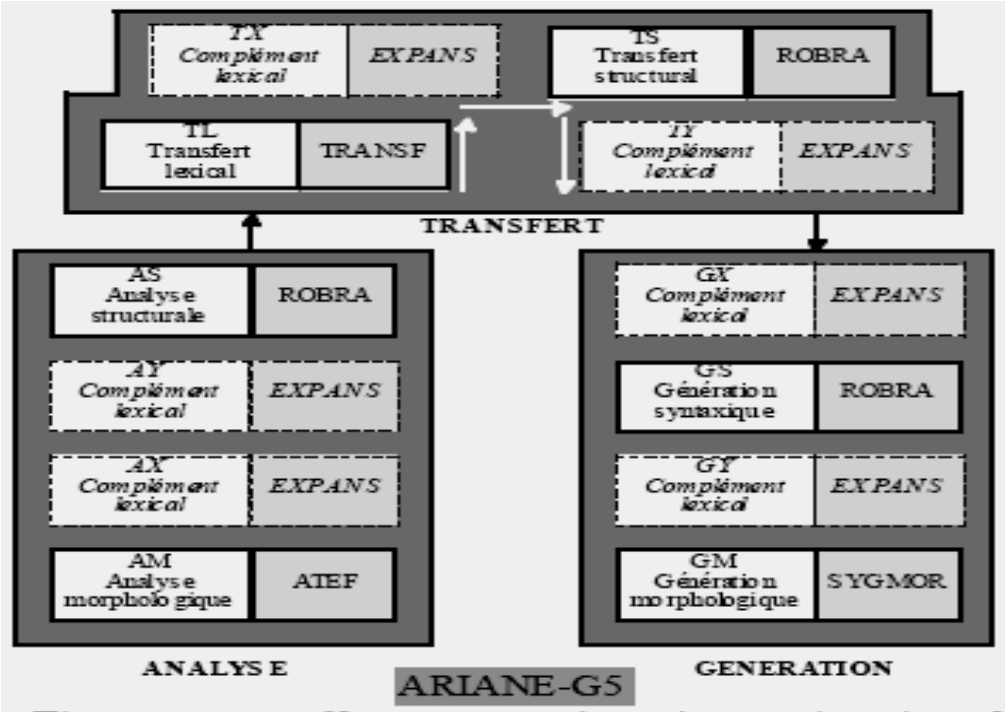


Figure 1: Étapes/Phases dans Ariane-G5

Si l'on suit la méthodologie de développement linguiciel de B.Vauquois (Figure 1), à l'aide de « planches de grammaires statiques » (GSCS) [Chappuy, 83 ; Vauquois et Chappuy, 85], un

corpus de test correspond aux phrases exemples d'une "planche", et éventuellement à d'autres phrases similaires.

Une planche est une règle de correspondance entre une famille de chaînes et une famille d'arbres, associées à un type de syntagme précis. Un treillis est défini sur les classes syntagmatiques (PHVB, PHSUB, PHINF, PHGER, GN, GA, GADV, etc.). Une *planche* correspond à un syntagme *élémentaire* (ne contenant pas de syntagme supérieur ou égal dans le treillis), *simple* (pouvant contenir un ou des syntagmes de même niveau, ex: "la ville de Paris"), ou *complexe* (pouvant contenir des syntagmes supérieurs, par exemple: "la ville qu'il voit").

METAL. METAL [Slocum et al., 1985] est un système de traduction automatique créé initialement par J. Slocum et son équipe à l'université du Texas à Austin en 1981-84, grâce à un contrat de Siemens (déçue de n'avoir pas obtenu de résultats satisfaisants pour l'allemand-anglais commandé à Logos les années précédentes). Il a d'abord été utilisé par Siemens et étendu à d'autres couples de langues, puis mis en service opérationnel en 1985[1].

Une grande qualité de ce système est que le développement modulaire non régressif est prévu dans le logiciel pour l'étape la plus difficile, l'analyse.

On construit successivement les grammaires de niveau 0, 1, 2…, N. La grammaire de niveau i contient toutes les règles de niveau j≤i. Pour analyser une phrase, on la soumet successivement aux grammaires G0, G1… jusqu'à obtenir un succès. Si Gk est la grammaire analysant ainsi la phrase S, on dit que S est de niveau k. Pour développer de façon monotone, on développe successivement les niveaux 0, 1,…, N, en construisant des suites de test pour chaque niveau. On est ainsi assuré qu'une phrase de niveau k donnera toujours la ou les mêmes analyses quand on passe au développement des règles de niveau k+1 [Slocum et al., 1985].

b. Corpus de textes pour les tests

Corpus de textes pour les tests. Les textes de ces corpus sont différents des suites de test, car il s'agit ici de *texte connecté* (paragraphe, section, page). On les utilise pour deux raisons:

- évaluation de la « qualité globale » d'un texte traduit. En effet, un texte reconstitué est toujours meilleur que la liste de ses phrases.
- dans certains cas, le système de TA peut traiter plus d'un segment dans une *unité de traduction* (par exemple plusieurs paragraphes ou plusieurs pages, comme en Ariane-G5).

c. Corpus applicatifs

Corpus du CETA (1961-1970). Les essais portèrent sur un corpus russe donné par la Rand Corporation (financée par l'USAF (US Air Force)), de près de 400000 mots, appliqué principalement à la TA russe-français.

Corpus du GETA et de B'Vital (1974-1992). Le GETA, issu du CETA, se tourna vers la TA de grande qualité de sous-langages, dans le cadre de contrats avec la DRET du Ministère de la

[1] De rares clients l'ont acheté à ce moment, pour l'allemand-anglais, seul couple disponible, comme SAP. Siemens a développé et fait développer d'autres couples de langues, concernant le français et l'espagnol, puis a cédé METAL à sa filiale SIETECH, qui l'a à son tour cédé à GMS (Gesellschaft für Multilinguale Systeme, Berlin) et à L&H (Lernhout & Hauspie, NL). Après la faillite de L&H, METAL a été racheté par Comprendium, lui-même racheté en 2008 par Lucy Software, dirigée par D. Grasmick, ex-directeur de la traduction pendant 20 ans chez SAP.

Défense2 (1974-87), puis du projet national de TAO (PN TAO, 1982-87), en liaison avec des industriels, dont la jeune pousse B'VITAL (1985-1992). Le GETA travailla sur de nouveaux corpus, saisis par le labo (bulletins signalétiques du "реферативный журнал" (referativnyij zhurnal, ou "journal référatif") du VINITI, articles scientifiques et techniques, et autres, choisis et transmis par le CEDOCAR de la DRET. Les industriels travaillèrent sur des corpus de type manuels d'entretien d'avions.

Dans ce genre de situation, un corpus peut correspondre à un lot de travail à réaliser (par exemple à un envoi de la DRET), et un texte à un document élémentaire (bulletin signalétique, presque toujours de moins d'une page, ou article de quelques pages).

Systran (1967—). On sait peu de choses à propose de gestion de corpus chez Systran. Il y a des suites de test, par phénomènes linguistiques. Il y a aussi des corpus de textes, et plus récemment de pages Web et de fichiers pdf.

Il semble que, tous types confondus, la taille d'un corpus de test (dans une langue source donnée, pour une version "généraliste", comprenne environ 1000 à 2000 pages (250K à 500K mots). D'après nos informations, il faudrait plusieurs mois pour dépouiller les résultats d'un test complet, prérequis de la mise sur le marché d'une nouvelle version (release), par exemple pour le passage de Systran v5 à Systran v6. Nous parlerons plus loin du support informatique.

d. Gros corpus parallèles bruts dans les systèmes de traduction statistique (SMT) (1990—)

Les EDL sont dans ce cas plus simples que pour les systèmes de TA experts. Mais ils offrent une gestion de gros et même très gros corpus. On peut trouver de gros corpus parallèles disponibles gratuitement, par exemple:

Nom de corpus	Taille en mots (total)	Nombre de langues	Taille moyenne par langue
EuroParl	407.069.444	11	37.006.313
Hansard	47.389.000	2	23.694.500
JR Acquis	1.055.583.954	22	47.981.089
XinHua News	29.000.000	2	14.500.000
OPUS	30.000.000	60	500.000

Table 1: Exemples de gros corpus de traduction

On trouve maintenant des environnements permettant de créer des systèmes de traduction probabiliste, tels que Pharaoh, Matrax, Moses, Joshua, etc., utilisant des corpus parallèles. Cependant, il n'y a pas de vrai support logiciel aux corpus fourni par ces systèmes, bien qu'on trouve des sites Web permettant de télécharger des corpus.

e. Corpus enrichis d'annotations linguistiques ou sémantiques (TA experte ou EBMT)

On utilise de gros corpus enrichis par des annotations plus ou moins complexes dans des variantes de TA « par les exemples », et de gros corpus multilingues contenant des représentations interlingues UNL dans des systèmes de TA « experte ».

Corpus de S-SSTC. Un corpus de S-SSTC (Synchronized Structured String-Tree Correspondences) (voir Figure 2) est directement utilisé dans le système "EBMT" (Example-Based MT) de l'Universiti Sains Malaysia (UMS), Penang.

[2] Ces bulletins ont une structure particulière, et on a construit une variante de l'analyseur du russe qui en tient compte pour appliquer des grammaires locales adaptées (sur l'arbre décoré représentant le texte) aux différents *sous-langages* rencontrés (titre, date, liste d'auteurs, références bibliographiques, et corps du texte).

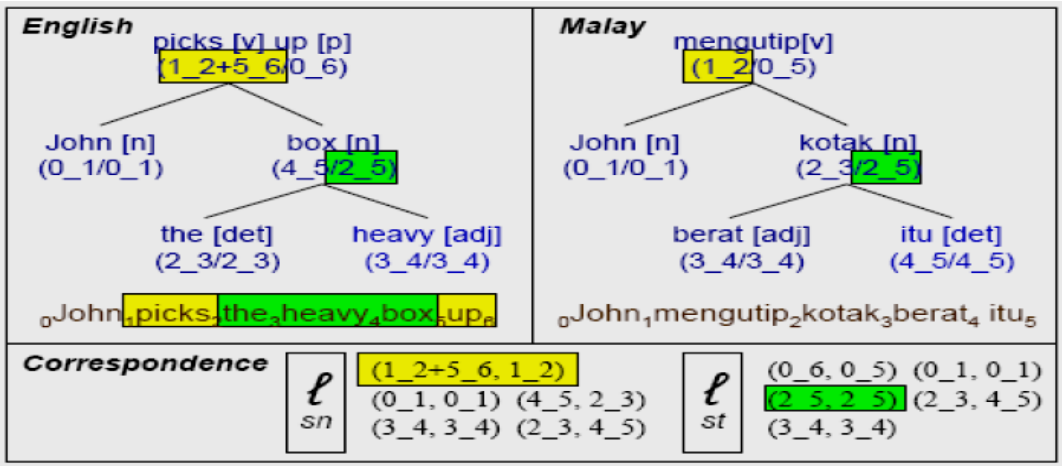


Figure 2: Exemple complet d'une S-SSTC

Des structures de correspondances S-SSTC ont été développées pour synchroniser environ 15000 exemples anglais-malais [Al-Adhaileh et Kong, 1999]. Une extension au malais-chinois est en cours (projet Brain Gain, 2010).

Corpus contenant plusieurs langues et des graphes UNL. Les corpus UNL (Universal Networking Language) sont construits pour valider des systèmes de TA à base d'enconvertisseurs et de déconvertisseurs vers et depuis UNL. Ils contiennent des hypergraphes UNL servant de « pivot » pour passer d'une langue à une autre (Figure 3). Un (hyper-)graphe UNL est une structure sémantique abstraite d'un énoncé anglais équivalent à l'énoncé à traduire [Tsai, 2004].

Voici l'exemple d'un segment dans un corpus enrichi multilingue au format UNL contenant 3 langues et un graphe UNL.

{org:es}
Los ríos dejaron prácticamente de llegar, taponados por presas.
{/org}
{unl}
obj(flow(icl>occur).@past.@entry.@not,
river.@def.@pl)
man(flow(icl>occur).@past.@entry.@not, almost)
rsn(flow(icl>occur).@past.@entry.@not, :01)
obj:01(block(icl>do).@past.@entry, river.@def.@pl)
agt:01(block(icl>do).@past.@entry, dam.@pl)
{/unl}
{fr} Bloquées par des barrages, les rivières ne coulaient presque plus.{/fr}
{el} Blocked by dams, the rivers almost stopped flowing.{/el}

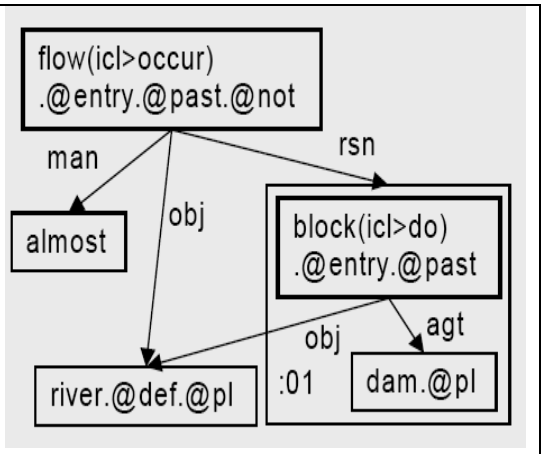


Figure 3: Exemple d'un graphe UNL

Du point de vue du support informatique à la gestion de tels corpus, il y a quelques fonctions dans CASH [Blanc, 1999 ; Nguyen, 2009], mais pas d'outil distribué par UNDL[3] ou U++[4] pour vraiment gérer les corpus UNL. Nous nous intéresserons donc à stocker, visualiser et

[3] UNDL : voir http://www.undl.org/

[4] U++ : voir http://www.unl.fi.upm.es/consorcio/index.php

éventuellement éditer des graphes UNL se présentant comme des annotations des énoncés (en langue naturelle) d'un corpus. Il devrait aussi être possible d'appeler les enconvertisseurs et déconvertisseurs disponibles pour créer les graphes UNL associés aux énoncés, puis obtenir leurs traductions dans diverses langues, avant de les évaluer.

CASH : ce « méta-EDL » de TAO a été écrit en HyperCard par Etienne Blanc [Blanc, 1999]. Il permet à l'utilisateur de disposer sur sa machine des copies des fichiers source des linguiciels présents dans la machine (IBM H30) sur laquelle tourne Ariane-G5 (grammaires, dictionnaires…). L'utilisateur peut modifier ces linguiciels, ou en créer de nouveaux, et les renvoyer ensuite à Ariane-G5 pour test ou exécution. Il peut également effectuer une mise à jour de ses copies en demandant à Ariane-G5 l'envoi des données de tel ou tel linguiciel. Le développement peut ainsi s'effectuer de manière déportée sur CASH, les échanges avec la machine se faisant par protocole SMTP (courriel), http (Web), ou par protocole spécifique (CommSwitch de CSTAR) sur des sockets [Blanchon, 1999].

I.1.1.1.2 L'exemple d'Ariane-G5

La gestion des corpus en Ariane-G5 est organisée à deux niveaux principaux, celui des corpus et celui des textes de chaque corpus.

a. Définition et manipulation

Nom. Un corpus dans Ariane-G5 est repéré par un nom de 1 à 8 caractères, et ses textes sont contenus dans des fichiers dont le nom est le même que celui du corpus[5]. Chaque texte contient des informations supplémentaires pour permettre de l'identifier. Par exemple, lors de l'expérience russe - français pour la DRET et le CEDOCAR (1980-1987), DRET C0302002 A1 représentait le texte numéro 002 dans le corpus nommé DRET (4 caractères), reçu le 03 février 1984 (C signifiait la 3ème année du contrat).

Méthode de structuration externe d'un corpus. Un corpus en Ariane-G5 contient la liste des textes source, et une liste de séparateurs qui définissent une structuration externe. Ces séparateurs sont des « occurrences » servant de marqueurs de fin de groupe. La segmentation des textes source en unités de traduction est faite à partir de cette structuration externe. À chaque fichier source est associé un ensemble d'objets dérivés: (1) le fichier transcrit avec sa structure externe, (2) les représentations intermédiaires, (3) les traductions, avec un fichier par variante utilisée, (4) et les révisions.

Il était prévu qu'à chaque texte source soit associés sa forme translittérée, et un fichier de hors-texte, sorte de dictionnaire local associant par exemple à $$EXPMATH_1 la chaîne $x \leq \sqrt[3]{y^{n+1}}$$[6]. Un résultat intermédiaire d'un texte est une liste d'arbres décorés accompagnée d'un descripteur (variante du linguiciel, liste des unités de traduction représentées). Une traduction d'un texte est un texte translittéré accompagné d'un descripteur similaire [Boitet, 2007].

Textes source. Les textes source dans un corpus d'Ariane-G5 sont représentés par une liste faisant la correspondance entre nom externe CMS (ftype, fnom) et nom interne (un code sur 2 ou 3 caractères). Chaque texte source est stocké dans sa forme translittérée. Utiliser une translittération peut réduire la taille de l'analyseur et du générateur morphologiques [Nguyen, 2009].

[5] Un fichier CMS est défini par un nom, un type, et un mode, par exemple NOMFICH TYPEFICH A1, le nom et le type étant des symboles de 1 à 8 caractères EBCDIC.

[6] Code TeX pour $x \leq \sqrt[3]{y^{n+1}}$.

b. Représentations intermédiaires

Un résultat intermédiaire d'un texte est une liste d'arbres décorés accompagnée d'un descripteur (variante du linguiciel, liste des unités de traduction représentées). Ce résultat intermédiaire est utilisé comme format d'échange entre les phases du processus de traduction.

c. Textes traduits et textes révisés

Une unité de traduction en Ariane-G5 consiste en général en un ou plusieurs paragraphes, voire même une ou plusieurs pages.

Les textes traduits « bruts » sont donc aussi des « blocs » de texte qui sont les textes translittérés accompagnés des descripteurs correspondants. Pour la révision d'un texte traduit brut, il y a un seul fichier de révision par langue et par chaîne de traduction. Il n'y a pas de gestion de versions, seule la dernière modification est prise en compte. Il n'y a pas non plus de réinsertion des éléments hors-texte, ce développement ayant été interrompu pour d'autres, liés aux nouveaux axes de recherche (TAFD[7] avec le projet LIDIA, puis traduction de parole, puis UNL, puis méta-EDL avec CASH et WICALE).

d. Prétraitements et post-traitements

Entrée. Une entrée du processus de traduction est une simple chaîne de caractères. Les 256 caractères EBCDIC[8] sont considérés comme atomiques, et tous sont admis pour former les chaînes des langages spécialisés. Le blanc (X'40') est utilisé comme séparateur d'occurrences.

Sortie. Dans la plupart des systèmes ou protocoles de TA, la sortie de la phase de génération morphologique (GM) est en transcription "minimale" (choix des linguistes), par exemple "A!2, *A!2" pour "à, À".

Ariane fait un peu plus en prenant en compte l'usage le plus fréquent: transcription "intermédiaire" pour visualiser les textes source (russe : maj/min mais transcription latine) et les textes traduits (français : maj/min mais séquences spéciales pour les diacritiques en traduction (a!2, A!2), et transcription normale pour la révision — avec les diacritiques : "à, À").

Conclusion

Il n'y a pas encore de système de gestion de corpus vraiment complet dans les EDL de TA. La gestion des suites de test y est organisée assez simplement. Pour les corpus parallèles bruts, il existe seulement des sites Web permettant de télécharger des fichiers, comme http://www.statmt.org/europarl/ et http://iwslt07.fbk.eu/menu/resources.html pour les corpus EuroParl, BTEC. Pour les corpus enrichis d'annotations linguistiques ou sémantiques, il y a quelques systèmes comme CASH et UNLdeco [Sérasset et Boitet, 1999] offrant un nombre restreint d'opérations sur les corpus contenant des graphes UNL.

Ariane-G5 contient un environnement de gestion de corpus de textes, mais il ne gère pas de corpus de documents, ni de mémoire de traduction (MT). On peut réviser et post-éditer les textes traduits bruts, mais il n'y a pas de gestion de version, et on ne peut pas gérer et post-éditer de « petits segments » individuellement. Surtout, on ne peut pas travailler de façon collaborative et contributive sur le corpus parce que ce système n'est pas une plate-forme Web.

[7] TA Fondée sur le Dialogue avec l'auteur [Blanchon, 94].

[8] EBCDIC (Extended Binary Coded Decimal Interchange Code) est un codage des caractères sur 8 bits créé par IBM.

I.1.1.2 Gestion et support informatique de campagnes d'évaluation compétitives

Un système de support informatique de campagnes d'évaluation compétitive doit supporter la construction de corpus d'évaluation, l'évaluation objective, et l'évaluation subjective pendant et après une campagne d'évaluation.

I.1.1.2.1 Historique

Depuis quelques années, des campagnes d'évaluation compétitive ont été organisées soit par les bailleurs de fonds (DARPA, UE, Académie des Sciences Chinoise), soit à leur incitation et avec leur soutien (NIST, ELDA/ELRA pour la campagne CESTA), soit par des consortiums dans le cadre de projets coopératifs (CSTAR en 1999, projet NESPOLE! en 2002 et 2003, CSTAR de nouveau avec IWSLT depuis 2004, TC-STAR en 2006). Au début, on n'a eu que des jugements humains. En 2002, on a ajouté les mesures automatiques BLEU, NIST, puis beaucoup d'autres [Blanchon et Boitet, 2007].

Il y a aussi des participations humaines à des calculs objectifs avec plus ou moins de post-éditions dans HWER (dans la campagne GALE).

I.1.1.2.2 Supports informatiques pour les campagnes d'évaluation compétitive (2002-2008)

a. Construction de corpus d'évaluation

Du point de vue du support informatique pour construire des corpus pour les campagnes d'évaluation, on peut distinguer les trois tâches suivantes.

- **Extraction de corpus source**. On construit souvent les corpus source pour les campagnes d'évaluation en extrayant des segments source à partir d'un ou plusieurs corpus existants. Par exemple, on a extrait des corpus source à partir du corpus BTEC pour les campagnes IWSLT de 2004 à 2009 [IWSLT, 2009]. Cependant, les organisateurs (ATR, NIST, etc.) ne semblent pas disposer de système facilitant cette extraction en fusionnant et visualisant plusieurs corpus pour faciliter la sélection, en fournissant des fonctions de filtrage des segments en fonction de critères quelconques, etc.
- **Construction de traductions candidates**. Les traductions candidates sont fabriquées par la soumission des textes source aux systèmes de TA à évaluer. Cette tâche est le plus souvent faite manuellement. On construit parfois des fichiers batch pour lancer les systèmes de TA. Par exemple, Ch. Boitet l'a fait en 2004 avec Systran pour tout le corpus BTEC, à l'occasion de la campagne d'évaluation IWSLT-04 [Boitet, 2004].
- **Construction de traductions de référence**. Les traductions de référence sont souvent créées par post-édition des traductions candidates. Cependant, les campagnes d'évaluation n'ont encore presque jamais fourni d'outil de support à la post-édition. C'est pourquoi l'on a souvent utilisé Excel ou Word pour cette tâche, alors que ces logiciels n'ont pas été conçus pour la post-édition. La campagne GALE est une exception : il y a un outil appelé *MT post-editor* (Figure 4) destiné à la post-édition, mais cet outil est assez simple. Il ne s'agit pas d'un vrai système de support à la post-édition. En effet, pour qu'il le soit, il faudrait qu'il offre les mêmes aides que les systèmes d'aides au traducteur comme Trados [Trados, 2005], Déjà Vu [DéjàVu, 2010] ou Similis [Similis, 2010].

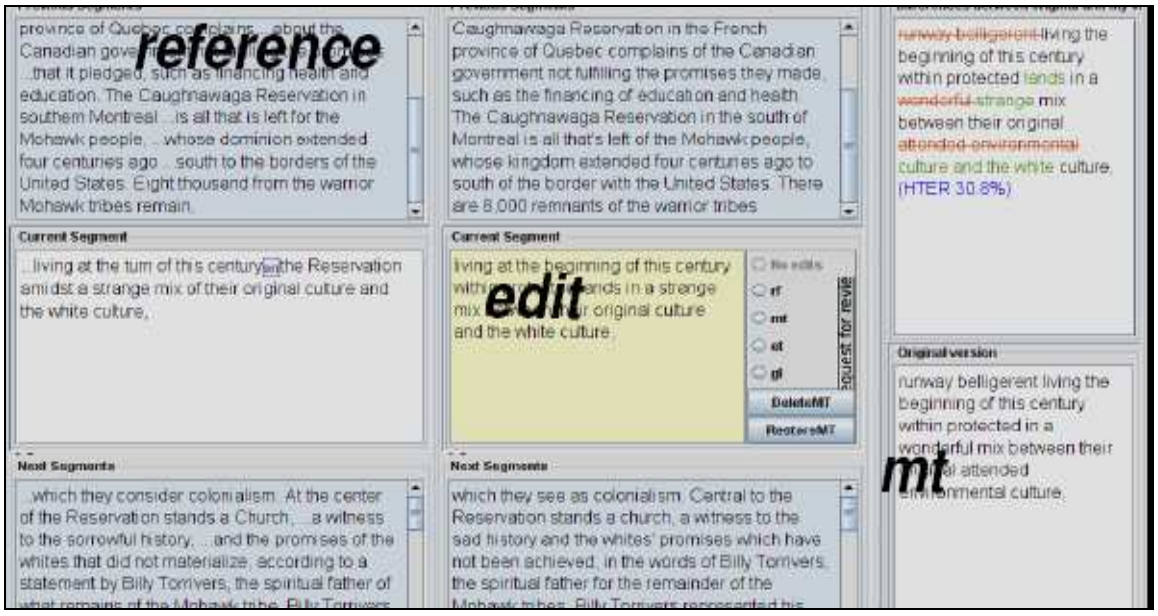


Figure 4: "MT posteditor" de la campagne GALE

b. *Evaluation objective*

La plupart des campagnes d'évaluation compétitive offrent des méthodes d'évaluation objective comme BLEU, NIST, mWER, etc. Cependant, les résultats détaillés de ces méthodes, associés aux données réelles, ne sont accessibles qu'aux organisateurs des campagnes d'évaluation. Ils sont inaccessibles aux participants qui ne peuvent donc pas voir les résultats sur les textes eux-mêmes (et pas seulement condensés dans des tableaux de chiffres), les tester ou les faire recalculer sur une portion de données ayant été mise à jour.

c. *Evaluation subjective*

Les campagnes d'évaluation mentionnées ci-dessus offrent des supports plus ou moins adéquats à l'évaluation subjective pendant et après les campagnes d'évaluation.

Pendant les campagnes d'évaluation telles que IWSLT, les participants doivent suivre un protocole bien précis exigé par les organisateurs. L'évaluation subjective dans ces campagnes d'évaluation est de deux types : fluidité et adéquation. Pour lancer ces évaluations, ces campagnes d'évaluation proposent généralement deux interfaces Web. La Figure 5 montre l'interface d'évaluation de la fluidité proposée par IWSLT-04.

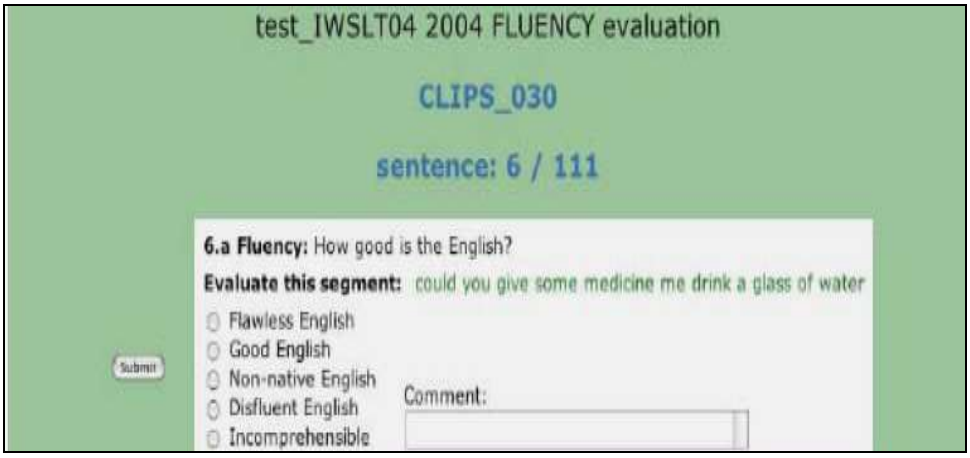


Figure 5: Interface d'évaluation de la fluidité pour IWSLT-04

Comme dans l'interface ci-dessus, un lot de phrases (111 phrases) est alloué à un évaluateur. A chaque fois, l'évaluateur ne voit qu'une phrase à évaluer (phrase 6). L'interface ne présente que la traduction candidate et elle cache la traduction de référence. L'évaluateur doit classer le niveau de langue de la traduction candidate en choisissant une valeur parmi : *Flawless English, Good English, Non-native English, Disfluent English, Incomprehensible*.

La Figure 6 montre l'interface d'évaluation de l'adéquation, qui présente à l'évaluateur humain la traduction de référence ainsi que la traduction candidate.

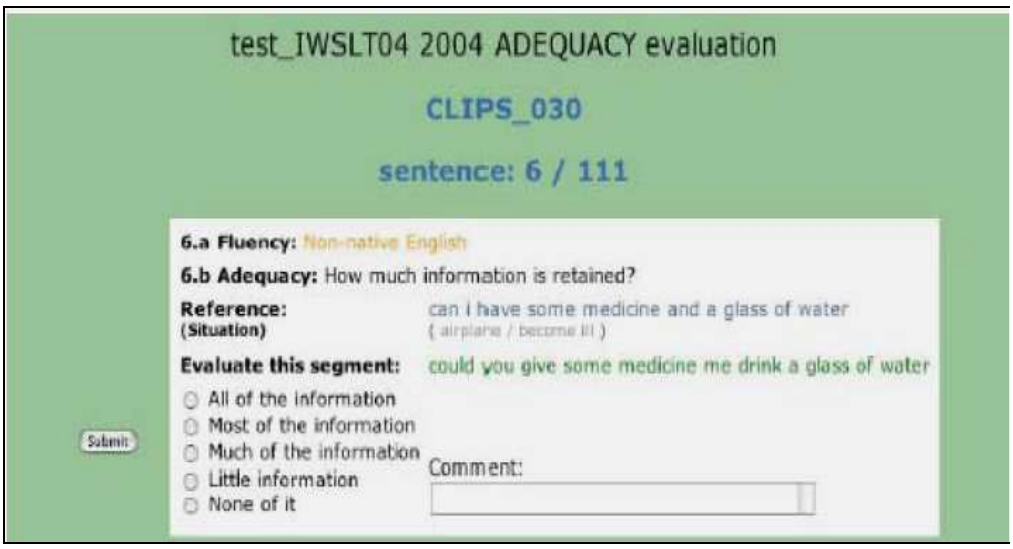


Figure 6: Interface d'évaluation d'adéquation pour IWSLT-04

L'évaluateur doit juger quelle proportion de l'information est transmise par la phrase candidate en la comparant avec une traduction de référence. Il doit choisir une valeur parmi: Toute l'information est transportée (*All of the information*), Presque toute l'information est transportée (*Most of the information*), La moitié de l'information est transportée (*Much of the information*), Peu d'information est transportée (*Little information*), et Aucune information est transportée ou contresens (*None of it*).

Ces interfaces, bien qu'elles soient fonctionnelles, présentent quelques limitations. La présentation est figée. L'interface présente à un évaluateur une phrase à la fois, ce qui rend la tâche d'évaluation très pénible et inefficace pour évaluer des phrases liées (à partir d'un dialogue ou d'un document).

Un défaut majeur est que, après des campagnes d'évaluation telles que IWSLT, on ne peut pas voir les résultats d'évaluation subjective. Les participants (et a priori les autres développeurs) ne peuvent pas voir ces résultats et les utiliser pour améliorer leurs systèmes.

Enfin, les serveurs d'évaluation de ces campagnes sont fermés rapidement, au plus après quelques mois. Cela pose un problème lorsqu'un évaluateur manque de temps (par exemple en période de vacances), et qu'il ne peut pas continuer après pour compléter sa partie. Ces campagnes d'évaluation ne permettent pas non plus de réutiliser les mêmes données pour refaire des campagnes d'évaluation.

Conclusion

Bien qu'il existe des systèmes pour le support des campagnes d'évaluation, ces systèmes présentent de nombreuses limitations.

Un système de support à l'évaluation de TA doit faciliter et accélérer la construction de corpus d'évaluation, et permettre aux organisateurs ainsi qu'aux participants de voir les

résultats détaillés d'évaluation objective associés aux données réelles. Il doit permettre aussi aux participants de configurer l'interface selon leur besoin (définir les boutons à choisir correspondant aux critères utilisés dans les campagnes, définir le nombre de segments visualisés sur une interface). Enfin, il doit permettre d'effectuer l'évaluation liée à la tâche par la post-édition de TA.

I.1.1.3 Support à l'évaluation de systèmes de TA en opération

Un support logiciel à l'évaluation de systèmes de TA en opération doit éventuellement permettre d'appeler k systèmes de TA sur le même texte ou le même document ou la même page Web, mais cela ne suffit pas. Il faut aussi pouvoir montrer les différences, permettre de post-éditer, et mesurer le temps passé, etc.

L'évaluation de systèmes de TA en opération est faite soit par les développeurs ou les éditeurs de ces systèmes, soit par l'extérieur.

I.1.1.3.1 Evaluation par les développeurs/éditeurs

a. Systran

Systran [Systran, 2009] fournit un outil appelé SYSTRAN Review Manager (SRM) [Costa et Panissod, 2003]. Le groupe de développement de Systran l'utilise en interne pour réviser et évaluer la qualité de son produit.

Ce système fournit une interface Web permettant à des utilisateurs disséminés géographiquement d'y accéder. Cette interface permet de réviser et d'évaluer un corpus traduit, segmenté en unités de traduction, en choisissant des critères définis par l'administrateur (Figure 7).

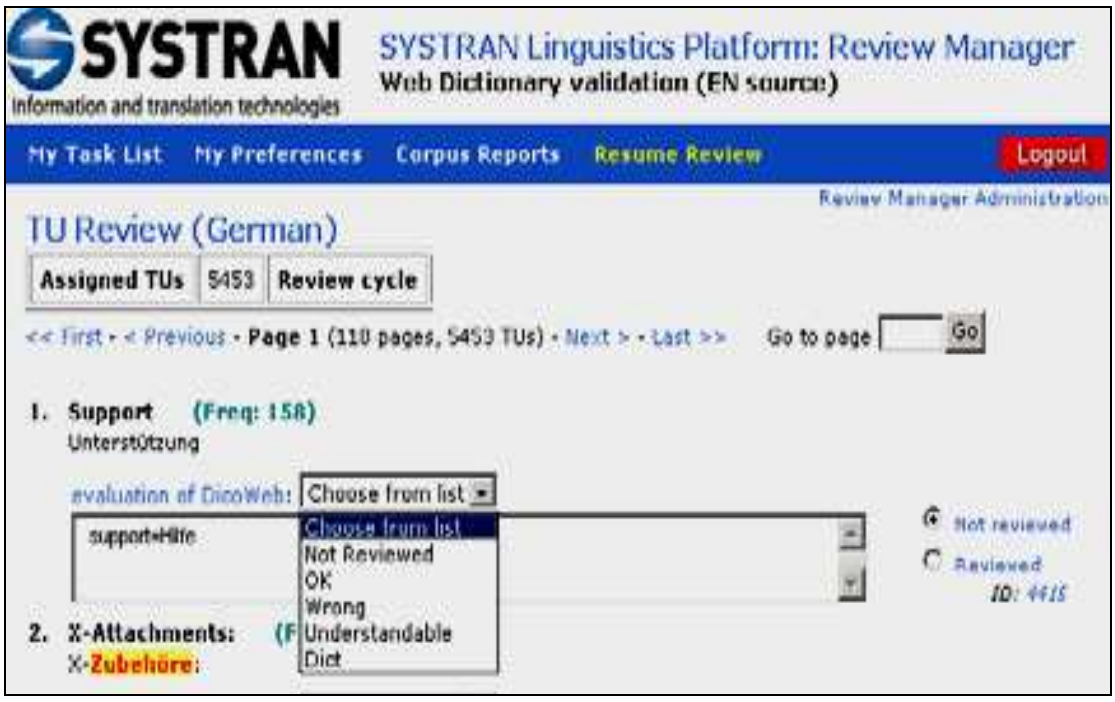


Figure 7: Interface de révision de SYSTRAN Reviewer Manager

Systran évalue son produit selon deux types de mesures, quantitatives et qualitatives.

a.i Mesures quantitatives

La plus simple forme de mesure quantitative que Systran utilise est booléenne : « traduction correcte » ou « traduction incorrecte ». Cette forme permet d'effectuer le jugement rapidement. La vitesse moyenne de révision est d'environ 150 UT par heure [Costa et

Panissod, 2003]. Cependant, le critère de « traduction incorrecte » est souvent raffiné et remplacé par les valeurs suivantes : Erreur non terminologique (*Non-terminology error*) ou Nécessite une entrée multilingue (*Requires a multilingual entry*), etc. (Figure 8).

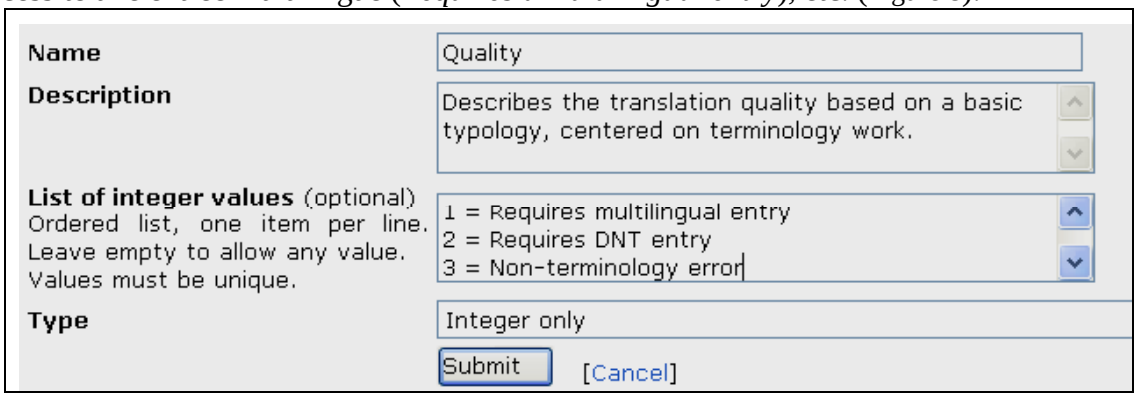


Figure 8: Mesures quantitatives de Systran

Systran utilise aussi une métrique fondée sur la messure de qualité de traduction de la Société Automotive Engineers (SAE) J2450 [SAE J2450, 1999]. Un score numérique pondéré représente la qualité de traduction moyenne d'une UT ; il est fondé sur le nombre d'erreurs de trois catégories : la grammaire, la terminologie et le format. Avec une telle typologie, la vitesse moyenne de révision est d'environ 80 UT par heure [Costa et Panissod, 2003].

a.ii *Mesures qualitatives*

La méthode de mesure qualitative que Systran utilise consiste à demander aux réviseurs de fournir des commentaires qualitatifs sur les traductions via un éditeur de dictionnaire bilingue, qui permet aussi d'ajouter de nouveaux termes. La vitesse moyenne de ce type de mesure est d'environ 50 UT par heure [Costa et Panissod, 2003].

Les nouveaux termes ajoutés sont compilés et utilisés pour retraduire le corpus source. De nouvelles traductions sont comparées rapidement avec les traductions précédentes en utilisant un algorithme de distance d'édition [Damerau, 1964 ; Levenshtein, 1966 ; Wagner et Fischer, 1974]. La différence entre les deux versions est visualisée comme sur la figure 9, ce qui permet aux réviseurs de faciliter la validation.

Figure 9: Comparaison entre deux versions de traduction

Via des communications personnelles, nous savons aussi que Systran utilise des fichiers XML contenant des suites de test et des matrices NIST, BLEU, etc. pour évaluer globalement son système.

b. *PAHO MTS*

PAHOMTS [PAHO, 2009] est un système de TA, développé par l'équipe de Marjorie Léon et maintenu par la PAHO (Pan American Health Organization, Washington) depuis 1980 [Vasconcellos et Léon, 1998]. On a trouvé des informations intéressantes relatives à l'évaluation de ce système par son développeur, la PAHO : PAHOMTS a traduit plus de 88 millions de mots depuis 1980. Les post-éditeurs post-éditent la sortie brute pour produire des traductions de bonne qualité avec un gain de 30-50% de productivité [PAHO, 2009].

Il y a plusieurs outils fournis par la PAHO pour le support à la post-édition des résultats de ce système, tels que des macros intégrées à Microsoft Word ou à PowerPoint, un outil de recherche et d'index, etc. [Aymerich et Camelo, 2006]. Cependant, PAHOMTS ne comporte apparemment pas d'outil de support à l'évaluation. Ce que nous avons trouvé est qu'on évalue le corpus traduit segment par segment, en utilisant un fichier « côte à côte » (side-by-side) contenant une table à trois colonnes : segment source, segment traduit, et commentaire. Les réviseurs peuvent associer des commentaires ou des suggestions à chaque segment, directement sur ce fichier [Aymerich et Camelo, 2009].

I.1.1.3.2 Evaluation par l'extérieur

a. Utilisateurs directs

L'évaluation d'un système de traduction peut être faite par les post-éditeurs en mesurant leur temps de post-édition des résultats de TA. Cette méthode d'évaluation a été appliquée par des utilisateurs directs. Par exemple, [Young, 1999] a montré des résultats automatiques aux utilisateurs, et mesuré leur efficacité à atteindre leur objectif. [Allen, 2001b ; Guerra, 2003] ont organisé le travail de traducteurs et mesuré le temps de post-édition et des éléments non fonctionnels comme le sentiment des utilisateurs: confort, fiabilité, interface, etc. [Sripada et al, 2003] a utilisé des données post-éditées pour évaluer SUMTIME-MOUSAM, un système de TA qui produit des prévisions météorologiques maritimes [Sripada et al., 2003].

On peut lire sur le site de J. Allen[9] que la post-édition de résultats de Systran fait souvent gagner au moins 50% du temps. Dans le projet EOLSS/UNL au GETALP, Ch. Boitet a post-édité 7000 segments anglais – français (environ 6500 segments sont des pré-traductions de Systran, et les autres sont des pré-traductions de Reverso) et a gagné environ 65% du temps.

Apparemment, les utilisateurs directs n'ont pas d'outil leur permettant d'organiser et de mesurer la post-édition, mais ils utilisent souvent des feuilles Excel.

b. Audit

Plusieurs consultants et cabinets d'audit (bureau Van Slype, Omnium…) proposèrent de nombreuses mesures liées à la qualité linguistique (fidélité, grammaticalité, parfois couverture) des résultats bruts et à la structure interne des systèmes, pour mesurer leur potentiel d'amélioration et le coût de cette amélioration (suites de tests, etc.). Ces mesures font essentiellement appel à des jugements subjectifs par des experts humains [Blanchon et Boitet, 2007].

Apparemment, ces consultants et cabinets d'audit n'ont pas d'outil informatique leur permettant de faciliter voire d'automatiser l'évaluation de résultats de TA.

c. Journaux

Il y a des journaux et des Workshop qui annoncent des comparaisons de qualité entre les systèmes en opération.

Par exemple, dans le Workshop EACL 2009[10] (Workshop sur SMT), Systran a été classé devant Google et les autres logiciels pour la qualité de ses traductions de l'anglais en français [Systran, 2009].

Dans le journal PC&I (Personal Computer & Internet), *@ProMT Professional 8.0 Multilingual* a remporté le prix dans la catégorie « Qualité » ainsi que le titre de « la solution la plus complète de traduction automatique ».

[9] Voir le site `http ://www.geocities.com/mtpostediting/` de J. Allen.

[10] http://www.statmt.org/wmt09/

Conclusion (avant juillet 2007)

Du point de vue du support informatique à l'évaluation des systèmes en opération, il n'y a presque aucun support logiciel proposé pour ou par l'extérieur (Jeff Allen n'a aucun logiciel et les autres non plus). Pour les développeurs, il n'y a que des outils en interne. Par exemple, au GETALP, on a proposé TRADOH [Vo-Trung, 2004b], un métasystème d'appel de systèmes de TA en ligne. Cependant, il n'y a pas de système existant disponible pour évaluer k systèmes S_1, S_2,..,S_k (avec paramètres, car il ne suffit pas de les appeler en parallèle).

I.1.2 Synthèse des objectifs et des problèmes

Notre objectif dans ce chapitre est de concevoir et d'implémenter un système offrant un support informatique à l'évaluation de systèmes de TA, et à l'exploitation de corpus de traductions (import, visualisation, export, etc.).

Pour pouvoir atteindre cet objectif, nous devrons traiter plusieurs problèmes émergents présentés ci-dessous.

I.1.2.1 Aspects importants pour relever ce défi

Définitions et de dénominations unifiées pour plusieurs notions. Les objets sur lesquels ce système travaille sont les corpus, les segments, les unités de traduction, etc. Cependant, les définitions actuelles de ces termes sont ambiguës, incomplètes, et souvent imprécises. Nous proposons donc des redéfinitions et des dénominations unifiées pour plusieurs notions (voir §I.1.3.1 et §I.2.1.1).

Modélisation et traitement d'entrées non textuelles. Dans les campagnes d'évaluation, les entrées ne sont pas seulement des textes, mais aussi des structures non textuelles. Les entrées peuvent en effet être des segments source sous forme de listes d'énoncés candidats avec scores, ou sous forme de graphes lexicaux directement produits par des reconnaisseurs de parole. De plus, certaines campagnes diffusent (et reçoivent) peut-être aussi des fichiers son comme dans les campagnes d'évaluation de synthétiseurs de parole. Il faut donc pouvoir traiter non seulement des entrées textuelles, mais aussi des entrées non textuelles. Il faut aussi modéliser les entrées (chaînes, treillis, etc.) pour pouvoir guider le système. Nous proposons des solutions à ce problème au §I.2.1.2.

Visualisation intuitive des données et des résultats des évaluations. Pour l'évaluation subjective, il faut visualiser les données de façon à ce que les juges puissent facilement les comparer et les évaluer. Par exemple, pour évaluer un corpus de dialogues ou de documents dont les segments sont cohérents, un sectra doit permettre non seulement de juger segment par segment, mais aussi de voir globalement les documents source et cible en parallèle, avec synchronisation au niveau des segments.

Pour l'évaluation objective liée à la tâche, un sectra doit visualiser intuitivement les différences entre une traduction automatique et sa postédition, avec une présentation du type *Track changes* de MS Word (voir Figure 10).

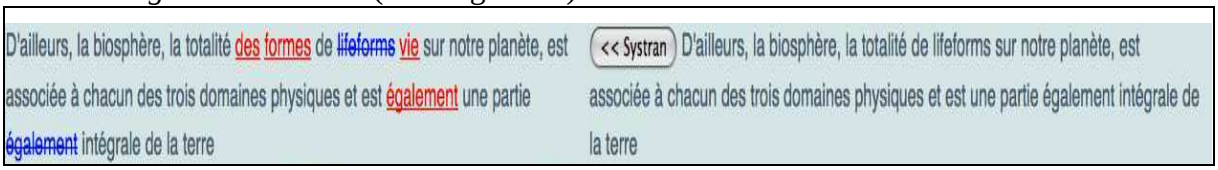


Figure 10: Présentation du type Track changes de MS Word

Pendant et après une campagne d'évaluation, le besoin est de visualiser les résultats de tel ou tel type d'évaluation sur tout ou partie du corpus. Le système doit les visualiser de façon

synthétique. Ce sont non seulement des tableaux de chiffres, mais aussi les données linguistiques elles-mêmes (segments source, traduits, éventuellement post-édités).

Les résultats d'évaluation subjective doivent être visualisés de façon à ce que les juges et les organisateurs puissent les comparer et les suivre.

Nous présentons et résolvons ce problème au §I.2.2.1.

Parcours et visualisation dans une masse de données, le « problème de l'ascenseur ». Un sectra doit aussi permet de travailler avec de grands corpus tels qu'EuroParl, ou avec des mémoires de traductions correspondant à plusieurs corpus. La navigation dans de tels corpus pose des problèmes tels que la lenteur, la conservation de la place sur l'écran, etc. Le problème de la lenteur se pose souvent quand on effectue des opérations telles que la sauvegarde, le chargement, etc. d'une masse de données. Le problème de la conservation de la place sur l'écran est relatif aux problèmes de présentation et d'ergonomie de l'interface, car une masse de données est constituée non seulement par des milliers de segments, mais aussi par de nombreuses occurrences liées à un segment source (une source peut avoir plusieurs traductions, plusieurs post-éditions, un graphe UNL, un fichier son, des scores BLEU, NIST, etc.). Nous présentons et proposons des solutions à ce problème au §I.2.2.2.

Flux de travaux. Il y a certaines tâches dans une campagne d'évaluation qu'il faut effectuer en plusieurs étapes successives. Par exemple, pour la tâche d'évaluation subjective, on devra d'abord attribuer des comptes et des droits aux juges, puis leur affecter des tâches et des données, et enfin passer dans une configuration permettant d'effectuer l'évaluation subjective. Une solution moderne consiste à utiliser un outil de flux de travaux (WorkFlow) pour définir et contrôler le déroulement de ces tâches. Nous présentons et proposons des solutions à ce problème au §I.2.3.2.

Gestion des utilisateurs. Un système de support d'évaluation de TA doit permettre de définir de façon adéquate plusieurs types d'utilisateurs, et leurs droits d'accès, d'introduire facilement de nouveaux types d'utilisateurs, etc. La gestion des utilisateurs doit aussi contribuer à assurer la sécurité des données ainsi que des campagnes d'évaluation. Nous présentons et proposons des solutions à ce problème au I.2.3.2.

I.1.2.2 Importance des aspects conceptuels, informatiques et de génie logiciel.

Les problèmes présentés ci-dessus peuvent être classés selon l'importance relative de leurs aspects conceptuels, algorithmiques, et programmatoires, comme le montre le tableau suivant.

Problèmes	Conceptuels	Informatique	Génie Logiciel
(Re)définitions et de dénominations unifiées pour plusieurs notions	••••	—	—
Modélisation et traitement d'entrées non textuelles	•••	••	••
Visualisation intuitive des données et des résultats des évaluations	••	•••	•
Flux de travaux (WF) : organisation des participants et des tâches	••	••	•••
Navigation et visualisation d'une masse de données, « le problème de l'ascenseur »	•	••	•••
Gestion des utilisateurs	•	••	••••

Table 2:Problèmes émergents classés en trois catégories

I.1.3 Notions unificatrices et principes généraux

I.1.3.1 Terminologie

Nous pouvons trouver les définitions ci-dessous dans l'Annexe 1.

I.1.3.1.1 *Phrase, segment, unité de traduction*

a. *Phrase*

Définition I-1. Une *phrase* est l'unité élémentaire d'un énoncé, formée de plusieurs mots ou groupes de mots et qui présente un sens complet. [TheFreeDictionary, 2010].

b. *Segments*

Définition I-2. Un *segment* est l'unité de traduction de base des traducteurs humains. Il s'agit d'une phrase, d'un titre, ou d'un terme dans une nomenclature.

On appelle aussi « segment » (en TMX ou en XLIFF) un segment source accompagné d'une ou plusieurs traductions. Il nous faut donc raffiner cette définition.

Définition I-3. Un *segment multilingue* est une liste qui se compose de N segments monolingues équivalents en N langues.

Dans le format TMX [11] (Translation Memory eXchange), un segment multilingue est représenté par un élément <tu>, par exemple :

```
<tu>
   <tuv xml:lang="en"><seg>How are you?</seg><tuv>
   <tuv xml:lang="fr"><seg>Comment vas tu?</seg><tuv>
   <tuv xml:lang="vi"><seg>Chú có khỏe không?</seg><tuv>
</tu>
```

Si N = 2, 3,… on parle de segments bilingues, trilingues, etc. On appellera donc *segment monolingue* un segment multilingue réduit à un seul segment (source).

Définition I-4. Un *segment monolingue* est un segment dont le contenu est en une seule langue.

c. *Diagramme traductionnel*

Définition I-5. Un *diagramme traductionnel* est un graphe de traductions dont les nœuds sont les langues successives (par exemple Ja, En, Fr), et donc les arcs portent éventuellement des métadonnées concernant l'opération de traduction (humaine ou automatique, personne ou systèmes, caractéristiques ou paramètres).

Voici ci-dessous un exemple de la traduction d'un texte source du français (Fr) à l'anglais (En), l'espagnol (Es), le vietnamien (Vi), le japonais (Ja), et le coréen (Ko).

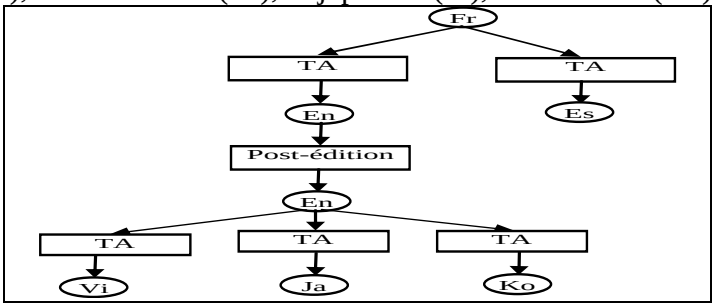


Figure 11 : Exemple d'un diagramme traductionnel

[11] http://www.lisa.org/fileadmin/standards/tmx1.4/tmx.htm#Misc_GroupTU

Le diagramme traductionnel (Figure 11) montre que les traductions en anglais et en espagnol sont obtenues directement à partir du texte source français, tandis que les traductions en vietnamien, japonais, et coréen sont obtenues à partir du texte anglais post-édité.

d. Unité de traduction

Définition I-6. Une *unité de traduction* est une partie d'un texte ou d'un document dont certains aspects sont traités globalement durant une opération de traduction.

Pour un traducteur humain, l'UT correspond à ce qu'on lui demande de "livrer". S'il traduit et fournit un document chapitre par chapitre, ses UT sont des chapitres. S'il fournit un livre complet, son UT est le livre.

Les opérations globales dont il s'agit sont multiples, et leur portée minimale doit être inférieure à la taille d'une UT pour qu'on puisse les traiter.

Par exemple, traiter la concordance des temps en français suppose que l'UT soit au moins un paragraphe. Il en va de même pour la résolution d'anaphores pronominales extraphrastiques.

Pouvoir traiter les sigles et acronymes créés dans un document suppose que l'UT couvre tout le document. En effet, beaucoup de sigles ou acronymes sont définis dès le début, puis utilisés tout au long du document.

Par exemple, on peut lire au début d'une thèse "…des écoles, telles que HEC, CNAM (Conservatoire National des Arts et Métiers)", et on saura ainsi que CNAM est masculin singulier dans la suite. Dans un autre document, on pourrait lire "…la Caisse Nationale d'Assurance Maladie (CNAM)…", et on saurait que CNAM est féminin singulier.

En TA, l'*unité de traduction* est presque toujours un segment, et nettement plus rarement un (petit) paragraphe. On trouve ainsi dans les entrées du BTEC[12] quelques UT formées de 2 ou 3 phrases. Par conséquent, il faut avoir recours à des phases de prétraitement ou de post-traitement global pour pouvoir traiter (au moins) la cohérence terminologique, la concordance des temps, et les sigles.

En ce qui concerne la cohérence terminologique, le maximum de ce qui est fait actuellement consiste à déterminer un "thème" ou un "domaine" pour chaque segment, et à le traduire en adaptant en conséquence la liste de priorité des dictionnaires de TA utilisés pour traduire ce segment (ou des tables de traduction dans le cas d'un système probabiliste).

En fait, il faudrait distinguer les *unités de traduction* et les *unités de traitement*, souvent plus grandes. Dans le cas d'un système de TA probabiliste, l'unité de traduction est le segment, mais l'unité de prétraitement peut être tout le texte à traduire, puisqu'on dispose d'outils permettant de traiter un très gros corpus comme un tout.

I.1.3.1.2 Textes et documents

a. Textes

Définition I-7. Un *texte* peut être une unité de longueur variable : un page, un livre, etc. Un texte est sous forme brute, ou sous forme formatée (html, xml, etc.). En TA, un texte est un fichier d'entrée ou de sortie.

[12] Le BTEC (Basic Travel Expression Corpus), construit par ATR et ses partenaires du projet CSTAR, contient au moins 163000 entrées, en plusieurs langues (japonais, anglais, chinois, coréen, allemand, français, italien, et un peu espagnol). Une entrée est normalement un énoncé tiré d'un livre de phrases pour touristes, mais parfois elle contient plusieurs énoncés, séparés par des "|", par exemple: "Good morning. | I would like some coffee. | Give me also two pancakes please."

b. Documents

Définition I-8. Un *document* est constitué par un texte et des objets satellites (images, formulaires, etc.). Un *document* peut donc comprendre plusieurs fichiers.

Par exemple, une page Web contient au moins un fichier .html représentant le texte et un répertoire contenant des fichiers représentant les objets satellites (icônes, images, audio, vidéo, etc.).

I.1.3.1.3 Types de traductions pour l'évaluation

a. Traductions candidates

Définition I-9. Une *traduction candidate* est le résultat d'un système de TA à évaluer.

b. Traductions de référence

Définition I-10. Une *traduction de référence* est une traduction de bonne qualité utilisable pour l'évaluation de *traductions candidates*.

I.1.3.1.4 Types d'évaluation

a. Evaluation interne

Définition I-11. Une évaluation est dite *interne* si elle vise à juger la conception de systèmes (architecture linguistique et architecture computationnelle) de TA et leurs perspectives d'amélioration et d'extension à de nouvelles langues, à de nouveaux types de documents, et à de nouvelles tâches (e.g. de l'assimilation à la dissémination).

b. Evaluation externe

Définition I-12. Une évaluation est dite *externe* si elle consiste à juger un système de traduction comme une *boîte noire.*

On juge les traductions sur des critères linguistiques (grammaticalité, fidélité, etc.) ou sur des critères d'usage (productivité, coût), et l'on juge aussi les aspects de coût, portabilité, fiabilité, convivialité du système en opération.[13]

c. Evaluation subjective

Définition I-13. Une évaluation est dite *subjective* si elle fait appel à des jugements qualitatifs humains.

Il peut s'agir soit de noter chaque traduction candidate en fonction d'un certain critère accompagné d'une *échelle de notation* (par exemple, fluidité avec note de 1 à 5), soit de comparer deux traductions candidates, là aussi avec une échelle précise (par exemple, { -1, 0, 1 }).

Pour définir une telle évaluation, il faut aussi savoir si l'on présente aux juges d'autres données que la ou les traductions candidates, comme le segment source, et/ou une traduction de référence.

Les critères utilisés dans les campagnes récentes sont l'adéquation et la fluidité, mais il y en a beaucoup d'autres possibles, et utilisés dans de nombreuses évaluations antérieures, comme la grammaticalité, la fidélité, l'utilité, la qualité d'usage, la qualité traductionnelle (jugée par des

[13] D'après (Hovy et al., 2002), les caractéristiques externes d'un système de TA à évaluer sont (1) ses fonctionnalités (pertinence, précision, etc.), (2) sa fiabilité (recouvrabilité, conformité, etc.), son utilisabilité (intelligibilité, opérabilité, etc.), (3) son efficacité (temps, ressources utilisées, etc.), (4) sa maintenabilité (analysabilité, testabilité, etc.), (5) sa portabilité, et (6) son coût.

professionnels de la traduction), etc. [White et al., 1994 ; ALPAC, 1966 ; JEIDA, 1992 ; IWSLT, 2004, 2006].

Pour chaque type d'évaluation, il faut définir une interface adaptée pour les juges. Il faut aussi une interface pour les personnes gérant les évaluations subjectives (définition des profils des juges, affectation des tâches, suivi, consolidation des résultats).

d. Evaluation objective

Définition I-14. Une évaluation est dite *objective* si elle ne fait pas appel à des jugements subjectifs. Elle peut être fondée sur des références ou liée à la tâche.

Définition I-15. Une évaluation *objective* est fondée sur des références si elle est réalisée par des programmes informatiques produisant des résultats numériques à partir d'un certain nombre de traductions de référence et des traductions candidates fournies en entrée.

Certaines méthodes, comme WER (Word Error Rate, ou distance d'édition au niveau des mots) [Levenshtein, 1966 ; Wagner et Fischer, 1974], permettent de noter chaque traduction candidate. Nous dirons qu'il s'agit de *notation individuelle.* D'autres, comme les célèbres BLEU et NIST [Papineni *et al.*, 2002], utilisant des calculs sur les n-grammes de mots, ne produisent qu'un score global pour un jeu d'essai donné (souvent de l'ordre de 500 segments). Nous dirons qu'il s'agit de *notation globale.*

Un environnement supportant ce type de mesure contient autant de programmes ("scripts", ou "plugins") que de méthodes d'évaluation. Divers organismes et laboratoires publient des scripts pour ces mesures. Un problème est que ces scripts sont le plus souvent écrits en Perl et ne donnent pas toujours les mêmes résultats quand on les utilise sur des plates-formes différentes avec les mêmes données [MT08, 2008].

Définition I-16. Une évaluation *objective* est liée à la tâche si elle est réalisée par des programmes informatiques produisant des résultats numériques ou symboliques à partir des segments source, de leurs prétraductions automatiques, et de mesures liées à la tâche (temps de compréhension, temps de post-édition, temps de réalisation d'une tâche (comme une réservation) en contexte bilingue).

I.1.3.1.5 Segments non textuels

Définition I-17. Un *segment non textuel* peut être un treillis ou un graphe de chaînes de mots ou de lemmes directement produit par un reconnaisseur de parole. Un segment non textuel peut aussi être une représentation de l'expression langagière multimodale (texte + parole + gestes).

I.1.3.1.6 Corpus

a. Notion de corpus

Dans cette thèse, nous ne nous intéressons pas à tous les types de corpus, mais nous nous concentrons sur les corpus de traductions, en particulier sur les corpus de TA, et sur leurs caractéristiques et structures spécifiques, pour pouvoir les exploiter.

Définition I-18. Les *corpus pour la TA* sont ceux qui sont utilisables pour la traduction, incluant la traduction humaine et la traduction automatique.

Définition I-19. Les *corpus pour la traduction humaine* (*corpus pour la TH*) sont les mémoires de traduction, les corpus de documents, etc.

Définition I-20. Les *corpus pour la traduction automatique* (*corpus pour la TA*) ne sont pas seulement des textes, mais aussi des annotations linguistiques.

Les corpus en TA n'ont pas seulement les caractéristiques des corpus en général, mais aussi des caractéristiques spécifiques de leur aspect traductionnel : (1) niveaux de parallélisme ou bien niveaux d'alignement entre les textes (niveau de document, de paragraphe, de phrases, de mots), (2) nombre des langues et systèmes d'écriture, (3) annotations linguistiques, qui sont soit internes au texte (balisage au fil du texte), soit externes (structures + correspondances) [Boitet, 2007].

Chaque type de corpus pour la TA est utilisable par un type de système de TA. Par exemple, les corpus parallèles sont utilisés dans les systèmes de TA empirique directe, le plus souvent de TA probabiliste. Les corpus enrichis par des annotations plus ou moins complexes sont utilisés dans les systèmes de TA indirecte « par les exemples » (système "EBMT" (Example-Based MT) de l'UTMK (USM, Penang)). Les corpus multilingues contenant des représentations interlingues UNL (Universal Networking Language) sont utilisés dans les systèmes de TA « experte », et les mémoires de traductions (phrases alignées) sont utilisées dans des outils d'aide aux traducteurs (THAM - Traduction Humaine Aidée par la Machine).

b. Document et corpus monolingue

Définition I-21. Un document est dit *monolingue* si tous ses segments sont dans une seule langue.

Définition I-22. Un corpus est *monolingue* si tous ses documents sont monolingues pour la même langue.

c. Document et corpus multilingues

Définition I-23. Un document est dit *multilingue* s'il comprend des segments en plusieurs langues.

Définition I-24. Un corpus est *multilingue* s'il contient des documents multilingues, ou des documents monolingues dans au moins deux langues.

Un corpus *multilingue* peut contenir des documents parallèles et des documents comparables.

d. Document et corpus parallèles

Définition I-25. Un document est dit *parallèle* s'il est multilingue et s'il est aligné au niveau des segments, de sorte qu'un segment est traduction de celui avec lequel il est aligné.

Définition I-26. Un corpus est *parallèle* si tous ses documents sont parallèles.

Les corpus parallèles sont utilisés dans de nombreux contextes. Ils sont parfois utilisés comme ressources pour l'enseignement d'une langue seconde, mais sont plus souvent utilisés pour le traitement automatique des langues.

e. Corpus comparables

Définition I-27. Un corpus est dit *comparable* s'il est composé de textes comparables dans des langues différentes, non alignés au niveau des segments, mais parlant d'un même sujet, à la même époque et dans un registre comparable.

Une sélection d'articles de journaux dans différentes langues, traitant d'une même actualité internationale et à la même époque, constitue un bon exemple de corpus comparable.
On peut trouver de nombreux corpus comparables, mais ils sont inutilisables tels quels pour la TA empirique, même si l'on peut en extraire des correspondances lexicales.

f. Corpus de phrases

Définition I-28. Un corpus est un *corpus de phrases* s'il ne contient que des *phrases*.

Le corpus BTEC [Kikui et al., 2003 ; Takezawa et al., 2002] en est un exemple. Le corpus BTEC contient des entrées qui correspondent aux énoncés que l'on rencontre dans des livres de phrases.

g. Corpus de textes

Définition I-29. Un corpus est un *corpus de textes* s'il ne contient que des *textes*.

Un corpus de textes n'a pas d'objets satellites (images, formulaires, etc.) associés. Il peut donc être représenté par un ou plusieurs fichiers de *texte*.

h. Corpus de documents

Définition I-30. Un corpus est dit *corpus de documents* s'il contient des *documents*.

Un corpus de documents doit donc en général être représenté non seulement par des fichiers de *texte*, mais aussi par des fichiers représentant des objets satellites (fichier contenants des hors-texte, des images, etc.).

I.1.3.2 Principes généraux

Un système de support informatique aux campagnes d'évaluation de TA doit être construit selon les principes généraux suivants : être unifié, être utilisable par tous les utilisateurs, être un service Web, et être programmable.

Support unifié. Un sectra doit fournir un support unifié à tous les types d'évaluation en développement, en campagne d'évaluation, et en utilisation.

En développement, le même système doit gérer les suites de test dans les EDL de TA, les corpus de développement, et les corpus d'apprentissage.

Pendant une campagne d'évaluation, un sectra doit permettre de : (1) réaliser l'*évaluation subjective* par des juges humains incluant l'évaluation d'adéquation, l'évaluation de fluidité, et l'évaluation par comparaison de deux traductions t1 et t2, (2) réaliser l'*évaluation objective* par n-grammes, par mesure de post-édition, et par mesure de compréhension.

Après une campagne d'évaluation, ce système doit permettre de : (1) visualiser toutes les données de façon personnalisable, (2) faire et refaire des expériences, et (3) ajouter des données (plus ou moins partagées).

Tous types d'utilisateurs. Un sectra doit aussi fournir un support unifié à tous les types d'utilisateurs (développeurs, experts, organisateurs de campagnes, juges, etc). Il doit offrir des possibilités adéquates de communication entre organisateurs et juges, entre post-éditeurs, etc.

Utiliser une BD unique accessible par le Web. Par conséquent, il faut une BD unique, gérant différent types de corpus et accessible via le Web (intranet pour le développement).

Offrir un niveau suffisant d'automatisation, de paramétrisation, voire de programmabilité. Un sectra doit aussi offrir un niveau suffisant d'automatisation, de paramétrisation, voire de programmabilité pour effectuer certaines tâches.

I.2 Traitement des problèmes liés à ce défi

I.2.1 Problèmes à dominante conceptuelle

I.2.1.1 Problème 1.1 : Définitions précises de nouvelles notions utiles

I.2.1.1.1 Motivations

Plusieurs termes largement utilisés sont ambigus, incomplets, et souvent imprécis. L'ambiguïté provient parfois du fait qu'un terme a un sens en TA et un autre en TH ou en THAM, et que nous voulons créer un cadre commun pour les corpus relatifs aux deux.

I.2.1.1.2 Etat de l'art

a. Exemples de termes ambigus

Unité de traduction. En TH, il s'agit le plus souvent d'un paragraphe, comme on peut le voir dans les fichiers au format TMX[14] utilisés dans les systèmes de THAM comme Trados ou Similis. Le nom de la balise <tu> (translation unit) refète cette intention.

En TA, par contre, il s'agit le plus souvent d'une phrase ou d'un titre, ce que les éditeurs de systèmes de THAM appellent plus volontiers des *segments.* En TA, on utilise très souvent *phrase* pour les deux notions (phrase ou titre).

Il existe aussi des systèmes de TA dont les unités de traduction sont plus grandes que des phrases ou des paragraphes. C'est en particulier le cas des systèmes écrits en Ariane-G5 du GETA (puis du GETALP), dont les unités de traduction (depuis 1978) peuvent couvrir plusieurs centaines de mots, jusqu'à 3 ou 4 pages standard (750 à 1000 mots).

Segment. On entend le plus souvent par *segment,* en traduction humaine, une phrase ou un titre, en langue source initialement, puis en langue cible après traduction. On parle donc de *segment source* et de *segment cible,* et chacun est monolingue.

Quand un document a été traduit, on peut le transformer en un document multilingue, balisé selon la convention TMX, si la traduction a été faite segment par segment. Au lieu d'un segment source, on trouve alors un *segment multilingue,* constitué de N_{lc} segments, un par langue cible, s'il y a N_{lc} langues cibles.

b. Exemples de termes absents

Segment multisource. Notons que, le plus souvent, l'information sur la langue source ne figure qu'au niveau du document, et pas de chaque segment. D'autre part, il arrive que deux segments (en langues L1 et L2) d'un segment multilingue puissent être considérés comme source, au sens où ils feront foi dans l'utilisation du document. Par exemple, certaines phrases importantes peuvent avoir été mises au point en parallèle dans plusieurs langues. Il faut donc prévoir le cas où un segment multilingue a "plusieurs langues source", et inventer un terme pour cela, par exemple *segment multisource*.

Segment multilingualisé. Par contre, en évaluation, un segment est toujours constitué par un seul segment source, accompagné de N_{ta} traductions automatiques, et de N_{rf} traductions de référence. On parlera alors de *segment multilingualisé*. Un tel segment est un objet éventuellement plus complexe qu'un segment multilingue, mais il n'aura, par définition, qu'une seule langue source.

Infrasegment. En traduction automatique, il arrive qu'une unité de traduction soit plus petite qu'un segment (au sens des traducteurs humains). Par exemple, un segment délimité et traité

[14] Voir le format TMX http://www.lisa.org/Translation-Memory-e.34.0.html

par Systran est le plus souvent une phrase, mais peut aussi être une partie de phrase, par exemple un élément d'une liste à puces contenue dans une phrase (sans parler des erreurs de segmentation dues à une interprétation erronée d'un point d'abréviation comme point de fin de phrase). On parlera dans ce cas d'*infrasegment*.

Supersegment. Il peut aussi arriver qu'un système de TA n'arrive pas à segmenter un paragraphe correctement, et qu'il traite deux phrases comme un seul segment. On parlera alors de *supersegment*.

Il ne faut pas confondre supersegment et unité de traduction. Une unité de traduction peut être volontairement constituée de plusieurs segments, comme c'est le cas pour les systèmes Ariane-G5 du GETA, et Sygmart de J. Chauché (LIRMM), qui permettent donc de traiter des phénomènes à portée plus large que le segment, par exemple la résolution d'anaphores extraphrastiques ou la concordance des temps.

Il nous faut donc préciser la notion de segment que nous retiendrons, et proposer des définitions précises pour les notions associées.

Au-delà des termes communs tels que *segment*, *unité de traduction*, etc., on manque de termes pour l'*unité de traduction multisegment*, la *structuration hiérarchique externe*, la *segmentation multiple*, la *segmentation récursive*, etc.

c. Exemples de termes imprécis

Voici quelques exemples de termes imprécis.

Fragment. Ce terme est souvent utilisé de façon neutre (une suite de mots quelconque contenue dans un texte), ou bien dans le sens d'une partie d'un segment (non nécessairement linguistiquement motivée). Nous l'utiliserons toujours dans le second sens.

Distance. On parle souvent de "distance" entre une traduction et une postédition, sans dire de quelle unité on parle (caractère, mot), sans donner de définition précise de "mot", et sans indiquer les opérations prises en compte (toujours insertion et suppression, souvent échange, plus rarement transposition, ou permutation binaire ou ternaire), et le coût de chacune, sachant qu'assez souvent il ne s'agit pas d'une distance, mais d'une pseudo-distance ou d'une similarité.

I.2.1.1.3 Termes retenus

Compétons les définitions données aux §I.1.3.1, §II.1.3.1, §III.1.3.1.

Définition I-31. Un *segment multilingualisé* sera pour nous un segment contenant une seule langue source.

Voici l'exemple de segment multilingualisé représenté en XML :

```
< segment type="multilingualise" id="seg01">
 <tuv type ="source" lang="en"> A burglar broke into my room. </tuv>
 <tuv type =”cible”  lang="fr"> Un cambrioleur est entré dans ma chambre.</tuv>
 <tuv type =”cible”  lang="vi"> Một thằng trộm lẳng vào phòng của tôi. </tuv>
</segment>
```

Définition I-32. Un *segment multisource* contient plusieurs langues source.

Voici l'exemple de segment multisource représenté en XML :

```
<segment type="multisource" id="seg00">
 <tuv type ="source" lang="en"> A burglar broke into my room. </tuv>
 <tuv type =”source” lang="fr"> Un cambrioleur est entré de force dans ma chambre.</tuv>
 <tuv type =”cible”  lang="vi"> Một thằng trộm lẳng vào phòng của tôi. </tuv>
</ segment-ms>
```

I.2.1.2 Problème I.2 : Modélisation et traitement d'entrées non textuelles

I.2.1.2.1 Préciser le problème

a. Listes et graphes lexicaux (sans le son)

Dans certaines campagnes d'évaluation de systèmes de TA de parole, les segments source sont sous forme de liste [d'énoncés candidats] avec scores, ou sous forme de graphes lexicaux[15], directement produits par des reconnaisseurs de parole.

A titre d'exemple, dans les campagnes d'évaluation IWSLT, on évalue des systèmes de TA de parole dont les entrées sont des graphes lexicaux directement produits par des reconnaisseurs de parole [Patry et al., 2007 ; Paul, 2009].

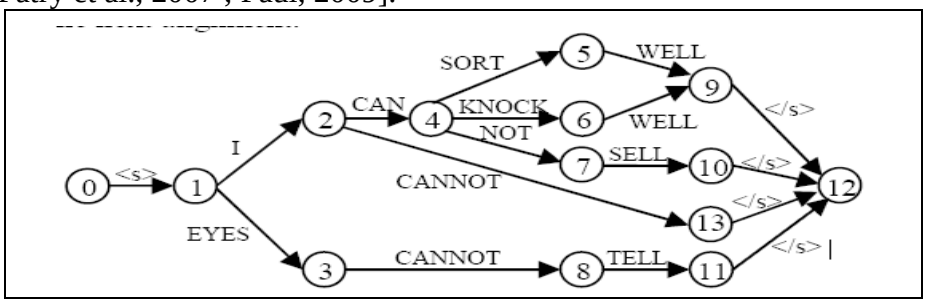


Figure 12 : Exemple d'un graphe lexical (ici un graphe de chaînes)

b. Fichiers son

Certaines campagnes diffusent (et reçoivent) aussi des fichiers son. Par exemple, les campagnes d'évaluation du projet EVALDA[16], ou les laboratoires tels que MICA[17], utilisent des entrées audio pour évaluer des synthétiseurs de parole.

De plus, un sectra devra aussi gérer des corpus ERIM, corpus de dialogues bilingues oraux interprétés. Un segment est constitué d'un tour de parole, et éventuellement d'un texte descriptif, et d'un texte transcrit.

I.2.1.2.2 Etat de l'art

Pour traiter et traduire un graphe lexical dans des systèmes de TA, on en extrait souvent une liste des N meilleurs phrases, puis on les traduit, et enfin on sélectionne la meilleur traduction parmi les traductions produites [Zhang et al., 2004 ; Quan et al., 2005]. On peut aussi utiliser une approche consistant à traduire directement le graphe lexical avec un décodeur spécialisé couplé avec le système de reconnaissance de la parole [Saleem et al., 2004 ; Matusov et al., 2005 ; Zhang et al., 2005 ; Mathias et Byrne, 2006 ; Bertoldi et al., 2007].

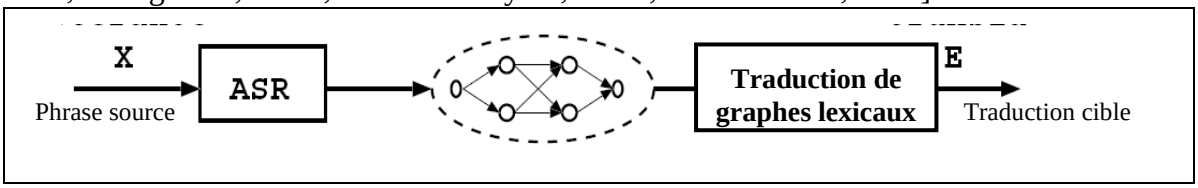


Figure 13 : Traduction de la parole

[15] Les graphes lexicaux utilisés sont soit des treillis (les lexèmes sont portés par les nœuds), soit des graphes de chaînes (les lexèmes sont portés par les arcs, comme dans les Figure 12 et Figure 14).

[16] http://www.technolangue.net/imprimer.php3?id_article=202

[17] http://www.mica.edu.vn/

On peut trouver maintenant plusieurs algorithmes et systèmes permettant de décoder un graphe lexical (voir [Saleem et al., 2004 ; Matusov et al., 2005 ; Zhang et al., 2005 ; Patry et al., 2007]).

Cependant, nous nous concentrons ici sur la représentation d'un graphe lexical comme entrée. Pour chaque graphe lexical, on énumère les nœuds dans un ordre topologique, avec pour chaque nœud la liste de ses arcs adjacents (sortants). Par exemple, Moses représente un graphe de chaînes lexical sous la forme ci-dessous (Figure 15).

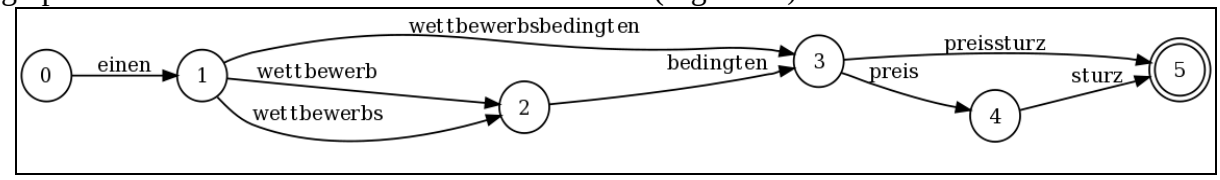


Figure 14: Graphe de chaînes lexical

```
(
  (
   ('einen', 1.0, 1),
  ),
  (
   ('wettbewerbsbedingten', 0.5, 3),
   ('wettbewerbs', 0.25, 2),
   ('wettbewerb', 0.25, 2),
  ),
  (
   ('bedingten', 1.0, 3),
  ),
  (
   ('preissturz', 0.5, 5),
   ('preis', 0.5, 4),
  ),
  (
   ('sturz', 1.0, 5),
  ),
 )
```

Figure 15:Représentation du graphe de chaînes lexical ci-dessus [Moses, 2009]

Le deuxième paramèttre est la probabilité associée à une transition, et le troisième est le numéro de nœud suivant.

En général, on écrit ce graphe sans espace, sur une seule ligne, comme suit:

```
((('einen',1.0,1),),(('wettbewerbsbedingten',0.5,3),('wettbewerbs',0.25,2),('wettbewerb',0.25,2),),(('bedingten',\
1.0,3),), (('preissturz',0.5,5), ('preis',0.5,4),),(('sturz',1.0,5),),)
```

En ce qui concerne le traitement des entrées audio, vidéo, image, etc., tous les systèmes multimédia ne sauvegardent que les chemins des fichiers contenant ces entrées dans leur base de données, mais les entrées elles-mêmes sont dans des dossiers du système d'exploitation.

I.2.1.2.3 Représentation d'entrées non textuelles dans un sectra

On pourrait systématiquement représenter une entrée comme un graphe pondéré, mais cela compliquerait inutilement le cas simple et le plus fréquent, où l'entrée est une simple chaîne de caractères.

Il semble donc préférable de définir plusieurs types pour la représentation de la partie "source" d'un segment.

Définition I-33. Un segment "source" multimédia est un graphe avec un type associé au corpus, par exemple simple chaîne, ou treillis de segmentation, ou treillis pondéré, ou automate fini, pondéré ou non.

Par défaut, ce sera une chaîne simple. On définira une écriture XML standard pour un segment qui pourra donc être multiple et peut-être multimodal.

Par souci de compatibilité avec les systèmes SMT modernes, on utilisera le format XML pour représenter l'entrée, et l'on définira un attribut *type* pour indiquer le type de l'entrée. Cet attribut recevra une valeur prise dans une collection de valeurs prédéfinies (« texte », « multimedia », « treillis », « graphe de chaînes », etc.). En fonction de la valeur de cet attribut, le système traitera l'entrée de façon adéquate.

Voici trois exemples :

```
<entry id="id00" type= "text"> This is text input </entry>

<entry id="id01" type= "multimedia"> file:///C:/Users/phap/Desktop/liglab44_en.html
</entry>

<entry id="id02" type= "lattice">
((('einen',1.0,1),),(('wettbewerbsbedingten',0.5,3),('wettbewerbs',0.25,2),('wettbewerb',0.25,2),),(('bedingten'
,\1.0,3),), (('preissturz',0.5,5), ('preis',0.5,4),),(('sturz',1.0,5),),) </entry>
```

Dans les exemples ci-dessus, si la valeur de l'attribut *type* est « text », le système importera le contenu de cette entrée dans sa base de données. Si cette valeur est « multimedia », le système copiera l'entrée sur ce chemin dans un répertoire du serveur, et sauvegardera le nouveau chemin dans la BD. Si cette valeur est « lattice », le système extraiera du treillis lexical d'entrée les phrases possibles et les sauvegardera dans sa BD.

I.2.2 Problèmes à dominante algorithmique

I.2.2.1 Problème 1.3 : Visualisation intuitive des données et des résultats des évaluations

I.2.2.1.1 Analyse du problème

Les interfaces destinées aux juges dans les campagnes d'évaluation sont simplement des grilles avec des boutons (voir Figure 6), et celles des éditeurs de traductions des systèmes de TA sont toujours des tableaux à la Excel (par exemple, tous les systèmes japonais tels que *ATLAS-II*, *Honyaku no O-Sama*, *Yakushite.net*, ou le système *BEYTrans*).

Le problème est qu'on montre non pas l'énoncé source usuel, mais une version normalisée (suppression des majuscules, séparation des ponctuations, etc.). Que faut-il montrer?

En ce qui concerne les résultats des évaluations, le problème est qu'on ne les voit jamais, puisque, comme déjà dit, on ne voit que des tableaux de chiffres: comment les montrer de façon synthétique, tout en montrant les données linguistiques elles-mêmes (segments source, traduits, éventuellement post-édités) ?

Comment rendre sensible le travail de post-édition pour le cas d'une évaluation objective par post-édition?

Un autre problème est que les organisateurs veulent souvent voir tous les résultats d'évaluation d'une campagne de façon globale (résultats d'évaluation subjective de tous les juges pour tout le corpus sur une interface). Dans le cas où le corpus de la campagne est assez grand (plus de 10000 segments), on rencontre le problème de chargement de grosses données sur une interface Web.

I.2.2.1.2 Etat de l'art

En ce qui concerne la visualisation sensible du travail de post-édition, plusieurs systèmes proposent une visualisation similaire à celle de la fonction *Track Changes* de Word (Figure 16).

Un cambrioleur est entré de force dans ma pièce.

Un cambrioleur ~~est~~ a ~~entré~~ forcé ~~de~~ ~~force~~ ~~dans~~ ma ~~pièce.~~ chambre.

Figure 16: Track changes dans Word

A titre d'exemple, des systèmes fournissant cette visualisation sont le système *SYSTRAN Review Manager* (étudié au I.1.1.3.1a), le système *Caitra* de traduction humaine (voir chapitre 2), et le système SECTra_w [Huynh et al., 2008a].

Pour produire cette présentation, ces systèmes ne suivent pas les actions d'édition de l'utilisateur. Ils utilisent la "trace" d'un algorithme de calcul de la distance entre chaînes, comme celui de [Wagner et Fischer, 1974], pour produire des suites les plus longues possibles d'insertions et de suppressions, en éliminant les substitutions. La distance d'édition est le coût minimal des opérations d'édition nécessaires pour transformer la première chaîne en la seconde. Les coûts attribués aux opérations sont typiquement les suivants (distance de Levenshtein) : 0 pour la conservation et 1 pour la substitution, l'insertion, et la suppression [Levenshtein, 1966]. L'algorithme a une complexité en O (n x m), où n et m sont les longueurs des deux chaînes.

```
Entier LevenshteinDistance(char s[1..m], char t[1..n])
    // d est un tableau de m+1 lignes et n+1 colonnes
    Entier D[0..m, 0..n] ;

    Pour i allant de 0 à m faire
      D[i, 0] := i ;
    Pour j allant de 0 à n faire
      D[0, j] := j ;

    Pour i allant de 1 à m faire
      Pour j allant de 1 à n faire
          si s[i-1] = t[j-1] alors C := 0
          sinon C := 1
          D[i, j] := minimum(
          D[i-1, j] + 1, // suppression
          D[i, j-1] + 1, // insertion
          D[i-1, j-1] + C // substitution
          )

    retourner D[m, n]
```

Algorithme 1: algorithme de calcul de la distance d'édition de Wagner et Fisher

En ce qui concerne le problème de la visualisation de milliers de segments, la plupart des systèmes utilisés dans les campagnes d'évaluation tels que celui de NIST traitent ce problème en découpant les corpus à évaluer en pages d'un nombre fixe de segments. Mais les utilisateurs voudraient pouvoir contrôler ce nombre.

I.2.2.1.3 Solutions proposées

Nous proposons d'abord de définir des pages logiques, de caractéristiques paramétrables (nombre de segments, affichage, cache, largeur et placement des colonnes, etc.) pour la visualisation de données assez grandes. Le problème de la navigation et de la visualisation en présence d'une très grande masse de données sera traité en détail au §I.2.2.2.

En ce qui concerne la visualisation des résultats d'évaluation, nous proposons de visualiser non seulement les tableaux de chiffres fournis par les mesures "à n-grammes", mais aussi les données linguistiques elles-mêmes (segments source, traduits, éventuellement post-édités).

Pour l'évaluation objective par post-édition, il vaudrait mieux visualiser les distances d'édition à trois niveaux (caractère, mot, phrase) entre une traduction automatique et une post-édition finale, avec la possibilité d'une visualisation rendant sensible le travail de post-édition, comme la fonction *Track changes* de MS Word.

De plus, un juge veut parfois voir les jugements d'autres juges sur les mêmes données (un segment peut être jugé par plusieurs juges). Cependant, les systèmes utilisés dans des campagnes d'évaluation telles que IWSLT ne le permettent pas. Un sectra devra permettre d'autoriser ou d'interdire cela, en fonction de l'étape de l'évaluation.

I.2.2.2 Problème 1.4 : Parcours et visualisation d'une masse de données, « le problème de l'ascenseur »

I.2.2.2.1 Analyse du problème

Bien que la solution consistant à définir dynamiquement des pages logiques puisse régler le problème de la visualisation de milliers de segments (cf. §I.2.2.1), cette solution n'est pas parfaite parce qu'il y a des cas où des segments découpés à partir d'un même paragraphe pourraient être distribués dans des pages différentes (page N, page N+1, etc.). Cela serait très pénible pour les utilisateurs, car ils ne verraient pas tous les segments d'un même paragraphe à la fois.

Permettre aux utilisateurs de paramétrer le nombre de segments d'une page logique n'est pas non plus une solution suffisante, car on rencontrera le même problème aux bornes de grandes pages logiques. Or, le besoin est clairement de pouvoir utiliser un « ascenseur » approprié pour faire défiler un million de segments, par exemple. Ainsi, dans le projet OMNIA, nous avons à traiter une collection de 500.000 petits textes de 60 mots, soit entre 1,5 et 2M segments.

Les problèmes ici sont la lenteur (chargement, navigation) et la conservation de la place sur l'écran. Nous appellerons ce problème le « problème de l'ascenseur ».

I.2.2.2.2 Etat de l'art

Ce problème a été plus ou moins traité dans des systèmes Web permettant la visualisation et le parcours d'une masse de données. A titre d'exemple, [Bey, 2009] a proposé dans sa thèse de diviser une masse de données en documents virtuels et d'utiliser l'ajustement du cache pour améliorer la performance sur le serveur. Avec cette proposition, il a annoncé que *BEYTrans* pouvait visualiser jusqu'à 50 K segments.

Un autre exemple est *Google Docs* [Google, 2009] qui permet de visualiser et de manipuler un document assez grand, probablement en utilisant le cache du navigateur du client. La taille de document que ce système accepte est 500Ko (~83 000 mots, soit 300 pages standard). Cependant, cette taille est petite par rapport à quelques corpus sur lesquels on veut travailler, comme le corpus Europarl (20M mots, soit environ 1M segments).

Un autre exemple est *Yahoo! Mail advance* qui permet de visualiser et de parcourir un nombre illimité de titres de courriers électroniques en faisant glisser le curseur d'un ascenseur (Figure 17).

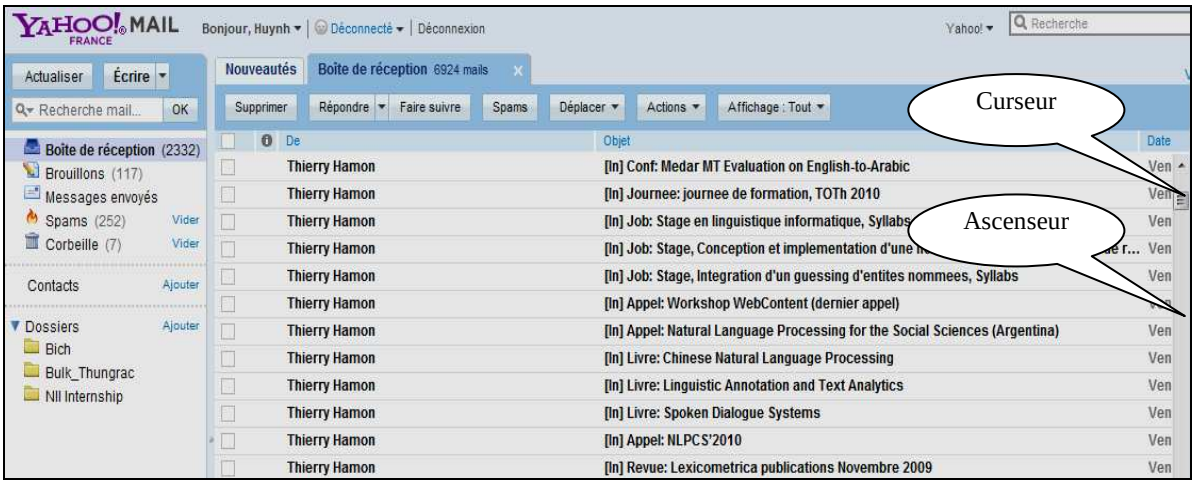


Figure 17 : Yahoo! Mail advance. Navigation dans tous les courriels en utilisant un curseur

Pour ce faire, *Yahoo!* groupe les titres de courriers électroniques en ensembles (chaque ensemble comprend par exemple 20 titres). Ensuite, il affecte chaque position sur l'ascenseur à un ensemble de titres de courriels. Enfin, il localise la position courante du curseur sur l'ascenseur pour déterminer quel ensemble de titres de courriels sera affiché sur l'interface. Dans le cas où un titre consiste en une chaîne très longue, il n'affiche qu'une portion de ce titre dans l'interface. Cette solution permet à *Yahoo!* de déterminer à l'avance un nombre de titres quelconque pour les grouper dans un ensemble.

Cependant, cette solution ne marche pas si l'on veut visualiser et parcourir les segments, parce qu'on veut toujours voir le contenu entier d'un segment pour pouvoir le manipuler. Bien entendu, *Yahoo!* utilise la technologie Ajax pour implémenter cet algorithme.

I.2.2.2.3 Solutions possibles

a. Solution 1

L'analyse des exemples présentés ci-dessus montre qu'on peut appliquer une méthode comme celle de *Yahoo!* en utilisant le curseur de l'ascenseur avec la technologie Ajax. Cependant, les longueurs des segments sont variables et différentes. On ne peut donc pas déterminer à l'avance un nombre de segments par ensemble. Par contre, on peut déterminer à l'avance un nombre de mots (Nm) affichés à la fois sur l'interface.

Nous proposons ici un premier algorithme de visualisation et de parcours d'une masse de données de corpus sur l'interface.

(1) Déterminer le nombre total (N) de segments de corpus qu'on veut manipuler.
(2) Numéroter les segments (S_1, S_2,..., S_i,…, S_N). On note l_i la longueur (vertical) de S_i.
(3) Déterminer la longueur verticale (L (en pixel)) de l'ascenseur. (Cette valeur peut être déterminée en se basant sur la résolution de l'écran),
(4) Diviser l'ascenseur en N portions (P_1, P_2,.., P_i,.., P_N) de même longueur $l=\frac{L}{N}$ avec une marge d'erreur de $\max_i \{lg(P_i)\}$ (Figure 18).

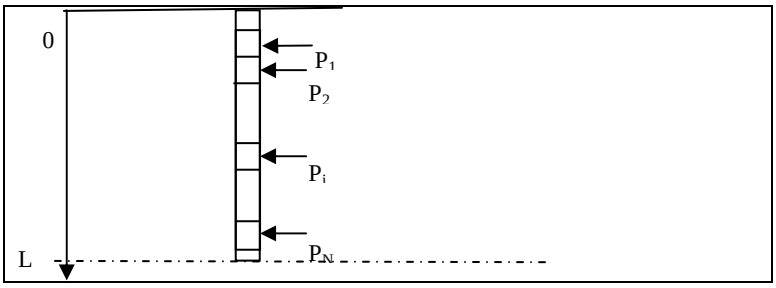


Figure 18 : L'ascenseur et ses portions

(5) Si le curseur est placé à la position P_i, le segment S_i sera le premier dans l'interface (Figure 19). Si l_i < Nm, afficher S_i, S_{i+1}, …, S_{i+m} où $\sum_{k=i}^{i+m} l_k$ >= (Nm).

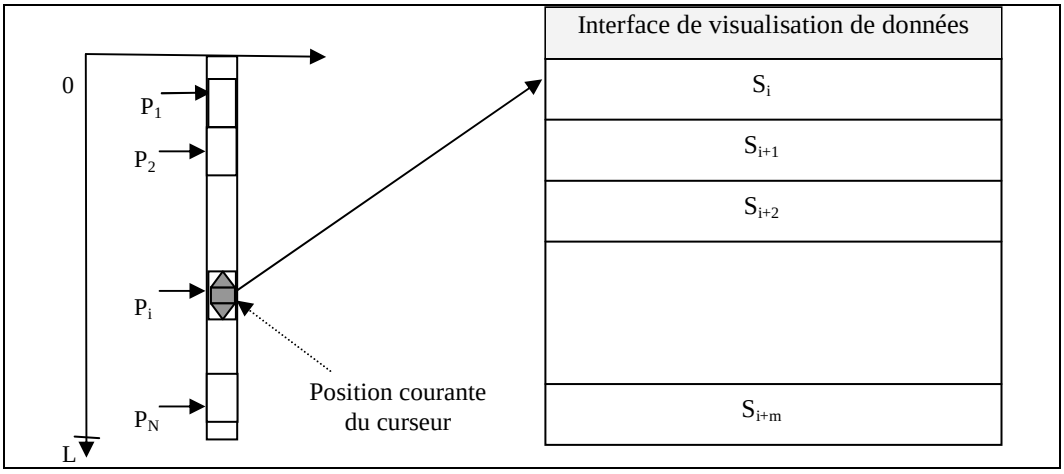


Figure 19 : Le curseur à la position P_i, le segment S_i sera le premier dans l'interface

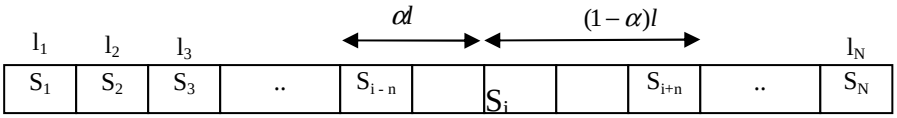
Cependant, cet algorithme posera un problème si la longueur de l'ascenseur est très petite (par exemple 500 pixel) par rapport au nombre total de segments (par exemple 100.000 segments), un pixel sera partagé par plusieurs segments (200 segments). Dans ce cas, on ne peut pas contrôler le curseur au segment correspondant qu'on veut. C'est pourquoi on devrait diviser un grand corpus en plusieurs sous-corpus logiques (C_1, C_2, …, C_K) (chacun comprend par exemple 500 segments).

Si le sous-corpus courant est C_i, le sous-corpus C_{i+1} (i<K) sera pris en compte quand le curseur sera passé dernière la position P_N, et en revanche le corpus C_{i-1} (i>1) sera pris en compte quand le curseur sera passé avant de la position P_1.

Un autre problème est qu'on ne peut pas respecter le "principe de localité", demandant que le segment courant soit toujours à la même place sur l'écran, en général avec plus de contexte au-dessus qu'au-dessous de lui.

b. *Solution 2*

(1) On a les segments S_1...S_N



avec $\sum_{j=1}^{N} l_j = L$ et $\alpha \in [0, 1]$

α est la position (verticale) relative du segment courant dans la partie de la fenêtre contenant les segments.

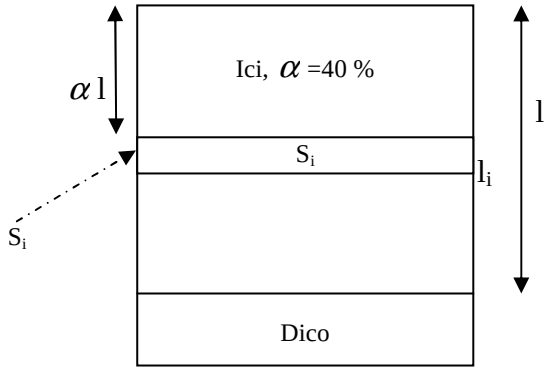
S_i est le segment courant.

(2) On suppose que l'IHM permet de visualiser un ensemble de segments de longueur $\leq l$ (verticale) et

$m = \max \{k \mid \sum_{j=0}^{k} l_{i+j} \leq (1 - \alpha) l \}$,

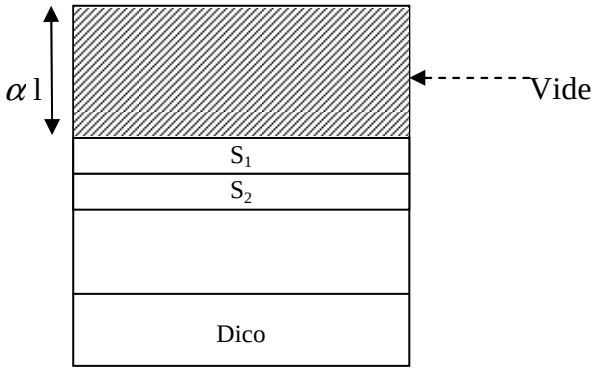
$n = \max \{k \mid \sum_{j=1}^{k} l_{i-j} \leq \alpha l \}$

(3) On suppose que le segment courant commence à αl à partir du haut dans l'IHM.



Alors l'IHM contient $\boxed{S_{i-n}, S_{i-(n-1)}, .., S_i, S_{i+1}, .., S_{i+m}}$

avec la convention que si $j \leq 0$ ou $j \geq N+1$, $S_j = \phi$. L'IHM n'affichera rien avant S_1, par exemple.



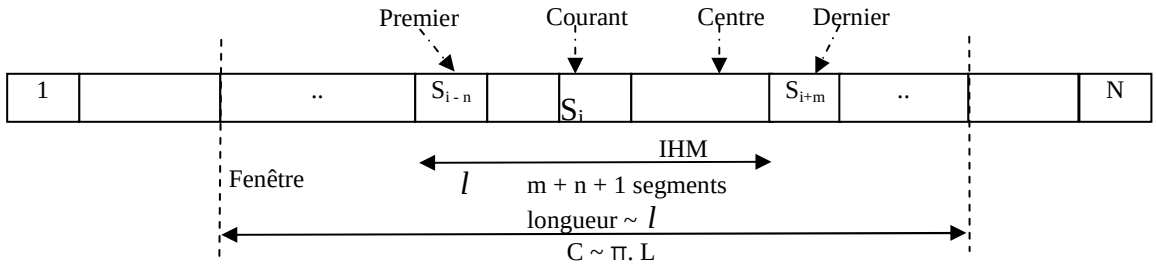
On veut qu'un clic sur l'ascenseur hors du curseur fasse incrémenter ou décrémenter i de k (k = 1 par défaut) et qu'on puisse régler ce nombre (par exemple, 4, ou 10, ou 20, ou ce qu'il faut pour changer de page sans "passer" un segment, donc

$s_i \downarrow \quad i \mapsto i + n$

$s_i \uparrow \quad i \mapsto i - m$

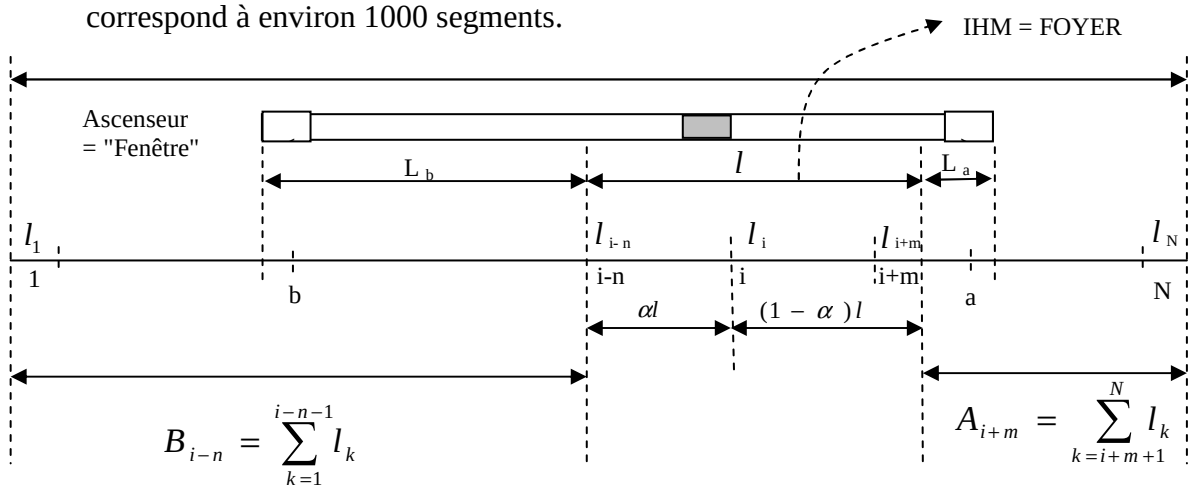
Reste le traitement du déplacement quand on "saisit" le curseur et qu'on le déplace.

On décide à chaque instant ce que représente l'ascenseur : tout (1 à N) ou bien une fraction (glissante) des segments. On a donc :



(longueur correspondant au nombre maximun de segments autour de S_i tenant dans la longueur π. L)

Avec, π est la proportion de L contenue dans la "fenêtre" représentée par l'ascenceur. Par exemple, si L correspond à N=1M segments, et si π = 0,1%, l'ascenseur correspond à environ 1000 segments.



B Before = longueur des segments avant l'IHM.

A After = longueur des segments après l'IHM.

Quand l'ascenseur est "recalé" (placé par rapport à l'IHM), on a :

$$b = \min \{k \mid \sum_{j=k}^{i-1} l_j \le \Pi \sum_{j=1}^{i-1} l_j \},$$

$$a = \max \{k \mid \sum_{j=i}^{k} l_j \le \Pi \sum_{j=1}^{N} l_{i+j} \},$$

On a toujours $\boxed{L = B_{i-n} + l + A_{i+m}}$

Principe (possible) : l'ascenseur montre π % du total, proportionnellement à la position du "foyer" (= ce qui apparaît dans l'IHM, à savoir (S_{i-n} … S_i …S_{i+m})).

Ce qui est montré dans l'ascenseur correspond à S_b … S_{i-n} …S_i …S_{i+m} … S_a

On aura C = π L par exemple C = 10%L.

Quand on recale la fenêtre $\begin{cases} L_b = \pi B_{i-n} \text{ (portion avant le foyer).} \\ L_a = \pi A_{i+m} \text{ (portion après le foyer).} \end{cases}$

La taille du curseur est supérieure ou égale à un minimum (il est au moins carré) et si possible proportionnelle à la taille l_i de S_i

On peut régler π.

On peut "recaler" l'ascenseur en fonction de la position du foyer (IHM) dans le tout (1..N).

Il y a recalage automatique si le foyer (l'IHM) arrive au début ou à la fin de l'ascenseur.

c. Solution 3

Nous proposons aussi une solution pour visualiser de l'endroit où on est dans les données. Pour cela, une technique d'arbre déployable avec ascenseur suffit (comme dans le Finder du Mac ou l'Explorer de Windows) (Figure 20).

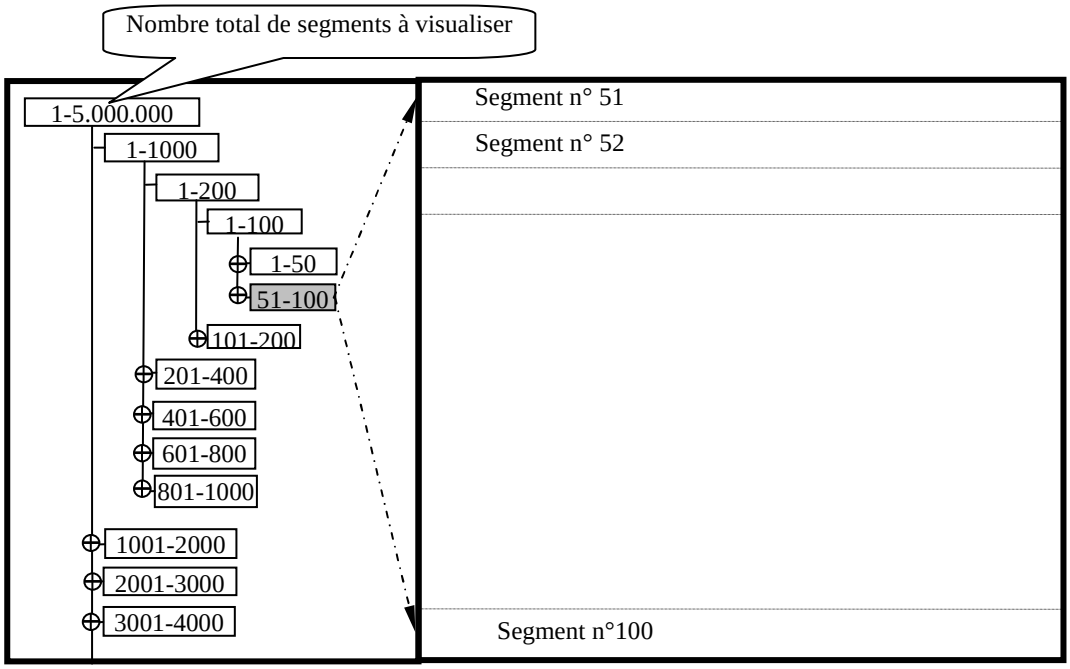


Figure 20:Visualisation d'une masse de données en utilisant la technique d'arbre déployable

Dans la Figure 20, nous voyons un arbre permettant de contrôler la visualisation d'une masse de données (5.000.000 segments). Chaque nœud de cet arbre représente une collection de données. Un nœud plus profond représente une collection plus petite. Un nœud feuille représente un segment.

I.2.3 Problèmes à dominante programmatoire

I.2.3.1 Problème 1.5 : Gestion des utilisateurs et de l'accès aux données

I.2.3.1.1 Analyse du problème

D'abord, il faut introduire la notion de projet dans un système d'évaluation de TA, puisque c'est ce qui est fait pour les campagnes actuelles: une campagne comme IWSLT-09 est un projet, avec des sous-projets (par exemple: évaluation objective, évaluation subjective). Deux projets doivent pouvoir concerner le même corpus. Par exemple, un projet peut consister à refaire une campagne après coup, avec les mêmes données mais avec d'autres systèmes de TA, ou d'autres juges, etc.

La gestion des utilisateurs et de leurs droits pose alors les problèmes suivants.

- Très classiquement, il faut pouvoir définir plusieurs types d'utilisateurs tels que les organisateurs, les évaluateurs, les post-éditeurs, les visiteurs, etc. Comment définir leurs droits d'accès de façon adéquate ? Par exemple, on veut pouvoir, selon l'étape de l'évaluation où l'on est, permettre ou empêcher qu'un juge voie les jugements d'un autre

sur les mêmes données que lui : il faut l'interdire pendant l'évaluation, mais il peut falloir le permettre ensuite, quand on veut analyser finement les désaccords entre juges.

- Il faut aussi pouvoir introduire facilement de nouveaux types d'utilisateurs, et adapter les profils des utilisateurs aux fonctionnalités et aux données existantes. Par exemple :
  - ✓ introduction du type "gestionnaire de projet de post-édition" alors qu'on n'a que "gestionnaire de projet d'évaluation".
  - ✓ à la création d'un nouveau corpus dans le cadre d'un projet existant, adaptation automatique ou semi-automatique des droits d'accès des membres du projet en fonction de leur droits sur les corpus du projet.
  - ✓ création du profil d'un contributeur pour un projet donné (ou pour un sous-projet, par exemple la post-édition vers une nouvelle langue), en fonction du profil général de ce contributeur (compétences linguistiques, domaines de spécialité).
- Quand deux projets utilisent un même corpus, comment séparer les utilisateurs dans chaque projet ? Par exemple, un utilisateur peut-il avoir les droits de travailler sur deux projets, mais un autre seulement sur un de ces deux projets ?
- Dans le cas d'un système permettant le travail contributif, un segment peut être modifié par plusieurs utilisateurs (par exemple, plusieurs utilisateurs post-éditent un segment pour améliorer sa qualité) : comment contrôler ses modifications en empêchant les utilisateurs de modifier les segments déjà post-édités par un utilisateur ayant un niveau traductionnel plus élevé ?
- Un sectra devra déléguer certaines tâches à d'autres systèmes. Comment gérer les utilisateurs quand on bâtit un système global avec des systèmes différents ayant chacun leur gestion des utilisateurs ? D'autre part, comment faire pour qu'un utilisateur puisse utiliser un seul compte pour travailler avec tous les systèmes mis en œuvre ?
- Un dernier problème est alors de savoir comment permettre à des personnes identifiées de se connecter au système d'évaluation, puis, sans devoir faire de nouveau login, à un autre système. Par exemple, comment entrer sur un sectra, et passer ensuite sur un "métasystème de TA" (comme TRADOH [Vo-Trung, 2004b]) pour lancer des traductions automatiques sur des serveurs distants, sans devoir s'identifier de nouveau ?[18]

I.2.3.1.2 État de l'art

Les problèmes relatifs à la gestion des utilisateurs ne sont pas nouveaux. Les fonctions très classiques comme définition, modification, suppression, etc., d'utilisateurs et de groupes existent dans la plupart des systèmes informatiques tels que les systèmes d'exploitation (Linux, Windows, Mac OS, etc.), les systèmes Web Wiki, etc.

Un exemple de gestion d'utilisateurs de services différents est *Google Dashboard*[19] qui permet l'identification automatique de comptes différents d'un utilisateur inscrit dans différents services en ligne de *Google* (Gmail, Youtube, Picasa Album, Google Documents, Google Reader, etc.). Ce service peut donc enregistrer tout l'historique des conversations sur Gtalk, des courriels dans Gmail, des albums photos sur Picasa, des vidéos sur YouTube, et l'historique des recherches Web fait par l'utilisateur (même avec des comptes différents).

[18] On verra plus loin d'autres cas de ce genre, mettant en jeu plus de systèmes, par exemple, pour la postédition contributive, SECTra_w, iMAG, EMEU_w et PIVAX.

[19] https://www.google.com/dashboard/?pli=1

Un exemple d'utilisation d'un seul compte pour la consultation de services différents est *EMEU_w* [Nguyen, 2009] qui est un moniteur utilisable par le Web fournissant un portail unique donnant accès à la fois à plusieurs composants : SECTra_w, PIVAX, TRADOH++, etc.

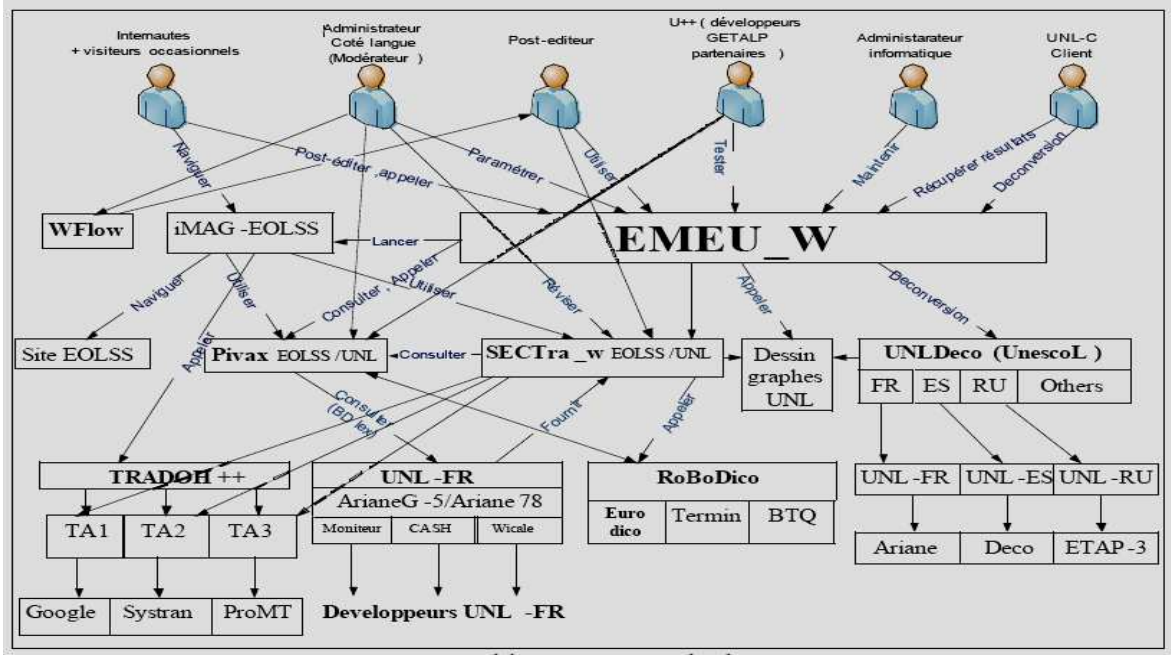


Figure 21: Architecture générale d'EMEU_w

I.2.3.1.3 Solution retenue

Notre solution consiste à :

- utiliser pour chaque système le même type de gestion (très classique) que XWiki, avec deux niveaux: individu et groupe, un individu pouvant appartenir à plusieurs groupes.
- avoir une gestion des utilisateurs pour chaque composant "agent" du système (par exemple, pour une iMAG et pour SECTra_w).
- créer un service unique faisant le lien entre ces gestionnaires d'utilisateurs, par exemple fondé sur un annuaire du genre LDAP, de façon à éviter de demander à un contributeur de faire plusieurs login alors qu'il a tous les droits nécessaires.
- associer à chaque utilisateur un profil général incluant des informations sur ses compétences et ses droits en général (par exemple, niveau traductionnel a priori pour chaque paire de langues, et droit à être administrateur d'un projet).
- associer aussi à chaque utilisateur un raffinement de son profil, pour chaque projet. Par exemple, quelqu'un peut être administrateur mais pas postéditeur dans un projet, et l'inverse dans un autre.

On propose l'idée d'utilisateur "représentant" un autre par délégation (surrogate), par exemple un sectra aurait un utilisateur "iMAG-site_user_fr_vn_3star_rights_ve_nb_2" (numéro 2). La personne physique n'aurait donc pas de compte sur le sectra, mais le relais en aurait plusieurs. On peut utiliser la même idée quand un sectra délègue certaines tâches à d'autres "agents", par exemple à TRADOH++ ou à SEGDOC ou à PIVAX, etc.

On peut aussi prévoir une sorte de LDAP partagé par tous les agents. Le sectra pourrait alors identifier quelqu'un arrivant sous le nom "iMAG-Demo_LIG_user_fr_en_3star_rights_ve_nb_1", par exemple, comme une personne X ayant un compte sur le sectra, et lui proposer de l'utiliser. Pour cela, il suffirait d'interroger le LDAP (annuaire) géré par tous ces agents.

I.2.3.2 Problème 1.6 : Flux de travaux (WF) : organisation des participants et des tâches

I.2.3.2.1 Problème

Nous avons montré au §I.1.2 que l'organisation des participants et des tâches dans une campagne d'évaluation est compliquée. On souvent demande d'effectuer des tâches différentes sur des corpus différents pour un même type d'évaluation. Voici ci-dessous un exemple d'organisation des tâches pour l'évaluation objective liée à la tâche (mesure d'effort de la post-édition).

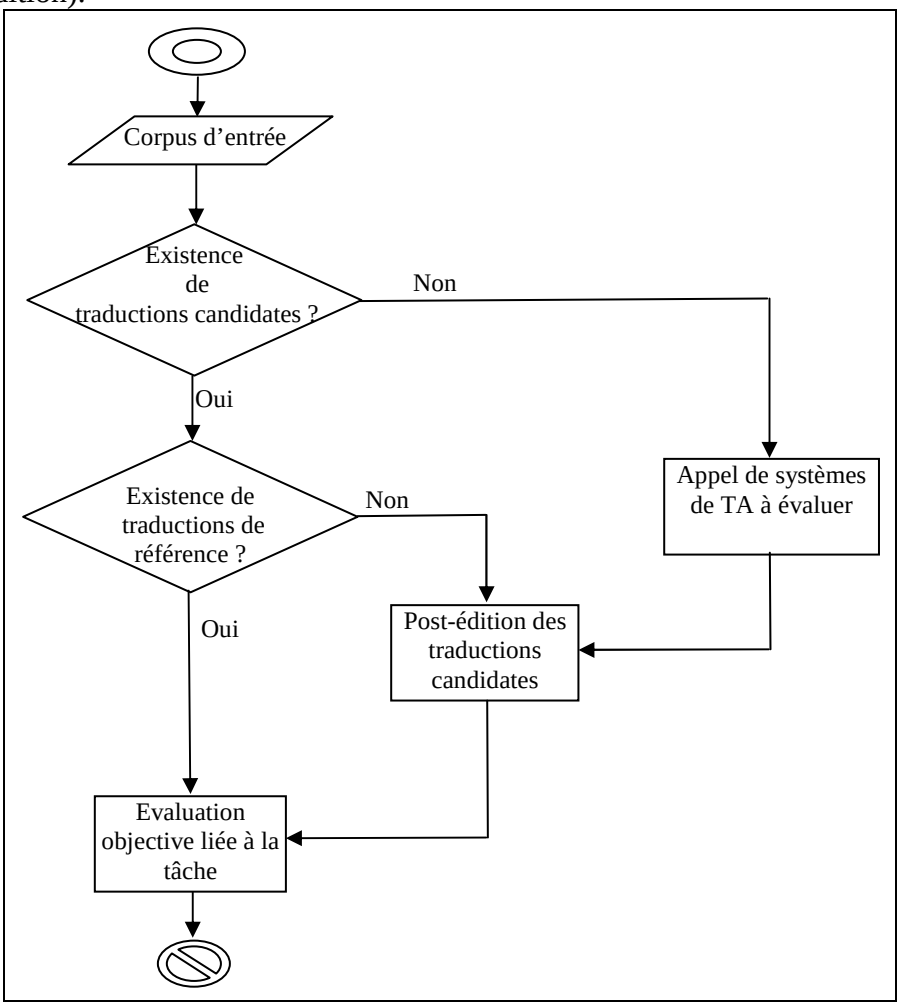


Figure 22 : Complexité d'organisation de l'évaluation objective liée à la tâche dans une campagne

Il serait donc utile de définir les flux de travaux pour effectuer certaines tâches, et puis d'étudier comment gérer les flux de travaux dans un système de support informatique à l'évaluation de résultats de TA. Il nous faut ensuite étudier comment intégrer et adapter des outils permettant la gestion de flux de travaux dans un tel système.

I.2.3.2.2 Etat de l'art

Il y a plusieurs types de flux de travaux. Ces différents types dépendent des objectifs et des besoins des organisations. Par exemple, il y a des flux de travaux pour la production (comme *OpenEDMS*[20]), des flux de travaux administratifs (comme *Relius Administration*[21]), des flux

[20] http://www.altimate.ca/flux de travaux.html

[21] http://www.sungard.com/products_and_services/ebs/relius_administration

de travaux coopératifs (comme *CWS (Collaborative Workflow System)*[22]), des flux de travaux ad hoc génériques (comme *OpenWfe*[23]), etc.

En ce qui concerne le problème de l'intégration des outils dans un autre système, nous pouvons citer le travail de R. Albatal dans le cadre de son M2R [Albatal, 2005]. Ce travail s'est limité à sélectionner des outils adéquats de gestion de flux de travaux pour l'intégration dans une plate-forme de construction collaborative de bases lexicales (Jibiki). Dans ce travail, R. Albatal a étudié comment mettre certaines fonctionnalités du dictionnaire LexALP [Brunet-Manquat et Sérasset, 2006] sous la surveillance d'un outil de gestion du flux de travaux pour améliorer sa performance et pour donner plus de souplesse afin de définir de nouveaux sous-projets ou pour changer les contraintes et les conditions (les variables du contexte) d'une tâche existante. Il a aussi décrit la tâche de l'édition d'un terme dans LexALP, et simulé cette tâche avec OpenWfe [John, 2006].

I.2.3.2.3 Solutions

Nous proposons de définir les flux de travaux suivants pour certaines tâches effectuées dans un système de support à l'évaluation de résultats de TA :

- Flux de travaux pour la tâche de construction de corpus d'évaluation

  (1) Import d'un ou plusieurs corpus source. (2) Extraction des segments source satisfaisant des critères quelconques. (3) Soumission des segments sélectionnés aux systèmes de TA à évaluer et récupération des traductions candidates. (4) Post-édition des traductions candidates et production des traductions de référence. (5) Construction de corpus d'entrées en récupérant les segments source, les traductions candidates, et les traductions de référence.

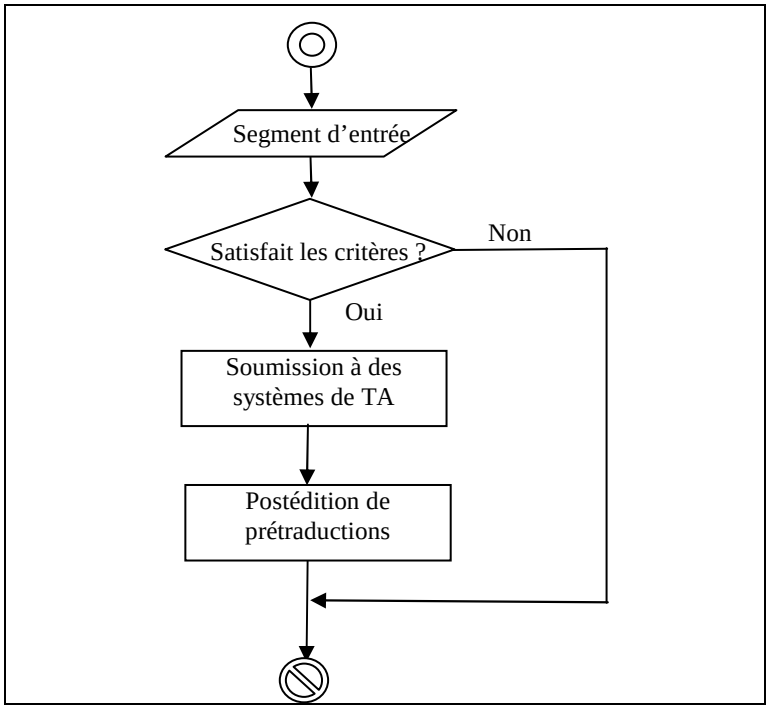


Figure 23: Flux de travaux de construction de corpus d'entrée

[22] http://www.triplearcplc.com/

[23] www.openwfe.org

- Flux de travaux pour la tâche d'évaluation subjective

  (1) Affectation de comptes et de droits aux juges ainsi qu'aux organisateurs. (2) Affectation des tâches par les organisateurs (par exemple, lancement des calculs BLEU, NIST, etc.), et définition de sélections arbitraires (par exemple, définition du nombre de segments d'évaluation alloués à un évaluateur). (3) Affectation des tâches et des données aux juges (par exemple, au juge A sont affectés 200 segments pour effectuer l'évaluation d'adéquation). (4) Réalisation de l'évaluation subjective.

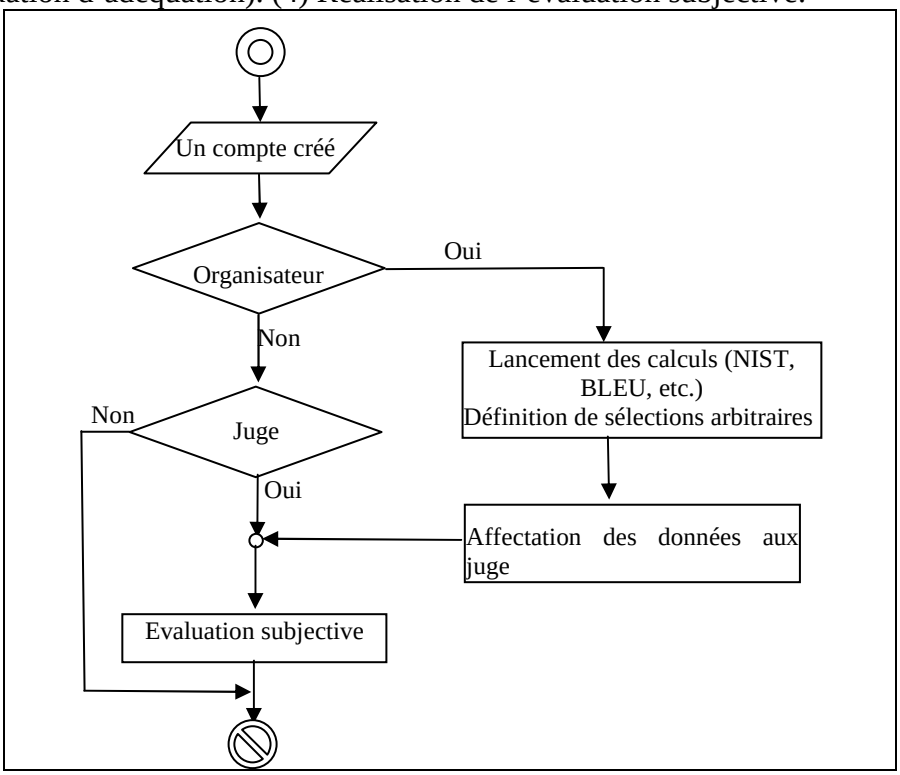


Figure 24: Flux de travaux d'évaluation subjective

- Flux de travaux pour la tâche d'évaluation objective liée à la tâche

  (1) Import de corpus. (2) Soumission à des systèmes de TA à évaluer. (3) Post-édition des résultats de TA. (4) Calcul du temps de post-édition en mesurant une distance d'édition adéquate entre la traduction candidate et le résultat de post-édition (voir Figure 22).

En dehors des flux de travaux présentés ci-dessous, il faut gérer des sous-flux de travaux dans certaines étapes. Par exemple, il faut un WF pour construire des contraintes, comme celle vue plus haut (un post-éditeur ne peut pas post-éditer les résultats d'un post-éditeur ayant un niveau traductionnel plus élevé).

En ce qui concerne la sélection d'un type et d'un outil de flux de travaux à adapter et à intégrer, nous pensons que les flux de travaux génériques et les outils de gestion génériques sont les plus utiles dans notre cas pour les raisons suivantes.

D'abord, ce type de WF permet de facilement personnaliser sa structure. Ensuite, il fournit un module pour gérer les comptes et les droits des utilisateurs, et un module pour spécifier des flux de travaux et les affecter aux différents utilisateurs, qui peuvent être facilement adaptés dans notre cas. Enfin, il fonctionne dans le même environnement logiciel (serveur, base de donné, JDK) que celui que nous utilisons pour développer un système de support à l'évaluation de TA. Nous aborderons encore ce problème au §III.2.2.2.

I.3 Implémentation, expérimentation et évaluation

I.3.1 Spécification et implémentation

Pour pouvoir expérimenter et évaluer les solutions proposées ci-dessus, nous spécifions et implémentons un sectra pour l'évaluation de résultats TA. Nous le baptisons "SECTra_w" pour indiquer qu'il s'agit dès sa conception d'un service Web.

I.3.1.1 Objectifs

Par rapport à l'évaluation de TA, les objectifs sont les suivants :

- support de la création et de la gestion de campagnes d'évaluation,
- support de la création et de l'exploitation de corpus d'évaluation,
- support de divers types d'évaluation subjective et objective,
- support de la visualisation, et de la synthèse des résultats de campagnes d'évaluation.

I.3.1.2 Architecture générale de SECTra_w (partie d'évaluation)

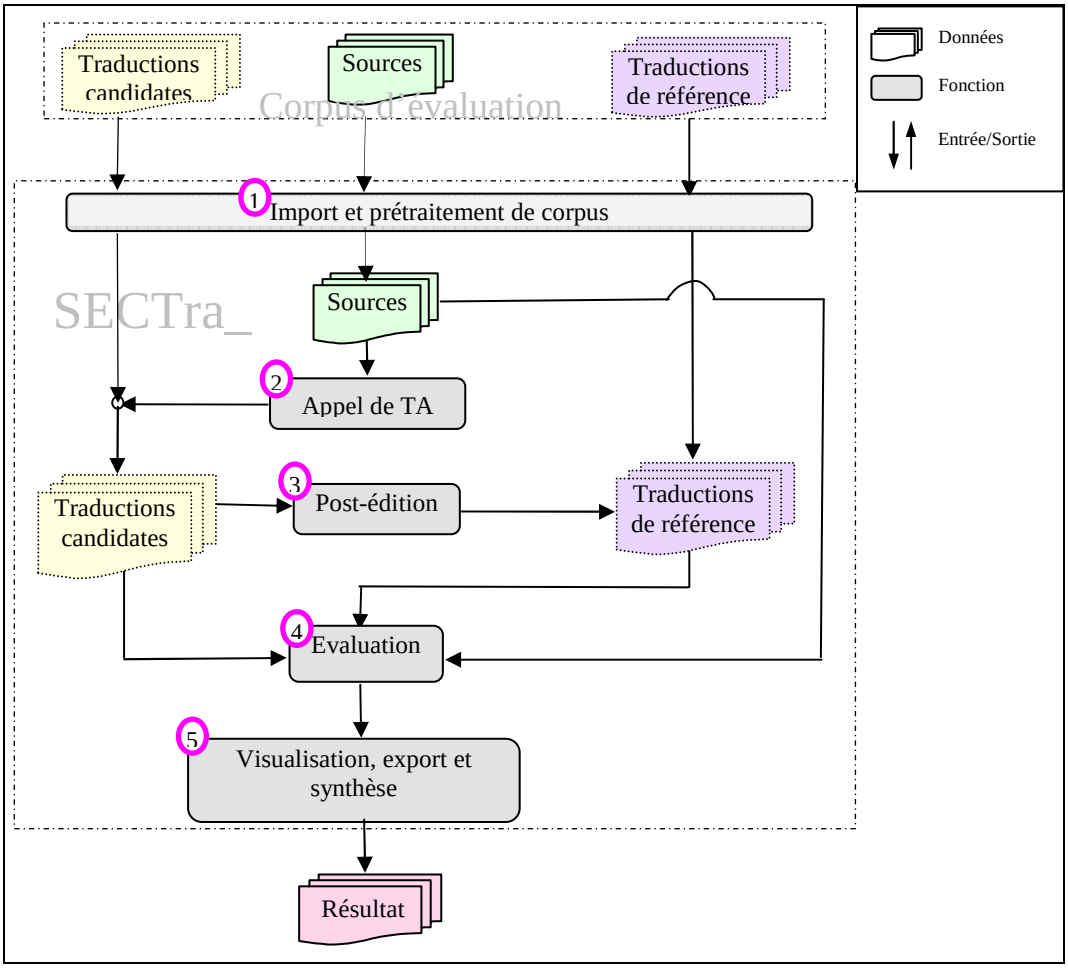


Figure 25: Boîtes fonctionnelles et schéma d'opération de SECTra_w

SECTra_w fournit les fonctionnalités suivantes :

- Import et prétraitement de corpus d'évaluation.
- Appel de systèmes de TA pour produire des traductions candidates.
- Post-édition de pré-traductions pour produire des traductions de référence.
- Outils d'évaluation subjective et objective.
- Interfaces de visualisation de données, et de résultats de campagne d'évaluation.

- Export et synthèse de résultats de campagnes d'évaluation.

I.3.1.3 Spécification des fonctions

I.3.1.3.1 Import et prétraitement de corpus

SECTra_w doit permettre d'importer divers corpus multilingues disponibles et faciliter la sélection des corpus source pour organiser des campagnes d'évaluation.

Nous convenons d'un codage, d'un format, et d'une structure générique pour les corpus à importer dans SECTra_w.

Codage. Un corpus à importer doit être converti en UTF-8.

Format. SECTra_w accepte des fichiers d'entrée en format texte (.txt).

Structure. Au niveau physique : chaque corpus peut être constitué par un fichier source contenant des segments source, un ou plusieurs fichiers candidats (correspondants à différents systèmes de TA ou à différentes langues) contenant des traductions candidates à évaluer, et un ou plusieurs fichiers de traductions de référence. Cependant, un corpus d'évaluation peut ne contenir que des segments source, car SECTra_w permet de produire des traductions candidates par appel de TA, et des traductions de référence par post-édition en ligne.

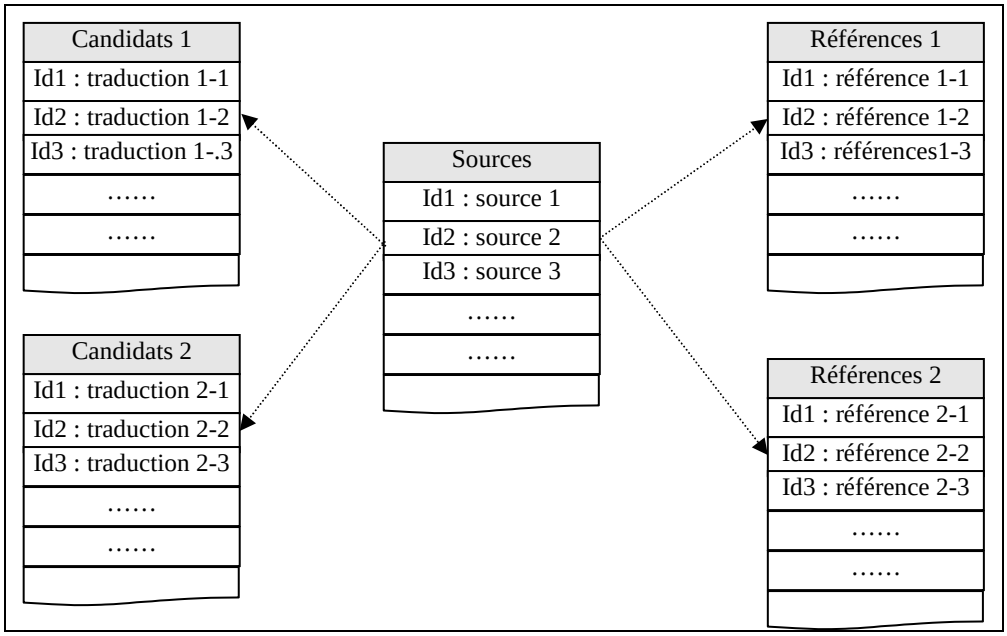


Figure 26: Structure d'un corpus à importer

Au niveau interne : les segments dans les fichiers (source, candidate, et de références) sont alignés les uns avec les autres (voir Figure 26). Chaque segment d'entrée suit la syntaxe suivante :

*id ":" texte ("|" texte)**

Par exemple : *BTEC0032 : Est-ce que je peux vous aider ? |Qu'est-ce que vous aimeriez ?*

I.3.1.3.2 Appel de systèmes de TA

SECTra_w doit pouvoir appeler des systèmes de TA à évaluer sur tout ou une partie d'un corpus. Nous verrons la spécification de cette fonction au chapitre 2, §II.2.2.2.

I.3.1.3.3 Fonction de post-édition de pré-traductions

SECTra_w doit offrir une fonction de post-édition pour produire des traductions de référence. Nous verrons aussi la spécification de cette fonction au chapitre 2, §II.3.1.

I.3.1.3.4 Outils d'évaluation

a. Outil d'évaluation subjective

L'interface d'évaluation subjective doit ne pas être figée, car il faut que les organisateurs puissent la définir selon les besoins de chaque campagne d'évaluation. Elle doit permettre d'effectuer un ou plusieurs types d'évaluation subjective (grammaticalité, fidélité, utilité, etc.) pour un ou plusieurs systèmes de TA à la fois. Elle doit aussi interdire ou permettre à plusieurs juges d'effectuer l'évaluation subjective sur la même unité de données (segment) selon des configurations.

On veut aussi pouvoir utiliser des boutons (radio) ou des étoiles (★★) pour représenter les valeurs des critères de chaque type d'évaluation subjective.

Pour définir un type d'évaluation subjective et ses critères, SECTra_w doit fournir à l'organisateur une interface graphique tabulaire pour faciliter la spécification. Voici un exemple de définition d'un type d'évaluation subjective.

Nom de type	*Adéquation*

Critères	**Valeurs**	**Symboles à visualiser**
Critère 1	*Aucune information n'est transportée, ou contresens*	★
Critère 2	*Peu d'information est transportée*	★★
Critère 3	*La moitié de l'information est transportée*	★★★
Critère 4	*Presque toute l'information est transportée*	★★★★
Critère 5	*Toute l'information est transportée*	★★★★★
Ajouter un autre critère		

b. Outil d'évaluation objective

Un outil d'évaluation objective doit fournir diverses méthodes d'évaluation objective telles que BLEU, NIST, WER, distance d'édition, etc. Il doit aussi être ouvert pour que les participants puissent facilement y accéder, et y intégrer d'autres méthodes d'évaluation objective.

Pour pouvoir intégrer d'autres méthodes d'évaluation objective, SECTra_w doit pouvoir exécuter des scripts en JavaScript, Velocity, Groovy, et des classes Java sous forme de JavaBean ou Plug-in.

SECTra_w doit aussi fournir des interfaces graphiques pour faciliter la sélection de données (tout ou partie d'un corpus), et le lancement des programmes de calcul des mesures d'évaluation objective sur les données sélectionnées.

I.3.1.3.5 Visualisation de données et de résultats de la campagne d'évaluation

e. Visualisation des données

La construction de l'interface doit suivre les principes de présentation suivants :

- Verticalité: tous les objets du même type doivent apparaître dans la même "colonne".

- Horizontalité: tous les objets liés au même segment source (y compris éventuellement ses corrections) constituent une "polyphrase" et sont présentés dans la même ligne.

L'interface doit permettre de changer la taille des colonnes, et de cacher/montrer des colonnes en utilisant la souris ou un tableau de configuration.

Les utilisateurs doivent également pouvoir choisir le nombre de segments affichés à chaque fois sur l'interface. Leurs travaux doivent être enregistrés automatiquement et les données déjà manipulées doivent se distinguer des autres au niveau du fond couleur (voir Figure 27).

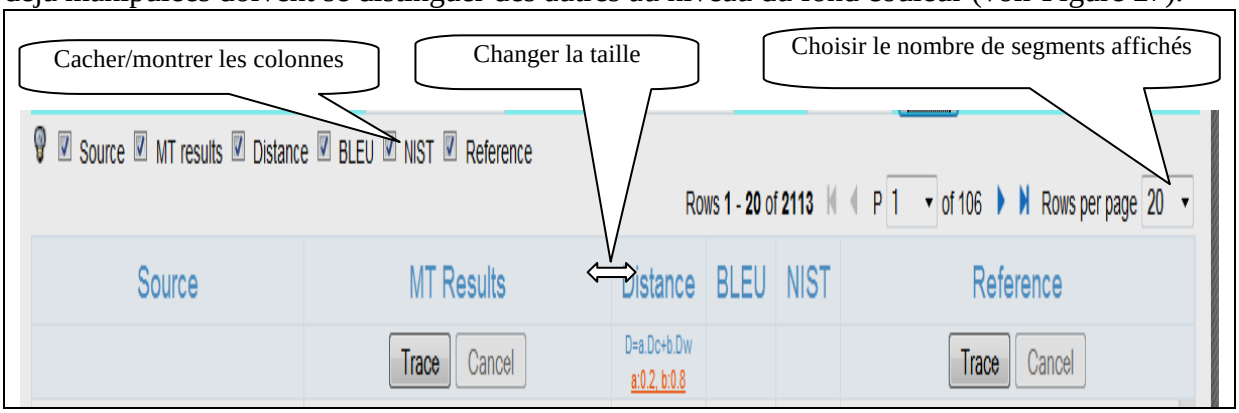


Figure 27: Spécification de la visualisation des données

a. *Visualisation des résultats de la campagne d'évaluation*

SECTra_w doit pouvoir montrer les résultats d'une campagne d'évaluation non seulement par des tableaux de chiffres, mais aussi par des données réelles (sources, résultats de systèmes de TA, traductions de référence, résultats post-édités).

Nous pouvons utiliser l'algorithme de calcul de la distance entre chaînes de [Wagner et Fischer, 1974] (voir §I.2.2.1), en introduisant des balises de formatage HTML pour rendre visible le travail de post-édition (voir Table 3).

Balises de formatages	Texte visualisé	Opération d'édition
 <strike>de force dans</strike>	~~de force dans~~	Suppression
 <u>J'aimerais un</u>	J'aimerais un	Insertion

Table 3: Balises de formatage pour rendre sensible l'effort de post-édition

SECTra_w doit aussi visualiser les résultats d'évaluation subjective d'une façon intuitive pour faciliter la comparaison des résultats fournis par différents juges sur les même données (voir Figure 28).

Traduction candidate	Traduction de référence	Résultats à la fluidité	Juges
J'aimerais quelque pain, s'il vous plaît.	J'aimerais du pain, s'il vous plaît.	★★★★ ★★★ ★★★★★	Xan Hervé Georges

Figure 28: Spécification de la visualisation des résultats d'évaluation subjective

I.1.1.1.2 Export et synthèse de résultats de campagnes d'évaluation

Cette fonction doit permettre aux organisateurs d'exporter les données et les résultats de campagnes dans une structure et dans un format prédéfinis.

On prévoit les options suivantes :

Options	Description
All	Exporter les données avec tous les résultats
Data only	Exporter seulement les données
Results only	Exporter les résultats subjectifs et objectifs
Subjective results only	Exporter seulement les résultats subjectifs
Objective results only	Exporter seulement les résultats objectifs

Table 4: Options d'export des données et des résultats d'une campagne d'évaluation

Cette fonction doit aussi permettre de calculer des statistiques sur les données (par exemple, nombre de mots par phrase : moyenne = 7,9, min = 1, max = 33) et sur les résultats d'évaluation, par exemple :

	Phrases		
	Dword	Dchar	Dsent
min	0	0	0
max	10	43	15,8
moyenne	1,24	5,06	2,09

Table 5: Données quantitatives sur les distances d'édition

I.3.2 Réalisation

Nous avons réalisé une première version de SECTra_w [Huynh et al., 2008b] pour le support à l'évaluation de systèmes de TA selon ces spécifications.

SECTra_w a été développé en utilisant la plate-forme XWiki, la technologie Ajax, et les langages de scripts Javascript et Velocity. Nous avons utilisé MySQL Server 5.1 pour la gestion des données.

Pour consulter les fonctions d'évaluation de SECTra_w, on peut accéder à http://eolss.imag.fr/xwiki/bin/view/Corpus/Evaluation.

Voici quelques éléments factuels.

Tâche	CCH	DSE	DSI	codage	import pour tests	testS 1	import pour la campagne	Tests	**Total**
Jours	10	15	6	30	0,5	0,5	1	2	**65**
Taille	2 pages	6 pages	4 pages	3000 lignes, C classes	200 seg.	2 STA	5000 seg.	2 STA	

Table 6: Taille et coût en temps de l'implémentation de SECTra_w (partie évaluation)

I.3.3 Expérimentation

I.3.3.1 Contexte

En novembre 2007, notre équipe, le GETALP du LIG a participé au projet TRANSAT dans le cadre d'un contrat avec France Telecom R&D. Dans ce projet, nous avons utilisé SECTra_w pour l'organisation d'une campagne d'évaluation ayant pour but d'essayer d'évaluer l'utilité des systèmes de traduction automatique commerciaux dans le domaine de la traduction de la parole. Le domaine privilégié dans cette étude était l'assistance au touriste en situation difficile. Une autre partie de l'étude portait sur le domaine de la restauration.

Les systèmes de TA à évaluer étaient le système Reverso (version 10 Pack Monde), et Systran (version 4) sur la paire de langues anglais-français.

I.3.3.2 Corpus d'évaluation

Dans cette campagne, nous avons utilisé des données issues du corpus BTEC pour construire deux corpus sur ces deux types de tâches relatives au tourisme, ainsi que des dialogues collectés par l'équipe Multicom du LIG [Blanchon et al., 1999]. Le premier corpus contient des tours de parole, relatifs aux problèmes de santé. Le second corpus est constitué de tours de parole, relatifs au domaine de la restauration.

I.3.3.2.1 Corpus de tâche d'assistance (FT Assistance)

a. Construction du corpus

Ce premier corpus source (anglais) a été extrait du corpus BTEC pour obtenir un ensemble de 2224 tours de parole tous différents, en accordant une priorité aux tours de parole relatifs aux problèmes de santé.

Les traductions candidates étaient les traductions de Reverso Translator 10, et les traductions de référence ont été produites par postédition de ces traductions candidates.

b. Composition du corpus

b.i Répartition des tours de parole dans les différentes sous-tâches

Santé : 50,85 %
Accidents, perte, vol : 24,51 %
Itinéraire : 16,28 %
Transports, et Location de voiture : 11,47 %
Non spécifique : 8,63 %

b.ii Tours de parole

Nombre de mots par tour de parole : moyenne = 6,2 ; min = 1 ; max = 30
Nombre de tours de parole à 1 phrase : 2083
Nombre de tours de parole à 2 phrases : 135
Nombre de tours de parole à 3 phrases : 5

b.iii Phrases

Nombre de mots par phrase : moyenne = 5,9 ; médiane = 5 ; min = 1 ; max = 28
Nombre de questions marquées : 717
Nombre d'affirmations (+ questions non marquées) : 1656

b.iv Vocabulaire

Nombre d'occurrences : 13833
Nombre d'occurrences différentes : 1319
Nombre des occurrences apparaissant 1 fois : 590 (45%)
Nombre des occurrences apparaissant 2 fois : 203 (15%)
Nombre des occurrences apparaissant 3 fois : 98 (7%)
Nombre des occurrences apparaissant 4 fois : 65 (5%)

I.3.3.2.2 Corpus de dialogues portant sur la restauration (FT Restaurant)

a. Construction du corpus

Le second corpus source (anglais), qui a aussi été extrait du corpus BTEC, est constitué de 2000 tours de parole dans le domaine de la restauration. Les traductions candidates étaient les traductions de Reverso Translator 10 et Systran (version 4). Les traductions de référence ont été produites par postédition de ces traductions candidates.

b. Composition du corpus

b.i Tours de parole

Nombre de mots par tour de parole : moyenne = 5.2 ; min = 1 ; max = 22
Nombre de tours de parole à 1 phrase : 1896
Nombre de tours de parole à 2 phrases : 97
Nombre de tours de parole à 3 phrases : 5

Nombre de tours de parole à 4 phrases : 2

b.ii *Phrases*

Nombre de mots par phrase : moyenne = 4,93 ; médiane = 5 ; min = 1 ; max = 19
Nombre de questions : 783
Nombre d'affirmations : 1330

b.iii *Vocabulaire*

Nombre d'occurrences : 10429
Nombre d'occurrences différentes : 855
Nombre des occurrences apparaissant 1 fois : 352 (41%)
Nombre des occurrences apparaissant 2 fois : 133 (16%)
Nombre des occurrences apparaissant 3 fois : 64 (7%)
Nombre des occurrences apparaissant 4 fois : 46 (5%)

I.3.3.3 Protocole d'évaluation pour le projet TRANSAT

I.3.3.3.1 Évaluation subjective à la NIST

Nous avons mis en œuvre un protocole (voir Figure 30) légèrement différent du protocole proposé par le NIST dans les campagnes TIDES d'évaluation de systèmes de traduction automatique[24].

a. *Fluidité*

Pour l'évaluation de la fluidité, H. Blanchon a choisi de proposer une échelle de 3 notes, au lieu de 5 dans le protocole NIST standard :

- (F1) formulation parfaitement compréhensible sans effort, que le style soit écrit ou oral.
- (F2) formulation acceptable à l'oral, éventuellement compréhensible en faisant un effort.
- (F3) formulation non acceptable.

b. *Adéquation*

Pour l'évaluation de l'adéquation (transport de l'information pertinente de la source vers la cible), nous avons utilisé une échelle de cinq valeurs, comme dans le protocole NIST standard :

- (A1) Toute l'information est transportée.
- (A2) Presque toute l'information est transportée.
- (A3) La moitié de l'information est transportée.
- (A4) Peu d'information est transportée.
- (A5) Aucune information n'est transportée, ou il y a un contresens.

I.3.3.3.2 Évaluation objective

a. *Évaluation objective fondée sur la distance d'édition*

Nous avons choisi trois mesures: la distance d'édition en mots, la distance d'édition en caractères, et une distance d'édition pondérée.

Les distances d'édition [Damerau, 1964 ; Levenshtein, 1966 ; Wagner et Fischer, 1974] en mots et en caractères considèrent les opérations d'insertion, de suppression et de remplacement en attribuant à chacune de ces opérations un poids de 1.

Ces mesures permettent de se rendre compte du travail de correction nécessaire à l'obtention de traductions utiles pour la tâche à partir des traductions candidates produites par le système.

[24] http://projects.ldc.upenn.edu/TIDES/Translation/TransAssess04.pdf

Ce genre de distance est aussi utilisé lors des évaluations du projet GALE dans le cadre de la méthode d'évaluation HTER [Przybocki et al., 2006 ; Snover et al., 2006].

La distance d'édition pondérée combine la distance d'édition en mots et la distance d'édition en caractères en donnant un poids de 0,2 à la première et un poids de 0,8 à la seconde. Il est en effet plus rapide de faire des manipulations sur les mots (un double clic pour la sélection) que sur des caractères individuels.

Cette méthode correspond à des propositions de [Blanchon et Boitet, 2007].

b. Évaluation objective n-gramme

Nous avons a intégré dans SECTra_w les scripts fournis par NIST pour calculer BLEU et NIST. Cependant, ils n'ont pas été expérimentés pendant la réalisation du projet TRANSAT lui-même, à cause de la limite de temps d'implémentation de SECTra_w.1, mais seulement un peu plus tard (fin décembre 2007).

I.3.3.4 Gestion de la campagne

I.3.3.4.1 Sécurité des données et gestion des utilisateurs dans SECTra_w

La gestion des utilisateurs a effectivement assuré la sécurité des données ainsi que des évaluations.

SECTra_w divise les utilisateurs en 4 groupes: administrateurs, organisateurs, contributeurs, et visiteurs.

Les administrateurs, qui sont les développeurs de SECTra_w, ont tous les droits sur le système et créent des organisateurs. Les organisateurs ont le droit de créer des campagnes d'évaluation en donnant les noms des campagnes, en important des corpus, en créant des comptes d'évaluateurs, en affectant les travaux aux évaluateurs, etc. Les contributeurs ont le droit d'effectuer l'évaluation subjective (fluidité, adéquation), et de post-éditer les prétraductions. Les visiteurs n'ont que le droit de voir les données, et pas le droit de changer ou d'éditer les données.

SECTra_w permet également aux organisateurs de définir des profils de post-éditeur, dans lesquels chaque post-éditeur a un nombre d'étoiles correspondant à son niveau traductionnel pour une paire de langues et relativement à un projet (chaque projet correspond à un domaine de traduction) (Figure 29), ainsi qu'une note (de 0 à 20) par défaut.

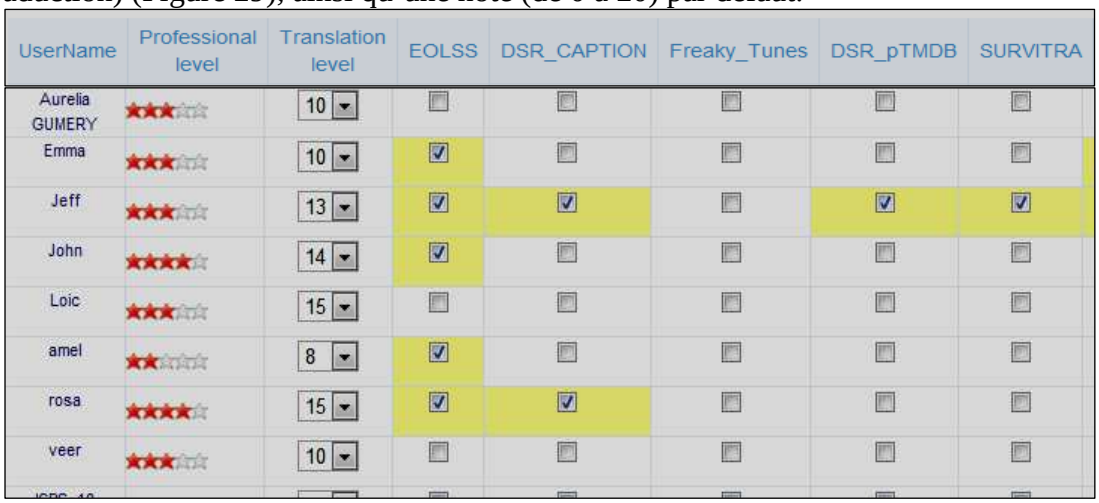


Figure 29: Interface de définition de profils de post-éditeurs

Notons que le niveau traductionnel et la note par défaut d'un post-éditeur peuvent être différents, non seulement selon les différentes paires de langues, mais aussi selon les différents projets.

I.3.3.4.2 Organisation des participants et des tâches

Dans cette campagne d'évaluation, Hervé Blanchon était le chef de projet ("organisateur" dans SECTra_w). Il a créé 8 évaluateurs (Hervé Blanchon, Georges Fafiotte, Chistian Boitet, Jean Philippe Guilbaud, Achille Falaise, Jean-Claude Durand, Jérôme Goulian, Didier Schwab), et leur a donné le droit d'effectuer l'évaluation subjective. La distribution des travaux aux évaluateurs a été faite facilement, en utilisant le fait que SECTra_w permet de diviser les corpus d'évaluation en ensembles de données dits "pages logiques". Chaque évaluateur avait la responsabilité d'effectuer l'évaluation subjective (fluidité, adéquation) sur un certain nombre de pages logiques, et pouvait définir lui-même le nombre de segments dans une page logique.

Chaque traduction candidate était jugée par plusieurs évaluateurs. Pour le projet TRANSAT, chaque traduction candidate a été jugée par 3 évaluateurs.

I.3.3.5 Visualisation des données et des résultats des évaluations

I.3.3.5.1 Visualisation des corpus

Pour cette campagne, l'interface générique a été paramétrée de la façon suivante : les textes source sont dans la première colonne, la colonne de post-édition est à droite, et les traductions candidates sont dans la colonne du milieu.

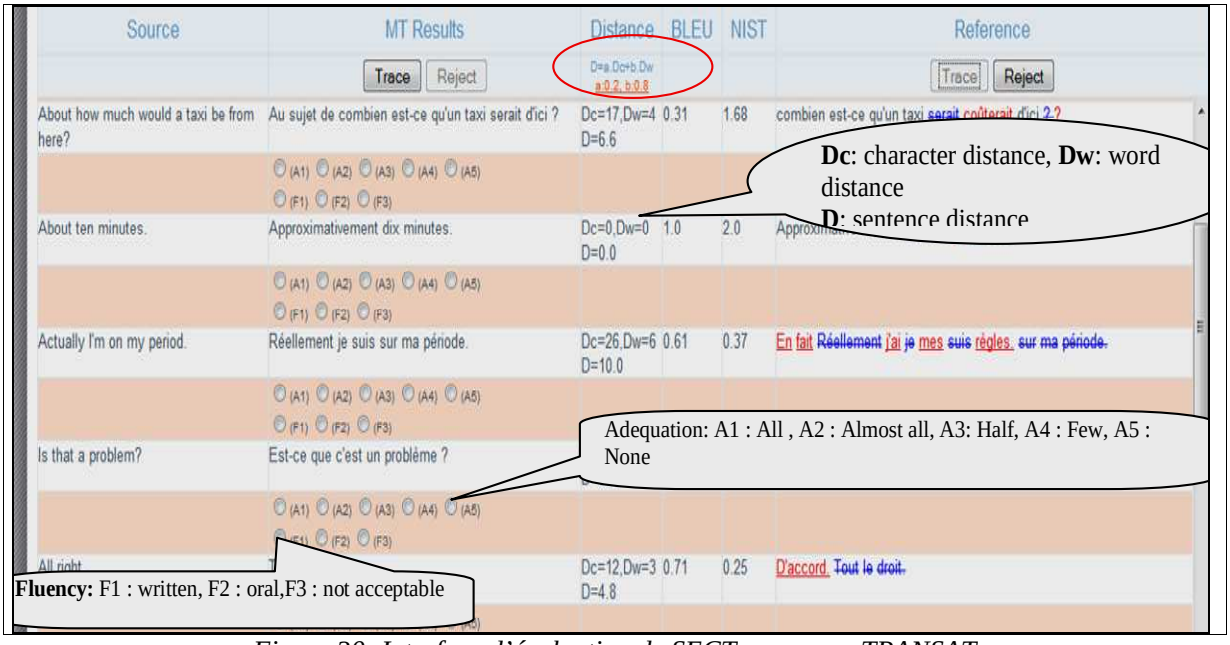


Figure 30: Interface d'évaluation de SECTra_w pour TRANSAT

Les évaluateurs n'ont pas souhaité la modifier, sauf pour régler la largeur des colonnes et le nombre de segments par page logique (par défaut, une page logique contenait environ 20 segments).

Par rapport à la toute première spécification, on a ajouté deux choses demandées par les évaluateurs : les modifications sont enregistrées automatiquement dès qu'on change de segment, et les segments déjà traités sont mis dans une couleur différente des autres.

I.3.3.5.2 Visualisation des résultats de la campagne d'évaluation

La visualisation des résultats de campagnes d'évaluation a été réalisée exactement selon les spécifications. Dans l'exemple suivant (Figure 31), toutes les cases de contrôle ont été cochées, et l'interface montre donc les scores BLEU, NIST, WER (« D_w »), « D_c» (distance basée sur les caractères, et D (distance de phrase : $D = \alpha Dc + \beta Dw$ (α est modifiable, et $\beta = 1 - \alpha$)) ainsi que les segments source, les résultats des deux systèmes de TA, et les traductions de référence.

Dans la Figure 31, on voit aussi la représentation intuitive par "trace" des différences entre les résultats de TA et la traduction de référence. Les chaînes insérées sont visualisées en rouge, et les chaînes effacées sont visualisées en bleu et sont barrées. Les résultats d'évaluation subjective de plusieurs juges sur un segment d'évaluation sont visualisés ensemble et en parallèle.

Le pourcentage d'achèvement du travail, le temps de travail, les mots source de post-édition, etc., d'un contributeur sont montrés aux organisateurs.

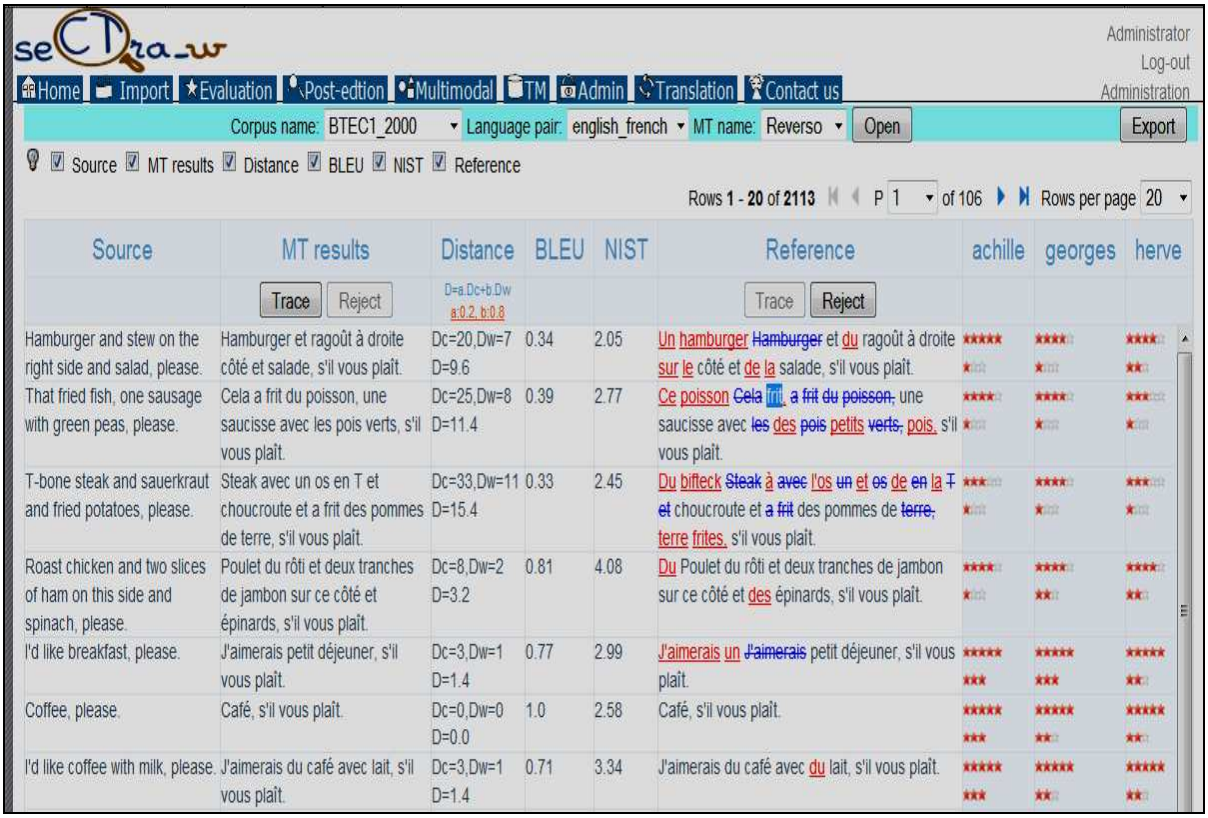


Figure 31: Visualisation des résultats de la campagne d'évaluation pour TRANSAT

Sur cette figure, on voit conjointement les résultats d'évaluation subjective de trois évaluateurs (Achille, Georges, Hervé), visualisés d'une façon intuitive. Pour chaque segment évalué, il y a deux lignes contenant des étoiles, correspondant respectivement au résultat d'adéquation et au résultat de fluidité. Le nombre d'étoiles montre le niveau de qualité de la traduction d'un segment.

I.3.3.6 Quelques résultats de la campagne d'évaluation TRANSAT

I.3.3.6.1 Évaluation objective

Pour chaque phrase de chaque tour de parole, SECTra_w a calculé la distance d'édition en mots (Dword), la distance d'édition en caractères (Dchar) et la distance d'édition combinée

(Dsent). Nous avons obtenu les résultats suivants en retenant les distances d'édition égales à 0, 1, 2 ou 3 pour les phrases et les tours de parole.

	Phrases			Tour
	Dword	Dchar	Dsent	Dsent
=0 / %	797 / 37,78	797 / 37,77	797 / 37,78	715 / 32,15
=1 / %	416 / 19,72	68 / 3,22	54 / 2,56	47 / 2,11
=2 / %	403 / 19,10	72 / 3,41	83 / 3,93	80 / 3,60
=3 / %	192 / 9,10	80 / 3,80	64 / 3,03	65 / 2,92

Table 7 : Distances d'édition pour la tâche de restauration avec Reverso

Voici le minimum, le maximum, la moyenne et la médiane, sur les phrases et les tours de parole pour les différentes distances.

	Phrases			Tour
	Dword	Dchar	Dsent	Dsent
min	0	0	0	0
max	16	64	25,6	25,6
moyenne	1,60	6,30	2,54	2,68
médiane	1	4	1,8	2

Table 8 : Données quantitatives sur les distances d'édition pour la tâche de restauration avec Reverso

I.3.3.6.2 Évaluation subjective

Voici les résultats d'évaluation subjective consolidés en évaluation stricte et en évaluation généreuse pour le corpus FT Restaurant traduit avec Reverso.

a. Évaluation stricte

a.i Pour les phrases

Valeur	Nombre	%
Yes	1374	65,03
No	208	9,84
A2	170	8,05
A3	103	4,87
A4	101	4,78
A5	157	7,43
Total	2113	100

Table 9: Accord consolidé par phrase pour l'adéquation

Valeur	Nombre	%
Yes	1889	89,40
No	0	0,00
F3	224	10,60
Total	2113	100

Table 10 : Accord consolidé par phrase pour la fluidité

a.ii Pour les tours de parole

Valeur	Nombre	%
Yes	1274	63,70
No	244	12,20
A2	165	8,25
A3	96	4,80
A4	92	4,60
A5	129	6,45
Total	2000	100

Table 11 : Accord consolidé par tour pour l'adéquation

Valeur	Nombre	%
Yes	1779	88,95
No	16	0,80
F3	205	10,25
Total	2000	100

Table 12 : Accord consolidé par tour pour la fluidité

Valeur	nombre	%
All	1756	87,80
No A	228	11,40
No F	0	0,00
No A&F	16	0,80
Total	2000	100

Table 13 : Accord consolidé par tour pour les 2 critères

b. Évaluation généreuse

b.i Pour les phrases

Valeur	Nombre	%
Yes	1615	76,43
No	137	6,48
A3	103	4,87
A4	101	4,78
A5	157	7,43
Total	2113	100

Table 14 : Accord consolidé par phrase pour l'adéquation

Valeur	Nombre	%
Yes	1889	89,40
No	0	0,00
F3	224	10,60
Total	2113	100

Table 15 : Accord consolidé par phrase pour la fluidité

b.ii *Pour les tours de parole*

Valeur	Nombre	%
Yes	1511	75,55
No	172	8,60
A3	96	4,80
A4	92	4,60
A5	129	6,45
Total	2000	100

Table 16 : Accord consolidé par tour pour l'adéquation

Valeur	Nombre	%
Yes	1779	88,95
No	16	0,80
F3	205	10,25
Total	2000	100

Table 17 : Accord consolidé par tour pour la fluidité

Valeur	nombre	%
All	1827	91,35
No A	157	7,85
No F	1	0,05
No A&F	15	0,75
Total	2000	100

Table 18 : Accord consolidé par tour pour les 2 critères

I.3.4 Évaluation de SECTra_w

Méthodologie

Nous évaluons maintenant SECTra_w dans l'esprit de ce qui précède, c'est à dire en essayant de mesurer dans quelle mesure les problèmes mentionnés ci-dessus ont été résolus par notre implémentation et par la façon de l'utiliser.

Evaluation du problème 1.2 : Modélisation et traitement d'entrées non textuelles. Dans la campagne d'évaluation TRANSAT, toutes les entrées étaient textuelles, ce qui ne permettait pas d'évaluer cet aspect.

Nous avons cependant pu l'évaluer en important une partie du corpus ERIM [Fafiotte, 2004] contenant des dialogues oraux français-vietnamien à 3 participants (agent, client, interprète) avec les fichiers son originaux, et des transcriptions écrites (voir les détails de ce corpus au chapitre 2, II.2.1.1). Nous avons importé environ 3 heures de ce corpus dans SECTra_w. On peut maintenant "rejouer" un dialogue, compléter les transcriptions, et traduire les segments (à l'écrit) dans d'autres langues. Tout cela fonctionne bien.

Autrement dit, ce problème a été traité au niveau des entrées sonores, mais pas encore au niveau des entrées sous forme de graphes (treillis ou graphes de chaines).

Evaluation du problème 1.3 : Visualisation intuitive des données et des résultats des évaluations. Pour juger de cet aspect, on peut évaluer la convivialité, l'utilisabilité, et l'ergonomie des interfaces pour la visualisation des données et des résultats des évaluations.

Les interfaces d'évaluation subjective ont été trouvées très conviviales et faciles à utiliser par les juges. Les points les plus positifs sont que les corpus sont présentés en parallèle, ce qui facilite beaucoup le travail de comparaison, et que les modifications sont sauvegardées automatiquement. Avoir deux couleurs différentes pour distinguer les segments évalués et non encore évalués est aussi très utile. Par contre, pouvoir modifier la taille et le nombre de lignes et de colonnes dans les interfaces n'a pas été jugé essentiel par nos évaluateurs, alors que l'organisateur a beaucoup apprécié cela.

La visualisation des résultats d'évaluation subjective a été améliorée au cours de la campagne à la demande de l'organisateur et est maintenant jugée excellente.

Enfin, la visualisation de type "trace" a reçu les félicitations de tous, y compris des visiteurs occasionnels. Elle est beaucoup plus intéressante que des tableaux de chiffes. La possibilité de la montrer dans les deux sens est jugée particulièrement originale et intéressante.

Evaluation du problème 1.4 : Le problème de l'ascenseur. Nous avons proposé deux algorithmes permettant de parcourir et visualiser une masse de données comme le corpus EuroParl, ou une grande mémoire de traductions fusionnée à partir de plusieurs corpus.

Nous avons implémenté le premier algorithme (voir I.2.2.2.3). Cependant, dans la campagne d'évaluation TRANSAT, le nombre de segments des deux corpus utilisés pour cette campagne est au total 5.000, ce qui est trop petit par rapport à celui que nous voulons tester. Nous avons donc importé une partie du corpus EuroParl, environ 100.000 segments (soit 1.500.000 mots), puis testé cet algorithme sur cette masse de données. Cet algorithme marche très bien.

Cependant, au niveau technique, cet algorithme nécessite d'utiliser la technologie Ajax, et le cache de données sur le client et sur le serveur. Par conséquent, il y a quelques limitations, par exemple la technologie *Ajax* ne marche pas avec tous les navigateurs tels que le navigateur *Internet Explorer* version 6 et antérieure. Si le navigateur sur le client ne permet pas de cacher des données, ces algorithmes ne peuvent pas marcher.

Enfin, cet algorithme ne permet pas de réaliser complètement le principe de localité, car il ne peut pas forcer le segment courant à rester à une place (horizontale) fixée à l'avance dans l'écran.

Evaluation du problème 1.5 : Gestion des utilisateurs. Les problèmes liés à la gestion des utilisateurs ont été traités de façon jugée très satisfaisante. Nous avons adapté toutes les fonctions classiques de gestion des utilisateurs fournies par XWiki et nous avons développé des fonctions avancées dans SECTra_w.

Evaluation du problème 1.6 : Flux de travaux (WF) : organisation des participants et des tâches. Nous avons défini des flux des travaux pour certaines tâches, ce qui a facilité l'organisation des participants et des tâches. Cependant, nous n'avons pas implémenté toutes ces propositions, mais seulement le flux de travaux pour la tâche d'évaluation objective liée à la tâche (voir I.2.3.2).

Conclusion

Il s'agissait dans ce chapitre d'étudier les problèmes posés par la conception et la réalisation d'un sectra (système d'exploitation de corpus de traductions) utilisable pour le développement et surtout l'évaluation de systèmes de TA.

Pour cela, nous avons étudié dans les EDL de TA la gestion des suites de test, et de plusieurs différents types de corpus (corpus pour les tests, corpus applicatifs, corpus parallèles, etc.). Nous avons aussi étudié les supports informatiques mis en œuvre dans des campagnes d'évaluation compétitives, et les supports informatiques à l'évaluation de systèmes de TA « en opération ».

Nous avons conclu qu'il n'y pas encore de vrai sectra pour la TA. La gestion des suites de test est organisée assez simplement dans les EDL. Pour la gestion de corpus parallèles bruts, il existe seulement des sites Web permettant de télécharger des fichiers comme les sites Web pour les corpus EuroParl, BTEC. Pour la gestion de corpus enrichis d'annotations linguistiques ou sémantiques, il y a seulement des systèmes comme CASH et UNLdeco [Sérasset et Boitet, 1999] offrant quelques opérations sur les corpus contenant des graphes UNL.

Parmi les EDL, nous avons présenté la gestion de corpus dans l'EDL d'Ariane-G5 comme un exemple intéressant. Nous avons conclu qu'Ariane-G5 offre un environnement de gestion de corpus de textes, mais souffre encore de plusieurs limitations.

Nous avons également étudié plusieurs problèmes intéressants et émergents qui nous ont conduit à proposer de nouvelles notions, des principes généraux, et des solutions pour construire un sectra unifié pour l'évaluation de systèmes de TA.

Nous avons construit un système appelé SECTra_w selon nos spécifications. Ce système a été utilisé avec succès, fin 2007, pour une campagne d'évaluation dans le cadre du projet TRANSAT de FT R&D. Avec SECTra_w, après avoir importé un corpus source, et éventuellement les traductions de référence, on peut appeler plusieurs systèmes de TA, stocker leurs résultats, et demander à des juges d'effectuer l'évaluation subjective (fluidité, adéquation). SECTra_w fournit plusieurs méthodes d'évaluation objective (NIST, BLEU, etc.), et permet aussi d'effectuer l'évaluation objective liée à la tâche, en permettant à des participants de post-éditer les résultats de systèmes de TA, et en mesurant un distance d'édition (et/ou le temps de post-édition). Les résultats post-édités peuvent être ajoutés à l'ensemble des traductions de référence, ou le constituer s'il n'y a pas de références.

Ce système a été évalué par ses utilisateurs et par les résultats qu'il a donnés. C'est un très bon environnement, et on nous le demande. Le défi a bien été relevé. Les résultats sont corrects et stables, utilisables de façon opérationnelle dès la première mise en service.

Cependant, plusieurs problèmes liés au support informatique à la post-édition de résultats de systèmes de TA n'ont pas encore été présentés dans ce chapitre. Nous les aborderons dans le chapitre 2.

En effet, un sectra complet devait permettre d'exploiter non seulement les corpus parallèles, mais aussi les corpus de documents multimodaux, et multi-annotés. Mais, nous rencontrons toujours des difficultés à cause de la variété et de la différence des structures et des formats des corpus de traductions.

Bien que SECTra_w soit un bon environnement, il faut encore rendre totalement générique la préparation et l'organisation d'une campagne d'évaluation.

Chapitre II
Support contributif au travail humain sur des corpus variés en contexte multilingue

Introduction

Le besoin de traductions de référence pour les campagnes d'évaluation de TA a été présenté dans le chapitre 1. On trouve aussi un besoin en corpus parallèles pour la TA « empirique » directe, surtout en TA statistique, et un besoin en corpus enrichis par des annotations et des représentations interlingues UNL dans des variantes de TA "empirique" indirecte, et de TA "experte".

En TA "empirique", on a besoin de 50M mots (selon K.Knight à CICLING-05) jusqu'à 200 M mots (selon Ph. Koehn à un séminaire de l'UE sur les recherches futures en TA début 2006). En TA "experte", il suffit de quelques centaines de pages en langue source, et de ressources terminologiques bilingues ou multilingues pour guider le développement. En TA indirecte par les exemples, on utilise des bi-textes "préparés", et les tailles nécessaires (selon la méthode) sont moins grandes, par exemple 30.000 phrases, soit environ 2.000 pages [Boitet, 2007].

Un réel problème est alors la nécessité de la mutualisation (collaboration) du travail humain pour obtenir ces corpus. En effet, pour construire des corpus parallèles, on peut utiliser des méthodes d'acquisition de textes multilingues à partir de ressources multilingues [Resnik, 1998, 1999 ; Koehn, 2005 ; Munteanu et Marcu, 2006 ; Do et al., 2009], ou étendre des corpus existants à d'autres langues en appelant des systèmes de TA. Cependant, on a besoin de beaucoup de travail humain pour la révision et la post-édition humaine (de 15mn à 45mn par page standard de 250 mots) pour obtenir des données utilisables.

Pour les corpus enrichis, on n'a pas besoin de corpus gigantesques, mais les problèmes qui restent sont l'édition manuelle ou assistée, destinée à améliorer la qualité, et l'hétérogénéité des annotations, car les structures et les annotations associées peuvent être très complexes. Le coût de la construction est donc élevé. Par exemple, dans le cas des S-SSTC (synchronized structured string-tree correspondences), il faut environ 15h par page de travail humain, à partir d'une première version calculée automatiquement. Pour la GDA (Global Document Annotation) proposée à MPEG-7 par Koichi Hashida, c'est "seulement" 8h/p.

L'idée générale dans ce chapitre est donc d'étudier le support collaboratif et contributif au travail humain sur des corpus de traductions. Les travaux humains sur les corpus de traductions sont la post-édition, l'annotation, l'évaluation, l'extension, etc.

Dans la première section de ce chapitre, nous étudions l'état de l'art des supports au travail collaboratif sur les corpus de traductions, éventuellement multimédia et multiannotés, ce qui amène à dégager 6 problèmes à résoudre pour atteindre cet objectif. Dans la deuxième section, nous proposons des solutions à ces problèmes. Nous terminons ce chapitre par l'implémention, l'expérimention et l'évaluation des solutions proposées dans la deuxième section. Parmi les activités du travail humain, nous choisissons l'implémentation, et l'expérimention de l'activité qui nous semble la plus indispensable : la post-édition collaborative et contributive.

II.1 État de l'art et problèmes émergents

II.1.1 État de l'art

II.1.1.1 Systèmes permettant la post-édition contributive de traductions automatiques

Il existe quelques outils permettant la post-édition de traductions automatiques. Par exemple, *MTpost-éditor* (voir §I.1.1.2.2), développé par le NIST, a été utilisé dans le projet GALE [Meghan et al., 2008]. *SYSTRAN Review Manager* (voir §I.1.1.3.1) est utilisé chez Systran. Dans quelques projets tels que *TraCorpEx, B@bel Unesco*, et les campagnes d'évaluation telles que *IWSLT*, on a utilisé des outils classiques, tels que MS Word, MS Excel, etc., comme outils de post-édition.

Cependant, ces outils ne nous intéressent pas, car ils ne nous fournissent qu'un service minimum pour la post-édition, et ne tournent qu'en local. Nous nous concentrons sur des systèmes qui tournent en ligne, et permettent la post-édition contributive et collaborative de traductions automatiques.

Apparemment, il n'y a pas de système complet de support à la post-édition contributive et collaborative de traductions automatiques. Cependant, dans quelques systèmes en ligne utilisés pour la traduction humaine, on trouve les prémices de tels supports.

Nous avons étudié plusieurs systèmes en ligne utilisés pour la traduction humaine, tels que *Google Translator Toolkit* [Google, 2009], *BEYtrans* [Bey, 2009], *Yakushite.net* [Kitamura et al., 2003 ; Yakushite.Net, 2009], *Translationwiki.net* [Translatewiki.net, 2010], *Traduwiki* [Traduwiki, 2010], *Caitra* [Koehn, 2009], etc. Les trois systèmes que nous analysons ci-dessous sont de bons exemples, car ils offrent effectivement un support à la post-édition en ligne. Ce sont *Google Translator Toolkit, Beytrans,* et *Caitra.*

Au niveau du support informatique et linguistique à la post-édition dans ces systèmes, nous étudions particulièrement les aspects suivants : l'ergonomie de l'interface de travail, le support des ressources linguistiques (suggestions, dictionnaires, glossaires, etc.), le support au travail collaboratif et contributif, et le support de fonctionnalités telles que le filtrage, le chercher-remplacer, etc.

II.1.1.1.1 Google Translator Toolkit (1/3 p.)

Google Translator Toolkit[25] (Figure 32) est un système pour la traduction de documents en ligne. Ce système est constitué à partir de deux composants importants : l'éditeur en ligne de *Google Docs* et le système de TA *Google Translate***.**

[25] http://translate.google.com/toolkit/, visité en janvier 2010.

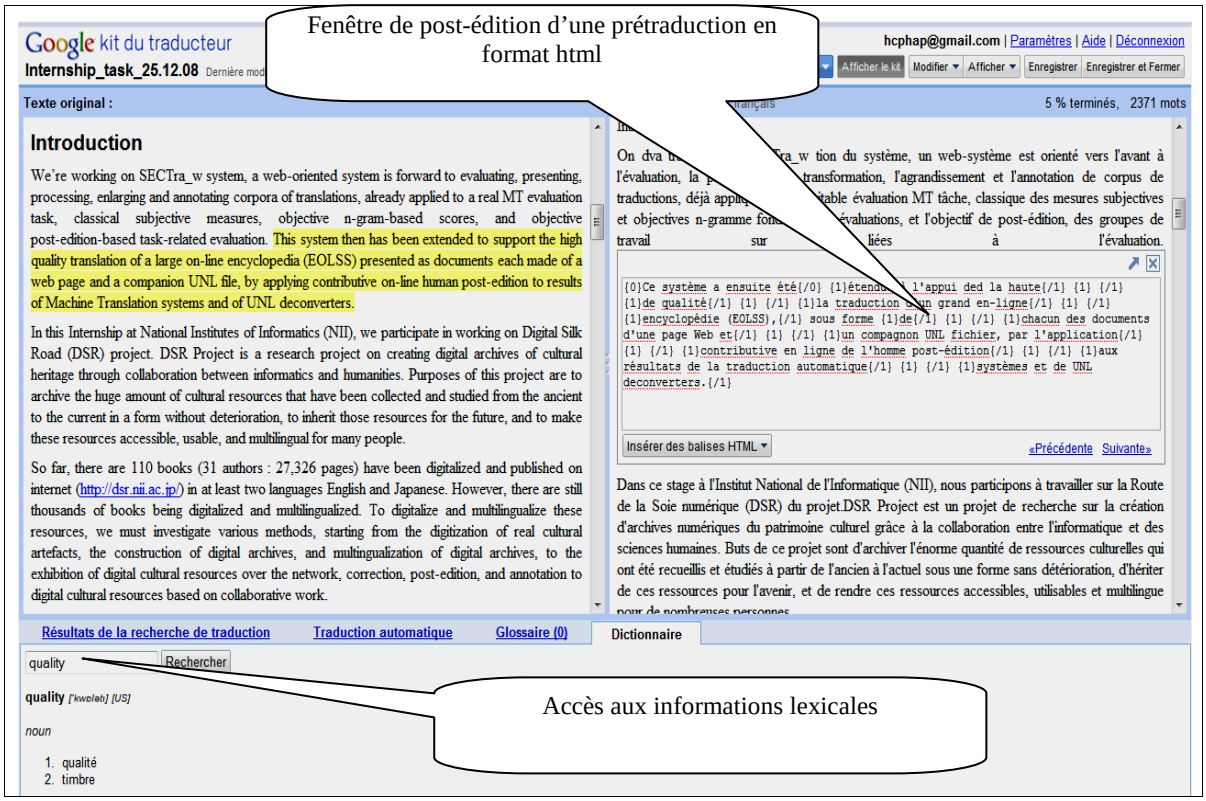


Figure 32: Interface de post-édition de Google Translator Tookit

Ce système offre aussi un support à la post-édition collaborative de documents. Après que l'utilisateur a importé un document source et choisi une paire de langues à traduire, ce système appelle *Google Translate* pour traduire le document source en langue cible (sachant que *Google Translate* traduit un segment en utilisant soit son moteur de traduction, soit sa mémoire de traductions). Il appelle ensuite *Google Docs* pour visualiser les documents source et cible en parallèle. Il permet enfin de post-éditer le document cible segment par segment, grâce à une fenêtre affichant la pré-traduction en format html.

La présentation de deux documents entiers (source et cible) en parallèle avec une visualisation intuitive synchronisée entre un segment source et une traduction facilite beaucoup la post-édition de documents contenant des segments cohérents.

Cependant, cette présentation n'est pas nécessaire, et est même pénible pour la post-édition de corpus de segments, dont les segments ne forment pas un cohérent, tels que le corpus *BTEC* [Kikui et al., 2003 ; Takezawa et al., 2002]. En effet, les post-éditeurs préfèrent souvent post-éditer des segments courts séparés qu'un grand paragraphe (cette remarque a été faite dans quelques projets de traduction effectués dans notre équipe par des traducteurs professionnels).

De plus, à cause de cette présentation, *Google Translator Toolkit* ne peut travailler qu'avec un document assez petit (seuls les fichiers ne dépassant pas 1 MB sont acceptés [Google, 2009]). Mais un système complet de support à la post-édition doit travailler non seulement avec des documents petits, mais aussi avec de grands corpus tels que *EuroParl*, *JR Acquis*, etc.

La fenêtre de post-édition présente une prétraduction sous forme de code HTML. Cela est pénible pour les post-éditeurs. Il vaudrait mieux que la version de la prétraduction montrée aux postéditeurs ne soit que le contenu textuel, sans code. Ou bien, il faudrait montrer aux post-éditeurs une version sans code, à côté de la version avec code à post-éditer.

Google Translator Toolkit peut enregistrer les modifications des post-éditeurs automatiquement. Cela est très important et utile pour les post-éditeurs, car ils sont toujours

sûrs que leurs travaux sont bien sauvegardés. Cependant, *Google Translator Toolkit* ne permet pas de sauvegarder seulement ce qui a été modifié, il sauvegarde le document entier (même s'il n'y a qu'un seul caractère modifié dans un seul segment). Cela est très inefficace pour la post-édition d'un document assez grand, puisque la sauvegarde d'un document entier dans un système en ligne implique le transfert de ce document du client au serveur.

Google Translator Toolkit fournit aussi plusieurs aides à la post-édition telles que le dictionnaire, le chercher-remplacer, etc. Par contre, il ne contient pas d'outil permettant aux post-éditeurs de communiquer, de partager leurs idées, expériences, etc.

Enfin, *Google Translator Toolkit* n'utilise que les résultats du système de TA *Google Translate* comme pré-traductions pour la post-édition. Les post-éditeurs ne peuvent donc pas choisir de partir des résultats d'autres systèmes de TA dans le cas où les résultats de *Google Translate* sont mauvais.

II.1.1.1.2 BEYTrans

BEYTrans [Bey, 2009] est un environnent de traduction collaborative destiné aux traducteurs bénévoles. Les utilisateurs peuvent importer des documents, pour permettre à des traducteurs bénévoles de les traduire, grâce à un éditeur de traductions couplé à des fonctions de support de la traduction humaine.

BEYTrans permet aussi la post-édition collaborative de traductions automatiques. Après avoir été importé, le document source peut être d'abord traduit en langue cible par *Google Translate* ou par une mémoire de traductions. Ensuite, BEYTrans visualise les deux documents source et cible en parallèle dans la partie gauche de l'interface. Enfin, il permet aux traducteurs bénévoles de post-éditer le document cible en cliquant et post-éditant segment par segment.

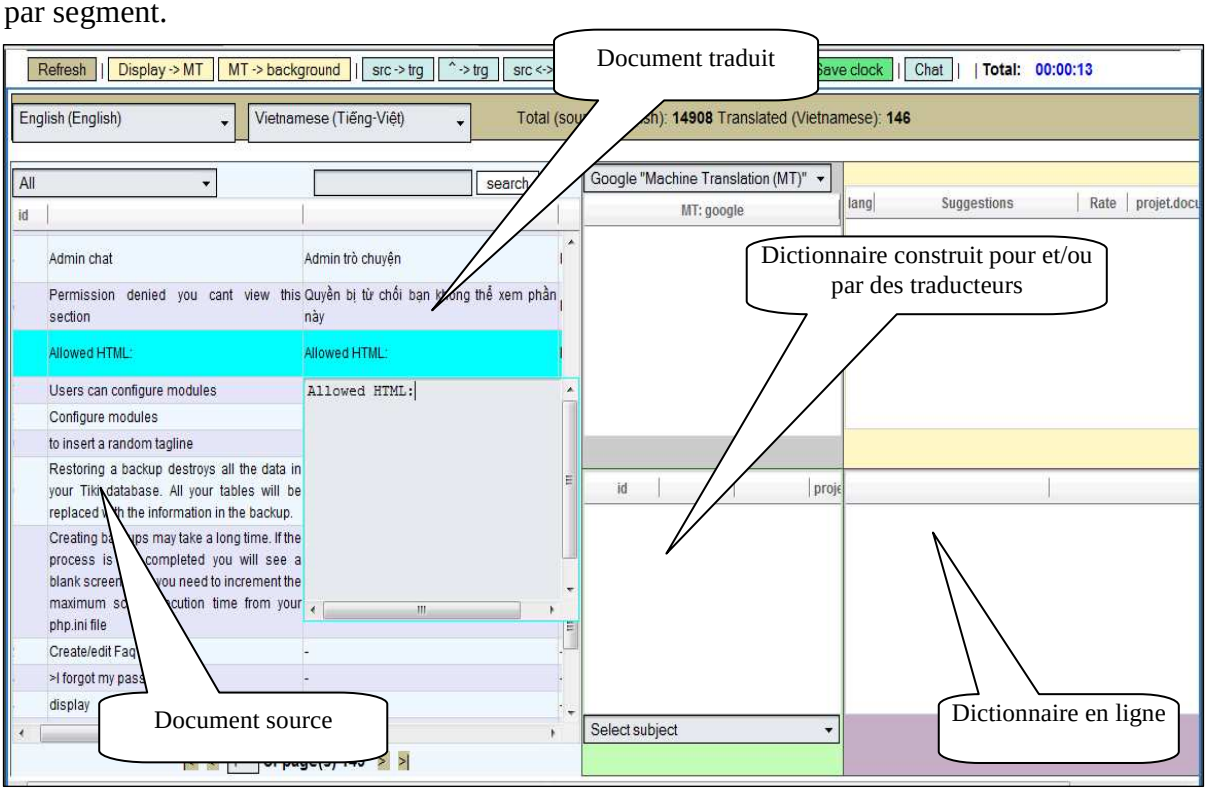


Figure 33: Interface de post-édition de BEYTrans

La présentation des documents dans *BEYTrans* est très semblable à celle de *Google Translator Toolkit (voir §II.1.1.1.1).* Il y a donc les mêmes avantages et les même limitations que *Google Translator Toolkit*, présentés ci-dessus.

Cependant, hors de ces limitations, l'interface de post-édition de *BEYTrans* a encore plus de limitations. L'espace où sont les traductions est petit, alors que les espaces à droite sont parfois vides, mais on ne peut ni les cacher, ni les diminuer.

Pour post-éditer un segment, on doit cliquer sur sa traduction deux fois, puis attendre l'apparition d'une fenêtre permettant la post-édition. Cela fait perdre du temps et finit par être exaspérant quand on répète souvent cette action sur un grand nombre de segments.

BEYTrans fournit plusieurs caractéristiques et fonctions permettant aux traducteurs de collaborer. Les traducteurs peuvent communiquer les uns avec les autres pour partager, échanger leurs idées, etc. via un tchat (ces échanges ne sont vus que par les intervenants). Par contre, on ne peut pas mettre des commentaires, des questions, des explications de façon à ce que toutes les personnes (organisateurs, post-éditeurs, réviseurs) dans le projet puissent les voir n'importe quand.

Plusieurs traducteurs peuvent post-éditer un document en même temps dans *BEYTrans*. Cependant, il n'y a pas de gestion de versions de post-édition sur un segment. Si deux post-éditeurs travaillent sur un segment, le post-éditeur qui finit le premier perdra son travail, même si sa version est meilleure que l'autre.

BEYTrans ne permet de voir qu'une langue source et une langue cible à la fois, et ne permet pas d'afficher d'autres langues comme langues de référence. Or, dans le cas où le post-éditeur ne comprend pas bien la langue source, ou bien où quelques segments source ne sont pas clairs, il lui serait très utile de voir le segment dans une langue de référence (pour lui), le mieux étant bien sûr d'utiliser une langue qu'il connait et pour laquelle la post-édition a été effectuée.

II.1.1.1.3 Caitra

Caitra [Koehn, 2009] est un système en ligne qui permet aux traducteurs humains de post-éditer des résultats de systèmes de TA.

Caitra segmente le document importé en unités de traduction, puis les traduit dans la langue cible en utilisant Moses [Koehn et al., 2007]. Enfin, *Caitra* permet aux traducteurs de post-éditer le document cible segment par segment (Figure 34).

Caitra permet de visualiser les documents source, cible, et post-édité en parallèle. Les segments équivalents (source, traduction automatique, post-édition) sont repérés par un numéro, mais ne sont pas présentés en tableau, ce qui rend la lecture assez difficile. *Caitra* permet aussi de rendre sensible les opérations d'édition effectuées afin de transformer une traduction automatique en post-édition. Cette présentation devrait être très utile pour un système de support à la post-édition, car elle facilite beaucoup la révision des résultats de la post-édition.

Les suggestions que *Caitra* montre aux traducteurs sont sous deux formes, la traduction automatique en phrases (forme 1), et la traduction automatique en mots ou fragments (forme 2). La suggestion en mots ou fragments est représentée en termes de graphes lexicaux qui permettent aux traducteurs de choisir en cliquant sur chaque mot ou fragment pour construire une meilleure prétraduction (*Caitra* ne permet pas de post-éditer directement la traduction automatique de forme 1). Cette façon de faire pose problème dans le cas où la traduction automatique de forme 1 est presque parfaite, car les post-éditeurs sont toujours forcés de choisir des mots pour construire leur post-édition, au lieu de simplement modifier un petit peu la traduction automatique de forme 1. Cependant, un aspect intéressant est que *Caitra* prépare à l'avance (de façon proactive) des informations lexicales pour le segment. Les post-éditeurs peuvent voir tout de suite ces informations quand un segment est actif.

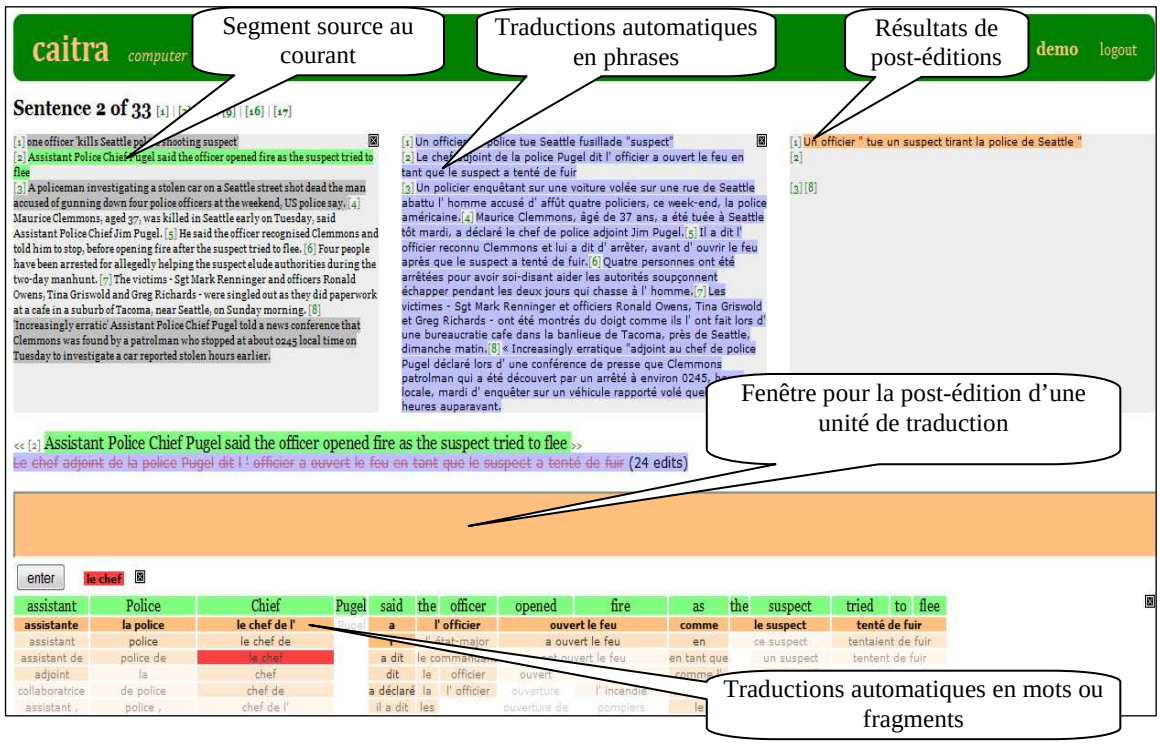


Figure 34: Interface de post-édition de Caitra

Bien qu'il y ait quelques points intéressants dans ce système, *Caitra* n'est pas un vrai système de support collaboratif et contributif à la post-édition de traductions automatiques.

- L'interface de post-édition n'est pas assez ergonomique.
- Les post-éditeurs doivent cliquer sur un segment et attendre assez longtemps pour pouvoir le post-éditer.
- Il n'y a pas de gestion des versions de post-édition.
- On ne peut pas appeler d'autres systèmes de TA pour obtenir d'autres suggestions dans le cas où les résultats de Moses sont mauvais.
- Il manque des fonctions permettant de communiquer entres post-éditeurs, et des fonctions aidant à post-éditer tels que le chercher-remplacer, le filtrage, etc.

Conclusion

Bien que les systèmes utilisés pour la traduction humaine présentés ci-dessus fournissent des supports à la post-édition, ce ne sont pas de bons supports informatiques à la post-édition collaborative et contributive. En effet, beaucoup de fonctionnalités y sont absentes ou trop limitées, en particulier en ce qui concerne l'ergonomie de l'interface, le support des informations linguistiques, le support à la communication entres utilisateurs, la gestion des post-éditions et la qualité des post-éditions.

Ces systèmes ne conviennent pas pour la post-édition de grands documents. La post-édition de corpus de segments n'est pas efficace dans ces systèmes. Notamment, ils ne conviennent pas pour l'affectation de tâches de post-édition aux post-éditeurs.

II.1.1.2 Systèmes de création et d'extension collaborative de corpus de traductions

Un système de création et d'extension collaborative de corpus de traductions doit permettre de créer un nouveau corpus, ainsi que d'étendre un grand corpus existant de façon collaborative. L'extension « *verticale* » permet de créer de nouveaux segments, et l'extension

« *horizontale* » permet d'ajouter de nouvelles langues dans un corpus existant en appelant des systèmes de TA, et puis en les post-éditant de façon collaborative.

Nous avons cherché en vain de tels systèmes. Nous avons seulement trouvé deux systèmes qui se rapprochent un peu de cet objectif et qui sont intéressants pour notre étude : *ERIM-Collecte* et *XTM-INLT*.

II.1.1.2.1 ERIM-Collecte : collecte de corpus de parole

ERIM-Collecte [Fafiotte, 2004] a été construit dans le cadre du projet ERIM (Environnement Réseau pour l'Interprétariat Multimodal), visant à créer un environnement en réseau pour l'aide à la communication orale multilingue, dans lequel on puisse collecter des corpus de dialogues parlés spontanés bilingues et multilingues.

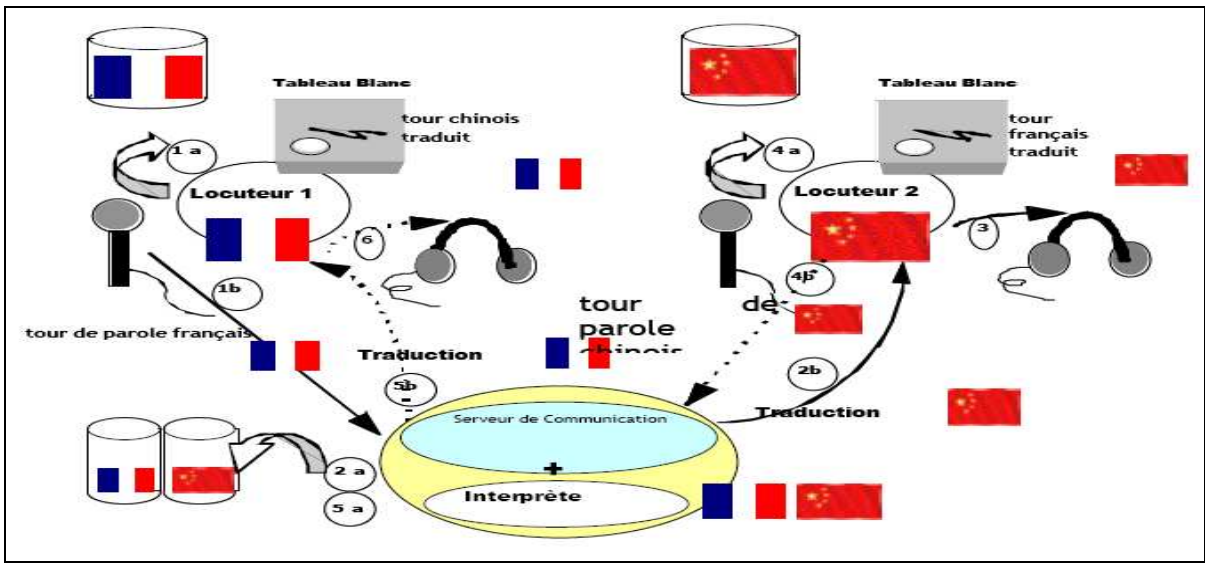


Figure 35: Architecture du système ERIM-Collecte

ERIM-Collecte permet l'enregistrement systématique des actes et données de l'interaction pour tous les participants (deux locuteurs ou plus, un interprète ou plus). L'enregistrement est fait localement lors de la conversation. En fin de dialogue, les descripteurs et fichiers produits localement sont transmis à un serveur de collecte, où ils sont regroupés et structurés [Fafiotte, 2004].

Bien que ERIM-Collecte permette de créer des corpus ERIM via le réseau, ses composants (ERIM Collecte clients, et ERIM Collecte serveur) doivent être installés en local.

Nous avons rendu ce corpus accessible et extensible horizontalement (par ajout de transcriptions, de traductions et d'annotations) dans notre système SECTra_w [Huynh et al., 2008a]. L'extension « verticale » s'identifie dans ce cas à la collecte, et nous ne l'avons pas abordée. ERIM la réalise d'ailleurs très bien.

II.1.1.2.2 XTM-INLT, suite complète pour la traduction (traditionnelle) sur le Web, avec WF

XTM-INLT [XMLINTL, 2009] est un système Web permettant de gérer et d'exploiter des mémoires de traductions pour traduire des documents. C'est un système du même genre que Similis[26] ou Trados[27]. Il a un serveur central, et le travail est distribué et récupéré via le Web. Il y a un outil pour gérer la terminologie (qui reste sur le serveur).

[26] http://similis.org/linguaetmachina.www/index.php

[27] http://www.trados.com/

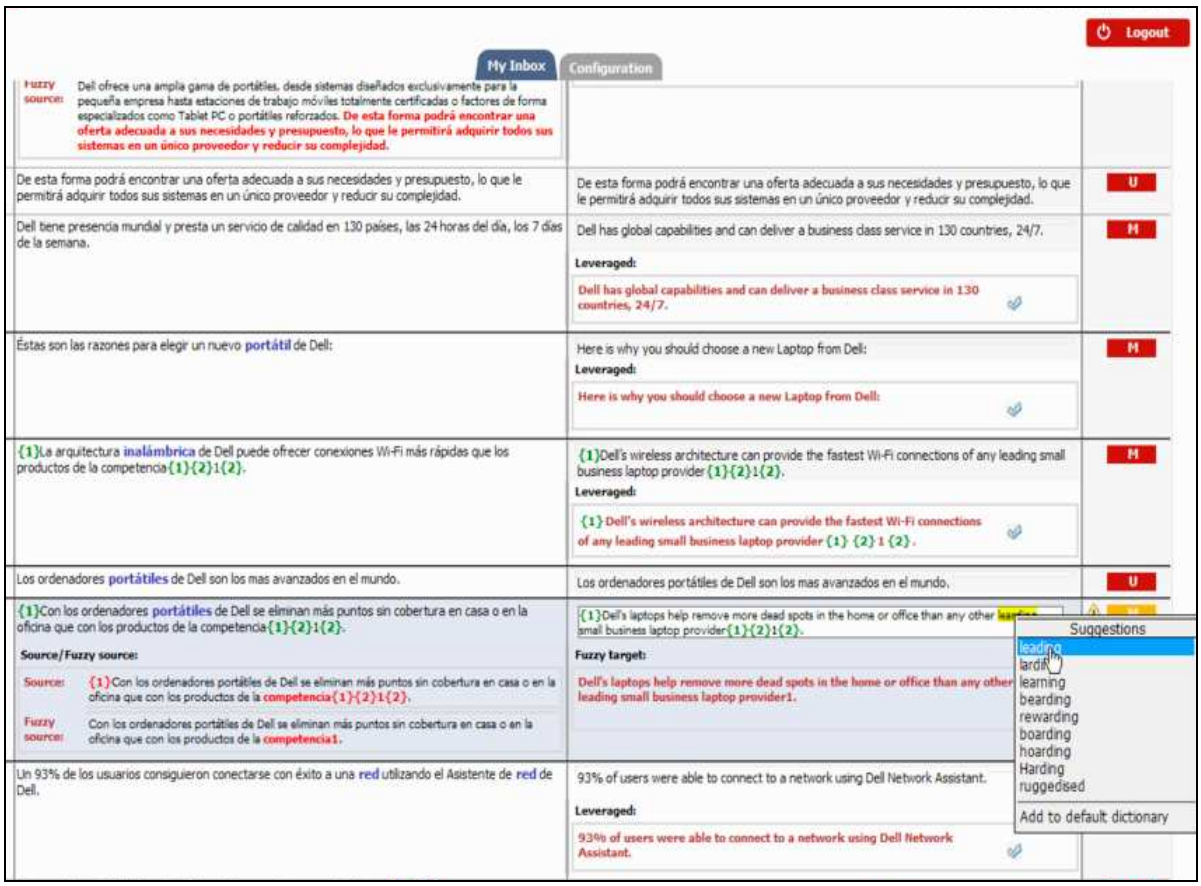


Figure 36: Editeur de traductions de XTM-INLT

Les points intéressants dans ce système sont son éditeur de traductions (Figure 36) qui permet aux traducteurs de travailler à travers un navigateur Web, et son composant « workflow » qui permet de traduire un document selon des flux de travaux.

Conclusion

Les deux systèmes *ERIM-Collecte* et *XTM-INLT* que nous venons de présenter ne sont pas des systèmes de création et d'extension collaborative de corpus de traductions. Cependant, nous nous intéressons à quelques aspects marquants de ces systèmes. Avec *ERIM-Collecte*, nous nous intéressons à rendre ce système accessible et disponible dans notre environnement, pour permettre de créer des corpus ERIM de façon collaborative. Avec *XTM-INLT*, nous nous intéressons à son éditeur de traductions permettant aux traducteurs de travailler à travers un navigateur Web, et à son composant de flots de travaux.

II.1.1.3 Systèmes d'étude et d'annotation collaborative de corpus de traductions

Un système de support à l'étude et à l'annotation collaborative de corpus de traductions doit permettre de visualiser et de disséminer des types variés de corpus de traductions pour les annoter de façon collaborative. Un tel système doit permettre de mutualiser les travaux humains pour enrichir une ressource par des métadonnées, des données multimodales, et des annotations allant du balisage morphosyntaxique à des représentations sémantico-pragmatiques, en passant par des analyses multiniveau, des catégories sémantiques, des mots universels (UW) et des graphes UNL, des représentations dialogiques, etc., et avec les correspondances entre les différents "canaux" (texte, voix, geste, vidéo…) pour les corpus multimodaux.

Nous avons cherché de tels systèmes, et n'en avons pas trouvé. Il y a cependant des outils qui contiennent des points intéressants pour notre étude : *OLLIE* et *Distrib-Dial & Annot-Dial*.

II.1.1.3.1 OLLIE

OLLIE [Cunningham et al., 2003] est une application client-serveur fournissant le support à l'annotation collaborative de corpus à travers un navigateur Web.

OLLIE est utilisé principalement pour l'annotation de documents et de corpus linguistiques. Bien que *OLLIE* ne semble pas avoir été utilisé (ni être utilisable) pour supporter des corpus de traductions, il contient plusieurs points intéressants pour nos recherches. Ce qui est intéressant est surtout son architecture logicielle qui permet à plusieurs utilisateurs de travailler sur un document en même temps de façon collaborative, et la façon dont il traite certains des problèmes qui nous intéressent, tels que le support à l'authentification des utilisateurs et des profils, la gestion de la synchronisation, etc.

II.1.1.3.2 Outils Distrib-Dial & Annot-Dial

Distrib-Dial & *Annot-Dial* [Fafiotte, 2004] ont été conçus par G. Fafiotte dans le cadre du projet ERIM pour la dissémination et l'annotation de corpus ERIM sur le Web.

La première version de ces outils a été développée en JSP (Java Server Page) par l'auteur dans le cadre de son M2R à Danang puis à Grenoble en 2006. Récemment, ce système a été intégré dans SECTra_w comme un composant avec des fonctions permettant de rejouer des dialogues, de recalculer différentes mesures, d'annoter et de transcrire des corpus, et de traduire des transcriptions.

Conclusion

OLLIE fournit une architecture logicielle permettant de mutualiser l'annotation de documents et de corpus linguistique. Il fournit aussi les supports à l'authentification des utilisateurs et des profils, et la gestion de la synchronisation. Cependant, *OLLIE* ne résout pas la plupart des problèmes qui nous intéressent, en particulier ceux liés à l'annotation collaborative de corpus de traductions, tels que la transcription de corpus de parole, la traduction des transcriptions, la conversion en graphes UNL, etc.

Distrib-Dial & *Annot-Dial* sont utilisés pour la transcription de corpus de parole, et pour la traduction des transcriptions. Cependant, ces outils sont spécialisés au corpus ERIM et manquent donc de généricité.

II.1.1.4 Intégration de l'accès (éventuellement contributif) à des dictionnaires

Un système de support à la post-édition de traductions automatiques doit fournir l'accès à des dictionnaires de façon proactive et contributive. Les post-éditeurs doivent pouvoir non seulement utiliser des dictionnaires, mais aussi contribuer aux dictionnaires. Nous étudions donc ce support dans les systèmes utilisés pour la traduction humaine et permettant aussi la post-édition de traductions automatiques, déjà présentés au §II.1.1.1.

II.1.1.4.1 Google Translator Toolkit

Google Translator Toolkit donne aussi aux traducteurs accès à un dictionnaire qui est disposé dans une zone fixée en bas de l'interface de traduction (voir Figure 32). Il y a une zone éditable permettant de taper un mot ou une expression à chercher.

Cependant, *Google Translator Toolkit* ne permet pas aux traducteurs de contribuer en ajoutant des mots qui n'existent pas dans le dictionnaire afin de les réutiliser plus tard ou de les partager avec d'autres traducteurs.

Il ne permet pas non plus d'importer des ressources lexicales pour construire des dictionnaires pour un projet.

Les informations lexicales montrées aux traducteurs sont pauvres et peu pertinentes, parce qu'elles ne sont ni traitées selon le domaine, ni fusionnées à partir de ressources différentes.

II.1.1.4.2 BEYTrans

BEYTrans donne accès à des dictionnaires construits pour et par les traducteurs humains. Chaque communauté de traducteurs a son propre dictionnaire. Pour consulter un dictionnaire, on doit indiquer la langue source, la langue cible, la communauté, et l'expression à chercher. Les traducteurs peuvent également ajouter une nouvelle entrée si un mot n'est pas dans le dictionnaire.

BEYTrans permet aussi aux traducteurs d'importer des ressources pour construire des dictionnaires adaptés à leurs besoins.

Pour la présentation des informations lexicales, *BEYTrans* utilise deux zones de l'interface de traduction pour visualiser les informations lexicales. La première zone contient les informations lexicales trouvées à partir d'un dictionnaire en ligne, et la deuxième est réservée au dictionnaire construit pour ou/et par la communauté (Figure 33).

Cependant, il reste encore quelques limites dans BEYTrans.

- Au niveau technique, la fonction permettant à l'accès aux dictionnaires ne marche pas bien. Les deux zones utilisées pour visualiser des dictionnaires sont donc souvent vides et les traducteurs ne peuvent ni les cacher ni les diminuer pour augmenter l'espace de traduction.
- Les utilisateurs doivent attendre très longtemps pour voir les informations lexicales disponibles, parce que *BEYTrans* ne les prépare pas à l'avance. Contrairement à la spécification annoncée dans la thèse [Bey, 2009], il n'y a donc pas la proactivité qui nous semble nécessaire. A chaque fois, *BEYTrans* doit lancer la recherche des informations sur le serveur ou sur Internet.

II.1.1.4.3 Caitra

Comme dit ci-dessus (§II.1.1.1.3), *Caitra* permet aux traducteurs de choisir des informations lexicales pour construire des post-éditions. Les informations lexicales dans *Caitra* sont visualisées dans une zone en bas de l'interface de traduction, en termes de graphes lexicaux (Figure 34).

Les informations lexicales que *Caitra* montre aux traducteurs sont simplement des traductions mot à mot produites par Moses. Les informations ne sont donc ni traitées ni normalisées, et n'offrent aucune garantie de fiabilité, en particulier au niveau terminologique.

Un point intéressant est que *Caitra* prépare à l'avance des informations lexicales pour un segment. Dès qu'un segment est sélectionné, les informations lexicales correspondantes sont montrées aux traducteurs.

Conclusion

Nous avons présenté le support des informations lexicales dans les systèmes permettant la post-édition de traductions automatiques. Les informations lexicales utilisées dans ces systèmes sont encore brutes ou simplement des traductions mot à mot d'un système de TA (Caitra). Elles ne sont pas filtrées selon le domaine, ni fusionnées à partir de ressources lexicales différentes. Sauf *Caitra,* ces systèmes ne fournissent pas les informations lexicales pour chaque segment de façon proactive, et permettent pas non plus aux post-éditeurs de consulter efficacement les informations lexicales.

II.1.2 Synthèse des objectifs et des problèmes

Notre objectif essentiel dans ce chapitre est de trouver comment construire un sectra qui supporte non seulement l'évaluation de résultats de TA, mais aussi le travail humain collaboratif sur des corpus variés en contexte multilingue, et principalement la post-édition collaborative. Le premier point qui nous a frappé est qu'il n'existe aucune définition générique précise, et utilisable pour construire des sectra, de notions qui semblent pourtant basiques, comme ce qu'est un corpus de traductions, un segment multilingualisé et contextualisé, un flux de travaux contributifs, etc. Nous précisons les aspects qui nous semblent les plus importats, puis précisons les notions "émergentes", et dégageons les problèmes les plus importants à résoudre pour relever ce second défi.

II.1.2.1 Aspects importants pour relever ce défi

Aspect générique de la définition des corpus. Un sectra doit permettre de travailler facilement avec les différents types de corpus utilisés dans les projets de post-édition, les campagnes d'évaluation, et les projets d'annotation. Ces corpus ont éventuellement des structures et des formats très variés.

Pour cela, ce système doit permettre de définir et de traiter ces corpus d'une façon générique. Cela permettrait aux organisateurs d'importer facilement leurs corpus, et de gérer leurs projets sans l'intervention d'un spécialiste informaticien.

Il semple aussi nécessaire de permettre aux organisateurs eux-mêmes d'optimiser les interfaces selon la structure de leur corpus. Par exemple, l'interface pour la post-édition de corpus parallèles comme ceux d'EuroParl (un segment source aligné avec une traduction (schéma 1-1)) doit être facilement adaptée aux corpus comme le BTEC (un segment source aligné avec plusieurs traductions (schéma 1-N)).

Enfin, ce système doit permettre aux utilisateurs d'exporter leurs corpus selon des formats définis par eux-mêmes.

Nous proposons des solutions à ce problème au §II.2.1.1.

Définition étendue d'un « contexte » de segment. La notion de contexte concerne non simplement le domaine de la traduction (santé, informatique, etc.), et la position d'un segment source dans un ou plusieurs textes source, mais la notion de contexte va bien plus loin que cela. Nous devrons donc étudier et donner une définition étendue d'un contexte de segment. Nous abordons ce problème au §II.2.1.2.

Support des informations lexicales. Un sectra devra fournir un support des informations lexicales aux contributeurs. Ces informations doivent être bien traitées, et normalisées. Elles doivent aussi être fournies de façon *proactive*.

Nous proposerons des solutions à ce problème au §II.2.2.1.

Appel de ressources extérieures. Un sectra doit utiliser et fournir à des contributeurs plusieurs types de ressources telles que des traductions automatiques, des informations lexicales, des graphes UNL, etc. Cependant, ce n'est ni un système de traduction, ni un système de gestion d'informations lexicales, ni un système de segmentation et de normalisation, etc. Il doit donc appeler des services externes pour réaliser des tâches et récupérer des ressources. Le problème est de déterminer comment appeler ces ressources extérieures.

Nous proposerons des solutions à ce problème au §II.2.2.2

Aspect contributif et ouvert. Pour la mutualisation du travail humain sur les corpus de traduction, il faut un service Web de type Wiki, mais la granularité ne conviendra pas (nos

grains ne seront pas des pages, mais des informations associées à des segments, comme 1..N TA, 1..P post-éditions).

De plus, un système d'exploitation de corpus doit permettre d'effectuer toutes les actions usuelles sur les corpus. Cependant, il est impossible de les implémenter toutes bien, parce qu'il y a trop de fonctionnalités différentes, chacune présentant de nombreuses difficultés et contraintes. On peut donc envisager d'en programmer certaines (évaluation, appel à de la TA, post-édition, etc.), mais un sectra doit permettre d'intégrer et de déléguer les autres.

Nous présentons ce problème au §II.2.3.1.

Sécurité des données, prévention du piratage. Le système à construire étant collaboratif et ouvert, il se pose plusieurs problèmes relatifs à la sécurité des données, à savoir la pollution et le vol de données.

Nous étudierons ce problème et proposerons plusieurs solutions au §II.2.3.12.

II.1.2.2 Importance des aspects conceptuels, informatiques et de génie logiciel

Les problèmes présentés ci-dessus peuvent être classés selon l'importance relative de leurs aspects conceptuels, algorithmiques, et programmatoires dans le tableau suivant :

Problèmes	Conceptuels	Informatique	Génie Logiciel
Aspect générique de la définition des corpus.	•••	•	•
Définition étendue d'un « contexte » de segment	•••	-	-
Support des informations lexicales	•	•••	••
Appel de ressources extérieures	•	••	•••
Aspect contributif et ouvert	••	••	••••
Sécurité des données, prévention du piratage	•	•	••••

Table 19: Problèmes liés à la post-édition collaborative classés selon trois aspects

II.1.3 Notions unificatrices émergentes et principes généraux

II.1.3.1 Notions

II.1.3.1.1 Mot typographique

Définition II-1. Un *mot typographique* est une suite finie de signes appartenant à un ensemble fini connu, et délimitée par des séparateurs appartenant eux-mêmes à un ensemble fini connu.

Par exemple, en français, les délimiteurs sont le blanc et les signes de ponctuation (point, virgule, point-virgule, deux points, point d'interrogation, point d'exclamation, tiret – mais pas l'apostrophe ni le trait d'union).

Cependant, cette notion ne s'applique qu'aux langues dont le système d'écriture contient des séparateurs permettant de délimiter des mots, comme les langues européennes (français, anglais, italien, etc.), ou des transcriptions de systèmes d'écriture sans séparateurs de mots comme le Pinyin (pour le Mandarin) ou la transcription Hepburn (pour le japonais).

Le mot est utilisé comme unité pour payer les traducteurs humains dans la plupart de projets de traduction. Cependant, on ne peut pas l'appliquer au chinois, au japonais, au thaï, au laotien, et au khmer. On prend alors comme unité une page standard (voir ***Définition II-4***).

Par exemple, dans le projet EOLSS/UnescoL (voir §II.3.3.1), les postéditeurs ont été rétribués selon la règle suivante :

- ✓ la 1ère post-édition est comptée comme de la traduction, et payée 3 euros pour 100 mots source.
- ✓ les post-éditions suivantes (par une autre personne qui le post-éditeur initial) sont comptées comme de la révision, 3 euros pour 100 mots différents entre la version précédente et la nouvelle.

II.1.3.1.2 Multisegment

Nous avons déjà présenté la notion de *segment* (voir ***Définition I-2***) et la notion de *segment multilingue* (voir ***Définition I-4***).

Il arrive souvent qu'un segment soit représenté en plusieurs langues et qu'on ne sache pas quelle langue est la langue source. Par exemple, on trouve de tels segments dans les livres de phrases.

Définition II-2. Un *multisegment* est un segment qui se compose de plusieurs segments dont la langue n'est pas indiquée comme source ou cible.

Par exemple, un *multisegment* est représenté par le format XML suivant :

```
<multisegment>
   <seg xml:lang="en">How are you?</seg>
   <seg xml:lang="fr">Comment vas tu?</seg>
   <seg xml:lang="vi">Chú có khỏe không?</seg>
</multisegment>
```

II.1.3.1.3 Segment multilingualisé

En évaluation, un segment est toujours constitué par un seul segment source, accompagné de N_{ta} traductions automatiques, et de N_{rf} traductions de référence. On peut appeler un tel segment *segment multilingualisé*.

Définition II-3. Un *segment multilingualisé* est un *segment multilingue* dans lequel une seule langue est considérée comme langue source.

Ainsi, un *multisegment* sera représenté dans une mémoire de traductions par autant de *segments multilingualisés* qu'il a de langues source.

II.1.3.1.4 Page

a. Page standard (page du traducteur)

Définition II-4. Une *page standard* contient 250 *mots typographiques* ou environ 1400 caractères (1 page A4, double interligne, Times 12) pour les alphabets et certains syllabaires (comme les langues européennes), et 400 caractères pour les idéogrammes (comme le japonais, le chinois, et le coréen).

On utilise souvent la notion de *page standard* en traduction professionnelle comme unité, et pour répartir le travail entre les traducteurs.

b. Page logique

Définition II-5. Une *page logique* est une unité de traitement informatique contenant un nombre spécifique de caractères, ou de mots, ou de segments, prédéfinie par les utilisateurs (administrateurs, post-éditeurs, évaluateurs, réviseurs, etc.) ou par les développeurs.

Dans les systèmes de traitement de corpus, on utilise souvent la notion de *page logique* pour diviser une masse de données en unités plus petites permettant d'effectuer quelques

opérations plus efficacement, telles que la visualisation, le chargement, etc. Dans quelques projets, on utilise aussi la notion de *page logique* pour faciliter l'affectation des tâches aux contributeurs faissant de la post-édition, ou de l'évaluation.

II.1.3.1.5 Type de corpus de documents

a. Corpus de documents TMX

Définition II-6. Un corpus est dit *corpus de documents TMX* si c'est un *corpus de documents* dont le texte est en format TMX.

TMX[28] (Translation Memory eXchange) est un standard XML ouvert pour l'échange de mémoires de traductions entre les outils et les vendeurs de traduction.

b. Corpus de documents « multifichier »

Définition II-7. Un corpus est dit *corpus de documents « multifichier »* si c'est un *corpus de documents* dans lequel un document est constitué de plusieurs fichiers.

c. Livre de phrases

Définition II-8. Un métasegment est un segment contenant des variables lexicales.

Par exemple.

Donnez-nous $nombre $contenant de $liquide

Avec $nombre = [1|2|3|..], $contenant = [tasse|verre] (s) et $liquide = [café|thé|lait].

Définition II-9. Un *livre de phrases* est un document dont les segments sont des métasegments multilingualisés et contextualisés.

Nous détaillons plus loin la notion de contexte.

II.1.3.2 Principes généraux relativement classiques

Hormis les principes généraux que nous avons présentés au §I.1.3.2, nous avons besoin des principes suivants pour construire un sectra supportant la post-édition collaborative en ligne.

II.1.3.2.1 Proactivité

Les données à accéder doivent être prêtes quand les contributeurs en ont besoin, ils ne doivent jamais avoir à demander au système de les rechercher.

Définition II-10. Le principe de *proactivité* consiste à préparer les données à l'avance afin de permettre aux contributeurs d'accéder à ces données sans attente.

II.1.3.2.2 Utilisation de boucles infinies pour les tâches en arrière-plan

La proactivité pour une masse de données peut être obtenue grâce à un travail de préparation qui peut s'étaler sur plusieurs jours, voire quelques mois. Il faut aussi mettre à jour les données régulièrement pour obtenir la synchronisation nécessaire entre le sectra et les systèmes délégués. Nous proposons d'utiliser des *boucles infinies* pour pouvoir effectuer ces tâches, comme M. Lafourcade et D. Schwab l'ont fait pour la création de grands ensembles de vecteurs conceptuels [Schwab, 2006 ; Lafourcade, 1998].

[28] http://www.lisa.org/Translation-Memory-e.34.0.html

II.1.3.2.3 Délégation

D'habitude, on sépare les fonctions réalisées à l'intérieur d'un système et celles réalisées par appel à des services extérieurs. Dans notre contexte, la plupart des fonctionnalités doivent pouvoir être réalisées des deux façons.

Par exemple, il faut que le système permette de transcrire un corpus oral, en le rejouant, segment par segment, mais il faut aussi qu'on puisse réaliser cette transcription à l'extérieur du système, à l'aide d'outils sophistiqués, intégrant entre autres un reconnaisseur de parole, que nous ne pouvons ni ne voulons intégrer dans un sectra.

Définitions II-11. Une fonction *délégable* d'un système S1 est une fonction réalisable par appel à un service d'un autre système S2, qu'elle soit ou non réalisée ou réalisable par un service interne au système S1.

Le *principe de délégation* consiste à faire en sorte que l'appel à chaque fonction délégable de S1 soit exactement le même, qu'on fasse appel à un service d'un autre système S2, ou à un service intégré à S1.

II.2 Problèmes liés à ce défi

II.2.1 Problèmes à dominante conceptuelle

II.2.1.1 Problème 2.1 : Aspect générique de la définition des corpus

Les corpus de traductions sont hétérogènes en théorie ainsi qu'en pratique, comme on l'a vu au chapitre 1, §I.1.1.1. Le tableau suivant en donne quelques exemples.

Corpus	Vue logique	Organisation physique	Organisation interne dans un corpus
Ariane-G5 brut	Ensemble de textes	Fichiers « à plat » avec convention de nommage.	Liste des unités de traduction (titres, phrases, paragraphes ou texte complet)
Ariane-G5 intermédiaire	Ensemble de phrases (d'Ariane)	Fichiers « à plat » avec convention de nommage liée à la chaîne d'exécution	Une unité de traduction avec un « arbre décoré »
TRANSAT	Ensemble de multisegments	Fichiers de texte par langue.	L'entrée contient des segments ou supersegments avec des séparateurs de segments et des scores.
EuroParl, Babel	Ensemble de multisegments	Fichiers de texte par langue.	Listes de paires d'énoncés <SSeg_L1, TSeg_L2>.
Moses	Suite de segments monolingues	Fichiers texte aligné monolingue	Chaque segment est sur une ligne.
MT	Ensemble de textes particuliers	Base de données	<anglais, français_i> : liste des occurrences pour cette traduction
ERIM	Ensemble de dialogues contenant des tours de parole avec leur descripteurs	Chaque dialogue correspond à un répertoire, contenant des fichiers son (.wav), des fichiers transcrits (.xml), et des fichiers de texte (.txt)	Un fichier son (.wav) attaché à un fichier de l'annotation selon une convention de nommage
EOLSS	Ensemble de documents	Chaque document correspond à un dossier, contenant 2 fichiers, .html et .unl.	Le fichier .html est le « fichier principal », le fichier .unl est le « fichier satellite », et est utilisé pour guider la segmentation du fichier .html.
Survitra	Livres de phrases, organisées selon le domaine.	Fichiers .xml ou fichier .xsl	Les phrases sont parfois des métasegments (voir ***Définition II-8***)

Table 20: Exemples de corpus, et de leurs organisations logiques, physiques, et internes

II.2.1.1.1 Objectif visé

Nous cherchons à définir un métamodèle unique permettant de décrire aussi bien des corpus de multisegments, de phrases, de textes (comme ceux des TRANSAT, BTEC, EuroParl, Babel, etc.), qu'un corpus de dialogues interprétés multilingues multimodaux comme ceux d'ERIM, qu'un ensemble structuré d'articles scientifiques comme EOLSS, ou qu'un livre de phrases multilingues comme Survitra, et donc des corpus de métasegments.

Si l'on arrive à définir un tel formalisme, on pourra unifier le traitement et la gestion de tous les types de corpus qu'on pourra décrire. On pourra aussi générer automatiquement des formats d'import/export et d'IHM à partir de la définition d'un corpus. Comme on l'a déjà fait pour les bases lexicales [Sérasset, 1994 ; Mangeot, 2001], on pourra aussi construire les représentations internes (tables en Postgres ou Mysql) et les interfaces de présentation et d'édition d'un corpus automatiquement, à partir de sa définition.

II.2.1.1.2 Etat de l'art

Il y a d'abord de nombreux travaux visant à trouver des standards permettant de représenter des corpus ou des données de traductions.

LISA (The Localisation Industry Standards Association) a défini le format *TMX* (Translation Memory eXchange) [LISA, 2009] dans le but de fournir un standard pour échanger les données de mémoires de traductions entre les outils d'aide aux traducteurs.

OASIS[29] a défini le format *XLIFF* (XML Localization Interchange File Format) [OASIS, 2009] comme standard pour l'échange de documents entre les outils de localisation, tels que les systèmes de MT et les systèmes de TA.

Le projet *Text Encoding Initiative* a défini le format *TEI* [TEI, 2009] comme format standard pour la représentation des données textuelles. Il a été utilisé pour représenter de nombreux corpus, tels que le *British National Corpus*[30], ou le corpus écrit développé par *Sinequa* [Sinequa, 2006], etc.

Au GETALP, on a aussi défini le format *CPXM* (Common Parallel eXample Markup), pour la représentation des corpus parallèles, qui a été utilisé dans le projet *TraCorpEx* [Hajlaoui et Boitet, 2003]. B. Bigi et V-B. Le ont aussi défini un format XML pour la normalisation de corpus français et vietnamien [Bigi et Le, 2008].

Ces formats visent à l'universalité, mais ils ne peuvent pas représenter tous les corpus que nous voulons décrire, par exemple les corpus ci-dessus.

II.2.1.1.3 Recherche d'une solution adaptée

Une première idée est de partir de descripteurs qui ont déjà été proposés pour certains types de corpus (corpus d'évaluation, ERIM, Survitra), pour voir ce qui semble au moins nécessaire.

Il s'agit de descripteurs XML qui correspondent à une vue logique des "documents" en question.

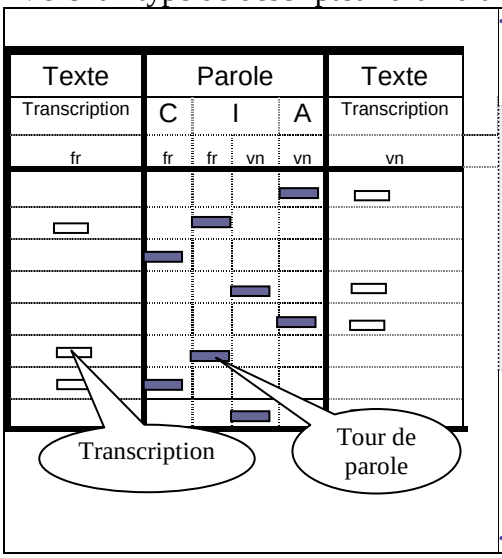
Voici d'abord un type de descripteur très simple pour un document du corpus d'évaluation de TA français-anglais. Pour chaque langue et chaque système de TA à évaluer, on a une en-tête donnant le nom du corpus, les langues (source, traduite), le nom du système à évaluer, puis une liste de segments d'évaluation. Chaque segment d'évaluation est constitué par une phrase source, une traduction candidate, une ou plusieurs traductions de référence, et des résultats

[29] http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html

[30] http://www.natcorp.ox.ac.uk/

d'évaluation subjective et objective (si possible). La description du corpus est donc la combinaison de descriptions partielles.

Voici un type de descripteur d'un dialogue du corpus ERIM français-vietnamien.




```
<Corpus nom="ERIM" structDc="erim.dtd">
  <dialog num="1" langs="Vi-Fr">
    <LOC nom="A">

    <TP num="1">
       <InfosTP lang="Vi" durée="13"
        Hdébut="10:20:00" Hfin="10:20:33">
       <InfosTP>
       <son>"A/TP1.wav"</son>
       <minitexte> "" </minitexte>
       <transc> Xin chào </transc>
       <trad lang= "fr"> Bonjour <trad>
     </TP>

    </LOC>
    <LOC nom="I"/>
     <TP num="2">
     .....
  </dialog>
</Corpus>
```

Figure 37: Exemple de structure et de description d'un dialogue du corpus ERIM

Un tel descripteur est en fait construit à partir de "sous-descripteurs" correspondant aux tours de parole de chaque intervenant, construits sur la machine utilisée, puis recopiés sur le serveur de collecte, ainsi que tous les fichiers son. On voit que le corpus ERIM (corpus multimodal) contient des dialogues (on considère un dialogue comme un document) dans lesquels chacun contient une liste de tours de parole. Une description de tour de parole contient un chemin d'un fichier son, une transcription, et une traduction de la transcription.

Voici maintenant un autre exemple de description de quelques "métaphrases" d'un livre de phrases du corpus Survitra (voir annexe 3). Le corpus Survitra (corpus multitraduits, postédités, et multiannotés) contient des entrées (on considère une entrée comme un document), chacune contenant une liste de segments. Chaque segment est décrit par une phrase source originale, une phrase source corrigée, N traductions automatiques en M langues, K post-éditions, un graphe UNL, etc.

Comme dans les exemples ci-dessus, nous voyons que les éléments principaux d'un formalisme générique de ces descriptions sont les descriptions de corpus, les descriptions des documents, et les descriptions des segments.

Il faut sans doute distinguer :

- la description d'un texte "usuel", considéré comme une représentation primaire (une suite de caractères) sur laquelle se greffent une ou plusieurs structures hiérarchiques (chapitres, sections…), dont les feuilles sont des segments ;
- la description d'un document, comportant un "texte principal" (fichier maître), une ou plusieurs décompositions hiérarchiques (les segments peuvent être différents de l'une à l'autre), des fichiers satellites (images, sons, vidéos, icônes…) et éventuellement un ou plusieurs fichiers compagnons ;
- la description d'une mémoire de traductions, considérée comme un ensemble de textes particuliers, dont les segments sont des segments multilingualisés et contextualisés représentant toutes les occurrences du contenu de leur partie source dans les textes ou documents "usuels" du corpus ;

- la description d'un corpus de textes ou de documents, considéré le plus souvent comme un ensemble non structuré ou peu structuré (date, domaine) d'éléments "réalisés" ou "réels", accompagné, pour nous, d'une mémoire de traductions, et aussi par la définition de la ou des structurations associées (comme c'est le cas dans Ariane-G5), et bien évidemment des métadonnées utiles.

La description d'un corpus de métadocuments, dont les documents contiennent des métasegments, et éventuellement certaines de leurs instanciations (segments "réels").

On voit aussi qu'un document usuel peut avoir des fichiers satellites et un ou plusieurs documents compagnons (par exemple un fichier .unl et un document XML le transformant en DAE — document auto-explicatif), mais les textes de la MT n'en ont pas. Par contre, un segment multilingualisé et contextualisé (mc-segment) pourra avoir un fichier satellite contenant les icônes etc. de tous ses contextes.

D'un autre côté, il faut décrire la structure des segments ou des "entrées" (éléments au-dessus) comme par exemple pour Survitra, avec l'idée qu'on peut avoir des "minidocuments" pour définir par exemple des classes de mots. Il faut ici donner les "composants" et leurs types (par exemple, contenu sonore, annotation de tel ou tel type comme graphe UNL, ou dessin…)

Il s'agit de valeurs d'attributs, à mettre dans les métadonnées. On peut décrire leurs valeurs pour un corpus: nom, attributs-clés (ex: année, journal, domaine, langues) et autres attributs (édition, auteurs principaux, référence, documents afférents…), type(s) de documents de ce corpus, éventuellement relations entre documents (sujets et corrigés, cours et livre d'exercices, livre et commentaires, livre et illustrations — DSR), nom et type de MT (ou des MT).

Nous proposons de considérer par principe qu'une MT est un *pseudocorpus* particulier, qui est formé d' autant de *pseudodocuments* que de langues source.

Les pseudodocuments en question ne sont pas connus à l'avance, mais sont construits au cours de la vie du système, et sont constitués non pas d'un document maître connu et d'un ou plusieurs découpages en segments (textuels en langue source, et formels pour la présentation), mais

- d'un document maître virtuel, construit à partir des segments-mc, présentés dans un ordre choisi au niveau du corpus (grâce aux attributs-clé),
- et de segments formels (le "sucre de présentation").

Nous permettrons d'avoir une même MT pour N corpus, et m MT pour un corpus. Sans doute faudra-t-il que les types des N corpus soient "compatibles". On s'inspire aussi de l'exemple d'IBM [Chenon, 2005] : on "construit" une MT pour un ou plusieurs nouveaux textes à traduire par union de MT calculées à partir de documents déjà traduits.

Nous devons donc définir ce que peut-être l'unification de 2 types de MT. C'est assez simple: union des (types de) "grains" ou "cellules". Un grain est décrit par un source (texte ou liste ou graphe ou son), une prétraduction simple, une prétraduction multiple (liste ou graphe), un ensemble de contextes, une postédition (texte ou graphe, si l'on veut permettre des choix) ou un ensemble de post-éditions avec les contextes associés, une ou des annotations, une mesure, une évaluation, une liste de références à des objets contenus dans le fichier satellite.)

On peut décrire les valeurs d'attributs d'un type de document: nom, attributs (auteur, langue(s), date…), nom et type de fichier maître.

On pourra s'inspirer de la solution de Jibiki et proposer un langage de description de macrostructures et de microstructures corporales. Si l'on considère qu'un corpus est similaire à un dictionnaire, un document est similaire à un volume de dictionnaire, et chaque segment

correspond à une entrée dans un volume. On peut définir la liaison d'un segment source vers plusieurs segments cible (traduits par un humain ou par un système de MT) via un volume de segments pivot. La structure interne d'un segment est sa "microstructure". Cette microstructure peut contenir autant d'attributs qu'on veut pour chaque segment [Nguyen, 2009].

II.2.1.1.4 Formalisation en XML s'inspirant de celle de G. Sérasset et M. Mangeot pour les bases lexicales

Au plus haut niveau, on propose un formalisme hiérarchique pour les corpus. Ce formalisme contient deux éléments principaux : une en-tête et un corps. L'en-tête contient des informations relatives au corpus, à des langues, à des noms de système de TA, à la date de création, etc. Le corps contient des informations sur le type de document (<doc>, <dialogue>, <entrée>, etc). Chaque type de document contient des descriptions de structures hiérarchiques (chapitres, pages, sections, <lgsource>, <LOC>…), et un descripteur de segment (<seg>, <TP>, , etc.). Le descripteur de segment contient des descriptions des occurrences (source, prétraductions, contextes, post-éditions, son, scores, graphe UNL, etc.).

Voici une DTD générique pour l'import et l'export dans le système SECTra_w. On appellera ce format *CCM* (Common Corpus Markup).

```
<!ELEMENT corpus(header, body) >                ;corpus
 <!ELEMENT header(name, date, domain, authors,  ;en-tête
          project, Nlang, lang*, othermeta*)>
    <!ELEMENT name (#PCDATA)>                   ; nom de corpus
    <!ELEMENT date (#PCDATA)>                   ; date de creation
    <!ELEMENT domain (#PCDATA)>                 ; domaine
    <!ELEMENT authors (#PCDATA)>                ; auteur
    <!ELEMENT project (#PCDATA)>                ; projet utilisant corpus
    <!ELEMENT Nlang (#PCDATA)>                  ; nombre de langues
    <!ATTLIST lang type CDATA>                  ; source ou cible
    <!ELEMENT lang (#PCDATA)>                   ; code de langue (eng, fra, etc.)
    <!ATTLIST othermeta type CDATA>             ; autre type de métadonnées
    <!ELEMENT othermeta (#PCDATA)>              ; nom de métadonnées
 <!ELEMENT body (doc*) #REQUIRED>               ; un corpus = n documents
    <!ATTLIST doc  type  CDATA>                 ; type de document (dialogue,
                                                classe de mots, etc.)
    <!ATTLIST doc  id  CDATA>                   ; id de document
    <!ATTLIST doc  Nsegmments  CDATA>           ; nombre de segments
    <!ATTLIST doc  name  CDATA>                 ; nom de document
    <!ELEMENT doc (article*)>                   ;1 doc = n articles
    <!ATTLIST article type CDATA>               ; structures hiérarchiques
                                                (chapitre, section, page,..)
      <!ELEMENT article (segment*)>             ; un article = n segments
        <!ATTLIST segment  id  CDATA>           ; id de segment
        <!ATTLIST segment  type  CDATA>         ;type de segment(TP, seg,etc.)
        <!ELEMENT segment (occurrence*)>        ; un segment = n occurences
         <!ATTLIST occurrence  type CDATA>      ; type d'occurrence (source,
                                                traduction automatique,post-
                                                édition, graphe UNL, son,
                                                etc.)traduction automatique,post-
                                                édition, graphe UNL, son, etc.)
         <!ATTLIST occurrence  lang CDATA>      ; langue
         <!ATTLIST occurrence version CDATA>    ; version
         <!ATTLIST occurrence  producer  CDATA> ; traducteur ou système de TA
         <!ATTLIST occurrence  level  CDATA>    ; niveau traductionel
         <!ATTLIST occurrence  rating CDATA>    ; note de qualité
         <!ATTLIST occurrence  date CDATA>      ; date de création
         <!ELEMENT occurrence (#PCDATA)>        ; contenu d'occurence
```

Figure 38: DTD du format commun d'import et d'export dans SECTra_w

Les exemples de format CCM pour les corpus ERIM, EOLSS, Survitra, TRANSAT etc. sont montrés dans l'Annexe 2.

II.2.1.2 Problème 2.2 : Définition étendue d'un « contexte » de segment

II.2.1.2.1 Analyse du problème et état de l'art

Un segment source peut apparaître à plusieurs endroits, et avoir des traductions (correctes) différentes pour certaines de ses occurrences.

Par exemple, "hai" en japonais peut se traduire en anglais par "yes" ou "no" selon qu'il s'agit d'une réponse à une interrogation positive ou négative.

Ou encore (exemple dû à J-M. Zemb, rapporté par Ch. Boitet) :

"Pierre bouscula Jean. Il lui dit « je ne t'ai pas bousculé exprès »" (Ich habe Dich nicht absichtlich gestoßen) (I did not hurt you on purpose).

"Pierre fit semblant de bousculer Jean. Il lui dit « je ne t'ai pas bousculé exprès »" (Ich habe Dich absichtlich nicht gestoßen) (I did not hurt you, on purpose).

De façon générale, pour produire une traduction de bonne qualité d'un segment, un traducteur (humain ou machine) doit tenir compte de son contexte. Qu'est-ce donc que le contexte d'un segment ?

D'abord, il semble que la notion de contexte concerne essentiellement l'occurrence du segment considéré dans un ou plusieurs textes source, et pas l'occurrence de telle ou telle traduction dudit segment dans tel ou tel texte cible.

Cependant, on pourrait argumenter que le choix d'une traduction précise dépend du contexte de ce choix dans la langue cible. Par exemple, en traduction d'une poésie, il se peut qu'on choisisse un équivalent plutôt qu'un autre en fonction de son nombre de syllabes, ou de sa dernière syllabe, pour obtenir le nombre de pieds ou la rime désirée.

Si nous suivions cette voie, nous devrions tenir compte de contraintes globales sur chaque texte traduit, ce qui est très difficile voire impossible car il est le plus souvent produit sans vue globale (par petites "unités de traduction"), et car, dans notre contexte, il évolue en fonction des corrections des contributeurs. Nous nous limiterons donc à une notion de contexte ne faisant intervenir que le texte en langue source, et des facteurs liés à la situation de traduction.

Il y a aussi un point très important : il faut pouvoir associer à une occurrence d'un segment une liste ordonnée de traductions possibles. Comme chaque traduction est (par nature) associée à un ensemble de contextes (ceux dans lesquels elle a été choisie pour traduire ce segment), il faut pouvoir comparer des ensembles de contextes. Ainsi, s'il y a dans le contexte "structurel" (même paragraphe, même section…) d'autres segments déjà traduits dans lesquels il y a un mot ou terme X pouvant avoir plusieurs traductions (t1, t2, t3) par exemple, et si X apparaît dans un segment à postéditer, on pourra calculer un ordre de préférence pour ces traductions, et cet ordre dépendra des choix faits dans ces autres segments.

Le calcul de cet ordre de préférence dépend en général du type de document. Par exemple, dans un document technique, on cherche à utiliser le plus possible le même équivalent pour le même mot (dans la même acception), alors que, dans une œuvre littéraire, on cherche plutôt à ne pas trop se répéter, et donc à utiliser des mots différents pour le même sens.

Un point important qui apparaît ici est que, si l'on parle des contextes de telle ou telle traduction, on s'intéresse uniquement aux *bonnes* traductions, et donc pas aux prétraductions automatiques non validées[31].

On distingue les types de contexte suivants:

- *contexte linguistique* (par exemple: "centres" de reprise anaphorique ou d'élision = noms avant, ou après pour la cataphore, dépendant de la position dans le document),
- *contexte dialogique* (actes de parole précédents, par exemple, question interronégative),
- *contexte structurel* (position dans la hiérarchie courante, qui rapproche ou éloigne les occurrences du même segment/mot du point de vue du choix de la traduction,
- *contexte par rapport à différents attributs* ou "traits" (auteur, domaine, genre du passage — manuel, référence, commercial, alerte, menu, aide…)

II.2.1.2.2 Propositions.

Les métadonnées d'un *segment-mc* (voir *Définition II-12*) devront donc comprendre tous ces attributs. Il sera donc possible en théorie de définir une fonction qui, à un ensemble de "vecteurs" de tous ces facteurs, associe un score à la traduction associée. En pratique, on pourra lui donner une forme standard (par exemple une combinaison linéaire normalisée), et chercher à apprendre ses coefficients automatiquement (machine learning) à partir de scores reflétant les préférences exprimées par les choix faits par les utilisateurs.

Définition II-12. Un *segment multilingualisé contextualisé* ou *segment-mc* est un objet formé d'un segment dans une langue source, de ses contextes d'apparition dans les documents où il est apparu, et des traductions proposées dans ces contextes, ainsi que des autres informations attachables à ce segment, par exemple un minidictionnaire construit à partir de la liste de ses lemmes, ou un ou plusieurs arbres d'analyse, ou un graphe UNL, etc.

Voici un exemple de *segment-mc* simple représenté en XML.

```
<segment id="eolss_enfr_0001_1709009.121122">
<occurrence type ="source" lang="en" docs = "D1|D2|D3" idseg="007|112|236"> How are
you? </occurrence>
<occurrence type ="cible" lang="fr" docs="D1" idseg="007"> Comment allez vous?
</occurrence>
<occurrence type ="cible" lang="fr" docs ="D2,D3" idseg="112|236"> Comment vas tu?
</occurrence>
</segment>
```

Dans cet exemple, la phrase source "*How are you?*" est apparue dans trois documents D1, D2, et D3. les identificateurs de cette phrase dans D1, D2, D3 sont respectivement 007, 112, et 236. Une traduction de cette phrase proposée pour le document D1 est "*Comment allez vous?*", et une autre pour les documents D2, D3 est "*Comment vas tu?*".

Nous proposons aussi la définition du contexte structurel d'un segment.

Définition II-13. Le *contexte structurel* d'un segment est la position du segment source dans la hiérarchie courante (paragraphe, section…).

II.2.1.2.3 Possibilités de représentation des contextes

Pour la représentation dans une BD "corporale", nous proposons d'utiliser un vecteur contenant des contextes pour chaque segment, ou bien, une structure factorisante pour un

[31] Il y a tout de même une proportion non négligeable de bonnes traductions dans les résultats de TA, par exemple pour Systran 25% pour le BTEC et 10% pour EOLSS, qui sont validées telles quelles, sans aucune modification.

ensemble de contextes. On pourrait par exemple décider de factoriser tous les attributs, souvent égaux, qui n'expriment pas la position exacte dans le texte.

On aurait alors par exemple pour Notepad++ "Mise en page"x{ens. de repères d'occurrences} et "Alertes"x{ens. de repères d'occurrences}.

En ce qui concerne le contexte structurel, nous proposons de faire comme dans TM/2[32], et de considérer qu'il y a un contexte par occurrence, représenté par la place de cette occurrence dans le document où elle est apparue.

II.2.2 Problèmes à dominante algorithmique

II.2.2.1 Problème 2.3 : Support des informations lexicales

a. *Deux sous-problèmes : la proactivité et la spécialisation*

Il est utile de permettre aux utilisateurs d'accéder à des ressources lexicales en ligne, par exemple en leur permettant de cocher des cases correspondant à ces ressources, et en créant automatiquement des onglets ouverts sur les ressources désirées. On peut aussi intégrer l'accès à l'interface d'une ressource particulière dans l'interface de postédition… mais on ne peut le faire que pour une!

Nous avons réalisé cela, mais ne considérons pas qu'il s'agit réellement d'un problème. Les deux "vrais" problèmes qui se posent sont liés à l'exigence de proactivité et de spécialisation (personnalisation) :

- **Proactivité.** Comment permettre à des contributeurs d'accéder pendant la post-édition à des ressources lexicales distantes de façon proactive, i.e. de façon qu'elles soient prêtes quand il en a besoin, et n'ait jamais à demander au système de les rechercher ?
- **Spécialisation**. Comment construire une ressource lexicale spécialisée à un projet de traduction d'un ou de plusieurs corpus, c'est-à-dire qui contienne les équivalents utilisés dans ce projet, et permette d'en "normaliser" certains (toujours par rapport à ce projet) ?

Pour la proactivité, la seule solution semble être de rechercher à l'avance, pour chaque segment, les informations lexicales potentiellement utiles, dans la liste des ressources indiquées, et de stocker le résultat dans une structure de données (minidictionnaire) associée à ce segment. Techniquement, cela pose plusieurs sous-problèmes :

- extraction à partir d'un segment des unités lexicales adéquates (mots, termes, etc.) à rechercher,
- construction d'une interface adéquate permettant la visualisation et la contribution.

En ce qui concerne la spécialisation, les sous-problèmes sont :

- la collecte et la fusion des informations lexicales concernant le ou les corpus à traiter à partir de différents ressources, pour construire une BDLM spécialisée au projet, dite BDLM "recyclée",
- la constitution d'une BDLM "du projet", qui peut se faire en combinant les apports directs des contributeurs humains (via une interface dictionnairique adaptée) et les apports indirects provenant de programmes d'extraction de terminologie bilingue à partir de couples <segment source, bonne traduction>.

[32] TM/2 n'est plus commercialisé depuis 1992, mais est toujours utilisé par IBM et ses contractants pour traduire 20M mots/an vers 40 langues (25 il y a quelques années).

- la gestion semi-automatique des poids associés aux différents équivalents (on peut utiliser les choix faits dans les postéditions comme des votes, et aussi permettre à des contributeurs de modifier directement ces poids).

b. Etat de l'art

Nous avons présenté au §II.1.1.4 certains systèmes Web qui sont utilisés pour la traduction humaine tels que *Google Translate Tookit*, *BEYTrans*, *Yakushite.net*, et qui permettent la gestion et le support des informations lexicales. Mais ces systèmes ne permettent pas d'utiliser les informations lexicales de façon proactive.

Une aide proactive à la consultation des informations lexicales est fournie par certains outils comme le système de support à la traduction humaine *Caitra* (Figure 34), et le système *Alexandria* (hypermedia multisource contextuel personnalisable) [Dutoit, 2010]. Dans ces systèmes, on construit à l'avance les informations lexicales à consulter à partir de ressources différentes. Ces informations lexicales sont donc toujours prêtes quand l'utilisateur en a besoin.

En ce qui concerne le problème de la construction d'une ressource adaptée à un projet, on peut mentionner le module *Robodico* [Nguyen, 2009] permettant la récupération des informations lexicales sur le Web en fouillant des dictionnaires en ligne comme *IATE*, *Wiktionary*, etc. pour la construction d'une ressource lexicale pour un projet. La technique de Robodico est d'envoyer une requête pour chaque mot afin de récupérer la page HTML contenant le résultat de la requête. À partir de la page HTML récupérée, on extrait les informations pertinentes et on les sauvegarde dans un format adéquat en local.

On peut aussi trouver des systèmes permettant l'élicitation de contributions lexicales des internautes, tels que *ITolDU* (Industrial Technical On Line Dictionary for University) [Bellynck et al., 2005], ou le travail en cours de *M. Daoud* (doctorant au GETALP-LIG) sur le système *SepT* (en cours de développement).

II.2.2.1.2 Solution retenue

Cependant, un sectra n'a pas à être un système de traitement des informations lexicales à proprement parler. Nous visons donc à déléguer la gestion et le traitement des informations lexicales à un système spécialisé à ces tâches.

Parmi les systèmes de gestion et de traitement des informations lexicales, nous choisissons le système *PIVAX* [Nguyen, 2009]. Avec *PIVAX*, on peut construire des BDLM spécialisées à des corpus ou à des sites Web et traiter des contributions lexicales des internautes. Pour chaque corpus ou site Web à manipuler, *PIVAX* récupère des informations lexicales à partir de ressources disponibles différentes (dictionnaires, sites Web, etc.) en découpant les segments en unités lexicales (mots, termes, expressions, etc.) et en cherchant les informations lexicales équivalentes de ces unités. Ensuite, il fusionne et sélectionne les informations lexicales trouvées pour construire les BDLM spécialisées aux corpus ou aux sites Web dans des « volumes » monolingues correspondants, en représentant les équivalences traductionnelles par des « axèmes » (liens monolingues) et des « axies » (liens multilingues). *PIVAX* offre également des interfaces permettant de visualiser les informations lexicales, d'y contribuer, et de les modifier.

Notre *SECTra_w* communiquera donc avec *PIVAX* afin de pouvoir préparer des minidictionnaires pour les segments à postéditer. Ces minidictionnaires devront être préparés à l'avance et toujours disponibles pour pouvoir être utilisés sans délai pour la post-édition. (Figure 39).

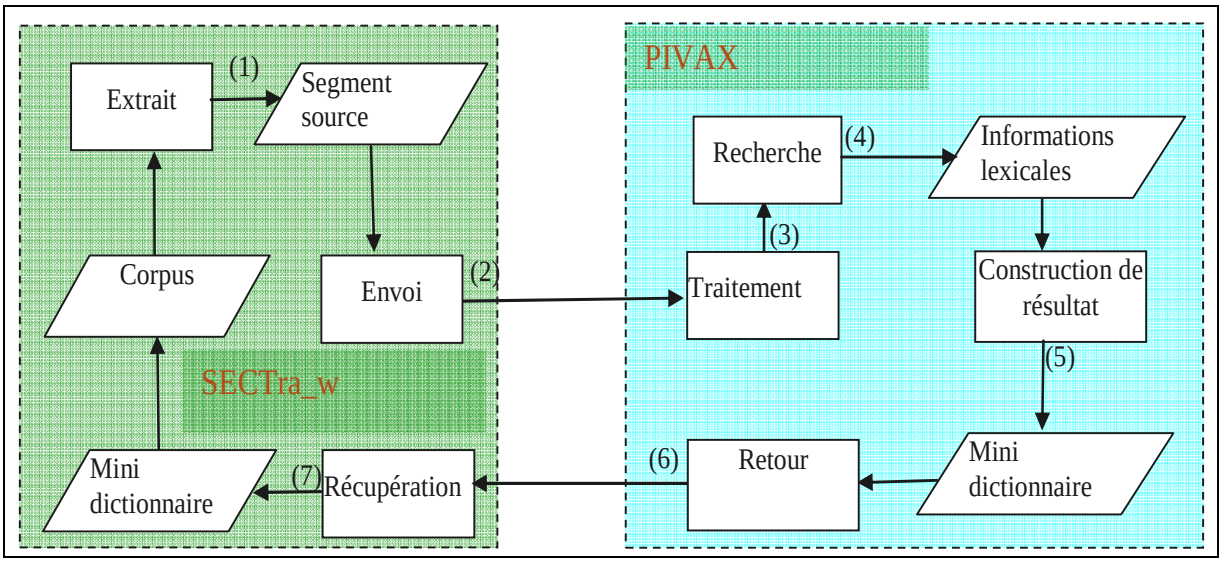


Figure 39: Schéma décrivant la communication entre SECTra_w et PIVAX pour la préparation du mini-dictionnaire d'un segment

Une fois un corpus ou un document à post-éditer importé, *SECTra_w* demande à *PIVAX* de construire un petit dictionnaire pour chaque segment. Le résultat (un fichier HTML pour chaque segment) est stocké dans *SECTra_w*. Quand l'utilisateur passe sur un segment, la consultation est déjà prête. Dans chaque article d'un minidictionnaire, on a les correspondances et les liens vers *PIVAX* pour afficher en détail cet article, et un bouton pour aller l'éditer sous *PIVAX*. Une fois qu'un article dans la partie "récupérée" a été édité par un humain, il est transféré dans la partie "projet". Selon la situation traductionnelle (couple de langues à traduire, seuil accepté pour afficher les entrées sur l'interface de post-édition…), *SECTra_w* utilise une *CSS* pour filtrer et choisir les informations à afficher, et montre immédiatement le résultat de la consultation quand l'utilisateur post-édite ce segment.

Comme *SECTra_w* gère plusieurs projets contenant plusieurs corpus, le nombre de segments-mc est très grand, et peut augmenter si l'on rajoute des documents. De plus, les ressources lexicales sont en constante évolution. Enfin, on peut rajouter de nouvelles langues cible à un projet de traduction, et il faut donc récupérer des équivalents pour ces langues, et mettre à jour les mini-dictionnaires en conséquence.

On ne peut donc pas créer ces minidictionnaires une fois pour toutes. Nous verrons dans la section suivante comment le faire de façon incrémentale, et en assurant une mise à jour automatique de tous les minidictionnaires, par une technique de "boucle infinie" (voir II.2.2.2 Problème 2.4 : Appel des ressources extérieures (BDLEX, systèmes de TA).

II.2.2.2 Problème 2.4 : Appel des ressources extérieures (BDLEX, systèmes de TA) en « boucle infinie » avec gestion des tâches

II.2.2.2.1 Préciser le problème, faire l'état de l'art

a. Motivations et première analyse du problème

On considérera qu'on a des *services* pouvant effectuer des *tâches*. Certains services peuvent s'appeler mutuellement, par exemple PIVAX doit pouvoir appeler SECTra_w pour stocker les exemples d'usage et faire faire leur "bonne traduction" par TA puis PE contributive.

On désire pouvoir faire exécuter des *commandes complexes* à des services, et souvent plusieurs en même temps. Par exemple, on peut avoir une *commande en boucle infinie* qui cycle dans tous les segments d'une MT et demande de les (re)traduire par TA, avec faible priorité et date limite non contraignante, en parallèle avec plusieurs *commandes fermées*

demandant la (re)traduction urgente des segments des pages sur lesquelles des contributeurs sont en train de travailler.

De plus, il faut que les gestionnaires des projets et les administrateurs du système puissent voir quelles sont les tâches en cours, et intervenir dessus, par exemple en les suspendant, en les détruisant, ou en changeant certains de leurs paramètres (urgence, adresses de livraison du ou des résultats…). Il faut aussi pouvoir définir des stratégies de choix de la tâche à accomplir, en cas de conflit.

b. Différents aspects de ce problème

On peut séparer différents aspects de ce problème:

- *envoi de commandes*: il faut pouvoir envoyer des commandes simples ou complexes, contenant des tâches devant être exécutées par plusieurs agents, et donc contenant la description d'un diagramme de tâches (une sorte de WF) — comme cela est fait dans CASH [Blanc, 1999] ou WICALE [Nguyen, 2009] pour le pilotage à distance d'Ariane-G5 (réseau LIDIA [Guillaume, 2003]).
- *traitement des commandes complexes*: on peut soit les traiter au niveau de chaque agent (par exemple, le système Ariane/LIDIA, considéré comme agent Web, exécute des commandes complexes, et alors il faudrait que d'autres agents de TA le fassent aussi), soit avoir un agent spécialisé pour les traiter (c'est ce qui est fait à l'intérieur du réseau LIDIA, où une machine virtuelle, sorte d'agent, s'occupe de ça — c'est ce qu'on a prévu de faire aussi au niveau d'Ariane-Y).
- *mode de communication*: par exemple, via des boîtes aux lettres implémentées comme des répertoires de fichiers (par exemple: utilisation du spool de zVM par Ariane/LIDIA).
- mode de transmission: Ariane/LIDIA utilise soit un protocole asynchrone (SMTP), soit des protocoles synchrones standard (HTTP) ou spécifiques via des sockets (comme CSTAR/Nespole!). On aimerait unifier cela, par exemple en utilisant toujours des REXXstacks [REXX, 2009] pour la transmission.
- *réception des résultats*: a priori, on doit pouvoir se contenter de boîtes aux lettres implémentées par des répertoires.
- *contrôle*: il faut une sorte de "poste de contrôle" montrant les commandes et les exécutions en cours (en fonction des droits et des choix), similaire au "moniteur d'activité" de MacOSX, ou au "tableau blanc" [Seligman et Boitet, 1993] de systèmes hétérogènes de reconnaissance de parole, avec la possibilité d'intervenir sur les files d'attente de commandes (comme dans un panneau de gestion d'une imprimante (MacOSX aussi).

II.2.2.2.2 Etat de l'art de l'appel de ressources externes

a. Techniques générales

On utilise souvent les modèles client-serveur et multiagent pour construire un système constitué de sous-systèmes pouvant se demander mutuellement des services. Des applications Web, ou des applications sur réseau tels que *Yahoo! Messenger*, *Skype*, etc. sont des exemples communs d'applications de l'architecture *client-serveur*. Le client est souvent responsable de visualiser les données et de fournir les interfaces d'interaction avec les utilisateurs. Le serveur est souvent responsable d'effectuer des calculs, de rechercher des données, de les traiter, et de répondre des résultats aux clients. Les protocoles utilisés sont le plus souvent TCP/IP, UDP, HTTP.

Il y a aussi des technologies communes permettant l'appel de ressources externes, telles que RMI (Remote Method Invocation) [Sun Microsystems, Inc., 2010], CORBA (Common

Object Request Broker Architecture) [CORBA, 2010]. RMI est un ensemble de classes permettant de manipuler des objets sur des machines distantes (objets distants) de manière similaire aux objets sur la machine locale (objet locaux). CORBA est une architecture logicielle pour le développement de composants et d'Object Request Broker ou ORB. Ces composants, qui sont assemblés afin de construire des applications complètes, peuvent être écrits dans des langages de programmation distincts, être exécutés dans des processus séparés, voire être déployés sur des machines distinctes.

b. Techniques déjà utilisées dans le domaine

Dans le domaine de la TA, des techniques d'appel de ressources externes ont été déjà utilisées dans des projets et des systèmes utilisés pour la traduction humaine. Par exemple, dans le projet *Eurolang Optimizer,* on a appelé le système de TA *LOGOS* (Logos-Optimizer). Le système *Yakushite.net* appelle le système de TA *PENSEE* [Shimohata et al., 1999]. Le système TRADOH [Vo-Trung, 2004b] permet d'appeler plusieurs systèmes de TA en ligne à la fois, et de montrer leurs résultats en parallèle. Ou encore, *Google* mobilise au moins 3000 machines pour traduire 51 paires de langues, dont certaines doubles, par exemple: FRA-DEU = FRA-ENG-DEU.

Un autre exemple intéressant est le réseau LIDIA [Guillaume, 2003]. La « maquette LIDIA » est un système de TA fondée sur le dialogue avec le rédacteur, réalisé en deux parties : l'une tourne sur Macintosh pour le dialogue de désambiguïsation, l'autre tourne sur un serveur IBM, pour toutes les phases d'Ariane-G5.

La communication entre les machines virtuelles a été faite par le mécanisme « spool » ou canal. Chaque machine range le résultat ou/et commande dans un spool pour qu'une autre machine les prenne et continue dans la suite des traitements [Guillaume, 2003].

LIDIA supporte l'échange des commandes et des données par les protocoles SMTP, HTTP, et socket.

II.2.2.2.3 Solutions proposées

Envoi de commandes. On a suggéré plus haut la possibilité de s'inspirer de ce qui est fait dans CASH ou WICALE pour le pilotage à distance d'Ariane-G5. Dans cette technique, une commande complexe est un fichier structuré contenant la description d'un diagramme de tâches (une sorte de WF) et des données à transmettre. Par exemple, on peut demander l'état des applications de TA exécutables sur une ou plusieurs machines virtuelles, et la compilation du dictionnaire de TL n°3 pour le couples RUS-FRA sur la machine GLC04.

On pourrait aussi utiliser un vrai langage de WF, réalisé en XML, et alors les données devraient aussi être représentées en XML (beaucoup plus verbeux mais permettant d'utiliser des outils tout prêts pour beaucoup de choses).

Mode de communication. Parmi les outils permettant de communiquer entre des systèmes, on a suggéré plus haut l'usage de REXXstacks.

REXXstacks est un outil permettant l'échange et la communication entre des systèmes sur réseau. Il est écrit en REXX (REstructured eXtended eXecutor), qui est un langage de programmation de haut niveau développé par IBM [Cowlishaw, 2009] depuis 1979 et mis en source ouvert depuis 1999. Ses versions ouvertes et très portables sont Regina et ooRexx (Rexx objet).

Une autre idée qui vient à l'esprit est de créer au niveau de la création d'une commande (simple ou complexe) un processus (thread) associé, qui créerait lui-même un socket établissant la liaison avec chaque agent nécessaire à la réalisation de la commande.

Mode de transmission. On a suggéré plus haut d'unifier les modes de transmission utilisés dans les systèmes existants : Ariane/LIDIA utilise soit un protocole asynchrone (SMTP), soit des protocoles synchrones standard (HTTP) ou spécifiques via des sockets (comme CSTAR/Nespole!). On aimerait unifier cela, par exemple en utilisant toujours des REXXstacks pour la transmission (le contenu pouvant correspondre à tout protocole de niveau plus haut comme HTTP ou langage de commandes d'Ariane ou langage de messages de CSTAR).

Réception des résultats. Pour la réception des résultats, notre solution est d'utiliser la technique de « boîte aux lettres». Nous utilisons des répertoires pour contenir les résultats reenvoyés par les systèmes externes. Il s'agit donc d'un scénario de communication asynchrone entre SECTra_w et les systèmes externes.

Contrôle. Pour le contrôle des processus, un sectra devra fournir une sorte de "poste de contrôle" montrant les commandes et les exécutions en cours (en fonction des droits et des choix), similaire au "moniteur d'activité" de MacOSX, ou au "tableau blanc" [Seligman et Boitet, 1993] de systèmes hétérogènes de reconnaissance de parole, avec la possibilité d'intervenir sur les files d'attente de commandes (comme dans un panneau de gestion d'une imprimante (MacOSX aussi).

II.2.3 Problèmes à dominante programmatoire

II.2.3.1 Problème 2.5 : Aspect contributif et ouvert

II.2.3.1.1 Préciser le problème, faire l'état de l'art

a. Précision du problème

La nécessité de permettre des contributions diverses, sur le Web, de façon ouverte, a été justifiée par ailleurs et au §II.1.2. Le problème qui nous occupe ici est de trouver comment supporter au mieux cet aspect dans le cadre d'un système d'exploitation de corpus de traductions.

Tout est assez clair aux niveaux conceptuels et algorithmiques, puisque nous nous situons dans le cadre du Web collaboratif (Web 2.0) et utilisons les outils associés. Par contre, l'aspect programmatoire pose des problèmes non négligeables.

a.i Aspect contributif

Pour l'aspect contributif, on a actuellement le choix entre diverses techniques pour réaliser un système contributif sur le Web:

- créer un service collaboratif en utilisant PHP et MySql, comme cela a été fait pour le système IToldU [Bellynck et al., 2005] pour l'apprentissage du vocabulaire anglais technique.
- partir de plates-formes génériques pour des classes d'applications ; par exemple, le système PIVAX [Nguyen, 2009] a été créé à partir de Jibiki pour des bases lexicales destinées à des systèmes de TAO hétérogènes.
- partir de plates-formes encore plus générales, comme des générateurs de Wiki (XWiki, Tikiwiki, Twiki, Docuwiki), comme cela a été fait pour la création des systèmes Translationwiki.net [Translationwiki, 2006], Wiktionary [Wiktionary, 2007], BEYTrans [Bey, 2009], XWiki Translation [XWiki Translation, 2010], etc.

Le choix de telle ou telle solution a des conséquences fortes sur le temps d'implémentation, la robustesse et la pérennité du système construit, sa maintenabilité, et sa capacité à passer à l'échelle.

a.ii *Aspect ouvert*

Nous avons abordé quelques aspects « ouverts » plus haut. Ainsi, l'ouverture aux différents types d'utilisateurs a été présentée dans le cadre du problème de la gestion des utilisateurs (Problème I.5). L'ouverture aux différents types de données a été présentée dans le cadre du problème de la modélisation et du traitement d'entrées non textuelles (Problème I.2). Enfin l'ouverture à de nouvelles langues a été présentée dans l'aspect multilingue (fondamental et de base) et dans l'aspect générique (Problème II.1).

Les aspects ouverts que nous voulons aborder dans cette section sont l'ouverture au niveau des programmes, et l'extensibilité.

Ouverture au niveau des programmes. Il s'agit ici non pas de "code source ouvert" pour les programmes, ce qui est plus ou moins imposé si l'on construit un système à partir d'une plate-forme Wiki, mais de "données ouvertes", comme dans Papillon : un utilisateur (ayant les droits adéquats) doit pouvoir récupérer les données (documents, segments-mc…) sous une forme (XML ou autre) préservant leur structure interne, et pas seulement sous une forme de présentation (html, pdf, etc.).

Extensibilité (programmatoire, pour l'intégration de nouvelles fonctionnalités). Il s'agit ici de permettre à des contributeurs d'ajouter des programmes dans le système. Dans notre cas, nous désirons limiter cela à des plug-ins, et à des programmes de niveau "utilisateur", par exemple permettre de programmer des flux de travaux (WF), car mettre un gros logiciel en évolution en source ouvert demande d'y passer beaucoup de temps, et nous n'avons personne pour suivre cela. Il faut donc trouver un bon équilibre, définir une API convenable, et structurer le logiciel en conséquence.

Par exemple, on voudrait pouvoir facilement créer un script pour sélectionner un sous-document formé de la page logique courante, des NbPrec pages logiques précédentes, et des NbSuiv pages logiques suivantes, et le visualiser sous telle ou telle forme avec en parallèle la version obtenue par TA seulement, et la version obtenue par TA et postédition. L'idée est d'enregistrer les actions de l'utillisateur, puis de lui permettre d'éditer le programme narratif obtenu (déjà un peu plus générique que la suite des actions telle quelle, par remplacement des constantes par des variables), de lui donner un nom et de le stocker dans un ensemble de "programmes utilisateur".

Un autre exemple très simple est de créer une macro à laquelle on donne une liste de couples <chaîne_1, chaîne_2> et une sélection, et qui itère sur la sélection le "chercher-remplacer", en mode interactif ou automatique, pour tous ces couples.

b. *Etat de l'art*

b.i *Gestion des conflits (aspect collaboratif)*

Les environnements collaboratifs non destinés à la post-édition, mais offrant l'édition collaborative et la gestion de documents multilingues, par exemple les CMS (Content Management Systems) collaboratifs libres gèrent cet aspect de plusieurs façons. Essentiellement, il faut définir ce qui se passe si deux contributeurs A et B veulent modifier la même unité d'information U (une page, un paragraphe, une phrase, selon la granularité désirée).

- **blocage en modification** : si A passe en mode d'édition de U, cela est signalé à B qui ne peut pas passer en mode d'édition. Les versions de U ont une structure linéaire (temporelle).
- **possibilité d'édition en parallèle** : A passe en mode édition, B en est averti ou non, et passe aussi en mode édition. Quand l'un des deux a fini (A ou B, par exemple B), U est

mis à jour, et l'autre (par exemple A) en est averti et peut voir l'état courant de U. Il peut alors décider d'abandonner, ou bien de continuer (en confirmant ou non tout ou partie des modifications faites par l'autre). Quand le second a fini, U est mis a jour.
Dans ce cas, les versions de U ont d'une part une structure linéaire (temporelle), et peuvent d'autre part avoir une structure arborescente. C'est ce qu'on désire dans le cas de la postédition d'un segment d'une mémoire de traductions, car on veut pouvoir produire des branches correspondant à des traductions différant selon les contextes (ou plutôt selon les classes de contexte).

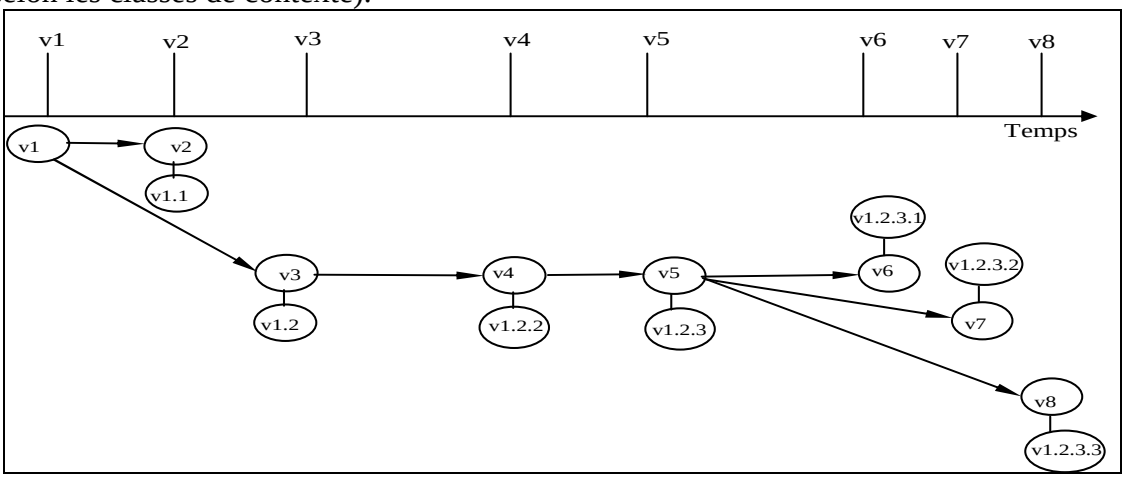


Figure 40: Illustration de l'évolution de versions dans le temps sur forme arborescente

b.i.1 Systèmes Web collaboratifs en général, et environnements Web de traduction ou post-édition collaborative

Les systèmes Web collaboratifs en général, comme Wikipedia et Wiktionary, et les environnements collaboratifs de traduction Traduwiki, BEYTrans, offrent un support au travail collaboratif au niveau des pages, mais pas au niveau des segments.

Le problème de conflit d'édition sur une page s'il y a plusieurs contributeurs travaillant en même temps a été traité dans ces systèmes par un mécanisme de gestion de versions. Ces systèmes gardent en effet plusieurs versions des modifications d'une page, correspondant aux éditions de plusieurs contributeurs. Ensuite les organisateurs ou les réviseurs autorisés comparent les différentes modifications pour produire une bonne version.

Cette idée est très bonne s'il s'agit de permettre aux organisateurs ou aux réviseurs de collecter des idées fournies par les contributeurs, dans le cas de la création d'un nouveau document.

Cependant, pour la post-édition de segments, cette idée n'est pas très bonne parce que la post-édition implique la modification et la correction de prétraductions. Une faute ne doit pas être corrigée de la même façon par plusieurs post-éditeurs. Pour éviter cela, un segment en cours de post-édition sera bloqué pour les autres post-éditeurs et dès que la post-édition sera terminée, ce segment pourra être amélioré par un autre post-éditeur. Pour chaque amélioration, il y a une version de post-édition correspondante.

b.ii Communication entre les contributeurs

Des moyens de discussion entre les contributeurs dans ces environnements collaboratifs sont les forums, ou des commentaires sur le projet. Par exemple, BEYTrans-v2 offre un forum pour la communication entre postéditeurs.

Cela semble nécessaire, mais n'est pas suffisant. En effet, il faudrait pouvoir mettre des commentaires, des questions, des explications directement sur un segment, et/ou sur chaque cellule d'un segment ou d'un pseudo-segment (=segment-mc).

b.iii Ouverture programmatoire

La technologie Wiki permet d'intégrer des plug-ins, ce qui donne une bonne ouverture au niveau de la programmation "classique".

En ce qui concerne l'ouverture à la programmation par des non-informaticiens, il s'agit essentiellement de proposer un langage spécialisé (L4G) dans lequel exprimer des sélections et des flux de travaux, ou encore de petits scripts basés sur les opérations élémentaires permises par l'interface de manipulation directe.

Il y a de nombreux exemples de tels L4G, par exemple JDF pour la gestion des flux numériques, ou encore des langages (parfois graphiques) pour la programmation par composants, etc. Le concept de "langage narratif" introduit par V. Bellynck dans sa thèse [Bellynck, 1999] semble aussi bien adapté à la programmation de scripts représentant des suites typiques d'actions d'un utilisateur.

Un programme de ce type peut être créé à partir des différences entre un état courant et un état final de données dont le traitement a été "délégué" à un agent externe (leur état courant peut être différent de leur état inital puisque la délégation d'un traitement n'implique pas nécessairement le blocage des données concernées), et son exécution effectue alors la synchronisation au retour.

II.2.3.1.2 Solutions proposées

a. Aspect contributif

Nous proposons d'utiliser un Wiki tel que XWiki, Tikiwiki, Twiki, Docuwiki, etc. comme plate-forme permettant de développer facilement et rapidement un système de support collaboratif au travail humain sur les corpus de traduction. Parmi ces Wikis, nous préférons XWiki.

Avantages. La technologie Wiki a été largement utilisée pour développer des environnements collaboratifs, XWiki est un Wiki Open-Source écrit en Java, permettant la gestion des versions du code source et des contributeurs au niveau de la programmation. Il offre tout à la fois les fonctions principales d'un wiki (édition collaborative, suivi d'information, gestion de l'accès des membres) et des possibilités de développement avancées (avec l'aide de langages de programmation utilisables directement au sein du wiki).

XWiki offre également la possibilité d'une gestion fine des droits d'utilisateurs, permettant de voir, d'éditer, de commenter, et d'administrer un espace ou même une page. Cela permet de nombreuses configurations : public ou privé, librement éditable ou non, ou un mélange des deux.

Inconvénients. Cependant, XWiki présente quelques inconvénients. Premièrement, XWiki ne permet la gestion de versions qu'au niveau des pages, alors que d'autres comme Docuwiki sont au niveau des paragraphes. Deuxièmement, XWiki n'est pas très compatible avec certains navigateurs tels que IE, car il utilise des fonctions évoluées (comme Ajax) qui ne sont pas encore standardisées sur les différents navigateurs.

Solutions. Nous devrons alors remodeler XWiki pour pouvoir l'adapter à un environnement de post-édition collaborative jusqu'au niveau des segments. Nous devrons aussi trouver divers moyens permettant la communication et l'échange entre utilisateurs, et remodeler la gestion des utilisateurs pour l'adapter à nos besoins.

b. Ouverture programmatoire

La solution que nous proposons a trois volets :

- délégation de certaines fonctionnalités, avec définition de formats (XML et spécifiques) pour l'envoi de commandes simples ou complexes et la réception des résultats ;
- définition d'une API permettant l'intégration de plug-ins programmés en Java ou en Groovy ;
- définition d'un langage narratif pour résoudre le problème de la synchronisation des données produites par un traitement effectué en délégation, et aussi pour écrire des sortes de "macros" utilisant les fonctions de base de l'interface.

II.2.3.2 Problème 2.6 : Sécurité des données, prévention du piratage, etc.

Nous avons abordé en partie le problème de la sécurité des données au §I.2.3.1 (problème I.6 : Gestion des utilisateurs). Nous nous concentrons ici sur la prévention de la pollution et du vol de données.

II.2.3.2.1 Le problème: prévenir la pollution et le vol de données

Il s'agit donc d'empêcher que des humains ou des programmes ne "polluent" les données, ou ne les volent si elles sont propriétaires.

a. Pollution des données

Les données peuvent être polluées pour plusieurs raisons. On peut ainsi :

- postéditer ou réviser en remplaçant une bonne traduction par une mauvaise ou une moins bonne, en général de façon inconsciente et pas par volonté de nuire ;
- introduire des faux sens ou des contresens de façon volontaire, pour changer plus ou moins fortement le message initial, dans la volonté de nuire à son auteur ;
- remplacer un texte par un autre, à caractère diffamatoire, grossier, pornographique, etc.

Comment empêcher cela, alors que ni l'auteur du message ni personne dans son environnement n'est le plus souvent en mesure de comprendre ce qui est écrit dans les langues cible ?

b. Vol de données

Les mémoires de traductions ont une valeur considérable, si on les utilise dans des contextes professionnels, non seulement car elles sont spécifiques, et utiles pour la traduction ou l'accès multilingue, mais aussi parce que l'analyse de la terminologie qu'elles contiennent peut renseigner la concurrence sur les produits en cours de développement et sur les techniques utilisées. C'est pour cela que, chez SITE/ITEP/Sonovision [Sonovision-ITEP, 2008], la rédaction et la traduction technique des documentations relatives aux avions Dassault se fait dans une partie sécurisée des installations.

Ce risque est évidemment d'autant plus fort que, comme dit plus haut, nous voulons une ouverture la plus grande possible au niveau des données.

II.2.3.2.2 État de l'art

a. Prévention de la pollution des données

La solution utilisée par Wikipedia est la "modération a posteriori", et a deux aspects:

- mise en place d'indicateurs automatiques alertant sur ce risque. Par exemple, on repère que B intervient toujours après A (ou après tout autre contributeur) pour modifier ce qu'il a écrit. On détecte alors un conflit.

- mise en place d'un modérateur pour les données concernées (en général, tout un article). Dans Wikipedia, il y a actuellement 900 000 articles pour la version francophone (le 15 janvier 2010)[Numerama, 2009], dont environ 30.000 articles ainsi modérés.
- annulation automatique des modifications faites par des contributeurs repérés comme "pollueurs" pour tout ou partie des informations.

Une solution plus draconienne est la "modération a priori". Elle consiste à de ne donner des droits en écriture à une personne qu'après avoir vérifié son profil, et lui avoir fait signer un engagement de bonne conduite, en la prévenant des conséquences éventuelles d'une pollution volontaire.

b. Prévention du vol de données

Une méthode de base utilisée dans la plupart des systèmes informatiques pour la prévention du vol de données est la gestion des utilisateurs. Avec une gestion des utilisateurs adéquate, le système peut autoriser ou empêcher un utilisateur de voir et de copier des données.

Certains systèmes Web tels que *Google Books*[33] empêchent la fonction de copie, fournie par le navigateur, qui permet de copier des données directement sur l'écran chez les clients (Figure 41).

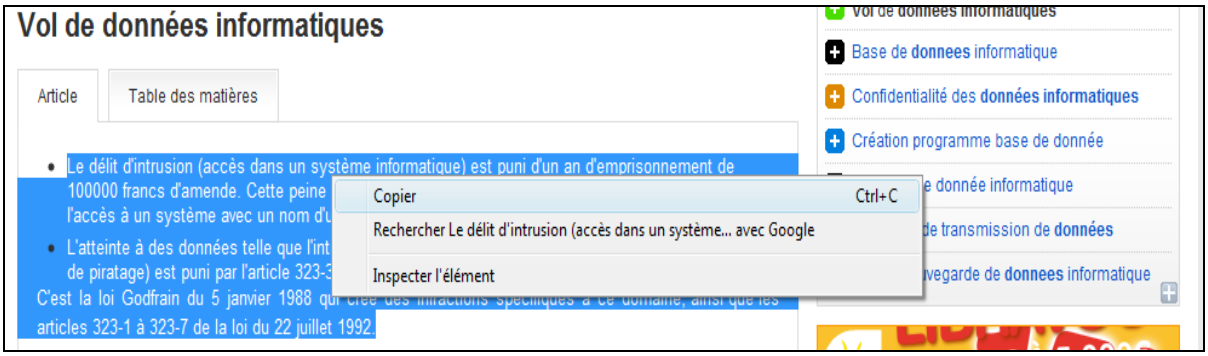


Figure 41: Fonction de copie de données fournie par le navigateur

Cependant, la prévention du vol de données effectué automatiquement par des programmes est un problème difficile. On vole souvent des données d'un site Web en envoyant automatiquement des requêtes avec des paramétrages adéquats, et en analysant des pages de résultat pour y récupérer des données intéressantes.

Pour prévenir de tels vols, certains systèmes comme *Google Translate*[34] exigent de fournir une adresse IP dynamique et des paramètres dynamiques (ces paramètres varient à chaque fois et sont générés automatiquement par le système). De plus, certains systèmes limitent le nombre des requêtes envoyées à partir d'une même adresse IP pendant une certaine durée.

II.2.3.2.3 Solutions retenues

a. Prévention de la pollution des données

Nous avons adopté les solutions suivantes.

- Pour la pollution "involontaire", on se contente d'interdire à un contributeur A de niveau traductionnel plus bas que celui d'un contributeur B (pour un projet ou un document donné) de modifier ce qu'a produit A.

[33] http://books.google.fr/books

[34] http://translate.google.fr/#

- L'organisateur d'une campagne, ou plus généralement l'administrateur d'un projet, peut bloquer les modifications sur toute sélection des données de son projet.
- Pour la pollution "volontaire", étant donné que nous désirons l'ouverture la plus grande possible au niveau des données, nous utiliserons une solution à la Wikipedia.

b. Prévention du vol de données

Pour la prévention du vol de données, nous proposons une gestion des utilisateurs adéquate, avec la gestion de leurs profils. Nous devrons aussi empêcher de copier des données par la fonction de copie fournie par les navigateurs. En permettant d'utiliser des programmes pour interroger automatiquement les données, nous proposons d'adopter la solution de *Google Translate*[35] qui consiste à fournir des requêtes contenant des paramètres secrets tels qu'un compte, une chaîne spéciale, etc. De plus, nous ne fournirons la fonction d'export de données que selon des droits adéquats.

II.3 Implémentation, expérimentation et évaluation avec SECTra_w

II.3.1 Spécification

II.3.1.1 Objectifs

Notre objectif est ici d'implémenter, expérimenter et évaluer les solutions proposées ci-dessus. Pour cela, nous avons utilisé le système SECTra_w, initialement développé pour le support à l'évaluation de systèmes de TA, et l'avons étendu en développant plusieurs fonctions et aspects nécessaires en support de la post-édition collaborative de corpus de traductions.

II.3.1.2 Architecture générale de SECTra_w pour la post-édition contributive

Les fonctions et composants suivants ont été développés :

- Création de nouveaux projets de post-édition/traduction de corpus.
- Import et prétraitement des corpus à post-éditer.
- Préparation des suggestions linguistiques servant à la post-édition.
- Conception et implémentation d'un éditeur avancé pour la post-édition collaborative.
- Export de résultats.

[35] http://translate.google.fr/#

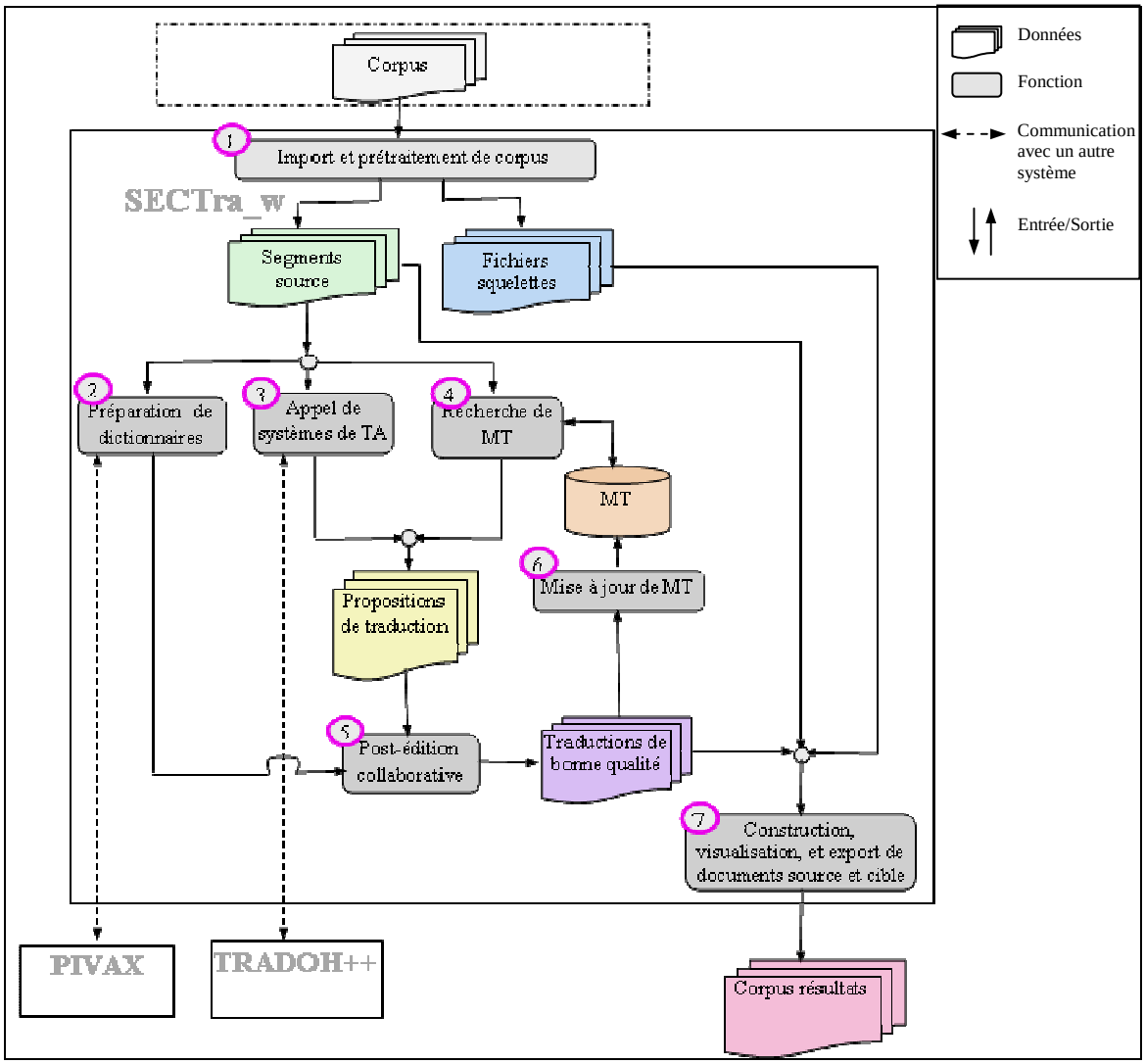


Figure 42: Architecture de SECTra_w pour la post-édition contributive

II.3.1.2.1 Import et prétraitement de corpus

SECTra_w doit permettre d'importer non seulement des corpus parallèles alignés au niveau des segments, mais aussi des corpus de documents qui sont soit des documents textuels, soit des pages Web. SECTra_w doit donc traiter le corpus d'entrée si nécessaire avant de préparer les aides linguistiques pour la post-édition. Ce traitement comporte les étapes suivantes :

1) Vérification et normalisation des documents d'entrée pour vérifier le codage et le format d'entrée. Dans le cas où le document d'entrée est une page Web, cette étape doit vérifier la cohérence du format HTML de la page d'entrée et il peut normaliser et corriger des erreurs si possible.

2) Segmentation des documents d'entrée pour les découper en unités de traduction (segments ou supersegments selon les systèmes de TA à appeler). Dans cette étape, un document d'entrée tel qu'une page Web est transformé en deux fichiers. Le premier contient une liste de segments source textuels, et le second est appelé « fichier squelette ». Le fichier squelette contient les codes du document, par exemple les codes HTML. Ce fichier est utilisé avec les traductions finales pour générer des fichiers de résultats après la post-édition. Grâce à ce fichier squelette, le fichier de résultat a la même structure que celui du fichier d'entrée, et le document traduit a exactement la même présentation que le document source.

3) Correction de segments source s'ils contiennent des fautes. Les segments source corrigées sont utilisés pour préparer des aides linguistiques, pour mettre à jour les mémoires de traductions, et pour générer les documents sources de résultats grâce aux fichiers squelettes.

II.3.1.2.2 Fonction de préparation des aides linguistiques

Après avoir importé et prétraité un corpus d'entrée, SECTra_w prépare des aides linguistiques pour tous les segments source des corpus à post-éditer. Cette préparation permet de proposer aux utilisateurs ces aides linguistiques de façon proactive. Les aides linguistiques pour un segment source sont des pré-traductions en N langues produites par plusieurs systèmes de TA, des suggestions à partir de la ou des MT, et un dictionnaire local (propre à chaque segment) contenant des informations lexicales.

Cette fonction est implémentée à l'aide de boucles infinies qui envoient les segments source depuis SECTra_w aux « agents » TRADOH+, PIVAX, et à la fonction de recherche dans la ou les MT. Le TRADOH+ est responsable d'appeler les systèmes de TA en ligne et hors-ligne pour produire des pré-traductions. PIVAX est utilisé pour construire un minidictionnaire pour chaque segment source.

La fonction de recherche dans la MT retrouve les traductions déjà postéditées dans la mémoire de traductions et est aussi appelée par une boucle infinie qui garantit que chaque segment sera revisité dans le futur à peu près autant de fois que tout autre segment. Ces boucles infinies sont aussi utilisées pour obtenir et mettre à jour les aides linguistiques fournies par des systèmes extérieurs.

L'utilisation de boucles infinies a pour objectif d'assurer que les aides linguistiques sont mises à jour dans le cas où les segments source sont corrigés et dans le cas où il y a de nouvelles langues ajoutées dans le corpus. De plus, cela permet d'effectuer tous les traitements de façon incrémentale, et pas par lots, ce qui prend un temps beaucoup trop important pour de gros corpus.

II.3.1.2.3 Editeur adapté à la post-édition collaborative et contributive

a. Ergonomie et mise en page de l'éditeur

L'éditeur doit permettre à plusieurs contributeurs de travailler simultanément sur la même collection de données (segments, pages, ou documents). Pour cela, l'éditeur doit gérer et contrôler les modifications des contributeurs. Si un segment est en cours de post-édition, le système doit changer son état par une marque spéciale pour le distinguer des autres segments, et ce segment doit être bloqué temporairement pour empêcher la post-édition par d'autres post-éditeurs.

SECTra_w doit également gérer les versions de post-éditions, mais il assure que la version suivante est l'amélioration de la précédente et que la version finale est en principe la meilleure.

L'éditeur doit également être mis en page selon les principes de présentation des interfaces de visualisation et d'évaluation présentées dans le chapitre 1.

- Verticalité: tous les objets du même type apparaissent dans la même "colonne".
- Horizontalité: tous les objets liés au même segment source (y compris éventuellement ses corrections) constituent un segment multilingualisé et contextualisé (segment-mc) et sont présentés dans la même « ligne ».
- Localité : les fonctions principales doivent résider toujours dans la même région. La post-édition est faite dans une zone fixe en haut où tout ce qui concerne les segments apparaît (texte source, texte post-édité, pré-traductions, suggestions de MT). Les informations lexicales relatives au segment actif sont localisées dans une zone en bas.

- Pro-activité : quand l'utilisateur clique sur le segment source, toutes les suggestions proposées par des systèmes de TA et des MT, et les informations lexicales correspondantes, apparaissent immédiatement.

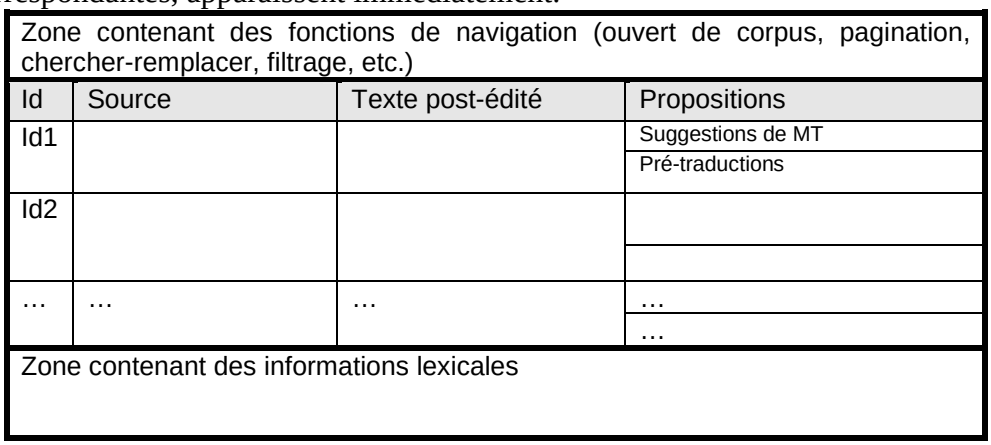


Figure 43: Spécification de l'éditeur de post-édition de SECTra_w

b. Modes de l'éditeur

L'éditeur a plusieurs modes de fonctionnement.

Mode édition. Dans ce mode, l'éditeur incite les contributeurs à travailler comme des post-éditeurs, c'est-à-dire qu'ils doivent d'abord cliquer sur le segment source et le lire, avant de post-éditer la traduction. La traduction proposée est la meilleure traduction disponible, c'est-à-dire soit le résultat d'une traduction ou d'une post-édition humaine, soit la "meilleure" des traductions automatiques candidates (déterminée pour l'instant de façon empirique).

Pendant la post-édition, l'utilisateur peut donner une note de qualité à sa version de post-édition, et il peut mettre des commentaires ou des remarques sur le segment en question. Les contributeurs peuvent choisir plusieurs langues comme langues de référence pour faciliter leurs travaux en sus de la langue source. L'éditeur doit donc permettre d'ajouter, de montrer, de cacher des colonnes.

Les contributeurs peuvent aussi choisir le nombre de segments à afficher dans une page logique. Dans ce mode, l'utilisateur peut également contribuer et/ou mettre à jour des informations lexicales dans le dictionnaire correspondant. Quand un contributeur a fini de post-éditer un segment, l'éditeur crée une nouvelle version de post-édition pour ce segment.

Mode lecture. Ce mode sert à visualiser une "trace" de l'effort de post-édition et à vérifier la qualité des résultats.

Pour l'observation des efforts de post-édition, l'éditeur met toujours à jour et montre le pourcentage d'achèvement de la post-édition. Il visualise aussi des opérations d'édition (insertion, suppression) pour transformer une pré-traduction en une post-édition comme la fonction «Track Changes» dans le Word.

Prétraductions	Post-éditions
Un cambrioleur est entré de force dans ma pièce.	Un cambrioleur ~~est~~ a ~~entré~~ forcé ~~de force~~ ~~dans~~ ma ~~pièce.~~ chambre.

Figure 44: Visualisation des opérations d'édition (insertion, suppression) pour transformer une pré-traduction en une post-édition

Enfin, pour faciliter la vérification de la qualité des résultats, SECTra_w doit permettre de visualiser les documents source et cible en parallèle, en maintenant horizontal l'alignement entre les segments source et cible correspondants (voir Figure 49).

c. Fonctions de support à la post-édition

c.i Support des suggestions de TA et de MT

Appel de TA. Pour chaque segment à post-éditer en une langue, il faut fournir plusieurs pré-traductions produites par plusieurs systèmes de TA. Actuellement, l'échange entre SECTra_w et les autres systèmes (PIVAX, TRADOH++, iMAG, Notepad++, etc.) se fait par le protocole http et le format est la page HTML de résultat.

Voici les caractéristiques d'une commande appelant TRADOH++ pour préparer des prétraductions en plusieurs langues pour chaque segment source.

Paramètre	Nom	Valeur par défaut	Note
Protocole	Protocol	http	Protocole de commande
Nom_serveur	Server	tradoh.imag.fr	Nom de serveur
Port	Port	80	Port
URL	url	/appelTA	Chemin sur le serveur
Id	id	" "	Identifiant du segment
Texte_source	Source_text	" "	Segment source à traduire
Délai	delay	200	Temps d'attente maximal en secondes, s'il est dépassé, on passe à une autre requête
Liste_TA	MTs	Systran, Reverso	Liste de noms de systèmes de TA
Langue_source	sl	en	Langue source
Liste_langue_cible	tls	fr, vi	Liste des langues cible

Table 21 : Paramètres de la fonction de recherche de MT

Le format de résultat renvoyé par TRADOH++ est représenté en format XML et est conforme à la DTD suivante.

```
<!ELEMENT segment (pretrans*)>            ; un article = n segments
<!ATTLIST segment  id  CDATA>             ; id de segment
    <!ATTLIST pretrans  lang CDATA>       ; langue
    <!ATTLIST pretrans  MT CDATA>         ; nom de système de TA
    <!ELEMENT pretrans (#PCDATA)>         ; contenu de prétraduction
```

Figure 45: DTD du format de résultat renvoyé par TRADOH++

Voici l'exemple des prétraductions du segment source « *Man and the Water Cycle* ».

```
<segment id="eolss_doc1_seg_01">
 <pretrans MT ="Systran" lang="fr"> Homme et le cycle de l'eau  </pretrans>
 <pretrans MT ="Reverso" lang="fr"> Man et l'Eau Font du vélo </pretrans>
 <pretrans MT ="Google"  lang="fr"> L'homme et le cycle de l'eau </pretrans>
</segment>
```

Recherche « exacte » dans les MT. SECTra_w ne fournit que la recherche exacte des segments source. En cas de succès, on trouve donc des traductions déjà post-éditées dans les MT et par hypothèse bonnes, ou bien des prétraductions automatiques déjà effectuées, ce qui réduit les appels immédiats à la TA. En cas d'échec, on appelle des systèmes de TA. On ne fait pas de recherche approchée car elle est très lente, et en général moins utile que la TA. Nous voyons la spécification de MT et de cette fonction au chapitre 3, §III.3.1.3.1.

c.ii Fonction de support des informations lexicales

Cette fonction doit fournir aux post-éditeurs des informations lexicales utiles de façon proactive. Ces informations lexicales doivent donc pouvoir être obtenues par fusion à partir de ressources différentes, et elles doivent être relatives au domaine du segment. Cette fonction doit fournir une interface permettant de visualiser, de proposer, et de normaliser des informations lexicales. Elle doit aussi permettre de mettre à jours des informations lexicales quand un segment source est corrigé.

L'appel de la préparation d'un minidictionnaire dans SECTra_w et PIVAX avec les paramètres de la requête http et le format du résultat HTML a la forme suivante.

Paramètre	Nom	Valeur par défaut	Note
Protocole	Protocole	http	Protocole de commande
Nom_serveur	server	eolss.imag.fr	Nom de serveur
Port	port	80	Port
URL	url	/pivax?QuickSearch.po	Chemin sur le serveur
Mot_à_chercher	headword	'test'	Mot ou suite de mots à consulter
Délai	delai	100	Temps d'attente maximal, s'il est dépassé, on passe à une autre requête
Lemma	lemma	yes	Demander de passer à la lemmatisation si elle existe
Langue_source	ls	en	Langue source
Langue_cible	lc	*	Langue cible
Segment_ID	segid		Identificateur de segment pour que SECTra_w réattache le résultat à ce segment

Table 22 : Paramètres de commande de préparation de minidictionnaire

Le résultat renvoyé par PIVAX est un fichier HTML sous le format simple de tableau. Sur ce fichier, on déclare les informations à extraire (par exemple en utilisant des XPath comme dans Jibiki) et à afficher sur l'interface simplifiée de PIVAX dans SECTra_w.

```
<html>
   <body>
      <table>
           <tr><td lang="en">test<td></tr>
           <tr><td lang="fr" cat="verb">tester<td></tr>
           <tr><td lang="fr" cat="noun">essai<td></tr>
           <tr><td lang="fr" cat="noun">test<td></tr>
      </table>
      <div>... autres informations ...</div>
   </body>
</html
```

MOT_SOURCE	/html/body/table/tr/td[lang="en"]	Récupération du mot source
MOT_CIBLE	/html/body/table/tr/td[lang="fr"]	Récupération de la liste des équivalents dans les langues cible

c.iii *Fonction de recherche et de filtrage de données*

Filtrage de données. Cette fonction permet de filtrer des traductions selon plusieurs critères. Par exemple, nous pouvons filtrer les traductions qui ne sont pas encore post-éditées, les traductions fabriquées par un post-éditeur quelconque, etc.

Voici un tableau de critères, correspondant aux méta-données associées à un segment.

Critères à filtrer	Valeurs de métadonnées	Description
Non-postédition	Etoiles = **	Filtrer toutes les traductions qui ne sont pas encore post-éditées
Bonne qualité	Etoiles = **** Note > 13	Filtrer toutes les traductions de bonne qualité.
Date	Date_cree = « today »	Filtrer toutes les traductions créées à une date quelconque.
Auteur	Auteur= « all »	Filtrer toutes les traductions produites par un auteur.

Recherche-remplacement de données. Cette fonction permet de chercher un texte en langue source ou/et en langue cible, et de remplacer le texte en langue cible par un autre texte. Par exemple :

Texte source à chercher	*please*
Texte traduit à chercher	*s.v.p*
Texte à remplacer	*s'il vous plaît.*

Elle doit fournir deux modes : le premier permet de vérifier segment par segment avant le remplacement. Le second permet de remplacer tous les segments à la fois.

Si le texte à remplacer est absent, cette fonction est la fonction de recherche seulement.

d. Fonction de support au travail collaboratif

d.i Fonction de communication et d'échange de connaissances entre les post-éditeurs

Cette fonction fournit des moyens de communication et d'échange de connaissances entre les contributeurs, tels qu'un tchat, un forum, et la possibilité de mettre des commentaires sur chaque page et sur chaque segment.

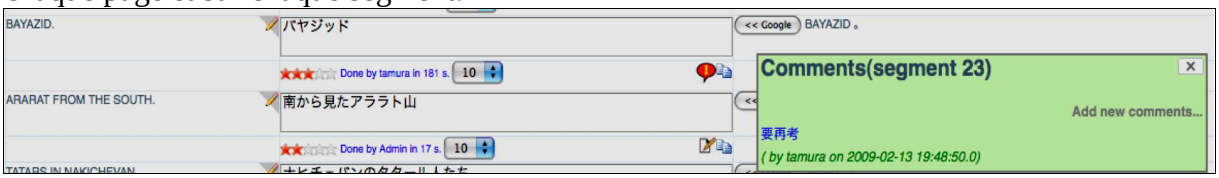


Figure 46: Commentaire sur chaque segment

d.ii Fonction de gestion de versions de post-éditions

SECTra_w doit permettre à plusieurs post-éditeurs d'améliorer une traduction. Il faut donc gérer des versions de post-édition. Cette fonction permet aussi de restaurer les post-éditions antérieures quand la version de post-édition courante est mauvaise à cause d'erreurs de saisie ou de traduction.

Figure 47: Gestion de versions de post-édition dans SECTra-w

d.iii Fonctions d'association des métadonnées au segment

SECTra_w doit permettre d'associer à chaque segment des métadonnées nécessaires. Ces métadonnées comprennent l'auteur (humain ou machine) qui produit le segment, le niveau traductionnel de son auteur (de 1 étoile à 5 étoiles), la note de qualité (de 0 à 20), le temps passé, etc. En se basant sur ces métadonnées, les post-éditeurs postérieurs peuvent plus facilement contribuer et améliorer la qualité des segments. Ces métadonnées permettent aussi à la mémoire de traductions de filtrer automatiquement les meilleurs segments pour le recyclage.

Figure 48: Métadonnées associées à une postédition

II.3.1.2.4 Fonctions de gestion d'effort et de progrès de post-édition

SECTra_w doit fournir des fonctions de gestion d'effort et de progrès de post-édition de projet. Cette fonction doit donc faire des statistiques sur la post-édition de chaque contributeur. Par exemple, elle doit fournir les statistiques sur les segments, mots, et caractères modifiés par un contributeur. Il faut aussi donner le temps total passé par chaque post-éditeur.

II.3.1.2.5 Fonctions de construction et visualisation de documents source et cible synchronisés

SECTra_w doit reconstruire des documents cible en utilisant les traductions disponibles. Il doit visualiser les documents source et cible en parallèle avec synchronisation des segments source et cible équivalents.

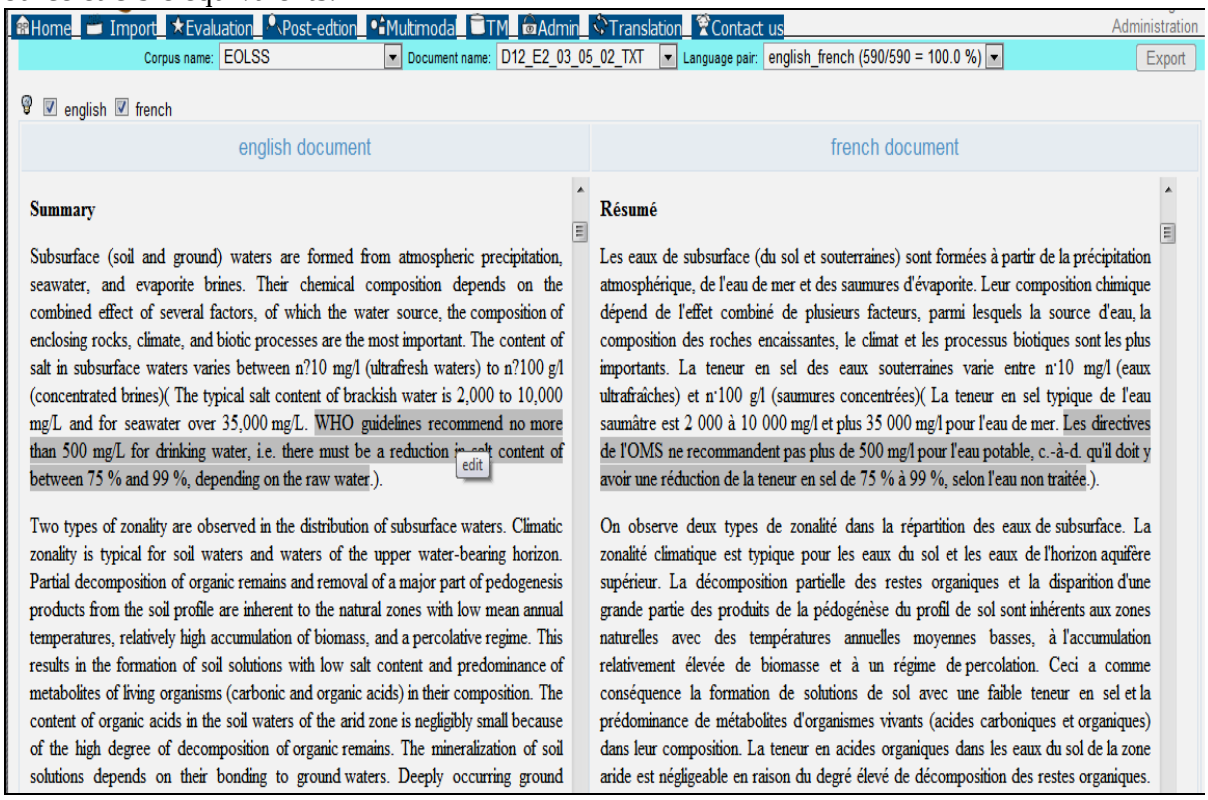


Figure 49: Visualisation des documents source et cible

Le système doit permettre aux post-éditeurs de retourner à l'éditeur de post-édition facilement, à l'endroit où ils étaient, une fois qu'ils ont fini de contribuer.

II.3.1.2.6 Organisation des données dans SECTra_w

Les données sont organisées selon un modèle hiérarchique pour pouvoir unifier le traitement et la gestion des corpus. Chaque projet de post-édition a une base de données et une mémoire de traduction. Chaque projet a un ou plusieurs corpus dans lesquels chacun a un ou plusieurs documents. Chaque document peut être organisé en plusieurs pages logiques. Chaque page logique est définie par un nombre de segments quelconques. Chaque segment source a un identifiant unique qui est utilisé pour le lier avec des occurrences (des traductions automatiques, des post-éditions, une liste de versions, un minidictionnaire, un graphe UNL, etc.).

II.3.2 Implémentation

II.3.2.1 Contexte

Nous avons étendu SECTra_w pour le support à la post-édition collaborative et contributive selon ces spécifications. Ce système a été expérimenté pour des utilisations réelles dans deux projets de post-édition, EOLSS , et DSR (voir II.3.3.1).

On a amélioré certaines fonctions existantes qui avaient déjà été développées pour le support de campagnes d'évaluation, telles que la fonction d'appel de systèmes de TA, la gestion des utilisateurs, etc., et développé et ajouté de nouvelles fonctions pour traiter de corpus de documents et pour permettre la post-édition collaborative et contributive.

Implémentation réalisée 6 mois de fin mars à fin septembre 2008.

Tâche	CCH	DSE	DSI	codage	Préparation. (Segmentation, Appel de TA)	import pour tests	testS 1	import pour la post-édition	Tests	**Total**
Jours	15	30	15	90	20	1	1	5	3	**180**
Taille	10 p.	20 p.	15 p.	150.000 lignes, Java classes	25 documents Appel de Systran,Google, Reverso	1000 seg.	3 post-éditeurs	15.000 seg.	10 post-éditeurs	

Table 23: Quelques éléments factuels sur l'implémentation du support à la post-édition collaborative.

II.3.2.2 Réalisation

On peut accéder à la fonction de post-édition de SECTra_w à

http://eolss.imag.fr/xwiki/bin/view/Corpus/PostEdit.

Pour le prétraitement des corpus d'entrée, nous avons développé un outil de segmentation basé sur des fichiers ou des segments de guidage. Les segments de guidage sont ceux des mémoires de traductions ou ceux contenus dans des fichiers compagnons associés avec les documents d'entrées (par exemple, des fichiers UNL). On a déjà développé des outils de conversion de corpus parallèles alignés au niveau des segments en formats XML prédéfinis.

On a mis en œuvre le principe de délégation pour permettre à SECTra_w d'utiliser Google Translate pour la segmentation de documents (en terme de pages Web), d'utiliser PIVAX pour préparer des informations lexicales, et d'utiliser TRADOH+ pour préparer des pré-traductions par TA.

On a créé un éditeur supportant la post-édition collaborative et contributive. Cet éditeur permet à plusieurs post-éditeurs de travailler simultanément sur la même collection de données. On a intégré dans cet éditeur plusieurs fonctions permettant de faciliter et d'accélérer la post-édition selon les spécifications ci-dessus.

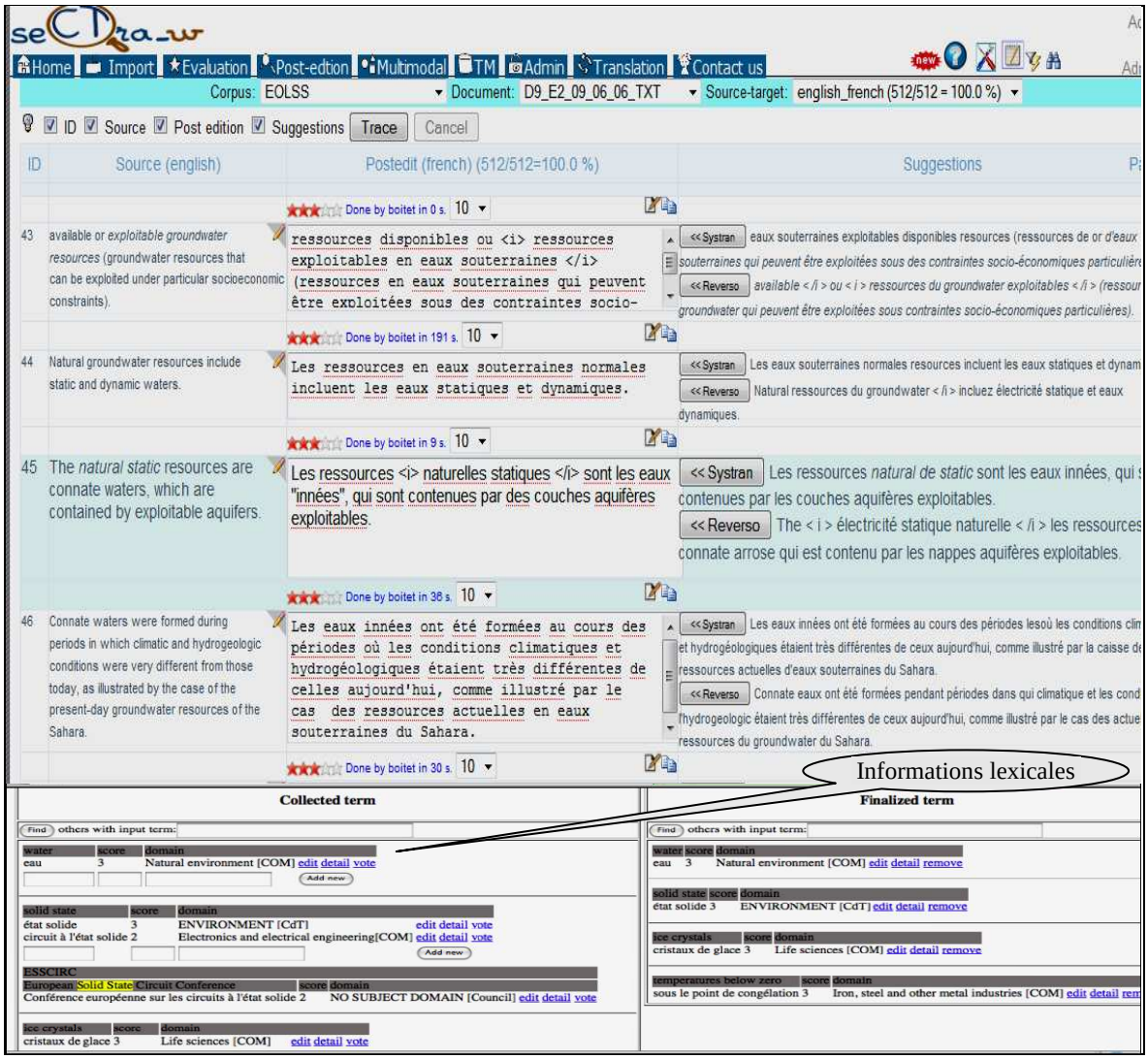


Figure 50: Editeur de post-édition de SECTra_w

II.3.3 Expérimentation

II.3.3.1 Contexte

SECTra_w a été utilisé dans deux projets réels de traduction et post-édition de corpus multilingues.

Le premier est le projet EOLSS/UnescoL réalisé de février 2008 à octobre 2008 dans le cadre d'un contrat entre l'Association Champollion et la fondation UNDL[36]. Dans ce projet, SECTra_w a été utilisé pour le support et la gestion de la traduction de bonne qualité de 25 articles de l'encyclopédie EOLSS (Encyclopedia of Life Support Systems), soit environ 220K mots (880 pages standard), ou 13676 segments. SECTra_w a fourni pour ce projet un environnent collaboratif en ligne pour la post-édition humaine appliquée aux résultats de systèmes de TA et de déconvertisseurs UNL.

[36] The UNDL FOUNDATION is a private Swiss law Foundation with head office in Geneva, Switzerland, legally representing the United Nations Organization in protecting its property rights pertaining to the UNL language and system, and legally representing the EOLSS Publishers and the UNESCO Joint Committee in translating the EOLSS with the use of the UNL technology

Le deuxième est le projet DSR (Digital Silk Road), réalisé par le NII (National Institute of Informatics, Tokyo, Japon) sous l'égide de l'UNESCO. C'est un projet de recherche visant à créer des archives numériques d'héritage culturel grâce à la collaboration entre l'informatique et les sciences humaines. Ces archives numériques ont été créées essentiellement à partir de ressources gigantesques collectées depuis des siècles et stockées dans la bibliothèque Tokyo Bunko. Cette bibliothèque contient environ 880.000 livres d'importance historique et 24.000 livres sur la Chine et l'Asie. En avril 2009, il y avait dans le site DSR 110 volumes (31 auteurs, 27.326 pages), correspondant à 52 livres, des livres, des recueils de photos, des cartes, tous contenant des textes plus ou moins volumineux, qui ont été numérisés et publiés sur Internet. La numérisation et la traduction de ces volumes sont en cours depuis 2002.

Actuellement, on vise à traduire les légendes des images contenues dans ce corpus pour servir à la recherche multilingue. Avant décembre 2008, la plupart des traducteurs humains dans ce projet utilisaient Excel ou MS Word comme outils pour leur traduction. Cela a provoqué beaucoup de difficultés pour eux et pour les organisateurs du projet, en particulier parce que les organisateurs ont été surchargés avec la gestion des fichiers. Ils devaient en effet envoyer des fichiers à traduire à chaque nouveau traducteur, et les traducteurs oubliaient parfois de rendre leurs fichiers de résultats aux organisateurs, qui ne pouvaient donc pas suivre la progression du projet. De ce fait, les traducteurs humains ne bénéficiaient d'aucun support informatique ni aide linguistique pour leurs travaux. C'est pourquoi, pendant mon stage au NII (de décembre 2008 à avril 2009), j'ai étendu et adapté SECTra_w pour le support de la traduction et de la post-édition de ce projet.

II.3.3.2 Prétraitement de données et préparation de projets

II.3.3.2.1 Import et traitement de corpus

Nous avons importé 25 documents sur l'eau et l'écologie de l'encyclopédie EOLSS, représentant environ 220K mots (13676 segments) ou 880 pages standard, tandis que le corpus EOLSS complet contient 6600 documents (62,5 M mots). Chaque document est constitué d'un fichier HTML (.aspx), d'un fichier compagnon .unl, et de fichiers satellites (images, icônes, et autres hors-texte). Le fichier .unl contient des graphes UNL représentant des segments découpés à partir du fichier .aspx correspondant.

Présentation de fichier HTML	Fichier compagnon .unl
The wave propagates from the source with a velocity of long gravity water waves in accordance with the **equation** $C_G = (g\ H)^{1/2}$, (1) **where** g is the acceleration due to gravity, and H is the depth of the **basin**. **Because** the average depth of the world ocean is 4 km, the typical velocity of tsunami in the ocean **is 200 m s-1** or **720 km h-1**. **Such a wave**, propagating with the velocity of an airplane, may traverse the Pacific ocean in 10-12 hours and bring down a wall of water 10 m high with a velocity of more than **70 km h-1** upon a calm ocean beach.	[S:44] {org} **The** … **equation** {/org} {unl} … {/unl} ;graphe unl représentant la phrase [/S] [S:45] **{org} where … basin. {/org}** {unl} … {/unl} [/S] [S:46] **{org} Because … is 200 HTM1 or 720 HTM2. {/org}** {unl} … {/unl} [/S] [S:47] **{org} Such a wave, … 70 HTM1 … ocean beach**. {/org} {unl} … {/unl} [/S]

Figure 51: Fichier HTML et fichier compagnon .unl

Nous avons utilisé notre propre segmenteur pour segmenter ces fichiers .aspx pour qu'on puisse à la fois récupérer des segments correspondant aux segments existant dans les fichiers .unl, et conserver leurs structures et leurs formats.

Ses entrées sont un fichier .aspx et un fichier .unl contenant les segments pour guider la segmentation. Ses sorties sont une liste de segments et un fichier squelette correspondant au fichier .aspx. Le ficher squelette contient des codes HTML et des « placeholders » utilisés pour insérer des segments de résultats.

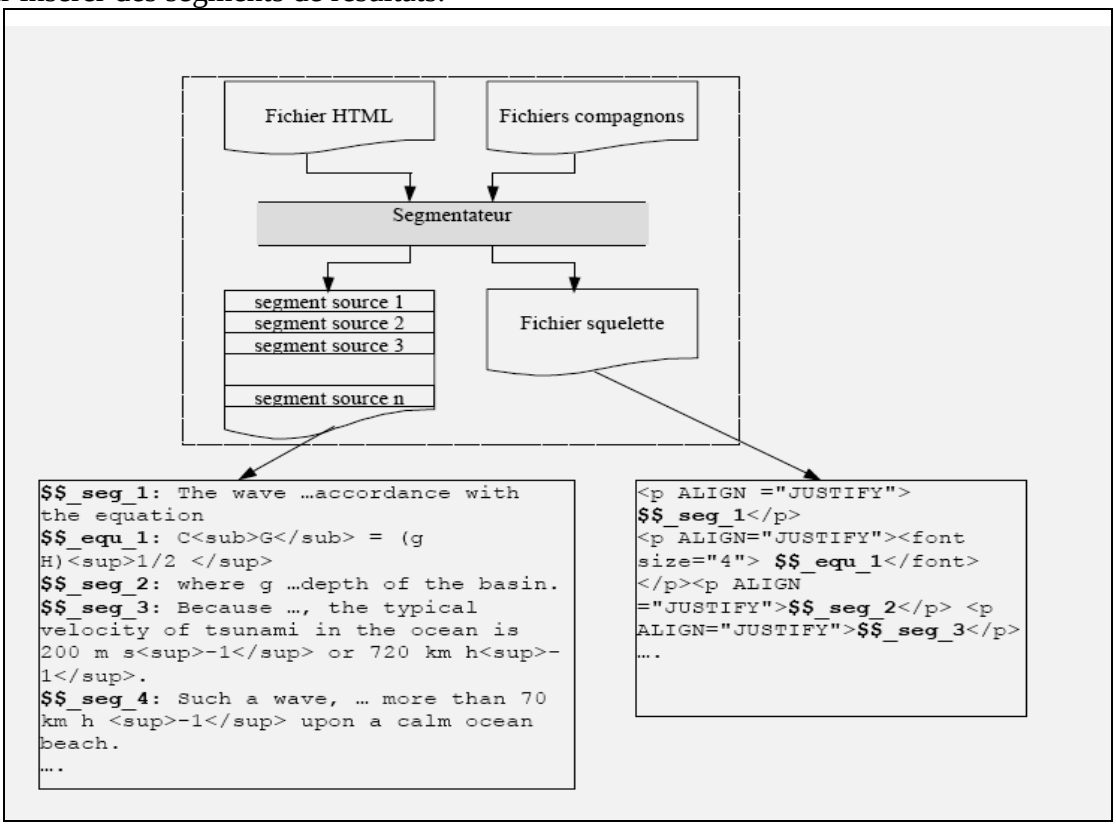


Figure 52: Segmentation basé sur des fichiers ou des segments de guidage

Un problème est que les segments dans les fichiers .unl sont normalisés et modifiés. En particulier, certaines balises de formatage, telles que <i>, <b>, etc., contenues dans les segments ont étés remplacées par des "occurrences" de type HTM1, HTM2, etc. Ces segments ont parfois été modifiés pour pouvoir être convertis en graphes UNL. Malgré ces différences, notre segmenteur a très bien fonctionné et a fourni un résultat correct pour ces 25 documents.

Nous avons aussi importé le corpus de légendes du projet DSR dans SECTra_w. Ce corpus comprend 46 documents constitués de 110 fichiers correspondant à 110 volumes. Ce corpus contient au total 11056 segments en 7 langues source : anglais, japonais, allemand, russe, suédois, italien, et français (voir les détails dans la le Table 24).

Langues / **Nombre de**	en	de	fr	ru	ja	it	sv	**Total**
documents	23	12	3	3	4	1	1	**46**
pages standard	292	149	33	18	32	17	8	**552**
segments	5840	2988	662	369	689	340	168	**11056**
mots	73000	37350	8275	4612		4250	2100	**138200**
caractères	438000	224100	49650	27675	12711	25500	12600	**791305**

Table 24: Corpus DSR

Dans ce corpus, chaque segment accompagne une image. Cette image est utilisée comme contexte pour la post-édition de la légende (voir Figure 53). Nous avons traité chaque image comme un fichier satellite qui a été aussi importé lors de l'import du corpus.

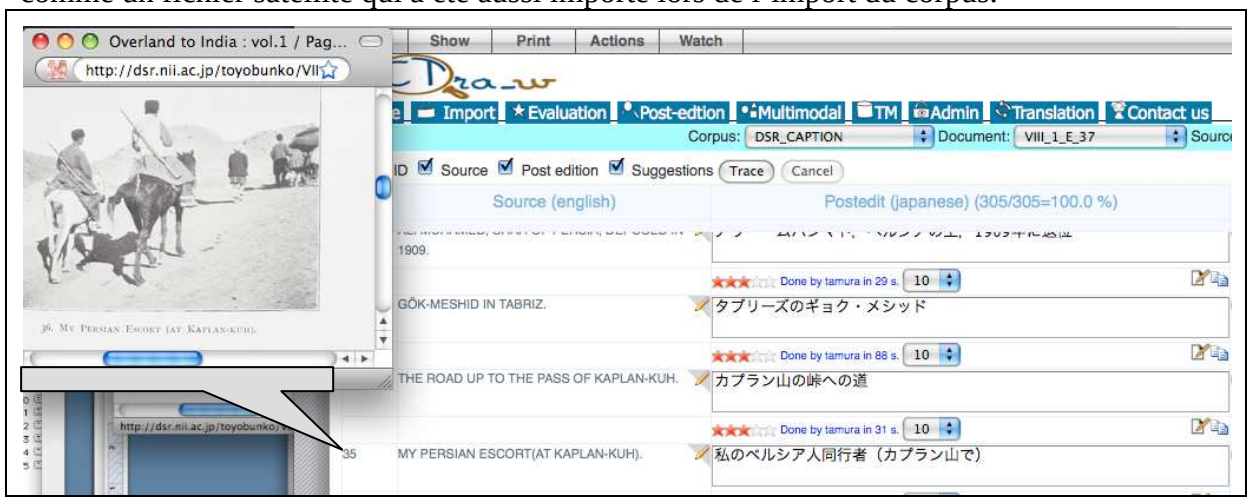


Figure 53: Relation entre le segment d'une légende et l'image associée

Avant l'utilisation de SECTra_w pour DSR, il y avait 92 volumes déjà traduits par des humains, et 18 volumes restant à traduire. Nous avons importé tous ces volumes comme 46 documents dans SECTra_w. En effet, avec les 92 volumes, on a voulu post-éditer de façon collaborative des traductions qui avaient été produites par une seule personne, et avec les 18 volumes restants, on a post-édité des pré-traductions de systèmes de TA.

II.3.3.2.2 Préparation des ressources linguistiques pour la post-édition

a. Appel de systèmes de TA

Nous avons appelé les systèmes de TA Systran et Reverso pour récupérer des pré-traductions en français pour tous les segments du corpus EOLSS. Avec la paire de langues anglais-français, le nombre de segments post-édités à partir des pré-traductions produites par Systran a été d'environ 10.000 (soit 76%), et par Reverso environ 3000 (soit 24%).

Pour les 18 volumes (2453 segments) de légendes du projet DSR, nous avons appelé les systèmes de TA ATLAS et Google Translate, pour récupérer des pré-traductions en japonais ou/et en anglais. La plupart des segments ont été post-édités à partir de pré-traductions de ATLAS, bien meilleur que GoogleTranslate.

b. Préparation d'informations lexicales

Nous avons délégué à une base lexicale PIVAX le support des informations lexicales pour la post-édition du corpus EOLSS. H-T. Nguyen [Nguyen, 2009] a traité les informations lexicales relatives au corpus EOLSS et construit trois collections d'informations lexicales, stockées dans PIVAX : entrées collectées, entrées proposées, et entrées normalisées.

- H-T. Nguyen [Nguyen, 2009] a collecté et mis dans PIVAX une masse de 120 Mo de données d'informations lexicales relatives aux mots-vedettes UW des documents EOLSS, à partir de sources gratuites disponibles telles que IATE.
- Les entrées proposées sont celles proposées directement par des post-éditeurs, ou extraites à partir de la paire <source, post-édition>.
- Les entrées normalisées sont déjà utilisées. Ces entrées sont adoptées et certifiées pour ce contexte, et sont utilisées pour construire des dictionnaires de déconversion. Elles sont affectées de poids plus élevés qu'un certain seuil (fixé par le gestionnaire du projet).

II.3.3.2.3 Création d'utilisateurs

Pour chaque projet de post-édition, SECTra_w permet de créer 3 groupes d'utilisateurs ayant des droits différents : organisateur, post-éditeurs, et réviseurs.

Nous avons créé plus de 80 post-éditeurs, 5 organisateurs, et plusieurs réviseurs dans SECTra_w pour la post-édition.

II.3.3.3 Post-édition de corpus

Environ 80 personnes ont utilisé et testé SECTra_w pour la post-édition, dont seulement 27 ont sérieusement fait de la post-édition. Précisément, dans le projet EOLSS, il y a eu 11 post-éditeurs français dans notre laboratoire, et 7 étudiants en traduction professionnelle. Dans le projet DSR, il y a eu 8 post-éditeurs japonais au Japon qui ont post-édité le corpus des légendes du site DSR. Voici le diagramme du nombre de mots source post-édités par chaque post-éditeur sur le projet EOLSS.

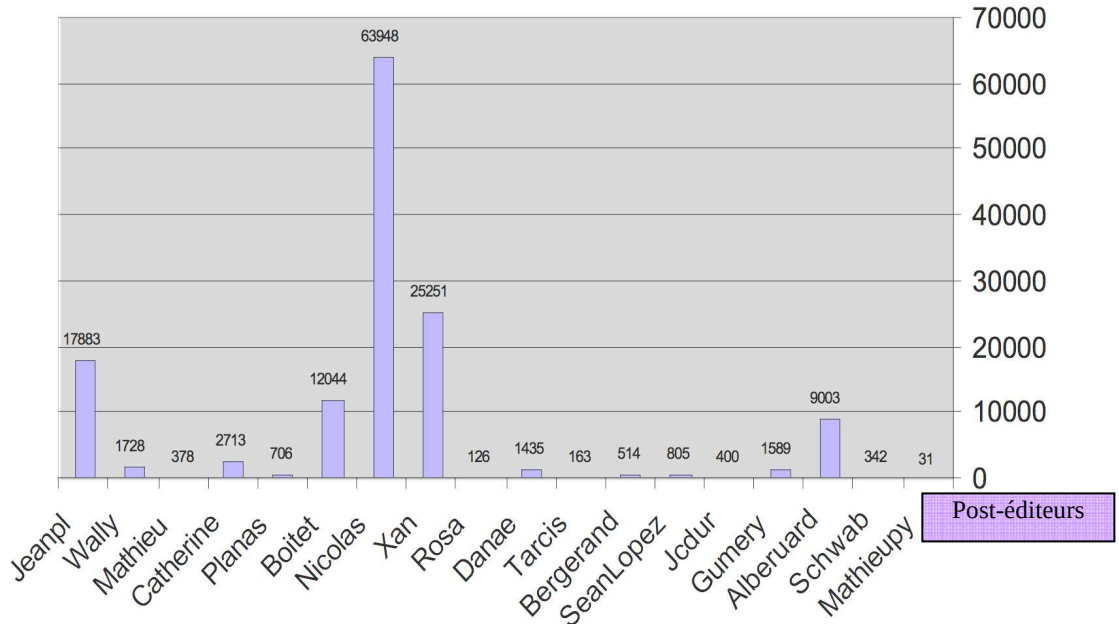


Figure 54:Nombre de mots source post-édités par les post-éditeurs dans le projet EOLSS

Pour le projet DSR, nous avons organisé la post-édition en deux étapes. Dans la première étape, la post-édition a été allouée aux post-éditeurs selon le nom de document ou/et le nombre de segments dans un document (par exemple, le post-éditeur A doit traiter 300 segments à partir du segment 100). Dans la deuxième étape, les résultats de post-édition obtenus dans la première étape ont été rendus disponibles pour la post-édition collaborative. Plusieurs post-éditeurs ont contribué à améliorer la qualité de traduction de chaque segment.

Voici le diagramme du nombre de segments et de caractères source post-édités (avec la première étape) par chaque post-éditeur sur le projet DSR (pour la langue japonaise, il est difficile de compter en nombre de mots, on compte donc en caractères).

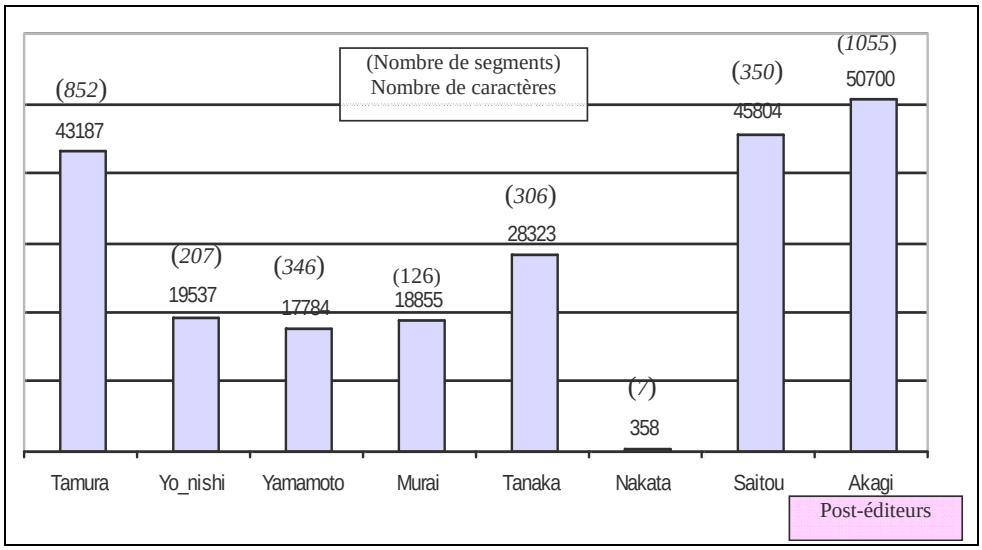


Figure 55: Nombre de segments et de caractères source post-édités par les post-éditeurs dans le projet DSR

SECTra_w fournit un tableau statistique de nombre de segments, de mots, de caractères source post-édités par chaque post-éditeur. Cependant, il est très difficile de mesurer de temps de post-édition pour chaque post-éditeur, parce que la post-édition d'un segment comprend plusieurs actions : lire le segment source, chercher des informations sur Internet, dans des dictionnaires, etc. Pendant la post-édition d'un segment, le post-éditeur peut aussi être interrompu par une autre activité, etc.

II.3.4 Évaluation

Méthodologie

Nous allons évaluer SECTra_w dans l'esprit de ce qui précède, c'est à dire essayer de mesurer dans quelle mesure les problèmes mentionnés ci-dessus ont été résolus par notre implémentation et par la façon de l'utiliser.

Evaluation du problème 2.1 : Génération automatique de formats d'import/export. Nous avons proposé le format générique *CCM* (Common Corpus Markup) pour l'import et l'export de corpus de traductions dans SECTra_w. Grâce à ce format, nous pouvons importer/exporter et gérer divers corpus dans notre système.

Nous avons développé un module permettant de facilement transformer des corpus parallèles, corpus d'évaluation, corpus de documents, corpus de phrases, le format TMX pour les mémoires de traduction en ce format.

On a également mis en œuvre l'idée d'avoir plusieurs formats équivalents, l'un XML, l'autre de type échange Excel, et le troisième encore plus simple (id: contenu EOL).

Pour l'export de corpus, l'utilisateur peut également définir un format, un codage et demander à notre système d'exporter.

Evaluation du problème 2.3 : Support des informations lexicales. Notre système a délégué à une base lexicale PIVAX, le traitement et le support des informations lexicales.

Seul la partie concernant les ressources lexicales collectées a pu être évaluée jusqu'à maintenant. De même, ce n'est qu'après les deux projets EOLSS et DSR que l'implémentation de ce qui concerne les minidictionnaires associés aux segments a été achevée. Elle n'a donc pas encore été évaluée. On peut seulement dire que les tests unitaires ont été positifs.

Evaluation du problème 2.4 : Appel des ressources extérieures. Pour l'instant, seule la délégation (totalement automatique) à PIVAX est effective. L'appel automatique (par boucle infinie) à TRADOH++ a été implémenté et devrait être mis en service et évalué dans un futur proche.

Evaluation du problème 2.5 : Aspect contributif et ouvert. L'aspect contributif est très complet, et très apprécié. Les utilisateurs l'ont plébiscité. L'aspect ouvert est encore embryonnaire, bien que certaines parties de l'API soient déjà utilisées et jugées satisfaisantes. Il s'agit surtout des appels externes de EMEU_w.

Evaluation du problème 2.6 : Sécurité des données, prévention du piratage. Pour la pollution "involontaire", nous gérons les profils des post-éditeurs, et nous nous contentons d'interdire à un contributeur ayant un niveau traductionnel plus bas de modifier ce qu'a produit un contributeur ayant un niveau traductionnel plus élevé. Notre système permet également à l'organisateur d'un projet de bloquer les modifications sur toute sélection des données de son projet.

Conclusion

Nous avons présenté dans ce chapitre nos recherches relatives aux problèmes de la construction d'un système de support à la post-édition collaborative et contributive.

Après avoir étudié le support à la post-édition et à l'accès à des dictionnaires dans les systèmes utilisés pour la traduction humaine, nous avons conclu qu'il n'existe pas de vrai système de support à la post-édition de traductions automatiques. Bien que dans ces systèmes étudiés, il y ait des supports informatiques et linguistiques à la post-édition, ceux-ci sont limités et simples.

Nous avons proposé plusieurs nouvelles notions, des principes généraux, et des solutions pour les problèmes émergents, et nous avons implémenté la plupart de ces solutions dans le système SECTra_w, à partir de spécifications précises. Ce système a été utilisé avec succès pour deux projets réels de post-édition et de traduction de corpus, EOLSS/UnescoL et DSR.

Notre système a été évalué selon plusieurs aspects :

- Gestion d'un projet : ce système permet aux organisateurs de gérer les corpus à post-éditer, la progression du projet, et la qualité des post-éditions.
- Efficacité de la post-édition : notre système permet d'accélérer la post-édition grâce à ses supports informatiques ainsi que ses supports linguistiques aux post-éditeurs.

Comme, à notre connaissance, il n'existe pas encore d'autre système de ce type, nous ne pouvons pas faire d'évaluation comparative. Nous pouvons seulement dire que :

- un contributeur expérimenté peut post-éditer l'équivalent d'une page standard (250 mots) en 15 à 20 minutes, en produisant un résultat équivalent à ce qu'il produirait en 1h par traduction manuelle traditionnelle.
- en moyenne (sur les 25 articles de EOLSS), la qualité produite par des contributeurs connaissant le domaine et postéditant vers leur langue, sans être des traducteurs professionnels, est au moins égale à celle d'un « premier jet » produit par un traducteur professionnel junior.

Le défi a donc bien été relevé.

Cependant, plusieurs problèmes liés au recyclage de traductions de bonne qualité et à la l'exploitation efficace de mémoires de traductions pour la post-édition au service d'applications novatrices comme les iMAG, Notepad++ n'ont pas encore été abordés. Ils feront l'objet du chapitre 3.

Bien que SECTra_w soit un bon environnement, il faudrait rendre plus aisée la préparation et l'organisation des projets de post-édition par les organisateurs non-informaticiens.

Chapitre III
Support informatique à l'exploitation de corpus de traductions dans des applications novatrices

Introduction

Dans les chapitres 1 et 2, nous avons présenté et résolu deux grands défis concernant le support à l'exploitation de corpus de traductions effectuées par l'humain. Plus précisément, ces deux chapitres ont porté sur les problèmes posés par la conception et la réalisation d'un système unifié, offrant sur le Web un support informatique à l'évaluation de résultats de TA et au travail humain sur des corpus variés en contexte multilingue. Cependant, il reste encore plusieurs problèmes importants à résoudre, tels que le support à la programmabilité du traitement des corpus à divers niveaux.

Dans ce chapitre, nous présenterons et résoudrons les problèmes non encore traités concernant ces deux défis, et attaquerons un autre grand défi lié au support à l'exploitation de ressources traductionnelles effectué par la machine. En effet, plusieurs applications novatrices, telles que les passerelles d'accès multilingue de bonne qualité à des sites Web élus (comme les iMAG (voir III.3.2.1)), les outils de support de la multilinguisation interne et en contexte des logiciels en source ouvert (comme Notepad++ (voir III.3.2.1)), ont besoin d'utiliser les ressources traductionnelles et de déléguer les fonctions de sectra. Les ressources traductionnelles sont les traductions post-éditées et les traductions automatiques gérées par un sectra. Les fonctions de sectra, déléguées par ces applications novatrices, sont la post-édition contributive et collaborative, la recherche et la mise à jour de traduction, etc.

Nous abordons donc plusieurs problèmes liés à la définition et la gestion d'une vraie mémoire de traductions, la normalisation d'une unité de traduction pour les appels à la TA, et la construction d'une architecture logicielle adéquate permettant la communication et l'échange entre les applications novatrices et un sectra.

Nous commençons ce chapitre par un état de l'art de la gestion des MT dans des systèmes existants, des méthodes de collecte de segments parallèles pour l'initialisation de MT, et de la programmabilité dans quelques systèmes connus. Ensuite, nous synthétisons et traitons des sous-défis, et proposons de nouvelles notions et des principes généraux de solutions. Nous expérimentons et évaluons les solutions proposées dans la spécification, et évaluons leur implémentation dans SECTra_w dans le cadre de trois projets, OMNIA, iMAG, et MIC-Notepad++.

III.1 État de l'art et problèmes émergents

III.1.1 État de l'art

III.1.1.1 Gestion de mémoires de traductions (MT) dans des systèmes existants

Liaison ou non avec les documents source. Il y a "liaison avec les documents" quand la MT pour deux langues est de forme {segment, références à ses occurrences, traduction}. Un segment apparaît autant de fois qu'il a de traductions, et on peut en principe reconstituer le texte d'un document (en langue source comme dans les langues cible) à partir de la MT.

Nous disons "lien avec le document source" car l'alignement entre source et cible peut ne pas être parfait au niveau des segments (N segments source → M segments cible).

Nous avons étudié de ce point de vue quelques systèmes d'aide au traducteur. Très peu ont cette liaison, ce qui empêche de proposer les traductions existantes dans un ordre de préférence reflétant leur adéquation a priori au contexte en cours.

Systèmes d'AT	Liaison avec les documents source ?	
	Oui	**Non**
TM/2 (IBM)	✓	
Trados (SDL)		✓
DéjàVu (ATRIL)		✓
Transit (STAR)	✓	
XTM (XML-INTL)		✓
Similis (Lingua & Machina)	✓	
EURAMIS (EC)		✓
XL8 (Globalware)		✓

Table 25: Systèmes d'aide au traducteur avec ou sans liaison avec les documents source

Il est un peu étonnant de voir qu'IBM a introduit cette liaison dès la conception de son produit TM/1, et que la plupart des éditeurs d'outils sur PC ne l'ont pas suivi, sauf assez récemment *Lingua & Machina* pour Similis. Le cas d'*EURAMIS* [Theologitis, 1997] est à part : c'est une très grosse base de données (plusieurs téraoctets) qui contient tous les renseignements possibles. Mais, dans la pratique de la traduction à la CEE (Commission Economique Européenne), on en extrait une MT non contextuelle de taille réduite, pour chaque document, en format acceptable par Trados.

Gestion explicite des contextes de traduction. Nous avons vu plus haut (§II.2.1.2) que la notion de contexte va bien plus loin que la simple position d'un segment source dans un ou plusieurs documents.

Quelques systèmes permettent une organisation en domaines et sous-domaines, ou types de documents (par exemple, Trados (SDL), DéjàVu (ATRIL), les MT dans BEYTrans, etc.), mais rien d'aussi précis que ce qui nous semble nécessaire (§II.2.1.2).

Prise en compte de la multiplicité possible des langues source et du diagramme de traduction de chaque cellule "cible". Aucun système ne le fait. Le diagramme de traduction (disant par quelle(s) langue(s) on est passé pour traduire vers telle langue) n'est jamais indiqué.

Pourquoi nous semble-t-il utile et même nécessaire de le faire?

- existence de sites Web développés à la fois dans plusieurs langues : il faut initialiser la MT en conséquence, et par exemple ne pas traduire du français vers l'allemand en passant par l'anglais (comme le fait Google) alors qu'une version en anglais de bonne qualité est disponible.

- nécessité politique: par exemple, de nombreuses organisations internationales ont plusieurs langues officielles, qui doivent "faire foi", mais souhaitent aussi être accessibles dans d'autres langues (UNESCO avec 6 langues, UIT avec 4 langues).

Gestion fine de la qualité des traductions brutes, des post-éditions et des révisions. À notre connaissance, aucun système ne propose quoi que ce soit. Or, dans un contexte participatif, cela nous semble nécessaire.

Nous présentons plus bas une solution assez simple et complète, qui permet aussi bien l'autoévaluation que l'évaluation externe.

Gestion de l'historique et des métadonnées associées aux versions des cellules. À notre connaissance, aucun système ne propose de montrer l'historique de la traduction d'un segment, ni les métadonnées associées (qui a fait quoi, en combien de temps, avec quel taux de modification…).

Or, dans un contexte où il peut y avoir des contributeurs spontanés, occasionnels et souvent bénévoles, aussi bien que des contributeurs organisés en projets et souvent rétribués, il nous semble nécessaire de le faire, et cela au niveau des cellules des segments, et pas seulement globalement au niveau d'un segment ou d'un document.

Conclusion

Les systèmes actuels de gestion de MT sont donc très limités, essentiellement parce que la notion même de MT qu'ils utilisent est trop réductrice, voire simpliste.

Il nous semble impératif de lever les limites notées ci-dessus. Tous les aspects sont importants, soit pour la conception même d'un sectra, soit d'un point de vue fonctionnel.

Le tableau suivant montre cette distinction.

Fonctionnalité	Aspect conceptuel	Aspect fonctionnel
Liaison ou non avec les documents source	••	•
Gestion explicite des contextes de traduction	•••	••
Prise en compte de la multiplicité possible des langues source et du diagramme de traduction	•••	••
Gestion fine de la qualité des traductions brutes, des post-éditions et des révisions	•	•••
Gestion de l'historique et des métadonnées associées aux versions des cellules	••	•

Table 26: Aspects importants pour le traitement des MT

III.1.1.2 Exploitation et mesures de sites Web multilingues pour l'initialisation de MT dédiées

Nous visons à permettre à des passerelles iMAG[37] (interactive Multilingual Access Gateway / Passerelle Interactive d'Accès Multilingue) d'exploiter des MT d'un système de gestion de corpus de traductions comme SECTra_w afin de traduire des sites Web *élus*. Dans plusieurs cas, les sites Web *élus* contiennent des pages Web dont le contenu a déjà été traduit partiellement en une certaine langue.

Il s'agit donc de permettre à l'iMAG de traduire ces pages complètement en cette langue avec la réutilisation des traductions de bonne qualité existant dans ces pages. Pour ce faire, nous devrons exploiter et mesurer ces sites Web pour l'initialisation de MT *dédiées*. Nous étudions donc les méthodes existantes permettant de le faire.

[37] http://eolss.imag.fr/xwiki/bin/view/imag/home

Méthodes d'extraction de segments multilingues à partir de textes et de leurs traductions. Une méthode efficace a été introduite par [Koehn, 2005] pour l'exploitation du corpus EuroParl [Koehn, 2005][38], un corpus bilingue aligné. Il l'exploite ensuite pour la traduction automatique. Il distingue au total cinq étapes: la collecte des données brutes, l'alignement des documents, la segmentation en phrases, la normalisation du codage et de l'encodage, l'alignement des phrases. Le point le plus important est l'alignement au niveau des phrases.

Cette méthode a permis à Ph. Koehn de construire avec succès le corpus parallèle EuroParl. Cependant, pour les sites Web, le problème est qu'on n'est pratiquement jamais sûr qu'il s'agisse de bitextes.

Méthodes d'extraction de bi-chunks à partir de corpus comparables. Cette méthode a été présentée par [Munteanu et Marcu, 2006] pour l'extraction de fragments parallèles à partir de bi-phrases provenant de corpus comparables.

Définition III-1. Nous appellerons *fragment* toute sous-chaîne [connexe] d'un segment (source ou cible), et *bi-fragment* tout couple de fragments en relation (potentielle) de traduction mutuelle.

Exemple: < "Let us know how to use X", "Apprenons à nous servir de X" >.

Ces méthodes généralisent les méthodes d'extraction de termes bilingues, sachant qu'on traite des corpus dans lesquels seule une toute petite minorité des segments source ont effectivement une traduction dans la langue cible considérée.

D'autre part, et un peu à cause de cette situation, la relation de traduction entre les fragments (chunks) source et cible des bifragments extraits peut être raisonnablement considérée comme symétrique.

Remarque: disposer d'une grande quantité de fragments bilingues ou mutilingues, si possible contextualisés (avec la même définition que pour les segments), serait très utile pour l'aide à la traduction et pour construire de meilleurs systèmes de TA.

Mais, pour l'instant, aucun système de MT ne gère ce genre de ressources, dont la nature est intermédiaire entre un dictionnaire et un corpus. Cette voie ne nous est apparue clairement que début 2008, à l'occasion de la rédaction d'une proposition à l'ANR. Elle est donc hors du cadre de cette thèse, mais nous prévoyons de l'explorer dès que possible.

Méthode d'extraction de MT à partir de corpus Web comparables contenant des traductions exactes. Cette méthode a été présentée par [Do et al., 2009] pour récupérer des phrases parallèles à partir de sites Web comparables. Elle est en deux étapes. La première consiste à identifier automatiquement des paires de pages Web parallèles en téléchargeant les pages Web à exploiter, puis en filtrant des pages parallèles selon plusieurs critères tels que la date de publication, certains mots spéciaux, ou le nombre de segments. La deuxième consiste à aligner des phrases. Après avoir segmenté deux pages Web parallèles en phrases, on aligne les phrases en utilisant plusieurs méthodes différentes, telles que la comparaison de la longueur des phrases [Brown et al., 1991], des informations lexicales [Kay et Roscheisen, 1993], des approches statistiques [Gale et Church, 1991], et l'utilisation d'un logiciel appelé Champollion.

[38] P. Koehn segmente les textes dans chaque langue puis les aligne 2 à 2 de façon à construire un corpus de segments parallèles pour chaque couple de langues.

Conclusion

Parmi les travaux présentés ci-dessus, nous nous intéressons au travail d'extraction de MT à partir de corpus Web comparables contenant des traductions exactes présenté par [Do et al., 2009]. Ce travail est proche de ce que nous désirons faire.

III.1.1.3 Programmabilité dans les systèmes de gestion de corpus

Ce qui suit peut être étendu à tous les types de corpus et pas seulement aux corpus de traductions.

Programmabilité dans Ariane-G5. Cet environnement de développement et d'exploitation de systèmes de TA a été présenté plus haut (§I.1.1.1).

En Ariane-G5 [Boitet, 1993b], l'utilisateur a seulement quelques possibilités très simples de programmation. Il s'agit plus de paramétrisation que de vraie programmation. On peut :

- définir ou redéfinir la liste des occurrences associée à un corpus, et utilisée pour produire la structure hiérarchique externe de chaque texte du corpus ;
- modifier l'intervalle [min..max] utilisé pour segmenter chaque texte en unités de traduction (§I.1.1.1.2) ;
- définir une sélection arbitraire de textes (appartenant à un ou plusieurs corpus), et lui appliquer une chaîne d'exécution ou une chaîne de production.

On peut aussi considérer que la définition d'une chaîne d'exécution ou de production est une sorte de programmation : on choisit un "chemin" dans le graphe des phases d'Ariane-G5, une "variante" pour chaque phase [Boitet, 1990 ; Nguyen, 2009], et des valeurs pour les paramètres contrôlant les types de trace (suivis d'exécution) et des sorties intermédiaires (arbres représentant les unités de transduction entre 2 phases). Il s'agit en fait plus de configuration, ou de paramétrage, que de programmation au sens usuel.

Programmabilité dans NooJ. NooJ [Silbersztein, 2004] est un environnement de développement linguistique muni d'un LSPL permettant d'écrire des transducteurs d'états finis sophistiqués, des expressions régulières à la Perl, des grammaires hors-contexte, etc., et de les appliquer à des corpus. Il permet de gérer les corpus de façon efficace, mais minimale (limitée au but recherché).

Beaucoup d'applications ont été écrites en NooJ (analyse morphologique, morphologique et syntaxique de nombreuses langues, extracteurs d'entités nommées, de collocations, calcul de concordances, etc.). Comme NooJ a été conçu pour une utilisation sur PC et pas sur le Web, il est installé à de nombreux endroits. Son auteur (Max Silbersztein) organise un ou deux colloques ou ateliers sur NooJ chaque année, et collecte sur son site tous les linguiciels et les corpus préparés par ses collègues, dans l'esprit du libre, et les rend disponibles.

NooJ fournit des commandes de traitement de corpus qui peuvent être utilisées directement par les utilisateurs dans la ligne de commande de Windows, ou d'Unix/Linux. Ces commandes sont de 3 types:

- information sur les corpus (liste des corpus, métadonnées, liste des noms des textes, liste des textes et de leurs différentes formes, etc.), comme dans Ariane-G5 ;
- import et export de corpus (ou de parties de corpus), avec choix des formats (inexistant en Ariane-G5) ;
- exécution d'une (suite d')applications linguistiques sur un ou plusieurs corpus, ou sur une sélection arbitraire de textes de même format d'entrée.

Conclusion

Bien que ces environnements fournissent un support à la programmabilité permettant de gérer et de traiter les corpus, ce support est encore assez simple. L'environnement Ariane-G5 permet seulement quelques possibilités très simples de programmation. Il s'agit plus de paramétrisation que de vraie programmation. NooJ fournit principalement ce support pour les corpus monolingues, mais pas pour les corpus de traduction.

Un sectra programmable devrait permettre d'effectuer non seulement autant d'opérations qu'Ariane-G5 et que NooJ, mais aussi des opérations motivées par l'utilisation dans des campagnes d'évaluation (par exemple, programmer certaines "mesures d'évaluation", comme de nouvelles variantes de BLEU, WER, HTER), ou les opérations motivées par l'utilisation dans des projets de post-édition (par exemple, appel à un ou plusieurs segmenteurs externes, appel en continu à des systèmes de TA distants, etc.).

III.1.2 Synthèse des objectifs et des problèmes

Pour fournir un bon support informatique à l'exploitation de ressources traductionnelles dans un sectra, ce qui précède montre qu'il faut traiter les points suivants.

Segmentation générique, multiple et récursive. Les documents (ou les pages Web) importés dans un sectra doivent être d'abord segmentés de façon générique, multiple et récursive. Nous étudions ce problème et proposons des solutions au §III.2.1.1.

Normalisation pour les appels à la TA. Le format d'une unité de traduction varie d'un système de TA à l'autre. La qualité de traduction d'un système de TA est donc aussi influencée par le format des segments qu'on lui fournit. Il faut donc les normaliser en fonction de chaque système de TA. Nous proposons des solutions à ce problème au §III.2.1.2.

Définition et gestion d'une vraie « mémoire de traductions ». Il faut organiser les MT en segments multilingualisés et contextualisés pour pouvoir proposer les traductions disponibles dans un ordre de préférence dépendant du contexte de l'occurrence à traduire. Nous étudions ce problème et proposons des solutions au §III.2.2.1.

Programmabilité du traitement des corpus. Un système de support informatique à l'exploitation de corpus de traductions doit être programmable à différents niveaux, en fonction des types d'utilisateurs. Nous étudions ce problème et proposons des solutions au §III.2.2.2.

Traitement de masses de données. Il faut pouvoir traiter de grosses masses de données, car non seulement les documents peuvent être gros, mais les mémoires de traductions elles-mêmes doivent être postéditables (et traduisibles dans de nouvelles langues) comme des documents, et elles contiennent souvent des millions de segments. Les problèmes concernent non seulement l'édition, la visualisation et la navigation dans une grande masse de données, mais aussi la préparation des ressources linguistiques (prétraductions, minidictionnaires, etc.) pour tous les segments de cette masse de données. Nous étudions ce problème et proposons des solutions au §III.2.3.1.

Architecture par agents. Un sectra doit déléguer certaines tâches à d'autres systèmes. Inversement, ces systèmes peuvent demander des services au sectra, pour l'exploitation de ressources traductionnelles et la post-édition. Dans ce cas, le sectra et les systèmes extérieurs fonctionnent en tant que serveurs ainsi que de clients. C'est pourquoi il nous a semblé préférable d'utiliser une architecture par agents plutôt qu'une architecture en client-serveur. Nous étudions et proposons des solutions pour construire cette architecture au §III.2.3.2.

Résumé

Les problèmes présentés ci-dessus peuvent être classés selon l'importance relative de leurs aspects conceptuels, algorithmiques, et programmatoires dans le tableau suivant :

Problèmes	Conceptuels	Informatique	Génie Logiciel
Segmentation générique, multiple et récursive	•••	•••	•
Normalisation pour les appels à la TA	•••	••	•
Définition et gestion d'une vraie « mémoire de traductions » (MT)	•••	•••	•
Programmabilité du traitement des corpus	•••	••••	•••
Traitement de masses de données	••	••••	•••
Architecture par agents	••	••	•••

Table 27: Aspects des problèmes liés aux utilisations novatrices d'un sectra

III.1.3 Notions unificatrices émergentes et principes généraux

III.1.3.1 Notions

III.1.3.1.1 Segmentation multiple

a. Motivations

La notion d'unité de traduction varie d'un système de TA à l'autre.

Dans certains cas, le système n'arrive pas à segmenter un paragraphe correctement, et une unité de traduction se trouve constituée de 2 ou 3 phrases. Nous dirons alors qu'il traite un *supersegment*.

Certains systèmes comme ceux écrits en Ariane-G5, en SYGMART ou en XIP sont construits de façon à traiter des unités de traduction les plus grandes possibles (compte tenu d'éventuelles limites d'implémentation). Ainsi, en Ariane-G5, une unité de traduction contient la plupart du temps quelques paragraphes (un seul arbre pour 300 à 500 mots), en SYGMART, on va jusqu'à une trentaine de pages, et en XIP on peut traiter un document XML de taille quelconque.

Dans d'autre cas, la segmentation est trop fine. Par exemple, le système traite chaque élément d'une liste à puces comme une unité de traduction. Nous dirons alors qu'il traite des *infrasegments*.

Comme un sectra de traductions doit pouvoir appeler différents systèmes de TA sur les mêmes documents, il faut qu'un document puisse être segmenté de façon multiple.

b. Supersegment

Définition III-2. Un supersegment est un morceau de texte qui contient plusieurs segments.

c. Infrasegment

Définition III-3. Un *infrasegment* est une portion d'un segment.

Par exemple, il peut s'agit d'un élément d'une liste à puces, ou de morceaux d'une phrase séparés par une formule ou une icône.

III.1.3.1.2 Segmentation récursive

Un segment peu contenir un sous-document.

Par exemple, un segment peut contenir un lien (<a href = "…") dont l'attribut TITLE contient un ou plusieurs paragraphes.

Ou encore, un segment en français peut contenir une citation dans une autre langue, composée d'une ou de plusieurs phrases, c'est-à-dire d'un ou de plusieurs segments, tout à fait "naturels".

Nous définissons donc les notions de *sous-document* et de *sous-segment.*

a. *Sous-document*

Définition III-4. Un *sous-document* est un document contenu dans un autre. Il peut aussi être contenu dans un segment.

Exemple:

Les thèmes scientifiques sont les infrastructures informatiques, l'interaction,etc.

Sous-document

b. *Sous-segment*

Définition III-5. Un *sous-segment* est un segment d'un sous-document.

Attention : un sous-segment n'est donc pas un fragment.

III.1.3.1.3 Métasegments et métadocuments

Dans un livre de phrases ou dans un fichier de messages de programmes, un segment peut contenir des variables. Chaque variable contient une valeur de classe de mots/termes.

Par exemple, dans le cas du corpus *Survitra*, une variable peut être signalée dans le segment par des crochets *[ ]* qui contiennent au choix :

- une simple liste lexicale (liste de mots/locutions) :

Exemple. *Avez-vous du vin [rouge / blanc / rosé] ?*

- l'identifiant d'une classe lexicale (liste nommée dont les éléments sont des mots ou des locutions), qui est détaillée ailleurs dans le corpus.

Exemple. *Je voudrais un verre de [$drinkGlass]*.

Définition III-6. Une *classe lexicale* (de mots ou locutions) est une liste nommée créée par ailleurs dans le corpus; une des valeurs de cette liste viendra instancier chaque occurrence de la variable lexicale correspondante, notée $nom_de_la_classe.

Dans la traduction, on affecte souvent des éléments (par exemple, son auteur) à un segment afin de le traiter ou d'évaluer sa qualité plus facilement.

III.1.3.1.4 Corpus contenant des relations entre textes, images et sons

On rencontre parfois des corpus contenant des textes accompagnés d'images, de vidéos, et de sons. Les textes dans ces corpus sont souvent utilisés pour décrire ou expliquer ces éléments

non textuels, et symétriquement ces éléments non textuels sont parfois utilisés comme contextes pour mieux comprendre les textes.

Par exemple, pour la traduction des légendes des corpus DSR ou OMNIA, on a besoin de voir aussi les images associées aux légendes. Ou encore, pour mieux comprendre un dialogue parlé dans le corpus ERIM, on a besoin de voir une transcription de ce dialogue.

III.1.3.2 Principes

III.1.3.2.1 Architecture par agents

On a dégagé un principe d'organisation par "agents" d'un système d'exploitation de corpus capable de relever ce défi.

SECTra_w devra donc être construit par des agents à gros grain qui marient autonomie et coopération avec d'autres agents. Ces agents sont mis en œuvre généralement avec des contraintes sur le temps. Les agents possèdent chacun des compétences, des informations ou des ressources limitées, mais, en regroupant leurs capacités, ils sont capables de résoudre des problèmes qu'un seul agent ne pourrait pas résoudre.

Les agents doivent être capables de communiquer, de synchroniser des données, et de coopérer l'un avec l'autre.

III.1.3.2.2 Gestion des corpus de TA dans les futurs systèmes de TAO hétérogène

a. Intégration de SECTra_w à Ariane-Y

Ariane-Y est la réingénierie du système Ariane-G5 vers un système plus moderne, plus portable, plus ouvert.

Dans l'environnement Ariane-G5, la post-édition et la révision de résultats de TA ne permettent en général de traiter ensemble qu'un ou plusieurs paragraphes, voire pages. Il n'y a pas de gestion de versions: la dernière modification est prise en compte. Il n'y a pas non plus de réinsertion des éléments hors-texte, pour la bonne raison qu'on n'avait pas à l'époque de l'implémentation d'Ariane-G5 de moyens de saisir en machine des images, des formules chimiques, ou même des formules mathématiques.

Nous désirons donc pouvoir intégrer notre sectra à Ariane-Y en tant que composant (agent) de gestion de corpus pour la TA, qu'il s'agisse de suites de test, de textes, ou de documents.

b. Délégation et idée de « serveur corporal »

Nous voulons permettre à Ariane-Y de déléguer des fonctions de gestion, et des interfaces graphiques à SECTra_w. Cette délégation laisse toutes les fonctions de contribution de ressources à un environnement collaboratif sur le Web.

III.2 Problèmes liés à ce défi

III.2.1 Problèmes à dominante conceptuelle

III.2.1.1 Problème 3.1 : Segmentation générique, multiple et récursive

Un sectra devra permettre la post-édition incrémentale de corpus de documents (des pages html, des fichiers textuels, etc.). Il s'agira donc de segmenter les documents en unités de traduction, puis de les soumettre à divers systèmes de TA. Cependant, la segmentation varie selon la langue, et selon le système de TA utilsé. Il faut alors une segmentation générique, multiple et récursive.

III.2.1.1.1 Problème et état de l'art

a. Comment effectuer la normalisation initiale en Unicode-UTF-8?

Il faut d'abord une normalisation initiale à laquelle se "raccrocheront" toutes les segmentations. On veut de l'UTF-8, et on veut remplacer les entités html (ex: "é") par leurs valeurs en Unicode (donc "é" ici). Pour cela, il faut identifier les passages du document initial homogènes en langue et en codage.

a.i Langues

On rencontre assez souvent des textes dont les segments sont dans des langues différentes, par exemple, des paragraphes en français dans un texte anglais. On a parfois des balises de changement de langue (par exemple en Word, mais seulement si le créateur du document a pris soin d'ajuster l'attribut de langue sur les différents fragments), mais pas toujours. En html, on n'en trouve le plus souvent une qu'au début du document.

Si l'on demande la traduction d'un tel texte en français, il faudra laisser les segments en français tels quels. Si l'on en demande la traduction en italien, il faudra soumettre les segments en anglais à des traducteurs anglais-italien, et les autres à des traducteurs français-italien.

a.ii Codages

Au niveau du texte original. Il faut reconnaître non seulement les langues utilisées, mais les codages des segments et des fragments écrits dans différentes langues. Cela n'est pas évident: pour le vietnamien par exemple, il y a plus de 40 codages sur un octet, et il y a 2 plages dans Unicode. On peut aussi trouver des pages Web dont les cadres (frames) utilisent différents codages.

Au niveau de l'appel à des systèmes de TA. Ce point concerne plutôt le problème 3.2, mais il est plus naturel de le traiter ici.

Il faut appeler les systèmes en codant les textes soumis dans des codages qu'ils acceptent. Nous avons décidé d'utiliser Unicode (UTF-8) pour coder toutes les chaînes à l'intérieur de SECTra_w, mais de nombreux systèmes de TA ne traitent le japonais qu'en JIS, JES, ou EUC-J, et le chinois qu'en GB2312-80 (en RPC), ou BigO-5 (à Taiwan).

Il faudra donc convertir les segments envoyés aux systèmes de TA dans les codages qu'ils attendent. Il ne s'agit pas d'un problème difficile, mais il faut en tenir compte.

b. Comment définir de façon générique une "segmentation multiple et récursive"

La segmentation d'un texte vise à maximiser la qualité de la traduction obtenue. En effet, la qualité de traduction est souvent influencée par la taille et le format de l'unité de traduction soumise au système de TA. Par exemple, Systran peut générer une bonne traduction d'une unité de traduction contenant des balises HTML et une très mauvaise s'il traduit indépendamment les fragments entre les balises. Dans certains systèmes comme ceux écrits en Ariane-G5, en SYGMART ou en XIP, une unité de traduction contient quelques paragraphes, des pages, ou voire même un document XML de taille quelconque.

Il s'agit donc d'une *segmentation multiple* qui permet de segmenter un texte en unités de traductions différentes selon les différents systèmes de TA, et il s'agit aussi d'une *segmentation récursive* qui permet de segmenter les sous-documents contenus dans les segments.

c. Comment la gérer?

c.i Représentation

Comment représenter la segmentation multiple ? On a le choix entre représenter explicitement le graphe des segmentations, et le représenter implicitement, par des liens entre segments (et infrasegments, et supersegments).

c.ii Opérations possibles

Il faut aussi permettre d'effectuer des opérations sur les segmentations.

Ajouter une nouvelle segmentation d'un document à une segmentation multiple. C'est nécessaire si l'on utilise plusieurs segmenteurs.

Corriger localement une segmentation (fusionner 2 segments ou diviser 1 segment)**,** comme ce que fait V. Satayamas pour la langue thaï avec son outil AnnotEd-W [Satayamas et al., 2007].

AnnotEd-W est un éditeur Web permettant de corriger la segmentation, en proposant des possibilités de segmenter un texte en mots sous forme des graphes de mots, et en permettant de vérifier et de choisir la meilleure segmentation.

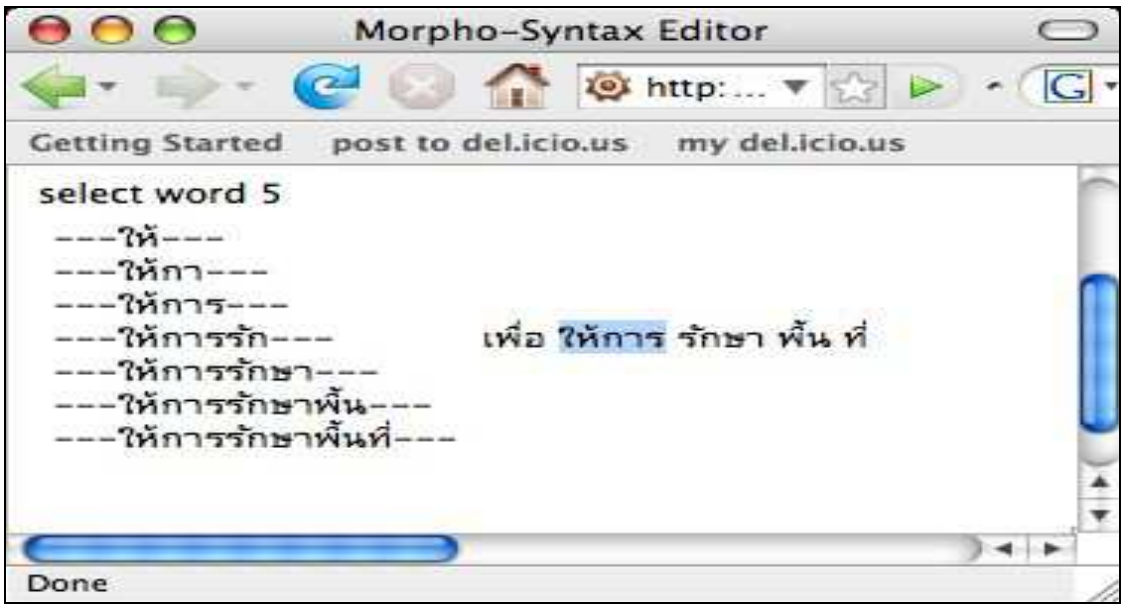


Figure 56: Interface de AnnotEd-W

III.2.1.1.2 Solutions possibles

a. Définition générique

Comme nous l'avons indiqué ci-dessus, l'unité de traduction soumise aux systèmes de TA est différente de l'un à l'autre au point de vue de la taille et du format. Avec la même page Web, la segmentation peut donc devoir être différente selon le système de TA à appeler.

Voici un exemple montrant qu'un texte html peut être segmenté de deux façons différentes pour être soumis à deux systèmes de TA.

```
<p>
The wave propagates from the
source with a velocity of long
gravity water waves in
accordance with the equation.
</p>
<p align="justify">where g is
the acceleration due to gravity,
and H is the depth of the basin.
Because the average depth of the
world ocean is 4 km, the typical
velocity of tsunami in the ocean
is 200 m s<sup>-1</sup> or 720
km h<sup>-1</sup>.
</p>
<p align="justify">Such a wave,
propagating with the velocity of
an airplane, may traverse the
Pacific ocean in 10-12 hours and
bring down a wall of water 10m
high with a velocity of more
than 70 km h<sup>-1</sup> upon a
calm ocean beach. The wave
velocity is decreased near the
coastline because of shallower
water and the slowing of the
wave by the roughness of the
bottom.</p>
```

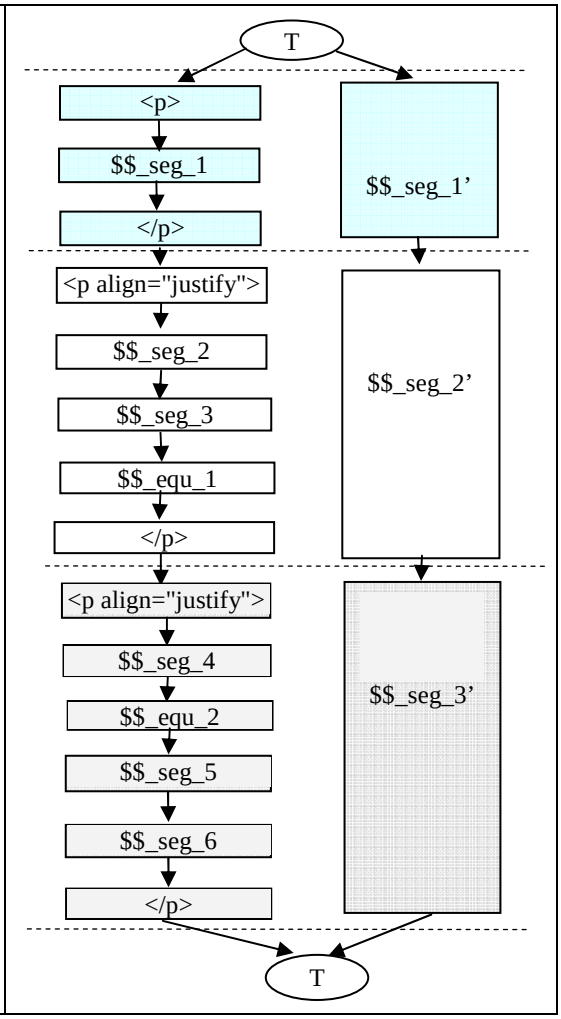


Figure 57: Deux façons différentes de segmenter un même texte

Avec:

$$_seg_1 = The wave propagates from the source with a velocity of long gravity water waves in accordance with the equation.

$$_seg_2 = where g is the acceleration due to gravity, and H is the depth of the basin.

$$_seg_3 = Because the average depth of the world ocean is 4 km, the typical velocity of tsunami in the ocean is

$$_equ_1 = 200 m s⁻¹ or 720 km h⁻¹.

$$_seg_4 = Such a wave, propagating with the velocity of an airplane, may traverse the Pacific ocean in 10-12 hours and bring down a wall of water 10m high with a velocity of more than

$$_equ_2 = 70 km h⁻¹

$$_seg_5 = upon a calm ocean beach.

$$_seg_6 = The wave velocity is decreased near the coastline because of shallower water and the slowing of the wave by the roughness of the bottom.

$$_seg_1' =' <p>' + $$_seg_1+' </p>'

$$_seg_2' ='<p align="justify">' + $$_seg_2 + $$_seg_3 + $$_equ_1 +' </p>'

$$_seg_3' = '<p align="justify">' + $$_seg_4 + $$_equ_2 + $$_seg_5 + $$_seg_6 +' </p>'

Un *segmenteur multiple* doit donc permettre de décrire des diagrammes guidant la segmentation. C'est dire que les entrées du *segmenteur multiple* doivent contenir non seulement le fichier à segmenter, mais aussi un fichier décrivant la segmentation ou les règles de segmentation, et un fichier décrivant la façon de normaliser les hors-texte (règles de construction des occurrences spéciales qui les remplcacent lors de l'appel à tel ou tel système de TA). Les hors-texte sont des éléments particuliers ayant un rôle linguistique dans les segments, comme les formules, les marques déposées, etc.

La description d'une segmentation multiple consiste en :

- un graphe des segmentations.
- un dictionnaire des segments de forme : nom, occurrence dans le fichier (origine et longueur), valeur normalisée.
- un dictionnaire des hors-texte de forme : occurrence spéciale, valeur. On référence à la valeur (cas d'images ou le sons).
- les valeurs des hors-texte non textuels sous forme d'un répertoire.

Voici l'exemple des fichiers d'entrée du *segmenteur multiple* pour la segmentation du texte dans l'exemple ci-dessus. Le premier est le fichier décrivant la segmentation, et le deuxième est le fichier contenant des hors-textes.

```
<p>
$_The wave propagates from the source with a velocity of long gravity
water waves in accordance with the equation. _$</p>
<p align="justify">$_ where g is the acceleration due to gravity, and H
is the depth of the basin._$ $_Because the average depth of the world
ocean is 4 km, the typical velocity of tsunami in the ocean i _$
$$_equ_1.
</p>

<p align="justify"> $_Such a wave, propagating with the velocity of an
airplane, may traverse the Pacific ocean in 10-12 hours and bring down a
wall of water 10m high with a velocity of more than _$ $$_equ_2 $_upon a
calm ocean beach_$. $_The wave velocity is decreased near the coastline
because of shallower water and the slowing of the wave by the roughness
of the bottom._$</p>
```

Figure 58: Fichier décrivant la segmentation du texte

Dans cet exemple, on utilise les chaînes de caractères ***$_*** et ***_$*** pour indiquer respectivement le début et la fin d'un segment.

Voici le fichier contenant les hors-texte correspondants.

```
$$_equ_1: 200 m s<sup>-1</sup> or 720 km h<sup>-1</sup>
$$_equ_2: 70 km h<sup>-1</sup>
```

Figure 59: Fichier de hors-texte

Grâce à ces fichiers, un *segmenteur multiple* peut segmenter un texte de plusieurs façons différentes. Chaque segmentation différente donnera un fichier squelette différent.

III.2.1.2 Problème 3.2 : Normalisation pour les appels à la TA

III.2.1.2.1 Problèmes

a. Variété des formats attendus par les systèmes de TA

Les différents systèmes de TA acceptent les textes à traduire dans des formats et des codages variés (xml, rtf, pdf et ASCII, UTF-8, EUC, JIS, etc). (voir Table 28).

Certains, comme Systran, publient des conventions ("flux XML") [Systran, 2009] permettant d'indiquer divers paramètres de traduction (fragment à ne pas traduire, nom propre, etc.). D'autres, comme les systèmes SMT générés par Pharaoh [Koehn, 2009], Moses [Moses, 2009] ou Joshua [Zhifei et al., 2009], attendent des segments normalisés d'une façon précise.

b. Balises de formatage et hors-texte

b.i Représentation dans les segments soumis à la TA

Comment traiter les balises de formatage, et les autres éléments non textuels (figures, icônes, formules…), souvent appelés "hors-texte" ?

La solution générale est de les remplacer par des "occurrences spéciales". Dans certains cas, comme pour les formules, qui peuvent avoir un rôle linguistique dans un segment, cette solution semble nécessaire. Dans d'autres cas, s'il ne s'agit que de présentation (italiques, gras) ou d'informations auxiliaires (comme une entrée d'index), il semble préférable de supprimer un hors-texte, et de le réinsérer dans la traduction "à sa place", ce qui impose d'aligner le texte de sortie avec le texte d'entrée.

b.ii Traitement au retour de la TA

Si l'on utilise la solution générale ci-dessus, comment réinsérer les balises et hors-texte aux endroits convenables dans le résultat brut de TA? Ce n'est facile, car, par exemple, un mot peut être traduit par deux mots non connexes (par exemple, un verbe français en italiques peut se traduire par un verbe avec particule séparable en allemand. Comment mettre les italiques sur les bons mots en allemand ?

Exemple: je lui *rends* son argent —> ich *gebe* ihm sein Geld *zurück.*

III.2.1.2.2 Etat de l'art

a. Variété des formats attendus par les systèmes de TA

Voici ci-dessous un tableau montrant la variété des formats attendus par quelques systèmes de TA.

	RTF	Excel	Word	PowerPoint	TXT	HTML	XML	SGML	PDF
SYSTRAN Business Translator	✓	✓	✓	✓	✓	✓	✓		✓
PAHOMTS	✓		✓	✓	✓	✓	✓	✓	
Google Translator Toolkit	✓		✓		✓	✓			
ATLAS		✓	✓			✓			✓

Table 28 : Formats attendus par quelques systèmes de TA

La normalisation d'un texte pour l'adapter à la variété des formats attendus par les systèmes de TA est effectuée par le système TRADOH [Vo-Trung, 2004b], un méta-système d'appel de systèmes de TA en ligne. TRADOH peut traiter un texte multilingue et/ou multicodage en utilisant le système SANDOH [Vo-Trung, 2004a] pour le diagnostic de la langue et du codage de chaque morceau du texte d'entrée. Ensuite, il les extrait et convertit leurs codages vers le système de codage accepté par le système de TA. Voici ci-dessous, par exemple, des codages attendus selon les langues par le système Systran [Systran, 2010].

Langue	Codage attendu	Autres codages supportés
English	ISO-8859-1	UTF-8
Dutch	ISO-8859-1	UTF-8
French	ISO-8859-1	UTF-8
German	ISO-8859-1	UTF-8
Greek	ISO-8859-7	WINDOWS-1253, UTF-8
Italian	ISO-8859-1	UTF-8
Portuguese	ISO-8859-1	UTF-8
Spanish	ISO-8859-1	UTF-8
Simplified Chinese	GB2312	UTF-8
Traditional Chinese	BIG5	UTF-8
Korean	EUC-KR	UTF-8
Japanese	Shift-JIS	EUC-JP, UTF-8
Russian	WINDOWS-1251	KOI8-R, UTF-8
Polish	UTF-8	
Swedish	UTF-8	
Arabic	UTF-8	

Table 29: Codages attendus selon les langues par Systran

b. Balises de formatage et hors-texte

b.i Représentation dans les segments soumis à la TA

Un premier exemple de traitement des balises et hors-texte dans les segments soumis à la TA est celui de MosesWeb [Moses, 2009]. MosesWeb utilise des occurrences spéciales MOSESOPENTAGi, MOSESCLOSETAGi pour remplacer les balises de formatage.

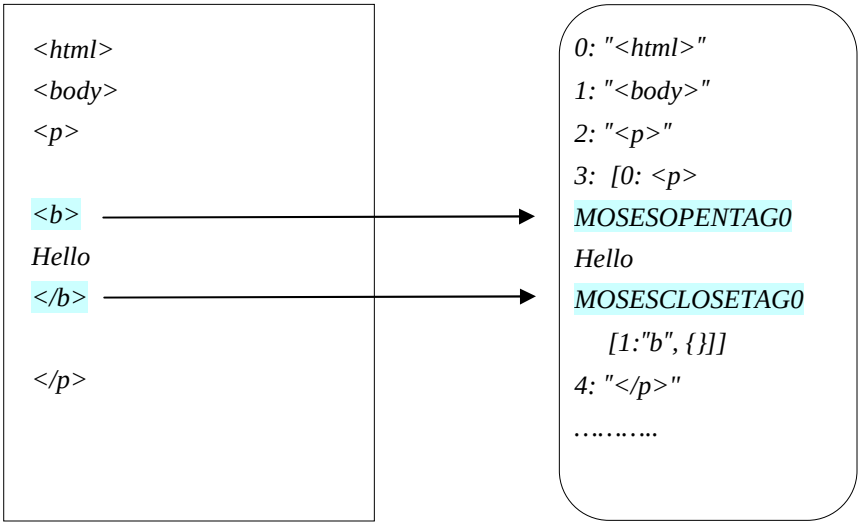


Figure 60: Exemple de la normalisation des balises de formatage dans MosesWeb

Un autre exemple est celui d'UNLC (à l'UNU). H. Uchida utilise des occurrences spéciales de forme HTMi pour remplacer les balises et les hors-texte dans le corpus EOLSS avant de l'envoyer à son enconvertisseur anglais-UNL.

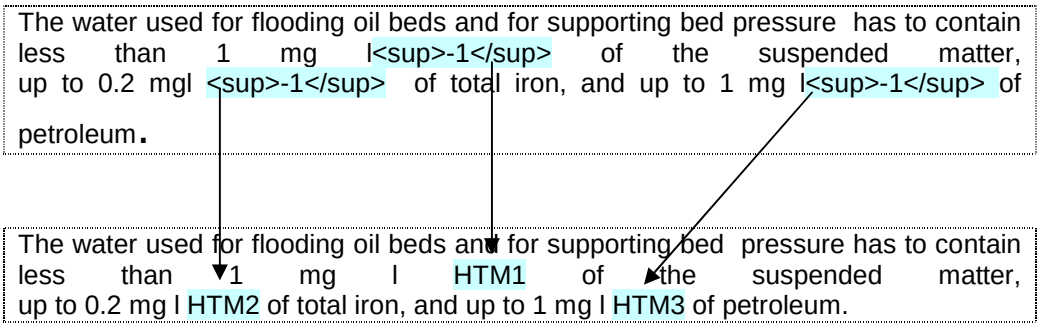


Figure 61: Exemple de normalisation des hors-texte dans les fichiers compagnons (.unl) du corpus EOLSS

III.2.1.2.3 Solutions proposées

Variété des formats attendus par les systèmes de TA. Le traitement de la variété des formats attendus par les systèmes de TA sera délégué à un agent appelé TRADOH++, généralisant le service Web TRADOH de Vo-Trung Hung [Vo-Trung, 2004b].

Identification de langue + codage. L'identification de la langue et du codage d'un texte sera déléguée à un agent appelé SANDOH++, généralisant le service Web SANDOH de Vo-Trung Hung [2004].

Balises de formatage et hors-texte. Il faut, au retour de la TA, stocker dans le sectra les résultats dans une forme normalisée. En effet, on veut pouvoir créer facilement les documents cible, et aussi pouvoir enchaîner deux systèmes de TA.

a. Proposition 1

On délègue le moins possible à TRADOH++.

- Dans le sectra, on définit une représentation normalisée des hors-texte (comme $$_expr_math_12 pour la douzième expression mathématique rencontrée).
- Dans le sectra, on associe à chaque segment (éventuellement infrasegment ou supersegment) un dictionnaire des hors-texte.
- Quand on appelle TRADOH sur un segment, on l'appelle pour un seul système de TA, et on lui "prépare le travail":
  - ✓ on lui donne pour chaque segment les paramètres de traduction pour ce système, et le segment également normalisé pour ce système de TA (c'est ce que fait MosesWeb pour Moses). Le sectra prépare et stocke donc toutes les formes normalisées correspondant à tous les systèmes de TA appelés.
  - ✓ TRADOH appelle les systèmes de TA sur les segments qu'il reçoit, et renvoie les résultats au sectra.
- Au retour, le sectra effectue la "dénormalisation" (par exemple, recapitalisation et recollage des ponctuations décollées, et remplacement des occurrences spéciales correspondant aux hors-texte par les siennes).
- Le sectra réinsère aussi les hors-texte supprimés lors de la normalisation (par exemple début et fin d'élément d'index).
- Le sectra effectue éventuellement un traitement d'erreur, par exemple quand on trouve dans la traduction un hors-texte qui n'était pas dans le segment original.

b. Proposition 2

On délègue le plus possible à TRADOH++.

- Dans le sectra, on définit une représentation normalisée des hors-texte ($$_expr_math_12).
- Dans le sectra, on associe à chaque segment (éventuellement infrasegment ou supersegment) un dictionnaire des hors-texte.
- Quand on appelle TRADOH, on lui donne une liste de systèmes de TA à utiliser, avec les paramètres convenables, et le dictionnaire de hors-texte du segment, sous la forme stockée par le sectra.
- TRADOH convertit la représentation des hors-texte dans celle attendue par chaque système de TA appelé (par exemple, $$_i_3 et $$_/i_3 (représentant la troisième occurrence de et) deviendront MOSESOPENTAG4 et MOSESCLOSETAG4 pour l'appel à MOSES, et HTM5, HTM6 pour l'appel au traducteur UNL de H. Uchida à l'UNU).
- TRADOH effectue aussi la normalisation fine de chaque segment, pour chaque système appelé.
- Au retour, TRADOH++ effectue la "dénormalisation" (par exemple, recapitalisation et recollage des ponctuations décollées, et remplacement des occurrences spéciales correspondant aux hors-texte par les siennes).
- TRADOH++ réinsère aussi les hors-texte supprimés (par exemple début et fin d'élément d'index, et renvoie le résultat au sectra). Sur demande, TRADOH++ renvoie aussi les formes normalisées du segment source et du segment traduit, pour qu'on puisse les stocker et les présenter dans le sectra.
- le sectra effectue éventuellement un traitement d'erreur, par exemple quand un hors-texte ne figure pas dans la traduction, ou quand on y trouve un hors-texte qui n'était pas dans le segment original.

III.2.1.2.4 Solution retenue

Nous avons choisi la proposition 2, parce que cela permet aussi à d'autres systèmes qu'un sectra (par exemple à un système de RI multilingue comme OMNIA) de déléguer le même genre de tâches à TRADOH++.

III.2.2 Problèmes à dominante algorithmique

III.2.2.1 Problème 3.3 : Définition et gestion d'une vraie « mémoire de traductions » (MT)

III.2.2.1.1 Décomposition du problème et état de l'art

Comme dit au §III.1.1.1, il semble impératif de lever les limites des systèmes actuels de gestion de MT.

D'abord, très peu de systèmes ont la *liaison avec les documents source*, ce qui empêche de proposer les traductions existantes dans un ordre reflétant leur adéquation a priori au contexte en cours.

Ensuite, la *gestion des contextes de traduction* est faite dans quelques systèmes, mais de façon trop limitée. Il y a une organisation en domaines et sous-domaines, ou en types de documents, comme ce qui est fait dans Trados (SDL), DéjàVu (ATRIL), ou dans les MT de BEYTrans, etc., mais rien d'aussi précis que ce qui nous semble nécessaire (§II.2.1.2).

Aucun système ne propose la *gestion fine de la qualité* des traductions brutes, des post-éditions et des révisions, ni la *gestion de l'historique et des métadonnées* associées aux versions des cellules. Or, dans un contexte où il peut y avoir des contributeurs spontanés, occasionnels et souvent bénévoles, aussi bien que des contributeurs organisés en projets et

souvent rétribués, il semble nécessaire de le faire, et cela au niveau des cellules des segments, et pas seulement globalement au niveau d'un segment ou d'un document.

Nous avons plus ou moins abordé et levé les problèmes liés à ces limites au point de vue conceptuel au §II.2.1.2. Dans cette section, nous cherchons à traiter ces problèmes aux niveaux algorithmique et fonctionnel. Les problèmes les plus importants sont le stockage et l'évaluation de la qualité des traductions.

Stockage. Le problème vient de la dualité d'une MT. D'une part, une MT peut être organisée et stockée comme une sorte de "dictionnaire de segments", elle doit donc contenir un seul "article" pour toutes les occurrences d'un segment. D'autre part, une MT peut être traitée comme un document, dont chaque segment doit avoir une seule langue source, et contenir ses traductions.

Évaluation de la qualité. L'évaluation de la qualité des traductions enregistrées dans une MT est indispensable pour permettre aux systèmes de classer les suggestions proposées à la post-édition.

Cependant, à notre connaissance, aucun système ne propose quoi que ce soit. Un raison vient peut-être de la limite du format des données utilisé dans les systèmes de gestion de MT. Ces systèmes représentent principalement la MT sous le format TMX, qui ne permet pas de représenter des métadonnées relatives à l'évaluation de la qualité d'une traduction. Une autre raison est peut-être que ces systèmes n'ont pas de gestion et/ou de définition adéquate des profils des producteurs, et qu'ils ne permettent pas non plus aux contributeurs de voter sur leurs résultats de traduction ou de post-édition.

III.2.2.1.2 Solution retenue

Structure de base. Nous représentons une MT comme un ensemble de documents particuliers, à savoir N *pseudo-documents* s'il y a N langues cibles.

Contextes. Chaque segment d'une MT pourra avoir des contextes multiples: occurrences (avec ou sans leurs contextes), peut-être aussi une relation avec une hiérarchie de domaines. Une traduction sera alors associée à un (ou plusieurs) sous-ensembles d'occurrences du segment.

Quanlité. Nous représentons la qualité d'une traduction par deux éléments: un **niveau** associé au type de production de cette traduction, et un **score** révisable.

Niveau	Type de production
*	Traduction mot à mot
**	Résultat de TA
***	Traduction ou post-édition par un locuteur natif de la langue cible
****	Traduction ou post-édition par un traducteur professionnel
*****	Traduction ou post-édition par un traducteur certifié

Table 30: Niveaux associés aux types de production des traductions

Le score révisable a une valeur de 0 à 20. En fonction du niveau traductionnel du producteur, un score par défaut est défini dans son profil et associé à chaque cellule qu'il produit. Cependant, le post-éditeur peut changer ce score, pour chaque résultat de traduction. Par exemple, s'il a un petit doute sur la traduction qu'il vient de produire, il peut lui mettre un score de 9/20. Ou encore, s'il trouve que la traduction est assez bonne, il peut lui donner un score de 15/20.

Autres métadonnées. A part le niveau et le score de qualité, il y a d'autres métadonnées associées à un segment, telles que l'auteur, la durée de fabrication, le taux de modification, la date de fabrication, etc.

III.2.2.2 Problème 3.4 : Programmabilité du traitement des corpus, avec synthèse entre flux de travaux et commandes complexes

III.2.2.2.1 Motivations et raffinement du problème : exemples d'opérations à "programmer"

Rendre programmable un sectra permettrait à l'utilisateur de définir lui-même, selon ses besoins et ses désirs, des traitements qu'il ne peut pas définir avec les interfaces graphiques.

Les opérations qu'il faut permettre de programmer concernent la gestion des corpus dans un système de TA, la mise en œuvre de campagnes d'évaluation, les projets de postédition de documents, et l'utilisation dans des systèmes novateurs (comme la RI multilingue dans OMNIA).

a. Opérations nécessaires dans un système de gestion de corpus d'un système de TA

Il faut pouvoir faire au moins autant qu'en Ariane-G5, et généraliser.

- ***Opérations directement reprises d'Ariane-G5*** : création d'un corpus, définition des séparateurs hiérarchiques, import de documents, sélection (manuelle, par édition d'une liste) d'un sous-ensemble de segments ou de documents, soumission à des chaînes d'exécution ou de production, export d'une sélection de documents sous différentes formes (par exemple source, traduction brute et révision).
- ***Opérations généralisant celles d'Ariane-G5*** : définition d'un corpus, de ses documents, et de ses segments multilingualisés et contextualités (pour que la TA puisse utiliser au mieux les MT), définition des "grains" d'information associés aux segments (par exemple: graphe UNL et pas seulement arbre décoré), programmation explicite des prétraitements et des postraitements (appel à des transcripteurs…).

b. Opérations motivées par l'utilisation dans des campagnes d'évaluation

Essentiellement, on a constaté :

- le besoin d'introduire une *gestion fine des utilisateurs*, la *notion de projet*, la possibilité de définir des *sélections arbitraires* grâce à des contraintes sur différentes valeurs de variables connues du système (et par conséquent le besoin de disposer d'une liste complète des variables et des commandes du système), et des *diagrammes de tâches* (ou flots de travaux). Ici, ce sont les gestionnaires de projet qui doivent programmer.
- le besoin de permettre à des utilisateurs de *programmer certaines "mesures d'évaluation"*, comme de nouvelles variantes de BLEU, WER, HTER, etc.
- le besoin de pouvoir *définir et contrôler de nombreux paramètres de l'interface*, comme la position et l'apparence (visible ou cachée) des colonnes, ou le nombre des boutons radio en évaluation subjective.

c. Opérations motivées par l'utilisation dans des projets de post-édition

- Appel à un ou des segmenteurs externes, représentation d'un graphe de segmentation, modifications locales du graphe de segmentation d'un document, appel en continu à des systèmes de TA distants, appel à des composants différents (comme le dessin des graphes UNL), appel à des transcripteurs…
- Import/export de mémoires de traduction et opérations sur ces pseudodocuments similaires aux opérations usuelles sur les documents.

d. Opérations motivées par l'utilisation dans des systèmes novateurs

- Appel en boucle infinie à des agents distants.
- Transformations complexes sur les données pour produire des composants linguiciels comme des minidictionnaires html pour EOLSS ou des systèmes-Q pour l'annotation de textes par des UW.

III.2.2.2.2 Types de "programmeurs" et de programmation

La programmabilité doit permettre à un sectra de servir plusieurs types d'utilisateurs avec des besoins différents. En fonction de leur compétence informatique, les utilisateurs de chaque type doivent avoir accès à un niveau différent de programmation.

a. *Paramétrage et programmation par les non-informaticiens*

Les non-informaticiens tels que les organisateurs doivent facilement mettre en œuvre des paramétrages « externes » : réglage de l'interface selon les besoins du projet, définition d'un format d'export, affectation d'un nom à une sélection de données, etc.

Il faut aussi fournir des supports pour effectuer certaines tâches par des flots de travaux.

On doit permettre de mettre en œuvre des paramétrages de commandes complexes préprogrammées, comme le lancement d'une boucle infinie d'appel à un ou plusieurs systèmes de TA, avec suivi et intervention possible depuis un panneau de contrôle, ou encore lancement d'un "chercher-remplacer" multiple sur une sélection arbitraire de segments (ou de documents), avec en paramètre une liste de couples ou de triplets <chaîne_1, chaîne_2 [, contexte]>.

On sait que des linguistes informaticiens arrivent très bien à écrire des scripts REXX pour un éditeur (XEDIT sous CMS, porté par Hessling en "THE"). Pourrait-on "ouvrir" ce type de programmation, ou est-ce trop risqué (effets de bord indésirables non détectables par compilation, ou possibilités d'intrusion…) ? Nous pensons que oui, mais cette idée n'a pas encore été réalisée, et donc n'a pas encore été évaluée.

b. *Paramétrage et programmation par des contributeurs informaticiens*

Les contributeurs informaticiens doivent bien sûr pouvoir effectuer tout ce qui précède, et aussi :

- des paramétrages "internes" plus délicats.
- l'écriture de scripts pour les traitements délégués (commandes complexes).
- l'écriture de programmes de service dans des LSPL, conçus pour éviter les problèmes de sécurité (par exemple, écriture de transcripteurs en LT pour l'import ou l'export de documents).
- la liaison avec de nouveaux "agents" introduits dans l'environnement (création ou adaptation d'une API, etc.).

III.2.2.2.3 Solution proposée

a. *Principes*

De façon générale, on se propose de suivre si possible deux idées de base:

- l'idée de la *programmation "narrative"*, c'est-à-dire qu'on puisse "programmer sans le savoir" grâce à une interface graphique à manipulation directe, et de façon équivalente, "programmer en le sachant" dans un langage d'apparence très classique, écrit sous forme textuelle, et dont les objets et processus primitifs sont exactement les mêmes que ceux de l'interface, tandis que les structures de contrôle sont les plus simples possible.
- *assurer l'innocuité des programmes* écrits par des contributeurs en faisant en sorte que les langages utilisés soient de même nature que les LSPL (langages spécialisés pour la programmation linguistique) classiques, c'est-à-dire qu'ils ne permettent strictement aucune action directe sur l'environnement (pas d'écriture sur fichier, ni en mémoire), ni aucun appel à des fonctions externes (pas d'appel système, en particulier), tout en offrant des structures de données puissantes et des structures de contrôle adaptées.

Définition. Un *langage narratif* [Bellynck, 1999, 2001] est un langage dont les objets et les actions primitifs sont ceux de l'interface graphique. Il offre les structures de contrôle minimales permettant de réaliser les actions visées.

b. Solutions pour programmer les différentes types de tâches

Voici comment ces deux idées peuvent être mises en œuvre au niveau d'un simple paramétrage, de la définition d'une sélection, de la programmation d'un flot de travail ou d'une commande complexe à un agent, d'une tâche de fond, et d'une transformation complexe de données, afin de les exporter sous une forme demandée par tel ou tel utilisateur ou service Web.

b.i Langage d'affectations : paramétrages

Une interface de paramétrage est toujours un formulaire très simple, et le langage narratif dont nous parlons est simplement une suite d'instructions de forme <paramètre> = <valeur>.

Supposons que nous exécutions une telle suite en faisant un `include` dans un script (en perl, php, ou rexx), nous aurions une faille de sécurité. Pour y remédier, nous préciserons les paramètres possibles, et les valeurs possibles pour chaque paramètre, et considérerons le langage obtenu comme un LSPL. Nous vérifierons donc tout paramétrage, simple ou complexe, à l'aide d'un compilateur (écrit en javacc ou en ANTLR si nous programmons en java) spécifique de ce petit langage.

b.ii Langage de conditions booléennes : définition de sélections

Pour le langage permettant la définition de sélections de données, il sera mieux de permettre en plus d'écrire une formule quelconque (dans un langage spécialisé comme pour WinFile, ou par un SELECT du langage SQL pour MySql ou autre).

b.iii Langage de flots de travail

Flot de travail ou commande complexe à un agent. On pense à l'approche de la programmation par composants, ou du réseau LIDIA, ou de l'outil Open Workflow [OpenWFE, 2009]. Voici un exemple de la description du flot de travaux pour un enseignant dans IToldU [Bellynck et al., 2005].

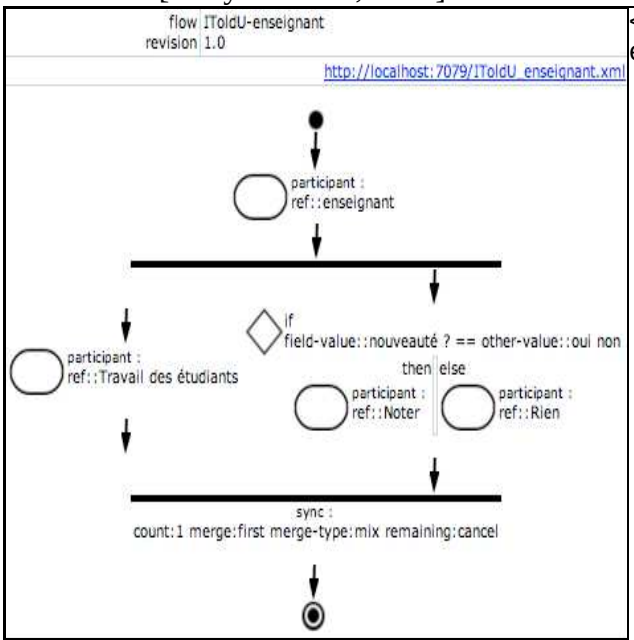


Figure 62: Diagramme de flot de travaux pour une opération simple dans IToldU

```
<process-definition name="IToldU-
enseignant" revision="1.0">
 <description>ItoldU
 Enseignant</description>
  <sequence>
   <participant ref="enseignant" />
    <concurrence
    sync="generic"
    count="1"
    merge-type="mix"
    remaining="cancel">
    <participant ref="Travail des
    étudiants" />
    <if>
    <equals field-
    value="nouveauté ?"
    other-value="oui non" />
   <participant ref="Noter" />
   <participant ref="Rien" />
   </if>
   </concurrence>
  </sequence>
</process-definition>
```

Ou encore, on propose la possibilité d'une interprétation directe des LSPL narratifs par plusieurs langages de scripts, comme pour les paramétrages. C'est ce qui est fait depuis longtemps dans Ariane-G5 avec le LIDS (langage interne de description de séquences).

Comme en Ariane-Y, on n'aura alors plus un formulaire paramétrable, mais une classe de diagrammes définie par les types de nœud et d'arcs, et manipulables par un éditeur graphique, sous un navigateur quelconque. Le programme narratif correspondant à un diagramme particulier est alors une suite d'instructions de l'éditeur permettant de le construire.

Tâche de fond. Nous proposons l'utilisation de commandes simples ou complexes pour contrôler des tâches de fond, par exemple, le lancement une boucle infinie d'appel de TA sur une sélection recalculée dynamiquement (ex: segments non traités depuis plus de 2 jours).

Langage narratif ou LSPL : programmation d'une transformation complexe de données. Il s'agit d'exporter des données sous une forme demandée par tel ou tel utilisateur ou service Web.

On propose l'approche par langage spécialisé (LSPL). Elle permettra en effet que les développeurs puissent définir eux-mêmes des tâches, tout en offrant des garanties de sécurité, la décidabilité intrinsèque (impossibilité de boucles infinies) et des bornes de complexité connues.

III.2.3 Problèmes à dominante programmatoire

III.2.3.1 Problème 3.5 : Traitement de masses de données

III.2.3.1.1 Motivations

On utilise maintenant des corpus gigantesques en TA empirique directe et statistique, et de gros corpus enrichis en TA « par les exemples » et TA « experte » [Boitet, 2007].

Un sectra rencontrera donc des problèmes de performance (édition, recherche, etc.).

La gestion de très gros corpus ne semble pas poser de problème spécifique par rapport à la gestion de grosses ou très grosses bases de données. Du point de vue de la taille, nous sommes très loin des tailles des données à gérer quand on traite des images ou de la vidéo, ou, encore pire, des résultats d'expériences de physique nucléaire.

Si l'on travaille en mémoire pour accélérer les traitements, il pourrait y avoir un problème de cohérence entre les données en mémoire et celles sur disque (en principe, gérées par une BD).

La solution simple que nous adopterions alors est d'envoyer une commande de mise à jour de la BD dès qu'une modificaiton est faite en mémoire, ou d'utiliser une technique standard de persistance des données (comme Hybernate).

III.2.3.1.2 Problèmes de lenteur des traitements

a. Recherche

Il n'y a aucun problème pour la recherche de coincidences exactes, du moment qu'on a assez de place pour préparer des structures d'accès de coût temps logarithmique, comme des arbres de recherche (AVL[39]) ou des tableaux de suffixes [Cromières, 2010].

Peut-être risque-t-on un gros problème de lenteur pour la recherche de coincidences floues. Cependant, pour la post-édition contributive, on ne souhaite pas l'utiliser, puisqu'on sait qu'elle est moins efficace que la TA [Blanchon et al., 2009 ; Allen, 2009]. Si l'on voulait

[39] http://www.labri.fr/perso/strandh/Teaching/MTP/Common/Book/HTML/node239.html

malgré tout l'utiliser, alors il faudrait faire une recherche "multiétage" comme E. Planas dans SIMILIS, puisque c'est au moins 40 fois meilleur que les algorithmes classiques sur la chaîne brute ("rez de chaussée"). Et, dans ce cas, on sait qu'on peut commencer par extraire une "sous-mémoire" formée des segments contenant un certain pourcentage des lemmes du segment à traduire. Il suffit de calculer des index sur les lemmes pour rendre cette extraction très rapide.

b. Traitements

De quels traitements s'agit-il? Nous allons distinguer ceux qui peuvent être effectués de façon incrémentale, à l'avance, et les autres.

b.i Traitements exécutables de façon incrémentale et à l'avance

Il s'agit :

- de la préparation de prétraductions par différents systèmes de TA ;
- de la lemmatisation en vue de la recherche dictionnairique ;
- d'annotations diverses, par exemple par des lexèmes interlingues (UW++) pour le projet OMNIA.

On a effectivement un gros problème si l'on veut traiter en une fois la totalité d'une grosse masse de données, comme par exemple 500000 textes compagnons (soit 30 M mots) de la base Belga-News[40] de CLEF09-CLEF10.

Par contre, si l'on travaille de façon incrémentale, en tâche de fond, le problème est résolu. L'annotation de la masse initiale de données prend certes 19h, mais ensuite l'annotation d'un nouveau texte ne demande que quelques secondes, de même que le traitement d'une requête.

b.ii Traitements non exécutables à l'avance

Ce sont tous ceux décidés par les utilisateurs au moment du traitement.

Filtrage. Ici, on a un vrai problème, si la condition de filtrage est arbitraire, puisqu'il faut alors, a priori, examiner tous les segments sur lesquels peut porter le filtrage, et qu'on ne peut pas précalculer les réponses.

Chercher-remplacer dans tout un corpus ou ensemble de corpus. S'il s'agit du chercher-remplacer interactif, le problème se ramène à celui du calcul d'une sélection.

S'il s'agit du chercher-remplacer automatique, le problème est de faire en sorte que l'utilisateur n'attende pas trop longtemps, et soit informé de l'avancement du traitement.

On sait réaliser cette opération en temps linéaire (en compilant un transducteur d'états finis à partir des paires (remplacé, remplaçant), voire légèrement sublinéaire [Morris et Pratt, 1970 ; Knuth et al, 1977 ; Boyer et Moore, 1977], mais ensuite le temps est "incompressible" (par exemple plusieurs secondes sur un fichier de 80Mo pour TextWrangler).

Une solution possible est alors de lancer une telle opération exactement comme une tâche de fond, sans bloquer l'utilisateur, en lui montrant une barre de progression, et en débloquant les segments modifiés au fur et à mesure qu'ils sont traités.

Calcul de scores globaux. Le calcul de scores locaux comme WER ne pose pas de problème puisqu'il se fait segment par segment.

[40] http://www.belga.be/FR/home.asp?lang=FR

Si l'on sait qu'on veut calculer des scores globaux tels que BLEU sur les corpus entiers, il est peut-être possible de trouver ou d'inventer des algorithmes qui les mettent à jour incrémentalement.

Par contre, si l'on veut calculer un score global sur une sélection arbitraire énorme, le temps de calcul sera certainement très long. Le seul palliatif semble être de "déléguer" ce calcul à un agent spécialisé, en lui exportant les données.

Import et export. Il n'y a pas de solution miracle. Il faut bien sûr utiliser des algorithmes rapides, mais la seule chose fondamentalement efficace est de diviser les données en sous-ensembles.

Tri. Ce problème a été abordé par Ch. Boitet à l'occasion de la thèse de Y.Bey [Bey, 2009]. La conclusion a été qu'on n'arriverait jamais à un temps de calcul raisonnable, même en divisant les données et en les triant sur des processeurs parallèles, puis en fusionnant les monotonies obtenues. Par contre, il a prouvé qu'on pouvait tout simplement maintenir les données triées à tout instant, par rapport à 1 ou N critères de tri connus, grâce à la technique des ensembles de chaînes de J.-C. Durand (développée dans le cadre du projet Ariane-Y).

Bien entendu, si l'on définit un nouveau critère de tri, le tri initial prendra un certain temps, mais il peut se faire en tâche de fond.

La solution reste partielle, puisqu'on ne pourra apparemment jamais trier instantanément un énorme ensemble de données par rapport à un nouveau critère de tri — ou alors, peut-être existe-t-il une solution "quantique" ?

Cette situation peut-elle se produire? Oui, si par exemple on veut trier tous les segments contextualisés d'une énorme MT par rapport à un critère portant sur les contextes. Il faudra donc "instrumentaliser" le sectra pour voir si cela se produit réellement et dans quels cas. En effet, peut-être y aura-t-il des solutions rapides pour ces cas, alors qu'il n'en existe pas dans le cas général.

c. *Problèmes de navigation et de visualisation*

Ces problèmes ont été traités au §I.2.1.2.

III.2.3.1.3 Résumé des solutions

Nous pouvons résumer les solutions aux problèmes posés par l'aspect "masse de données" des très gros corpus de traduction en distinguant ce qui concerne le client (navigateur sur le PC de l'utilisateur) et le serveur.

a. *Client*

Pour mémoire (car cela a été détaillé au §I.2.1.2), le problème de visualisation et de navigation semble résolu en :

- utilisant une pagination virtuelle ;
- divisant les données en blocs plus petits, par exemple des documents virtuels de taille inférieure à une limite paramétrable, et/ou de nombre de segments inférieur à une certaine limite ;
- modifiant la page d'interface par la technique Ajax ;
- ayant recours à un cache (sur le client) si l'utilisateur "navigue" dans des parties éloignées du corpus (ou d'un énorme document).

b. *Serveur*

Le problème de performance venant de la taille des données est résolu ou contourné en :

- créant des index ;
- maintenant les données triées selon les critères connus ;
- transformant les traitements non incrémentalisables en tâches de fond.

III.2.3.2 Problème 3.6 : Architecture par agents

III.2.3.2.1 Préciser le problème

On a décidé d'utiliser une architecture par agents plutôt qu'une architecture en client-serveur.

Notre problème ici est de savoir comment mettre en œuvre une telle architecture dans notre contexte. Les sous-problèmes à résoudre sont alors :

- déterminer quels seront les agents, s'ils pourront être fixes, ou s'il faudra en créer dynamiquement ;
- déterminer la façon d'implémenter l'aspect "agent" : que ce soit par implémentation directe ou par utilisation d'une "boîte à outils", il faudra sans doute choisir un langage, et un ou des protocoles ;
- essayer d'utiliser les compétences disponibles, et les expériences préalables.

Nous pouvons sans doute tirer parti de certains travaux antérieurs. Il ne s'agit donc pas ici de faire une étude exhaustive des outils et méthodes de la programmation par agents, mais simplement de trouver rapidement une façon de réaliser notre système "par agents", en utilisant le plus possible les expériences préalables de chercheurs qui peuvent nous aider directement.

III.2.3.2.2 Etat de l'art

a. Aspect "agents" dans le réseau LIDIA

Le réseau LIDIA [Guillaume, 2003] conçu et implémenté par P. Guillaume sous zVM/CMS autour d'Ariane-G5 est en fait un réseau d'agents, où chaque agent est une "machine virtuelle" dédiée à un certain type de tâche (information sur les données, exécution totale ou partielle d'une traduction, etc.).

Une commande complexe est envoyée à ce réseau sous forme d'un "kit" (une sorte d'archive non compressée) contenant les commandes à exécuter et les données à prendre en compte. Elle est acheminée par un agent "routeur" vers un agent (une machine virtuelle) pouvant réaliser la prochaine tâche à exécuter. Cet agent la réalise, met le résultat dans le "kit", et renvoie au routeur.

b. BLEXISMA

D. Schwab a réalisé pour sa thèse sur la désambiguïsation lexicale par vecteurs conceptuels [Schwab, 2006] le système d'agents BLEXISMA. Il a ensuite utilisé la boîte à outils pour un projet en coopération avec l'USM (Penang) sur le même sujet. Il semble que les boîtes à outils sont encore trop fragiles (ce sont presque toujours des prototypes de recherche) et qu'il vaille finalement mieux programmer directement son système, au moins dans le cas d'un système à gros grain, BLEXISMA2 [Schwab et Lim, 2008].

c. Les techniques de communication et les "boucles infinies"

c.i CommSwitch des projets C-STAR II et Nespole!

En ce qui concerne la communication , on peut s'inspirer de la technique du "CommSwitch" utilisée lors du projet Nespole![41].

[41] http://nespole.itc.it/

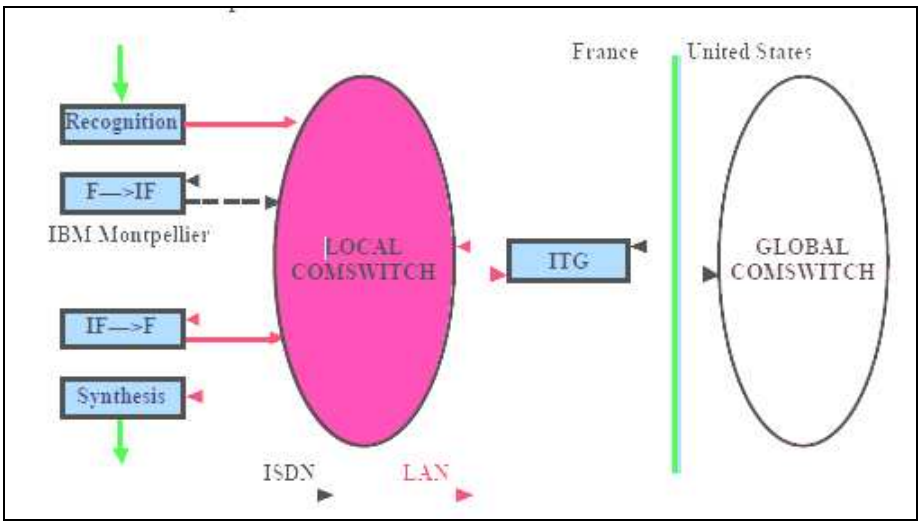


Figure 63: Architecture of NESPOLE !

Dans ce projet, tous les composants du CLIPS++ (reconnaisseur du français, enconvertisseur français → IF, etc.) sont des serveurs. Ainsi, aucun d'entre eux ne peut communiquer avec un autre. Ils sont tous connectés à un CommSwitch local .

c.ii *Aspect "agents" dans l'architecture en "tableau blanc"*

On peut s'inspirer du prototypage d'une architecture à "tableau blanc" [Boitet et Seligman, 1994] pour des systèmes de TA de parole hétérogènes (les agents étaient des reconnaisseurs, des analyseurs, des traducteurs, des générateurs, et des synthétiseurs).

On n'utilisait que les systèmes de gestion de fichiers (que des boîtes aux lettres, pas de stream etc.), et l'implémentation était faite sous KEE (en Common Lisp + bibliothèque de KEE). On transformait tous les composants en serveurs, mais pas encore en "agents".

c.iii *REXX et les REXXstacks*

Au niveau de l'implémentation, le GETALP a une expérience très positive avec REXX (utilisé pour le moniteur d'Ariane-G5, pour le réseau LIDIA, et pour l'éditeur transformationnel d'arbres TTEDIT de J.-C. Durand [Durand, 1988]) : il est totalement portable (ses versions actuelles sont Regina, ooRexx et NetRexx) et est intégré à THE[42] (The Hessling Editor, un éditeur compatible avec Xedit implémenté par Hessling).

De plus, REXX fournit un mécanisme simple mais efficace de communication entre applications ou machines par échange de listes de chaînes de caractères (commandes, données). À la réception, on peut gérer les commandes par les piles-files (REXXstacks) implémentées dans REXX.

Le mécanisme de REXXstack est compatible avec l'extension de notre architecture vers des serveurs/services puis des agents. Le coût d'implémentation par REXXstack n'est pas élevé, et l'on obtiendrait un système complet et homogène.

[42] http://hessling-editor.sourceforge.net/

III.2.3.2.3 Solutions retenues

Déterminer quels seront les agents. On en aura une liste prédéfinie de types d'agents, et du nombre normal de leurs instances :

Type d'agent	Nombre d'instances de ce type
SECTra_w	1
TRADOH++	1
PIVAX	1 ou plus
SegDoc	1
Traducteurs automatiques (agents "serveurs")	Nombre de serveurs de TA disponibles
Relais-iMAG	1
iMAG	Nombre de sites élus
Tableau blanc	1

Table 31: Déterminer quels seront les agents

La liste des types d'agent sera fixe, mais on prévoit de pouvoir ajouter un agent d'un type donné (par exemple un serveur de TA ou un agent dictionnaire) de façon dynamique.

Déterminer la façon d'implémenter l'aspect "agent". Vu l'expérience préalable de D. Schwab avec l'implémentation directe et avec l'utilisation d'une "boîte à outils", et les spécificités de notre problème, nous choisissons de réaliser une implémentation directe assez simple.

Utilisation de REXX, des REXXstacks, et des systèmes de fichiers pour les boîtes aux lettres. Vu l'avantage de REXX et des piles-files de REXX (REXXstacks) présenté ci-dessus, nous choisissons d'utiliser REXX, des REXXstacks, et des systèmes de fichiers (pour les boîtes aux lettres), pour implémenter la communication entre SECTra_w et les agents.

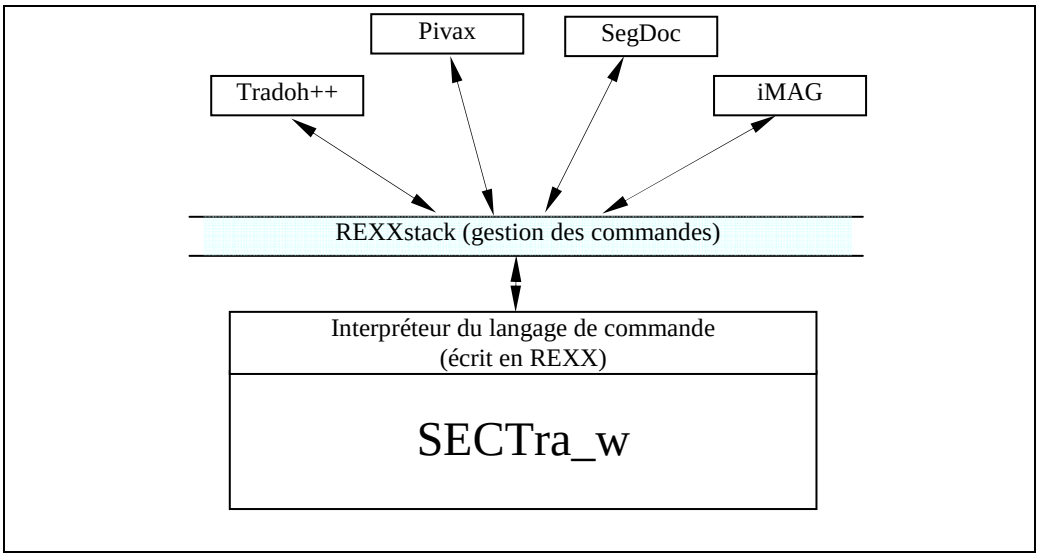


Figure 64 : REXX et REXXstack pour la communication entres les agents

III.3 Implémentation, expérimentation et évaluation avec SECTra_w

III.3.1 Spécification et implémentation

Pour pouvoir expérimenter et évaluer les solutions proposées ci-dessus, nous les implémentons dans SECTra_w.

III.3.1.1 Objectifs

Notre but ici est d'étendre SECTra_w pour qu'il devienne un service permettant à des applications novatrices (comme les iMAG (voir §III.3.2.1) et NotePad++ (voir §III.3.2.1)) d'exploiter ses ressources.

Pour cela, nous devons d'abord construire et gérer dans SECTra_w des mémoires de traductions dédiées à ces applications, ensuite développer des fonctions permettant à ces applications de les exploiter, et enfin permettre à SECTra_w et à ces applications de communiquer les unes avec les autres en construisant une architecture logicielle par agents.

III.3.1.2 Architecture générale

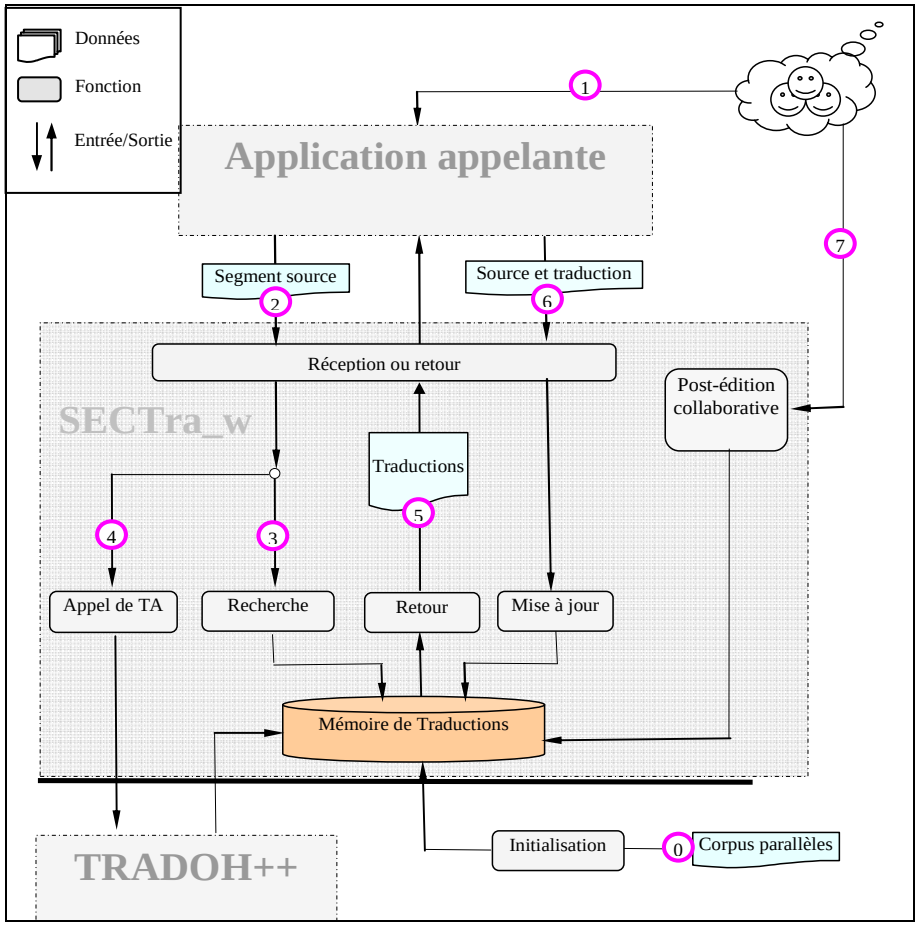


Figure 65: Architecture générale de SECTra_w intégré à des systèmes novateurs

Ce schéma montre que SECTra_w sera utilisé comme le support de ressources traductionnelles pour plusieurs applications novatrices. Chacune exploitera les ressources traductionnelles de SECTra_w comme suit.

- Pour chaque projet réalisé dans l'application appelante, on crée une MT dédiée dans SECTra_w. Cette MT peut être initialisée par des segments parallèles récupérés du projet.
- Supposons qu'un internaute veuille accéder à un document (une page Web, une interface de Notepad++, etc.) via une application appelante dans une certaine langue cible. L'application doit d'abord segmenter ce document, puis envoyer à SECTra_w les résultats de la segmentation, accompagnés de la demande de traduction (site élu, langue cible, et préférences associées communiquées par le relais, sous la forme d'une liste ordonnée de systèmes de TA, avec leurs paramétrages préférés).
- Après avoir reçu les requêtes de l'application appelante, pour chaque segment source, SECTra_w doit d'abord chercher des traductions pour ce segment source dans la mémoire de traductions dédiée correspondante. S'il n'existe aucune traduction équivalente dans la MT, ce segment doit être envoyé à TRADOH++ pour récupérer des traductions automatiques. Ces traductions automatiques sont aussi sauvegardées dans la MT dédiée.
- Après avoir obtenu les traductions pour ce segment source, SECTra_w les renvoie à l'application applante.
- L'internaute peut voir le document cible et le post-éditer directement segment par segment via l'interface de l'application appelante. L'application appelante envoie chaque post-édition à SECTra_w, qui met à jour la MT correspondante.
- L'internaute peut aussi post-éditer la MT dédiée directement sous SECTra_w. Cela demande donc une synchronisation de la gestion des utilisateurs entre SECTra_w et les applications appelante.

III.3.1.3 Spécification des fonctions

III.3.1.3.1 Construction de MT dédiées pour les applications utilisant SECTra_w

a. Architecture d'une MT

Nous construisons une MT dédiée à chaque projet réalisé par l'application appelante.

Comme dit plus haut (§II.2.1.1), elle est implémentée comme autant de *pseudodocuments* que de langues cibles. Chacun contient un segment-mc (segment multilingualisé et contextualisé) unique pour chaque chaîne (normalisée en UTF-8) apparaissant dans une occurrence de segment source du site Web élu (ou du corpus associé au projet consiéré).

On peut donc elle-même la post-éditer, l'étendre en la faisant prétraduire puis post-éditer dans d'autres langues, et la soumettre à divers traitements dont les résultats seront stockés comme des annotations (graphes UNL, mini-dictionnaires, etc.). Chaque segment-mc est bien sûr accompagné de métadonnées (niveaux et notes de qualité (voir §II.3.1.1.3.d) pour la traduction), et de ses contextes d'apparition.

b. Initialisation d'une MT dédiée

La MT dédiée à un site Web élu est si possible préalablement initialisée par les segments multilingualisés ayant été détectés et récupérés à l'avance à partir de ce site Web.

C'est un cas assez fréquent, qui se produit quand le site élu a plus d'une langue de diffusion (par exemple, 6 langues pour le site support de Systran), et qu'on s'aperçoit que les informations dans toutes les langues sauf une sont obsolètes ou incomplètes… et que c'est une bonne idée de renoncer à la diffusion en N langues avec mise à jour immédiate et haute qualité, et qu'on choisit un service d'accès multilingue.

Le problème ici est de déterminer quels bisegments sont réellement en relation de traduction. Pour cela, on peut utiliser des techniques d'extraction de corpus parallèles à partir de corpus comparables, comme l'a fait [Do et al., 2009] (§III.1.1.2).

III.3.1.3.2 Recherche et mise à jour

Toutes les fonctions suivantes peuvent être utilisées via des requêtes http (POST pour mettre à jour, GET pour récupérer de l'information).

SECTra_w fournit deux fonctions importantes : la recherche et la mise à jour de traductions dans les MT. SECTra_w ne fournit que la recherche exacte. En effet, comme on l'a dit plus haut, la recherche approchée classique est nettement moins utile que la TA (l'utilisation de MT gagne au plus 50%, tandis que la post-édition de TA gagne environ 65%), et la recherche exacte peut être faite très efficacement, même sur des mémoires gigantesques.

Il ne serait en fait justifiable d'intégrer la recherche approchée dans un tel système que si elle était "à l'état de l'art", c'est-à-dire si elle était fondée sur une représentation "en étages" des segments à traduire… et des segments de la MT, comme cela est fait dans SIMILIS. Nous ne l'avons donc pas intégrée dans notre implémentation actuelle.

III.3.1.3.3 Post-édition de MT

SECTra_w permet aux utilisateurs des applications novatrices de post-éditer les mémoires de traductions dédiées à leurs projets. La post-édition doit être faite dans deux cas :

- post-édition d'un document virtuel correspondant à un document (une page Web, une interface d'une application, etc.) ;
- post-édition de toute la MT correspondant à un projet.

Cela exige une gestion adéquate des utilisateurs entre SECTra_w et les systèmes appelants. Nous devrons implémenter l'idée d'utilisateur "représentant" un autre par délégation (surrogate) (voir §I.2.3.1).

III.3.1.3.4 Récupération de bonnes traductions à partir une ou plusieurs MT

SECTra_w doit permettre de récupérer de bonnes traductions à partir d'une mémoire de traductions dédiée. Il devra fournir une fonction de filtrage selon quelques critères afin de sélectionner les bonnes traductions. Les critères sont, par exemple, les noms d'auteurs de traductions et leurs niveaux traductionnels, un seuil de note de qualité associé à des traductions (par exemple, > 14), etc.

Cette fonction doit aussi permettre de fusionner plusieurs MT, et de sélectionner des segments parallèles selon un sujet (par exemple, santé, éducation, culture, etc.) pour les utiliser dans des campagnes d'évaluation ou pour tester des systèmes de TA, etc.

III.3.1.3.5 Construction : architecture par agents

On construit SECTra_w et les systèmes extérieurs (auxquels SECTra_w délègue ou avec lesquels il communique) selon une architecture par agents où chaque système fonctionne indépendamment sur chaque site. Les systèmes participent à un flot de données, et la communication et la synchronisation sont faites par un agent implémenté sur chaque site.

L'échange entre SECTra_w et les systèmes extérieurs se fait soit par le protocole http, soit par une solution à la WICALE dans laquelle, pour chaque système, on définit les commandes et le format des résultats associés dans un métalangage simple. Si l'on ajoute un autre système ou s'il y a un changement dans la syntaxe de commande d'échange ou de format, on s'adapte très vite.

III.3.1.4 Réalisation

Nous avons implémenté ces spécifications dans le cadre de deux projets réels : iMAG et Notepad++ (voir III.3.2.1).

L'implémentation a été réalisée en 3,5 mois, de décembre 2008 à avril 2009.

Tâche	CCH	DSE	DSI	codage	Acquisition de données	Initialisation de MT	Tests	**Total**
Jours	15	15	15	30	20	20	1	**116**
Taille	10 p.	20 p.	15 p.	4000 lignes, Java classes	2000 segments parallèles anglais-français du site Web du LIG	2000 segments + les corpus BTEC, EuroParl, B@bel		

Table 32:Quelques éléments factuels sur l'implémentation (troisième défi)

Nous utilisons MySQL Server 5.1 pour construire les MT. Pour chaque MT, nous utilisons un tableau principal contenant les segments-mc et des tableaux secondaires pour gérer les occurrences et les métadonnées.

Pour utiliser la fonction de recherche de MT, nous avons les paramètres spécifiés dans la table suivante.

Paramètre	Nom	Valeur par défaut	Note
Protocole	Protocol	http	Protocole de commande
Nom_serveur	server	eolss.imag.fr	Nom de serveur
Port	port	80	Port
URL	url	/xwiki/bin/view/TM/ExactSearch	Chemin sur le serveur
Texte_source	texte_source	« «	Phrase_à_chercher
Délai	delay	100	Temps d'attente maximal en secondes ; s'il est dépassé, on passe à une autre requête
Nom_MT	MT	iMAG_LIG	Nom de mémoire de traduction dédiée
Langue_source	Sl	en	Langue source
Langue_cible	tl	*	Langue cible
Etoile	etoile	>3	Recherche de traductions produites par des auteurs ayant un niveau traductionnel à partir d'un niveau précis.
Note_qualité	note	>10	Recherche de traductions ayant une note de qualité à partir d'une note précise.
Nom_auteur	nom_auteur	«HCPhap»	Recherche de traductions créées par un traducteur.
Date	date	> 2008-04-11 16:25:50	Recherche de traductions réalisées à partir d'une date précise.

Table 33 : Paramètres de la fonction de recherche de MT

Les applications appelantes peuvent donc exploiter leurs MT gérées par SECTra_w en lui envoyant des requêtes dans le format de la Table 33. Voici l'exemple d'une requête de recherche des traductions dans la MT dédiée au site Web du LIG.

http://eolss.imag.fr/xwiki/bin/view/TM/ExactSearch?texte_source=Ethics%20and%20Science&delay=100&MT=iMAG_LIG&sl=en&tl=fr&date=2008-04-11.

Le résultat renvoyé par SECTra_w est un fichier HTML sous le format simple de tableau. Les applications appelantes doivent analyser ce fichier pour récupérer des traductions.

Pour utiliser la fonction de mise à jour de MT, nous utilisons les paramètres spécifiés dans la table suivante.

Paramètre	Nom	Valeur par défaut	Note
Protocole	Protocol	http	Protocole de commande
Nom_serveur	server	eolss.imag.fr	Nom de serveur
Port	port	80	Port
URL	url	/view/TM/UpdateTM	Chemin sur le serveur
Id_Segment	id	lig_page1_16_phap_0 90411062550	Id unique du segment (par exemple, il est créé par le nom de MT, de page, d'auteur, le numéro de segment, et la date de création)
Texte_source	texte_source	« «	Texte source à mettre à jour
Texte_cible	texte_cible	« «	Texte cible à mettre à jour
Code_erreur	code_erreur	0	Code retour de la mise à jour correspondant à un succès ou à un type d'erreur
Nom_MT	mémoire de traduction	iMAG_LIG	Nom de mémoire de traductions dédiée
Langue_source	ls	en	Langue source
Langue_cible	lc	*	Langue cible
Nom_auteur	auteur	«traducteur»	Nom d'auteur: qui a traduit ce segment (système ou personne)
Date	date	2008-04-11 16:25:50	Date de réalisation de cette traduction
Durée	duree	30	Durée de réalisation de cette traduction en secondes
Etoile	etoile	3	Recherche de traductions produites par des auteurs ayant un niveau traductionnel supérieur ou égal au niveau précisé
Note_qualité	note	10	Recherche de traductions ayant une note de qualité supérieure ou égale à une note précise

Table 34 : Paramètres de la fonction de mis à jour de MT

Le résultat renvoyé par cette fonction est un nombre, correspondant à un succès ou à un type d'erreur.

Nous permettons aux applications appelantes d'intégrer l'appel de la fonction de post-édition de SECTra_w dans leurs interfaces. (Figure 66).

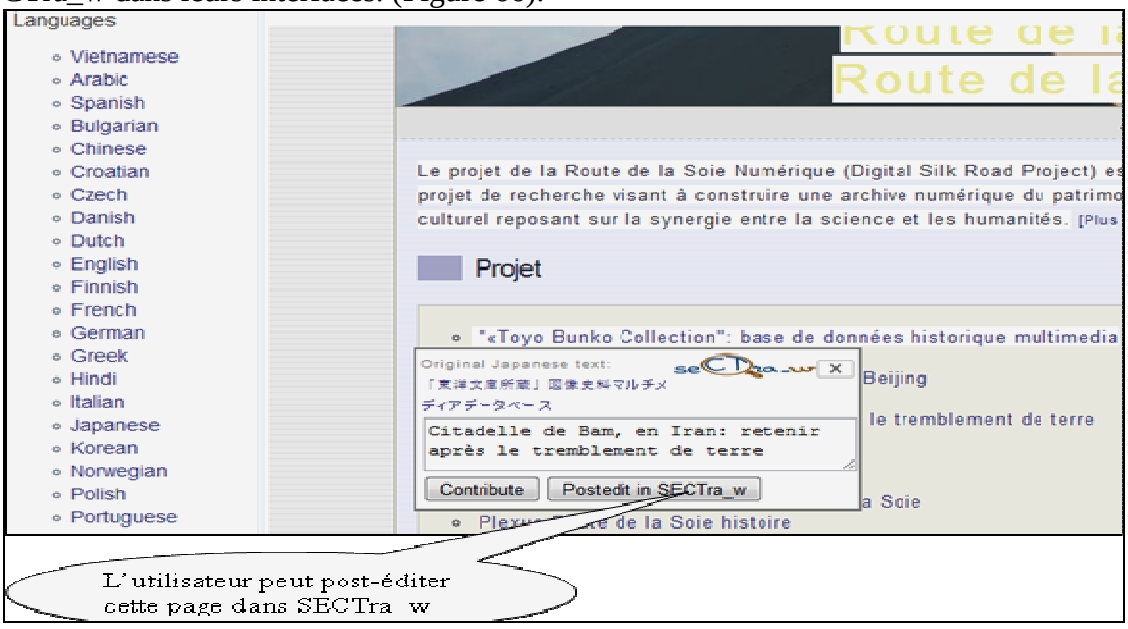


Figure 66: Interface d'une iMAG pour le site DSR

Nous avons construit une architecture en agents à gros gains pour SECTra_w et les autres systèmes (PIVAX, TRADOH++, iMAG, Notepad++, etc.). Actuellement, l'échange entre les agents se fait par le protocole http et le format est la page HTML de résultat. Tous les paramètres de la requête http et le format du résultat HTML sont définis par des conventions précises entre les agents.

Par exemple, voici le fichier de paramètres de SECTra_w par la déclaration d'une commande de préparation d'un minidictionnaire dans PIVAX:

Paramètre	Nom	Valeur par défaut	Note
Protocole	Protocole	http	Protocole de commande
Nom_serveur	server	eolss.imag.fr	Nom de serveur
Port	port	80	Port
URL	url	/pivax?QuickSearch.po	Chemin sur le serveur
Mot_à_chercher	headword	'test'	Mot ou suite de mots à consulter
Délai	delai	100	Temps d'attente maximal. S'il est dépassé, on passe à une autre requête
Lemma	lemma	yes	Demande de lemmatisation préalable, si elle existe
Langue_source	ls	en	Langue source
Langue_cible	lc	*	Langue cible
Segment_ID	segid		Identificateur de segment pour que SECTra_w réattache le résultat à ce segment

Table 35 : Paramètres de commande de préparation de minidictionnaires

Le résultat renvoyé par PIVAX est un fichier HTML sous le format simple de tableau. Dans ce fichier, on déclare les informations à extraire (ex. par XPath) et on l'affiche sur l'interface simplifiée de PIVAX dans SECTra_w.

```
<html>
   <body>
      <table>
           <tr><td lang="en">test<td></tr>
           <tr><td lang="fr" cat="verb">tester<td></tr>
           <tr><td lang="fr" cat="noun">épreuve<td></tr>
           <tr><td lang="fr" cat="noun">essai<td></tr>
           <tr><td lang="fr" cat="noun">test<td></tr>
      </table>
      <div>... autres informations ...</div>
   </body>
</html
```

MOT_SOURCE	/html/body/table/tr/td[lang="en"]	Récupération du mot source
MOT_CIBLE	/html/body/table/tr/td[lang="fr"]	Récupération de la liste des équivalents dans les langues cible

III.3.2 Expérimentation

III.3.2.1 Contexte

Nous avons expérimenté SECTra_w dans le cadre de trois projets réels, ANR OMNIA, iMAG, et MIC-Notepad++.

L'expérimentation de la possibilité de traitement de masses de données par SECTra_w a été faite dans le cadre du projet ANR OMNIA.

Dans les deux projets iMAG et MIC-Notepad++, nous utilisons SECTra_w comme serveur fournissant et gérant des ressources traductionnelles pour les passerelles iMAG et MIC-Notepad++, et comme environnement permettant la post-édition de MT dédiées à ces projets.

Projet ANR OMNIA. Le projet ANR OMNIA [OMNIA, 2009] a pour but la recherche et l'indexation d'images accompagnées de textes. Nous utilisons SECTra_w pour la post-édition d'un gros corpus de légendes de ce projet.

Projet iMAG. Le projet iMAG a pour but la gestion de l'accès multilingue de bonne qualité à des sites Web élus. Dans ce projet, nous avons construit des passerelles appelées iMAG (interactive Multilingual Access Gateway, ou Passerelle Interactive d'Accès Multilingue) pour une trentaine de sites Web, dont celui du LIG (voir Table 36). Chaque iMAG que nous avons construite apparaît comme un Wiki permettant l'accès multilingue à un site Web élu. Essentiellement, l'iMAG fournit une interface interactive permettant aux utilisateurs de voir (et de post-éditer) le site Web élu en plusieurs langues, mais en arrière-plan tous les processus de gestion de la traduction du site Web sont réalisés par SECTra_w et par SegDoc (système de segmentation).

Projet MIC-Notepad++. Le projet MIC-Notepad++ (multilingualisation interne et en contexte) a pour but la localisation et la multilinguïsation de Notepad++, un logiciel en source ouvert écrit en C++. Ce projet est en cours de réalisation par A. Fraisse-Zaïri dans le cadre de sa thèse chez *GETALP* et chez *Winsoft*[43]. Dans ce projet, on permet l'accès multilingue aux interfaces de Notepad en contexte, i.e. pendant qu'on utilise l'application sur son PC. Les segments du "glossaire" (textes de l'IHM et messages) sont d'abord traduits par des systèmes de TA, et puis post-édités par leurs utilisateurs. Comme dans le projet iMAG, SECTra_w est utilisé dans ce projet comme un système de support à la traduction et à la post-édition du logiciel Notepad++.

III.3.2.2 Traitement de masses de données (passage à l'échelle)

Import de divers corpus parallèles bilingues et multilingues. Pour pouvoir faire une expérimentation sur la possibilité de traitement de masses de données par notre système, nous avons importé un corpus de grande taille du projet OMNIA. Ce corpus consiste en 500K images et 500K textes compagnons de 40 mots en moyenne (soit 20M mots et environ 1,5M segments). Nous utilisons SECTra_w pour le support à la traduction de ce corpus vers d'autres langues en appelant des systèmes de TA, et en le post-éditant.

De plus, nous avons importé le grand corpus EuroParl (20M mots en 11 langues, soit 11M segments de 20 mots en moyenne) pour construire une très grande mémoire de traductions dans le but de tester la possibilité de visualisation, de navigation, et de recherche et remplacement dans une grande masse de données.

Visualisation et édition de corpus. Pour la visualisation, SECTra_w permet de diviser un grand corpus en pages logiques de taille paramétrable (par défaut celle d'une page standard). Avec cette solution, on ne rencontre pas de problème de visualisation d'une masse de données.

Nous avons implémenté la solution proposée au §I.2.1.2 (associer chaque position sur l'ascenseur à un ensemble de données, la technique Ajax, le maintien des données chez serveur et client, etc.) pour expérimenter la possibilité de visualisation d'une masse de données dans une interface. Avec cette solution, SECTra_w peut visualiser plus de 100 K segments (soit 1.500.000 mots) dans une interface.

[43] http://www.winsoft-international.com/en/

Pour les opérations d'édition sur un grand corpus, nous utilisons la technologie Ajax qui permet d'intervenir sur une grande page Web de façon incrémentale. Le transfert de données entre le client et le serveur est fait seulement sur les portions de données qui ont été modifiées. Grâce à cette technologie, on ne rencontre plus de problème d'édition sur une masse de données.

III.3.2.3 Support et gestion de mémoires de traductions dédiées aux iMAG

Nous avons introduit le projet iMAG au §III.3.2.1.

Deux « maquettes » d'iMAG avaient été implémentées auparavant, par Mohammad Daoud dans son M2R [Daoud, 2007], et par Carlos Ramisch dans son stage ENSIMAG [Ramisch, 2008]. Cependant, ces deux maquettes n'étaient pas fonctionnelles et ont seulement servi à l'étude du concept d'iMAG proposé par Ch. Boitet et V. Bellynck.

Un premier prototype opérationnel a été implémenté par l'auteur dans le cadre du contrat Baabel de l'ISCC (voir http://eolss.imag.fr/xwiki/bin/view/imag/home). Ce prototype est en cours d'amélioration par la société AXiMAG. Il se présente comme une extension de SECTra_w.

Dans le futur, il faudra impérativement implémenter les iMAG de façon séparée, et ajouter un service Web "relais de traduction" pour contrôler la communication entre SECTra_w et les passerelles iMAG.

Jusqu'à présent, 30 passerelles iMAG ont été créées. Chacune a une mémoire de traductions dans SECTra_w.

LIG laboratory	Digital Silk Road
Danang city	Da Nang University of Technology
TOL	ISCC
Unesco/B@bel	Systran
La Métro	Forum Lyon
Mica	Campus France
GETALP	Floralis
TechniLang	Ordinaide
Homerica	aikicorenc
MT 25 Years On	Essilor
Michelin	Winsoft
ARDI-RA	UNDL-foundation
Getalp presentation	UNESCO_Babel
XD-consulting	ARDI Rhône
Bull support site	LeMonde.fr mobile

Table 36 : Sites Web élus des iMAG

Exploration et mesure de sites Web élus pour initialiser leurs MT dédiées. Nous avons utilisé la technique d'extraction de corpus parallèles à partir de sites Web, comme l'a fait [Do et al., 2009] (§III.1.1.2) et avons récupéré quelques centaines (sur 2000) de segments bilingues français-anglais du site Web du LIG. Nous avons aussi utilisé SECTra_w pour

vérifier et post-éditer les segments bilingues récupérés avant de les recycler pour la traduction de ce site.

Exploitation des MT. Nous avons expérimenté les fonctions (de recherche et de mise à jour) permettant aux iMAG d'exploiter les mémoires de traductions dans SECTra_w. Ces fonctions sont utilisées sous forme du protocole http avec des paramétrages (voir §III.3.1.4). Elles sont facilement intégrées dans les iMAG. La recherche dans la MT est rapide, parce qu'une MT dédiée à un site Web est assez petite et qu'on effectue des recherches exactes et pas approchées.

III.3.2.4 Aspect architectural

Du point de vue de l'architecture, on construit des instances spécifiques de composants de SECTra_w (pour la MT), PIVAX (pour le dictionnaire) et une iMAG dédiée. Il y a une connexion entre ces instances locales avec une instance centrale de données afin d'initialiser les données pour un nouveau site.

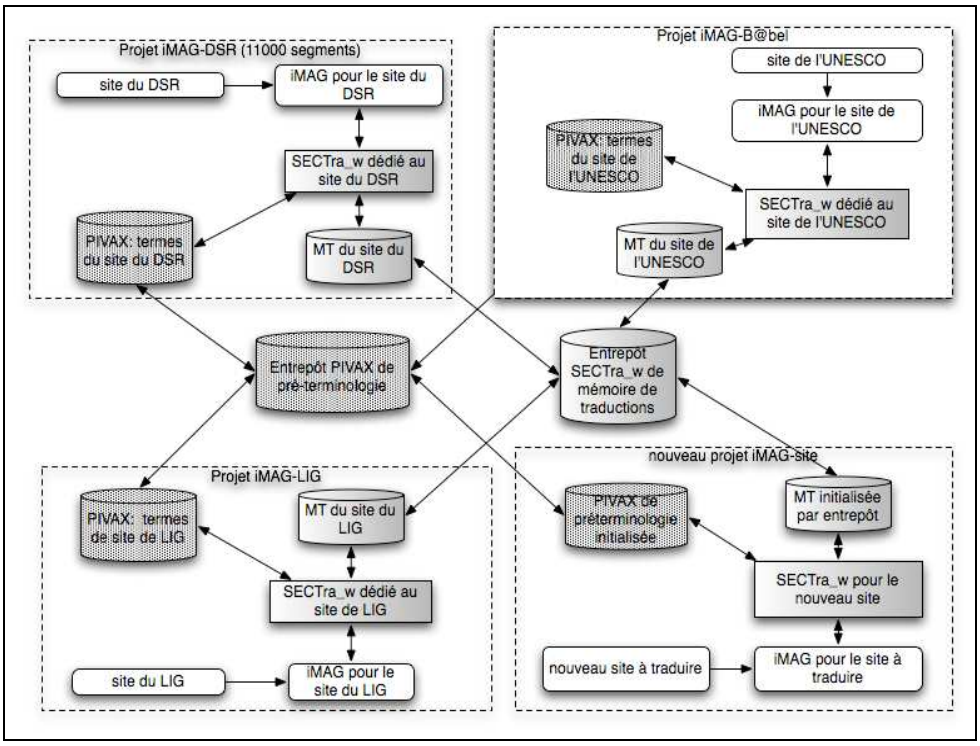


Figure 67: Architecture par agents de SECTra_w, iMAG, PIVAX [Nguyen, 2009]

Dans cette architecture, SECTra_w, PIVAX, et les iMAG communiquent les uns avec les autres pour effectuer des tâches.

SECTra_w communique avec PIVAX pour demander des mini-dictionnaires, et en revanche PIVAX communique avec SECTra_w pour extraire des unités lexicales à partir des corpus.

Les iMAG communiquent avec SECTra_w pour demander des traductions, soit à partir de MT, soit à partir de systèmes de TA, et inversement SECTra_w communique avec les iMAG pour demander des segments source, des fichiers squelette, et des dictionnaires de hors-texte.

III.3.3 Évaluation

Méthodologie

Nous évaluons maintenant SECTra_w dans l'esprit de ce qui précède, c'est à dire en essayant de mesurer dans quelle mesure les problèmes mentionnés ci-dessus ont été résolus par notre implémentation et par la façon de l'utiliser.

Evaluation du problème 3.2 : Normalisation pour les appels à la TA. Nous avons décidé plus haut que la normalisation du segment pour les appels à la TA devait être faite dans TRADOH++. SECTra_w ne traite donc pas ce problème. L'évolution et l'évaluation de TRADOH++ sont en cours, en collaboration avec Hong-Thai Nguyen.

Evaluation du problème 3.3 : Définition et gestion d'une vraie « mémoire de traductions ». Nous avons effectivement modifié l'implémentation initiale de façon à structurer une MT comme un pseudo-document. La MT est constituée de segments-mc.

Nous avons créé et initialisé plusieurs MT dédiées aux passerelles iMAG (voir III.3.2.3). Nous avons aussi construit plusieurs fonctions permettant d'exploiter et de gérer les mémoires de traductions. Ce sont la recherche, la mise à jour, la post-édition, etc., de MT.

Evaluation du problème 3.4 : Programmabilité. Au niveau du paramétrage et de la configuration par les non-informaticiens, nous avons construit SECTra_w pour permettre de mettre en œuvre des paramétrages « externes » :

- Optimisation de l'interface d'évaluation subjective selon les critères d'évaluation d'une campagne, et le réglage de l'interface (nombre de segments, de colonnes, de langues, etc.).
- Filtrage de données post-éditées selon la date, l'auteur, la qualité (niveau traductionel, note de qualité), etc.
- Recherche et remplacement interactive dans une sélection arbitraire (ensemble de segments, ensemble de documents).
- Récupération de données à partir de MT selon des paramétrages.
- Lancement automatique des appels de systèmes extérieurs (TRADOH++, PIVAX, etc.).

Au niveau du paramétrage et la programmation par des contributeurs informaticiens, SECTra_w permet d'écrire facilement des scripts pour le traitement de données, tels que des scripts pour l'import et l'export de corpus. Par exemple, A. Falaise a écrit un script permettant l'appel depuis l'extérieur de la fonction d'import de SECTra_w pour importer un corpus du projet Survitra et un autre du projet OMNIA.

Cependant, nous n'avons pas encore développé les langages (langage de conditions booléennes, langage de flots de travail, langage narratif) dans la version actuelle de SECTra_w. Nous prévoyons de les intégrer dans la prochaine version.

Evaluation du problème 3.5 : Traitement de masses de données. SECTra_w est en cours d'utilisation pour la post-édition du corpus OMNIA, un très grand corpus. Nous avons aussi importé plusieurs corpus (EuroParl, ERIM, BTEC, EOLSS/UNL, etc.) dans SECTra_w.

Evaluation du problème 3.6 : Architecture par agents. Nous avons construit une architecture logicielle dont SECTra_w, PIVAX, les iMAG, TRADOH++ sont les agents. Actuellement, la communication entre ces agents se fait par le protocole http, et le format d'échange est en HTML. Cependant, pour les agents non Web tels que Ariane-G5, nous sommes en cours d'implémentation en utilisant REXX, des REXXstacks, et des systèmes de fichiers (pour les boîtes aux lettres) comme moyens de communication.

Conclusion du chapitre 3

Nous avons étudié dans ce chapitre le problème de la gestion et de l'exploitation de MT. Les systèmes actuels de gestion de MT sont limités, essentiellement parce que la notion même de MT qu'ils utilisent est trop réductrice, voire simpliste. Nous avons présenté des solutions pour lever ces limites en proposant une structure adéquate pour les MT, permettant le stockage, et la gestion de contexte et de qualité de traduction. Nous avons construit plusieurs MT dédiées aux sites Web élus des iMAG, et étudié plusieurs méthodes existantes permettant l'exploitation de sites Web multilingues pour l'initialisation de ces MT dédiées.

En ce qui concerne l'exploitation des appels à la TA, nous avons étudié les problèmes de segmentation générique, multitiple et récursive, et de normalisation des unités de traduction.

Nous avons aussi étudié le problème de la programmabilité dans un système d'exploitation de corpus de traductions. Nous avons vu qu'un tel support est fourni dans quelques systèmes de gestion de corpus, tels que Ariane-G5 et NooJ, de façon encore assez limitée. Nous avons donc proposé plusieurs solutions permettant à différents types d'utilisateurs de programmer à plusieurs niveaux pour l'exploitation de corpus de traductions par SECTra_w.

Nous avons aussi étudié le problème de la construction du système global formé d'un sectra proprement dit, et des systèmes qu'il utilise, dans une architecture par agents. Dans cette architecture, le sectra est un serveur permettant aux autres systèmes d'exploiter ses ressources traductionnelles, et à ses utilisateurs de post-éditer leurs MT, et ainsi qu'à un client de demander les ressources extérieures (prétraductions, informations lexicales, etc.).

Nous avons expérimenté SECTra_w et cette architecture logicielle sur deux projets en cours, iMAG et Notepad++. Jusqu'à présent, 30 passerelles iMAG d'accès multilingue interactif ont été réalisées dans le projet iMAG.

Cependant, il reste encore quelques problèmes qui ne sont pas encore traités complètement. Pour la programmabilité, et pour l'architecture par agents, nous avons proposé plusieurs solutions au niveau des spécifications, mais nous n'avons pas encore implémenté toutes ces spécifications dans la version actuelle.

Conclusions et perspectives

Nous nous sommes intéressé aux défis posés par la conception et la réalisation d'un « système d'exploitation de corpus de traductions », abrégé en « sectra ». Un sectra vise à fournir un support informatique unifié à l'exploitation de corpus de traductions effectuées à la fois par l'humain et par la machine.

Le principe général d'un sectra est inspiré de celui des systèmes d'exploitation Windows, Linux, etc. La réalisation que nous avons proposée, SECTra_w, permet d'exécuter des tâches de base (la gestion des utilisateurs, l'import et l'export, etc.), offre les services les plus importants en TA (l'évaluation de TA, la post-édition, etc.) et fournit une plate-forme facilitant :

- l'intégration de services (accès à des mesures d'évaluation objectives, appel à des systèmes de TA pour produire des pré-traductions, etc.),
- la délégation (la segmentation, la gestion et le traitement des informations lexicales, etc.),
- l'extension à d'autres fonctionnalités, ainsi que des possibilités de paramétrage et de programmation.

En ce qui concerne le support informatique à l'exploitation de corpus par l'humain, nous avons étudié les aspects liés à la mutualisation du travail (évaluation et post-édition collaborative) sur les corpus en contexte multilingue, et les aides informatiques et linguistiques.

En permettant l'exploitation de corpus par la machine, nous nous sommes concentré sur les aspects de la construction d'une architecture logicielle par agents à gros grain, la définition et la gestion de ressources exploitables par les applications novatrices (par exemple, les iMAG, Notepad++, etc.).

Cette thèse a apporté des réponses théoriques et pratiques à trois grands défis. Le premier défi consistait à offrir un support informatique unifié à l'évaluation des systèmes de TA. Le deuxième défi concernait le support contributif et collaboratif au travail humain sur des corpus variés en contexte multilingue. Le troisième défi était la construction d'un support informatique à l'exploitation de corpus de traductions dans des applications novatrices comme l'accès multilingue à des sites Web (iMAG) et la recherche d'informations en contexte multilingue et multimédia (OMNIA).

Plusieurs notions nouvelles ont été précisées (comme *segment multilingualisé et contextualisé*, *pseudo-document*, *métadocument*, etc.), et plusieurs principes généraux (*proactivité*, *délégation*, etc.) ont été introduits. Nous avons dégagé six problèmes associés à chacun de ces trois défis, à dominante conceptuelle (par exemple, définition étendue d'un « contexte » de segment), algorithmique (par exemple, programmabilité du traitement des corpus), et programmatoire (par exemple, traitement de masses de données), que nous avons totalement ou partiellement résolus.

SECTra_w a été construit et expérimenté avec succès dans le cadre de plusieurs projets réels d'évaluation de TA, de post-édition, et de multilinguïsation de sites Web et d'applications grand public en source ouvert.

Perspectives

Bien que nous ayons déjà traité plusieurs problèmes complètement, il reste encore des problèmes partiellement traités ou pas du tout traités.

Notre travail de thèse ouvre donc des perspectives intéressantes à différents niveaux.

A court terme. Au niveau conceptuel, nous continuerons à traiter les problèmes liés à l'aspect générique de la définition des corpus, et à la segmentation générique, multiple, et récursive.

Au niveau programmatoire, nous implémenterons les flux de travaux (WF) facilitant l'organisation des participants et des tâches comme nous l'avons proposé au §I.2.3.2. Nous continuerons à importer plusieurs grands corpus (tout le corpus EuroParl, B@bel Unesco, etc.) et implémenterons le second algorithme de parcours et de visualisation de masse de données que nous avons proposé au §I.2.2.2. Enfin, nous continuerons à rendre SECTra_w plus générique, pour que des organisateurs de projets puissent facilement organiser des campagnes d'évaluation, et des projets de post-édition.

A plus long terme. Nous désirons faire évoluer SECTra_w pour qu'il devienne un sectra complet, implémentant l'intégralité des solutions et autres idées proposées dans cette thèse. Pour cela, il faudra permettre une programmabilité à différents niveaux (paramétrage, définition de sélections, programmation, etc.), pour plusieurs types d'utilisateurs (les non-informaticiens, les contributeurs informaticiens). Il offrira des LSPL (langage d'affectations, langage de conditions booléennes permettant la définition de sélections de données, langage de flux de travaux, etc.) permettant de faciliter l'exploitation des corpus par des non-informaticiens. Il contiendra divers outils permettant de traiter divers types de corpus.

Il faudra aussi mettre en place une API « souple » (générique), pour permettre l'accès à de nouveaux types de services. Nous pensons par exemple à des segmenteurs pour des langues peu dotées comme le vietnamien, le lao et le thaï, le khmer, etc.

Nous pensons aussi que SECTra_w pourait être utilisé comme un système de THAM (Traduction Humaine Aidée par la Machine) permettant de traduire des corpus de documents ou des sites Web en exploitant d'importantes ressources traductionnelles, et son éditeur de traductions.

Perspective personnelle. A la suite de cette thèse, j'aimerais faire un stage postdoctoral d'un ou deux ans au GETALP ou ailleurs pour poursuivre mes recherches. Ensuite, je rentrerai au Vietnam pour continuer mon enseignement et ma recherche à l'Ecole Polytechnique de Danang, et pour contribuer au renforcement du laboratoire DATIC, en coopérant avec le laboratoire MICA (Hanoi), le GETALP, et d'autres laboratoires en France et ailleurs.

Bibliographie

1. [Aalst, 1999] Aalst V-D. (1999) *The application of Petri Nets to workflow management.* The Journal of Circuits, Systems and Computers, pp. 21–66.
2. [Agirre, 2000] Agirre E., Arregi X., Artola X., Diaz De Ilarraza A., Sarasola K., Soroa A. (2000) *A Methodology for Building Translator-oriented Dictionary Systems.* Machine Translation. Vol. 15, pp. 295-310.
3. [Akiba et al., 2004] Akiba Y., Federico M., Kando N., Nakaiwa H., Paul M., Tsujii J-I. (2004) *Overview of the IWSLT04 Evaluation Campaign.* Proc. IWSLT 2004, Kyoto, Japan, September 30-October 1, 2004, vol. 1/1, 12 p.
4. [Albatal, 2005] Albatal R. (2005) *La prise en compte des flux de travaux pour la construction collaborative des bases lexicales multilingues.* Mémoire de M2R, GETALP, LIG, UJF (Grenoble 1), 62 p.
5. [Allen et Hogan, 2000] Allen J., Hogan C. (2000) *Toward the development of a post-editing module for Machine Translation raw output: a new productivity tool for processing controlled language.* In Proc. 3rd International Workshop on Controlled Language Applications (CLAW), Seattle, Washington, pp. 62-71.
6. [Allen, 2001a] Allen J. (2001a) *Postediting: an integrated part of a translation software program.* In Language International magazine, April 2001, Vol. 13, No. 2, pp. 26-29.
7. [Allen, 2001b] Allen J. (2001b) *Post-editing or no post-editing?* In International Journal for Language and Documentation, Issue 8, December 2000/January 2001. pp. 41-42.
8. [ALPAC, 1966] ALPAC (1966) *ALPAC Language and Machine : Computers in Translation and Linguistics, n° 1416. November 1966.* Automatic Language Processing Advisory Committee, Division of Behavioral Sciences, National Academy of Science - National Research Council. 124 p.
9. [Al-Adhaileh et Kong, 1999] Al-Adhaileh M., Kong T-E. (1999) *Example-Based Machine Translation Based on the Synchronous SSTC Annotation Schema.* Proceedings of the Machine Translation Summit VII. Singapore, pp. 244-249.
10. [Al-Assimi, 2000] Al-Assimi A-B. (2000) *Gestion de l'évolution non centralisée de documents parallèles multilingues.* Thèse UJF, Grenoble, 200 p.
11. [Al-Assimi, 2001] Al-Assimi A-B., Boitet C. (2001) *Management of Non-Centralized Evolution of Parallel Multilingual Documents.* Proc. Internationalization Track, 10th International World Wide Web Conference, Hong Kong, May 1-5, 2001, 7 p.
12. [Augar, 2004] Augar N., Raitman R., Zhou W. (2004) *Teaching and Learning Online with Wikis.* In Proceedings of the 21st Australasian Society of Computers. Learning in Tertiary Education Conference, ASCILITE-04. Australia, pp. 95-104.
13. [Aymerich et Camelo, 2006] Aymerich J., Camelo H. (2006) *Post-Editing of MT Output in a Production Setting: Experiences at the Pan American Health Organization.* AMTA 2006, Cambridge, MA, 36 transparents.
14. [Aymerich et Camelo, 2009] Aymerich J., Camelo H. (2009) *The machine translation maturity* model at PAHO. Proceedings of the twelfth Machine Translation Summit, Ottawa, Ontario, Canada, pp. 403 - 409.
15. [Ball, 2003] Ball S. (2003) *Joined-up Terminology - The IATE system enters production.* In Proceedings of the 25th International Conference on Translating and the Computer. London, UK, Vol. 5, 5 p.

16. [Bastien, 1993] Bastien J.M.C., Scapin D.L. (1993) *Critères ergonomiques pour l'évaluation d'interfaces utilisateur (version 2.1)*. Technical report N° 156, May 1993. INRIA. Programme 3 Artificial intelligence, cognitive systems, and man-machine interaction.

17. [Bellynck, 1999] Bellynck V. (1999) *Introduction d'une vue textuelle synchronisée avec la vue géométrique primaire dans Cabri-II.* Thèse UJF (Grenoble 1), 250 p.

18. [Bellynck, 2001] Bellynck V. (2001) *Langage « narratif » : 3 exemples pour convaincre*. IHM-HCI 13èmes Journées sur l'ingénierie de l'Interaction Homme-Machine (AFIHM), 15th annual conference of Human-Computer Interaction (group of British Computer Society), Poster session, Lille, France, septembre 2001, Vol.2, pp. 171-174.

19. [Bellynck et al., 2005] Bellynck V., Boitet C., Kenwright J. (2005) *ITOLDU, a Web Service to Pool Technical Lexical Terms in a Learning Environment and Contribute to Multilingual Lexical Databases.* Alexander F. Gelbukh (Ed.): Proceedings of Computational Linguistics and Intelligent Text Processing, 6th International Conference, CICLing 2005, Mexico City, Mexico, February 13-19 2005, pp. 324-332.

20. [Bertoldi et al., 2007] Bertoldi N., Zens R., Federico M. (2007) *Speech translation by confusion network decoding.* In Proceedings on the 32nd IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, Hawaii, USA, April 2007, pp. 1297-1300.

21. [Bey et al., 2005] Bey Y., Kageura K., Boitet C. (2005) *A Framework for Data Management for the Online Volunteer Translators' Aid System QRLex.* In Proceedings of the 19th Pacific Asia Conference on Language, Information and Computation (PACLIC19). Taipei, Taiwan, pp. 51-60.

22. [Bey, 2009] Bey Y. (2009) *Aides informatisées à la traduction collaborative bénévole sur le Web.* Thèse UJF, GETALP, LIG, Grenoble, 265 p.

23. [Bigi et Le, 2008] Bigi B., Le V-B. (2008) *Normalisation et alignement de corpus français et vietnamien : format et logiciels*. Journées internationales d'Analyse statistique des Données Textuelles (JADT'08), Lyon, France, 8 p.

24. [Blanc, 1999] Blanc É. (1999) *An interactive hypertextual environment for MT development.* Journal of Chinese Language and Computing, Communication of COLIPS (Chinese and Oriental Languages Information Processing Society), Volume 9, Issue 1, pp. 67-81.

25. [Blanc, 2001] Blanc É., Sérasset G. (2001) *From Graph to Tree: Processing UNL Graphs using an Existing MT System.* In Proceedings of the first UNL open Conference, Suzhou, China, 10 p.

26. [Blanchon, 1994] Blanchon H., Boitet C. (1994) *Multilingual dialogue-based MT for monolingual authors : the LIDIA Project and a first mockup.* Machine translation ISSN 0922-6567, vol. 9, no. 2, pp. 99-132.

27. [Blanchon, 1999] Blanchon H. (1999) *CLIPS++ Contribution within the C-STAR II Consortium.* Proc. C-STAR Workshop, Schwetzingen (Germany), 6 p.

28. [Blanchon et al., 1999] Blanchon H., Boitet C., Caelen J. (1999) *Participation francophone au consortium C-STAR II.* La Tribune des Industries de la Langue et du Multimédia, Vol. 31-32 (August-December 1999) : pp. 15-23.

29. [Blanchon et al., 2004] Blanchon H., Boitet C., Besacier L. (2004) *Spoken dialogue translation systems evaluation: results, new trends, problems and proposals.* Interspeech 2004: ICSLP Satellite Workshop, Proceedings, pp. 95-102.

30. [Blanchon et al., 2004] Blanchon H., Boitet C., Brunet-Manquat F., Tomokiyo M., Hamon A., Vo-Trung H. and Bey Y. (2004) *Towards Fairer Evaluations of Commercial MT*

Systems on BTEC corpora. In Proceedings of the International Workshop on Spoken Language Translation (IWSLT-04). Kyoto, Japan, pp. 21-26.

31. [Blanchon et Boitet, 2007] Blanchon H., and Boitet C. (2007) *Pour l'évaluation externe des systèmes de TA par des méthodes externes fondées sur la tâche.* TAL, vol. 48, 32 p.

32. [Blatt, 1998] Blatt A. (1998) *EURAMIS: added value by integration.* Terminologie et Traduction, 1, Commission des Communautés européennes, Bruxelles, pp. 59-73.

33. [Bläter, 1992] Bläter B., Schwall U., Storrer A. (1992) *A Reusable Lexical Database Tool For Machine Translation.* Proc. of COLING-92, Nantes, Aug. 23-98, 1992, pp. 510-517.

34. [Boitet, 1990] Boitet C. (1990) *Towards Personal MT: general design, dialogue structure, potential role of speech.* Proc. COLING-90, Helsinki, Finland, August 20-25, 1990, vol. 3/3, pp. 30-35.

35. [Boitet, 1993a] Boitet C. (1993a) *Crucial open problems in Machine Translation & Interpretation.* Proc. Symposium on Natural Language Processing in Thailand, Bangkok, 29 p.

36. [Boitet, 1993b] Boitet C. (1993b) *TA et TAO à Grenoble... 32 ans déjà!*. T.A.L (revue semestrielle de l'ATALA), vol.33/1-2, Numéro spécial du Trentenaire, pp. 45-84.

37. [Boitet et Seligman, 1994] Boitet C., Seligman M (1994) *The "Whiteboard" Architecture: A Way to Integrate Heterogeneous Components of NLP Systems.* COLING 1994, pp. 426-430.

38. [Boitet, 1996] Boitet C. (1996) *La synergie entre THAM, réseau et TA comme facteur de progrès théoriques et pratique en TAO.* NLP+IA 96 / TAL+AI 96, Université de Moncton, Canada, 12 p.

39. [Boitet, 2003] Boitet C. (2003) *Approaches to enlarge bilingual corpora of example sentences to more languages.* In Proceedings of Papillon-03 seminar, Hokkaido university, Sapporo, 3-5 July 2003, 13 p.

40. [Boitet, 2004] Boitet C. (2004) *Progress report on building the French BTEC and participating in the MT evaluation campaign (CSTAR project).* (rapport pour ATR), GETA, CLIPS, & ATR, 10 p.

41. [Boitet, 2005] Boitet C. (2005) *Gradable Quality Translation Through Mutualization of Human Translation and Revision, UNL-based MT and Coedition.* Research in Computing Science, IPN-CIC. Mexico. (presented at the 2nd Workshop on UNL and Other Interlinguas) (Gelbukh, A. ed.). In Book "Universal Networking Language, advances in theory and applications", Mexico, pp. 393-410.

42. [Boitet et al., 2006] Boitet C., Bey Y., Tomokiyo M., Cao W., and Blanchon H. (2006) *IWSLT-06: Experiments with Commercial MT Systems and Lessons from Subjective Evaluations*. In Proceedings of the International Workshop on Spoken Language Translation. Kyoto, Japan, pp. 23-30.

43. [Boitet, 2007] Boitet C. (2007) *Corpus pour la TA : types, tailles, et problèmes associés, selon leur usage et le type de système*. Revue française de linguistique appliquée. Vol. XII – 2007, pp. 25-38.

44. [Boitet et al., 2007] Boitet C., Boguslavskij i., Cardeñosa I. (2007) *An Evaluation of UNL Usability for High Quality Multilingualization and Projections for a Future UNL++ Language.* Proc. of Computational Linguistics and Intelligent Text Processing (CICLING-2007), February 18 to 24, 2007, Mexico City, Mexico, pp. 361-373.

45. [Boitet, 2008] Boitet C. (2008) *Les architectures linguistiques et computationnelles en traduction automatique sont indépendantes.* JEP-TALN 2008, Avignon, 9-13 juin 2008, 10 p.

46. [Boyer et Moore, 1977] Boyer R. S., Moore J. S. (1977) *A fast string searching algorithm.* Communications of the Association for Computing Machinery, 20 p.

47. [Brown et al., 1991] Brown P.F., Lai J.C., Mercer R.L. (1991) *Aligning sentences in parallel corpora.* Proceedings of 47th Annual Meeting of the Association for Computational Linguistics, 8 p.

48. [Bryant, 2005] Bryant S. L., Forte A., Bruckman A. (2005) *Becoming Wikipedian: Transformation of Participation in a Collaborative Online Encyclopedia.* In Proceedings of the GROUP International Conference on Supporting Group Work. Sanibel Island, Florida, US., pp. 1-10.

49. [Brunet-Manquat et Sérasset, 2006] Brunet-Manquat F., Sérasset G. (2006) *Création d'une base terminologique juridique multilingue à l'aide de la plate-forme générique Jibiki : le projet LexALP*. Communication présentée à la Conférence TALN2006, 10-13 avril 2006, Louvain la Neuve (B), 8 p.

50. [Buffa et Gandon, 2006] Buffa M., Gandon F. (2006) *SweetWiki: semantic web-enabled technologies in Wiki.* In Proceedings of the International Symposium Wikis 2006, pp. 69-78.

51. [Buffa, 2006] Buffa M. (2006) *Intranet Wikis.* In Proceedings of the Intranet Web Workshop (WWW Conference). Edinburgh, Scotland, UK, pp. 18-28.

52. [Burghart, 2004] Burghart M. (2004) *Annotation collaborative d'un corpus de documents médiévaux : outils pour l'analyse de la structure et du contenu des sermons de Jacques de Voragine.* Le Médiéviste et l'ordinateur, 43 p.

53. [Callison-Burch et al., 2006] Callison-Burch C., Osborne M., and Koehn P. (2006) *Reevaluating the Role of BLEU in Machine Translation Research*. In Proc. EACL-06, Trento, ITC/irst, ed., 8 p.

54. [Carl et al., 2002] Carl M., Way A., Schäler R. (2002) *Toward a Hybrid Integrated Translation Environment.* Proceedings of AMTA (The Association for Machine Translation in the America), October 8-12, 2002, Tiburon, California, pp. 11-21.

55. [Carpena, 2004] Carpena V. (2004) *WICALE, une interface cliente générique pour le pilotage de serveurs linguistiques.* Mémoire d'ingénieur CNAM, GETA, CLIPS, IMAG, 86 p.

56. [Chappuy, 1983] Chappuy S. (1983) *Formalisation de la description des niveaux d'interprétation des langues naturelles. Etude menée en vue de l'analyse et de la génération au moyen de transducteurs.* Thèse de 3ème cycle, INPG, Grenoble, 227 p.

57. [Chen, 2000] Chen J., Nie J.-Y. (2000) *Automatic Construction of Parallel English-Chinese Corpus for Cross-Language Information Retrieval.* Seattle, Washington, USA, pp. 21-28.

58. [Chenon, 2005] Chenon C. (2005) *Vers une meilleure utilisabilité des mémoires de traduction, fondée sur un alignement sous-phrastique.* Thèse UJF, GETA, CLIPS, IMAG, Grenoble, 221 p.

59. [Chris et Robert, 1998] Chris M., Robert D. (1998) *Evaluation in the context of natural language generation.* Computer Speech and Language 12, pp. 349-373.

60. [Cromières, 2010] Cromières F. (2010) *Vers un plus grand lien entre alignement, segmentation et structure des phrases.* Thèse, Laboratoire d'Informatique de Grenoble (LIG), GETALP, UJF, 310 p.

61. [Colmerauer, 1967] Colmerauer A. (1967) *Les Systèmes-Q: un formalisme pour analyser et synthétiser des phrases sur ordinateur*. Groupe TAUM, Université de Montréal, Montréal, Canada, 28 p.

62. [Costa et Panissod, 2003] Costa J-C., Panissod C. (2003) *SYSTRAN Review Manager*. MT Summit IX, New Orleans, Louisiana, USA, 4 p.

63. [Cunningham et al., 2003] Cunningham H., Tablan V., Bontcheva K., Dimitrov M. (2003) *Language engineering tools for collaborative corpus annotation*. Proceedings of Corpus Linguistics, 9 p.

64. [Damerau, 1964] Damerau F. J. (1964) *A technique for computer detection and correction of spelling errors.* in Communication of the ACM. vol. 7(3): pp. 171-176.

65. [Daoud, 2007] Daoud M. (2007) *Towards interactive Multilingual Access Gateways (iMAG*). Rapport de M2R, UJF, Grenoble, France, 60 p.

66. [Daoust, 2006] Daoust F., Yves M. (2006) *Logiciels d'analyse textuelle : vers un format XML-TEI pour l'échange de corpus annotés.* Actes des 8ème Journées Internationales d'Analyse statistique des Données Textuelles (SADT-06). Besançon, France : Presses Universitaires de Franche-Comté, pp. 327-340.

67. [Désilets, 2006] Désilets A., Gonzalez L., Paquet S., Stojanovic M. (2006) *Translation the Wiki Way*. In Proceedings of the WIKISym 2006. NRC 48736. Odense, Denmark, pp. 19-32.

68. [Dinh et al., 2008] Dinh Q-T., Le H-P., Nguyen T-M-H., Nguyen C-T., Rossignol M., Vũ X-L. (2008) *Word Segmentation of Vietnamese Texts: a Comparison of Approaches.* In Proceedings de conférence de LREC 2008, Marrakech, Maroc, 4 p.

69. [Do et al., 2009] Do T.N.D, Le V.B, Bigi B., Besacier L., Castelli E. (2009) *Exploitation d'un corpus bilingue pour la création d'un système de traduction probabiliste vietnamien - français.* TALN'09-Conférence sur le Traitement Automatique des Langues Naturelles, Paris, 10 p.

70. [Doan-Nguyen, 1998] Doan-Nguyen H. (1998) *Techniques génériques d'accumulation d'ensembles lexicaux structurés à partir de ressources dictionnairiques informatisées multilingues hétérogènes.* Thèse INPG, GETA, CLIPS, IMAG, Grenoble, 180 p.

71. [Durand, 1988] Durand J-C. (1988) *TTEDIT: un éditeur transformationnel d'arbres décorés.* Thèse UJF (Grenoble 1), mars 1988, 120 p.

72. [Erl, 2005] Erl T. (2005) *Service-Oriented Architecture (SOA): Concepts, Technology, and Design.* Prentice Hall PTR (August 12, 2005), ISBN-10: 0131858580. 792 p.

73. [Fafiotte, 2004] Fafiotte G. (2004) *Interprétariat à distance et collecte de dialogues spontanés bilingues, sur une plate-forme générique multifonctionnelle.* in Actes de TALN 2004, 8 p.

74. [Fluhr et al., 2006] Fluhr C., Möellic P-A., Hede P. (2006) *Usage-Oriented Multimedia Information Retrieval Technological Evaluation.* Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR 2006, October 26-27, 2006, Santa Barbara, California, US, pp. 301-306.

75. [Gale et Church, 1991] Gale W.A, Church K.W. (1991) *A program for aligning sentences in bilingual corpora*. Proceedings of the 29th annual meeting on Association for Computational Linguistics, June 18-21, 1991, Berkeley, California, pp. 177-184.

76. [Gotti et al., 2006] Gotti F., Langlais P., Coulombe C. (2006) *Vers l'intégration du contexte dans une mémoire de traduction sous-phrastique : détection du domaine de traduction.* In Proceedings of the Conference sur le Traitement Automatique des Langues Naturelles (TALN). Leuven, Belgium, pp. 483-492.

77. [Gow, 2003] Gow F. (2003) *Metrics for Evaluating Translation Memory Software.* In Partial Fulfillment of the Requirements for the Degree of MA (Translation). School of Translation and Interpretation University of Ottawa, Faculty of Graduate and

Postdoctoral Studies of the University of Ottawa, Ottawa, Canada: University of Ottawa, 135 p.

78. [Guerra, 2003] Guerra L.M. (2003) *Human Translation versus Machine Translation and Full Post-Editing of Raw Machine Translation Output.* Master's Thesis, Dublin City University, 137 p.

79. [Guilbaud, 1988] Guilbaud J-P. (1988) *Projet Eurotra: Réalisation en Ariane d'un transfert des structures produites par l'analyseur du français de B'VITAL vers des structures interface IS Eurotra.* Rapport de contrat, GETA, juin 1988, 205 p.

80. [Guillaume, 2003] Guillaume P. (2003) *Le réseau LIDIA.* Document interne GETA, octobre 2003, 5 p.

81. [Hajlaoui et Boitet, 2003] Hajlaoui N., Boitet C. (2003) *A pivot XML-based architecture for multilingual, multiversion documentsnobreakspace: parallel monolingual documents aligned through a central correspondence descriptor and possible use of UNL.* Convergences'03, Alexandria, 2-6 December 2003, 17 p.

82. [Hamon, 2007] Hamon O. (2007) *Campagne d'Evaluation des Systèmes de Traduction Automatique.* Rapport du Projet CESTA, ELDA/ELRA, Paris, 98 p.

83. [Hamon et al., 2008] Hamon O., Popescu-Belis A., Hartley A., Mustafa El Hadi W., Rajman M. (2008) *CESTA : la campagne d'évaluation des systèmes de traduction automatique.* Chapitre 4 du livre intitulé « l'évaluation des technologies de traitement de la langue: les campagnes Technolangue », 24 p.

84. [Heyn, 1996] Heyn M. (1996) *Integrating machine translation into translation memory systems.* In: EAMT Workshop, pp. 111-124.

85. [Hovy et al., 2002] Hovy E., King M., Popescu-Belis A. (2002) *Principles of Context-Based Machine Translation Evaluation.* Publisher: Springer, Machine Translation, Volume 17, Number 1, pp. 43-75.

86. [Hutchins, 1992] Hutchins W-J., Somers H-L. (1992) *An introduction to machine translation.* London: Academic Press, 1992. [ISBN: 0-12-362830-X], 362 p.

87. [Jacobson, 2002] Jacobson M. (2004) *Les outils modernes pour la transcription de corpus de parole*. Revue Parole 22, 23, 24, pp. 213-229.

88. [Jacobson, 2004] Jacobson M. (2004) *Gestion de corpus oraux annotés : méthodes et outils*. Journées d'étude sur la parole (JEP), Fez, Maroc, 19-22 avril 2004, actes pp. 73-76.

89. [James et Bruce, 1997] James L., Bruce P. (1997) *Developing and empirically evaluating robust explanation generators: the KNIGHT experiments.* Computational Linguistics, pp. 65-103.

90. [JEIDA, 1992] JEIDA (1992) *JEIDA Methodology and Criteria on Machine Translation Evaluation. November, 1992.* Japan Electronic Industry Development Association. 129 p.

91. [John, 2006] John M. (2006) *The OpenWFE Manual, Open source workflow Environment.* Lausanne, Suisse, 2006, 197 p.

92. [Kay et Roscheisen, 1993] Kay M., Roscheisen M. (1993) *Text - translation alignment.* Association for Computational Linguistics.

93. [Kikui et al., 2003] Kikui G., Sumita E., Takezawa T., Yamamoto S. (2003) *Creating Corpora for Speech-to-Speech Translation.* Proc. of European Conference on Speech Communication and Technology, 2003, pp. 381–382.

94. [Kit et Wong, 2008] Kit C., Wong T-M. *Comparative Evaluation of Online Machine Translation Systems with Legal Texts.* Law Library Journal, Vol 100(2), pp. 299-322.

95. [Kitamura et al., 2003] Kitamura M., Murata T., Sukehiro T., Shimohata S., Sasaki M., Matsunaga T., Nakagawa T. (2003) *Technology and Development on Collaborative Translation Environment "Yakushite.Net"*. In Proceedings of IPSJ-03, 65th Annual Conference of Information Processing Society of Japan (IPSJ). Vol. 5, pp. 319-322.

96. [Knuth et al, 1977] Knuth D. E., Morris J. H., Pratt V. R. (1997) *Fast pattern - matching in strings.* SIAM J. Comput., 6 p.

97. [Koehn, 2005] Koehn Ph. (2005) *Europarl: A Parallel Corpus for Statistical Machine Translation*. In Proc. of the 10th Machine Translation Summit, Phuket, Thaïland, pp. 79–86.

98. [Koehn et al., 2007] Koehn P., Hoang H., Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C. J., Bojar O., Constantin A., Herbst E. (2007). *Moses: open source toolkit for statistical machine translation.* In Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Prague, Czech Republic, pp. 177–180.

99. [Koehn, 2009] Koehn P. (2009) *A Web-Based Interactive Computer Aided Translation Tool.* In Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, Suntec, Singapore, pp. 17-20.

100. [Kraif, 2001] Kraif O. (2001) *Constitution et exploitation de bi-textes pour l'aide à la traduction.* Thèse UNSA (Université de Nice Sophia Antipolis), 549 p.

101. [Lafourcade, 1998] Lafourcade M., Chauché J. (1998) *Ficus - un agent dictionnaire coopératif et extensible*. NLP+IA'98, August 18-21, 1998, Moncton, New-Brunswick, Canada, 8 p.

102. [Laurian, 1984] Laurian A.M. (1984) Machine Translation: What Type of Post-Editing on What Type of Documents for What Type of Users. In Proc. COLING-84. Stanford (Ca.). pp. 236-238.

103. [Leuf, 2001] Leuf B., Cunningham W. (2001) *The Wiki Way: Quick Collaboration on the Web.* Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©2001 ISBN: 0-201-71499-X, 435 p.

104. [Levenshtein, 1966] Levenshtein V. I. (1966) *Binary codes capable of correcting deletion, insertions and reversals*. Soviet Physics Doklady, vol. 10, n° 8, 1966, pp. 707-710.

105. [Mathias et Byrne, 2006] Mathias L., Byrne W. (2006) *Statistical phrase-based speech translation*. in IEEE Conference on Acoustics, Speech and Signal Processing, 2006, 6 p.

106. [Mangeot, 2001] Mangeot M. (2001) *Environnements centralisés et distribués pour lexicographes et lexicologues en contexte multilingue.* Thèse UJF (Grenoble 1), 296 p.

107. [Manquat et al., 2006] Manquat B., Francis B-M., Sérasset G. (2006) *Création d'une base terminologique juridique multilingue à l'aide de la plate-forme générique Jibiki : le projet LexALP*. CLIPS, Grenoble, France, avril 2006, 8 p.

108. [Matusov et al., 2005] Matusov E., Kanthak S., Ney H. (2005) *On the integration of speech recognition and statistical machine translation.* In Proceedings of the 9th European Conference on Speech Communication and Technology (Interspeech), September 2005, 4 p.

109. [Meghan et al., 2008] Meghan L-G., Stephanie S., Lauren F., Haejoong L., and Shawn M. (2008) *Management of Large Annotation Projects Involving Multiple Human Judges: a Case Study of GALE Machine Translation Post-editing.* In Proc of the Sixth International Language Resources and Evaluation, Marrakech, Morocco, 4 p.

110. [Merkel, 1999] Merkel M. (1999) *Understanding and enhancing translation by parallel text processing.* Linköping studies in science and technology, dissertation no. 607,

Linköping University, Department of Computer and Information Science, Linköping, Sweden, 1999.

111. [Michael et al., 2004] Michael P., Hiromi N., Marcello F. (2004) *Towards Innovative Evaluation Methodologies for Speech Translation*. Working Notes of the NTCIR-4 2004 Meeting, Supplement Volume 2, Tokyo, Japan, 2004, pp. 17-21.

112. [Morris, 1970] Morris J.H., Pratt R. (1970) *A linear pattern-matching algorithm.* Rapport technique, University of California, Berkeley, 40 p.

113. [Munteanu et Marcu, 2006] Munteanu D.S., Marcu D. (2006) *Extracting parallel sub-sentential fragments from non-parallel corpora*. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 81-88.

114. [Nguyen, 2009] Nguyen H-T. (2009) *Des systèmes de TA homogènes aux systèmes de TAO hétérogènes.* Thèse UJF, GETA, CLIPS, IMAG, Grenoble, 241 p.

115. [Papineni et al., 2002] Papineni K., Roukos S., Ward T., Zhu V. (2002) *BLEU : a Method for Automatic Evaluation of Machine Translation.* Proc. ACL-02, Philadelphia, USA, July 7-12, 2002, vol. 1/1, pp. 311-318.

116. [Patry et al., 2007] Patry A., Langlais P., Béchet F. (2007) *MISTRAL: a Lattice Translation System for IWSLT 2007*. Proceedings of IWSLT 2007, Genoa, Italy, 2007.

117. [Paul, 2009] Paul M. (2009) *Overview of the IWSLT 2009 Evaluation Campaign*. Proceedings of IWSLT 2009, Tokyo – Japan, 18 p.

118. [Phan, 1991] Phan H-K. (1991) *Contribution à l'informatique multilingue. Extension d'un éditeur de documents structurés.* Thèse de Doctorat, Université des sciences et techniques de Lille Flandres Artois, mai 1991.

119. [Pouliquen, 2008] Pouliquen B. (2008) *Similarity of Names Across Scripts: Edit Distance Using Learned Costs of N-Grams.* Proceedings of Advances in Natural Language Processing, 6th International Conference, GoTAL 2008, pp. 405-416.

120. [Pouliquen, 2003] Pouliquen B., Ralf S., Camelia I. (2003) *Automatic Identification of Document Translations in Large Multilingual Document Collections.* In proceedings of the international conference recent advances in natural language processing (RANLP'2003), Borovets, Bulgaria, 10 - 12 September 2003, 8 p.

121. [Przybocki et al., 2006] Przybocki M., Sanders G., Le A. (2006) *Edit Distance: A Metric for Machine Translation Evaluation.* Proc. LREC 2006. Genoa, Italy. May 24-26, 2006. pp. 2038-2043.

122. [Quan et al., 2005] Quan V-H., Federico V., Cettolo. M.. (2005) *Integrated n-best re-ranking for spoken language translation.* In Interspeech 2005 - Eurospeech - 9th European Conference on Speech Communication and Technology, 4 p.

123. [Ramisch, 2008] Ramisch C. (2008) *Développement d'un site Web iMAG générique et instanciation sur des sites iMAG concrets.* Rapport de projet de fin d'études, INP Grenoble-ENSIMAG, 44 p.

124. [Resnik, 1999] Resnik P. (1998) *Parallel Strands: A Preliminary Investigation into Mining the Web for Bilingual Text.* Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup, pp.72-82.

125. [Resnik, 1999] Resnik P. (1999) *Mining the Web for bilingual text.* Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, Maryland, USA, pp. 527-534.

126. [Rouillard et al., 2007] Rouillard J., Vantroys T., Chevrin V. (2007) *Les architectures orientées service. Une approche pragmatique des SOA.* Editions Vuibert, Collection : Génie Logiciel, ISBN-10: 2711748685, 317 pages.

127. [Rouillard et al., 2006] Rouillard J. (2006) *Web services and speech-based applications*. ICPS'06, IEEE International Conference on Pervasive Services, Lyon, 2006, 4 p.

128. [Saleem et al., 2004] Saleem S., Jou S-C., Vogel S., Schultz T (2004) *Using word lattice information for a tighter coupling in speech translation systems.* In Proceedings ICSLP, Jeju Island, Korea, Oct 2004, 4 p.

129. [Satayamas et al., 2007] Satayamas V., Boitet C., Kawtrakul A. (2007) *AnnotEd-w, a specialized editor for annotating word boundaries collaboratively.* In Proceedings of the seventh international Symposium on Natural Language Processing (SNLP) 2007, Pattaya, Chonburi, Thailand, 6 p.

130. [Scapin, 1997] Scapin D.L., Bastien J.M.C. (1997) *Ergonomic criteria for evaluating the ergonomic quality of interactive systems.* Behaviour and Information Technology, 6 (4-5), pp. 220-231.

131. [Schwab, 2006] Schwab D. (2006) *Approche hybride-lexicale et thématique - pour la modélisation, la détection et l'exploitation des fonctions lexicales en vue de l'analyse sémantique de texte*. Thèse, Université Montpellier II, 390 p.

132. [Schwab et Lim, 2008] Schwab D., Lim L-T. (2008) *Blexisma2: a Distributed Agent Framework for Constructing a Semantic Lexical Database based on Conceptual Vectors.* DFMA-2008 : International Conference on Distributed Frameworks & Applications 2008, 21-22 October 2008, Penang, Malaysia, pp. 102-110.

133. [Seligman et Boitet, 1993] Seligman M., Boitet C. (1993) *A "whiteboard" architecture for automatic speech translation.* Proc. International Symposium on Spoken Dialogue. Waseda University, Tokyo, 1-12 novembre 1993, 4p.

134. [Seng et al., 2009] Seng S., Bigi B., Besacier L., Castelli E. (2009) *Segmentation multiple d'un flux de données textuelles pour la modélisation statistique du langage*. TALN 2009, Senlis, 10 p.

135. [Sérasset, 1994] Sérasset G. (1994) *SUBLIM : un système universel de bases lexicales multilingues et NADIA : sa spécialisation aux bases lexicales interlingues par acceptions.* Thèse UJF (Grenoble 1), 205 p.

136. [Sérasset et Boitet, 1999] Sérasset G., Boitet C. (1999) *UNL-French deconversion as transfer & generation from an interlingua with possible quality enhancement through offline human interaction.* Proc. MT Summit VII, Singapore, 13-19 September 1999, Asia Pacific Assoc. for MT, J.-I. Tsujii, pp. 220-228.

137. [Sérasset, 2004] Sérasset G. (2004) *A Generic Collaborative Platform for Multilingual Lexical Database Development.* In Proceedings of the Post-COLING 2004 Workshop on Multilingual Linguistic Resources (MLR2004). Geneva, Switzerland, pp. 73-79.

138. [Schneiderman, 1998] Schneiderman B. (1998) *Designing the user interface : strategies for effective human-computer interaction.* 2nd edition Addison Wesley.

139. [Shimohata et al., 1999] Shimohata S., Murata T., Ikeno A., Fukui T., Yamamoto H. (1999) *Machine Translation System PENSÉE: System Design and Implementation.* Proceedings of MT Summit VII "MT in the Great Translation Era", 13th-17th September 1999, Kent Ridge Digital Labs, Singapore, pp. 380-384.

140. [Silbersztein et Tutin, 2004] Silbersztein M., Tutin A. (2004) *NooJ : un outil TAL de corpus pour l'enseignement des langues et de la linguistique.* Journée ATALA, 9p.

141. [Silberztein, 2004] Silberztein M. (2004) *NooJ: an Object-Oriented Approach.* In INTEX pour la linguistique et le traitement automatique des langues. C. Muller, J. Royauté, Max

Silberztein eds. Cahiers de la MSH Ledoux, Presses Universitaires de Franche-Comté., pp. 359-369.

142. [Silberztein, 2005] Silberztein M. (2005) *NooJ: a linguistic annotation system for corpus processing.* In Proceedings of HLT/EMNLP, Human Language Technology Conference, pp. 10-11.

143. [Sinclair, 1996] Sinclair J. (1996) *Preliminary recommendations on Corpus Typology.* EAGLES: Expert Advisory Group on Language Engineering Standards, (Rap. tech.). May 1996, CEE, 62 p.

144. [Slocum et al., 1985] Slocum J., Bennett W-S, Whiffin L. Norcross E. (1985) *An Evaluation of METAL: the LRC Machine Translation System.* Proceedings of the Second Conference of the European Chapter of the Association for Computational Linguistics, Geneva, pp. 62 - 69.

145. [Snover et al., 2006] Snover M., Dorr B., Schwartz R., Micciulla L., Makhoul, J. (2006) *A Study of Translation Edit Rate with Targeted Human Annotation.* Proc. AMTA 2006. Cambridge, MA, USA. August 8-12, 2006. vol. 1/1: pp. 223-231.

146. [Sripada et al., 2005] Sripada S., Reiter E., Hawizy L. (2005) *Evaluation of an NLG System using Post-Edit Data: Lessons Learnt.* In Proceedings of European Natural Language Generation Workshop, pp. 133-139.

147. [Sripada et al., 2003] Sripada S-G., Reiter E., Davy I. (2003) *SUMTIME-MOUSAM: Configurable Marine Weather Forecast Generator.* Expert Update, 6 p.

148. [Streiter, 2005] Streiter O., Mathiasser M. (2005) *Open Source Framework for Multilingual Computing.* In Proceedings of the Lesser Used Languages & Computer Linguistics. European Academy Bozen, Italy, pp. 189-207.

149. [Takezawa et al., 2002] Takezawa T., Sumita E., Sugaya F., Yamamoto H., Yamamoto S. (2002) *Towards a Broad-coverage Bilingual Corpus for Speech Translation of Travel Conversation in the Real World.* Proc. LREC-2002, Las Palmas, Spain, May 29-31, 2002, vol. 1/3, pp. 147-152.

150. [Theologitis, 1997] Theologitis D. (1997) *EURAMIS, the platform of the EC translator.* In: EAMT Workshop (1997), pp. 17-32.

151. [Torlone, 2003] Torlone R., Atzeni P. (2003) *Chameleon: an Extensible and Customizable Tool for Web Data Translation.* In Proceedings of the 29th International Conference on Very Large Data Bases. Berlin, Germany. Vol. 29, pp. 1085 -1088.

152. [Tsai, 2004] Tsai W-J. (2004) *La coédition langue - UNL pour partager la révision entre langues d'un document multilingue.* Thèse UJF, GETA, CLIPS, IMAG, Grenoble, 307 p.

153. [Vasconcellos et Léon, 1998] Vasconcellos M., León. M. (1988) *SPANAM and ENGSPAN: Machine Translation at the Pan American Health Organization.* Computational Linguistics 11, pp. 122-136.

154. [Vauquois et Chappuy, 1985] Vauquois B., Chappuy S. (1985). *Static Grammars: a formalism for the description of linguistic models.* Proceedings of the conference on theoretical and methodological issues in machine translation of natural languages, Colgate Université New York, August, 25 p.

155. [Vo-Trung, 2004a] Vo-Trung H. (2004) *Méthodes et outils pour utilisateurs, développeurs et traducteurs de logiciels en contexte multilingue.* Thèse INPG, GETA, CLIPS, IMAG, Grenoble, 224 p.

156. [Vo-Trung, 2004b] Vo-Trung H. (2004) *Réutilisation de traducteurs gratuits pour développer des systèmes multilingues.* In Proceedings of the Conférence Rencontre des Étudiants

Chercheurs en Informatique pour le Traitement Automatique des Langues (RECITAL). Fès, Maroc, pp. 111-117.

157. [Wagner et Fischer, 1974] Wagner C.-K., Fischer M.-J. (1974) *The String-to-String Correction Problem.* in Journal of the ACM. vol. 21(1): pp. 168-173.

158. [White et al., 1994] White J. S., O'Connell T., O'Mara F. E. (1994) *The ARPA MT Evaluation Methodologies : Evolution, Lessons and Further Approaches.* Proc. Technology Partnerships for Crossing the Language Barrier (the First Conference of the Association for Machine Translation in the Americas), Columbia, Maryland, USA, October 5-8, 1994, 13 p.

159. [Young , 1999] Young R-M. (1999) *Using Grice's maxim of quantity to select the content of plan description.* Artificial Intelligence 115, pp. 215-256.

160. [Zhang et al., 2004] Zhang R., Kikui G., Yamamoto H., Watanabe T., Soong F., Lo W-K. (2004) *A unified approach in speech-to-speech translation: integrating features of speech recognition and machine translation.* In Proceedings of the 20th International Conference on Computational Linguistics, COLING - 04, Geneva, 2004, 7 p.

161. [Zhang et al., 2005] Zhang R., Kikui G., Yamamoto H., Lo W-K (2005) *A decoding algorithm for word lattice translation in speech translation.* In Proceedings of 2005 International Workshop on Spoken Language Translation, 2005, 4 p.

162. [Zhifei et al., 2009] Zhifei L., Chris C-B., Chris D., Juri G., Sanjeev K., Lane S., Wren T., Jonathan W., Omar Z. (2009) *Joshua: An open source toolkit for parsing - based machine translation.* In Proceedings of the Fourth Workshop on Statistical Machine Translation, Athens, Greece, March. Association for Computational Linguistics, pp. 135–139.

Netographie

1. [AVL, 2010] Arbres AVL. *http://www.labri.fr/perso/strandh/Teaching/MTP/Common/Book/HTML/ node239.html*. visité en février 2010.
2. [Allen, 2009] Allen J. (2009) http://www.geocities.com/mtpostediting/, visite en février 2009.
3. [Boitet, 2008] Boitet C. (2008). *Traduction automatique: ça marche ou non?*. http://www-clips.imag.fr/geta/christian.boitet/pages_personnelles/zArticles_sur_la_TAO_pdf/Interstices-TA&TAO-Boitet.v3.pdf.
4. [CORBA, 2010] *CORBA - Common Object Request Broker Architecture*. http://www2.lifl.fr/~merle/corba/. Visité en février 2010.
5. [Cowlishaw, 2009] Cowlishaw M. (2009) *IBM REXX Brief History*. http://www-01.ibm.com/software/awdtools/rexx/library/rexxhist.html. visité en avril 2009.
6. [DéjàVu, 2010] DéjàVu (2010) *un outil commercial d'aide à la traduction. http://www.atril.com*.
7. [Dutoit, 2010] *Système Alexandria*. http://www.memodata.com/. Visité en février 2010.
8. [Google, 2009] *Google (2009) Google Translator Toolkit - The Technology Guide*. http://www.tothetech.com/blog/tools-and-utilities/google-translator-toolkit.html.
9. [IWSLT, 2009] IWSLT2009 (2009) *Campagne d'évaluation IWSLT2009*. http://mastarpj.nict.go.jp/IWSLT2009/2009/12/evaluation-campaign.html.
10. [IWSLT, 2006] IWSLT2006 (2006) *Campagne d'évaluation IWSLT2006*. http://mastarpj.nict.go.jp/IWSLT2006/.
11. [IWSLT, 2004] IWSLT2004 (2004) *Campagne d'évaluation IWSLT2004*. http://mastarpj.nict.go.jp/IWSLT2004/.
12. [Koehn, 2009] *PHARAOH, a beam search decoder for phrase-based statistical machine translation models*. http://www.isi.edu/licensed-sw/pharaoh/. visité en février 2009.
13. [Kraif, 2006] Kraif O. (2006) *Corpus multilingues — multilingual corpora*. 3 p. http://w3.u-grenoble3.fr/kraif/index.php?option=com_content&task=view&id=20&Itemid=36.
14. [LISA, 2009] LISA (2009) *TMX format*. http://www.lisa.org/Translation-Memory-e.34.0.html.
15. [Moses, 2009] *Système Moses*. http://www.statmt.org/moses/. visité en février 2009.
16. [MT08, 2008] MT08 (2008) *Scripts Perl de NIST et BLEU*. http://www.itl.nist.gov/iad/mig//tests/mt/2008/scoring.html.
17. [NIST, 2002] NIST (2002) *The NIST 2002 Machine Translation Evaluation Plan (MT-02)*. http://www.itl.nist.gov/iad/mig/tests/mt/2002/doc/mt02_evalplan.v1.3.pdf.
18. [Numerama, 2009] Numerama (2009) *Wikipédia fête ses neuf ans avec 900 000 articles*. http://www.numerama.com/magazine/14869-wikipedia-fete-ses-neuf-ans-avec-900-000-articles.html.
19. [OASIS, 2009] OASIS (2009) *XLIFF format*. http://docs.oasis-open.org/xliff/xliff-core/xliff-core.html.
20. [OMNIA, 2009] OMNIA (2009) *Le projet OMNIA*. http://www.ellemme.org/, visité en octobre 2009.
21. [OpenWFE, 2009] OpenWFE (2009) *Open Workflow*. http://www.openwfe.org, visité en octobre 2009.

22. [PAHO, 2009] PAHO. (2009) *Machine translation at the Pan American Health Organization.* http://www.paho.org/ENGLISH/AM/GSP/TR/MACHINE_trans.htm.

23. [REXX, 2009] REXX (2009) *The Rexx Language Association.* http://www.rexxla.org/. visité en août 2009.

24. [SAE J2450, 1999] *SAE J2450 Society of Automotive Engineers Task Force on Translation Quality Metric (1999).* http://www.sae.org/TECHCMTE/j2450pl.htm et http://www.sae.org/TECHCMTE/j2450p2.htm.

25. [Satayamas, 2008] Satayamas V. (2008) *Word Segmentation Utility for Thai Language.* http://sourceforge.net/projects/thaiwordseg/.

26. [Similis, 2010] Similis (2010) *un outil commercial d'aide à la traduction. http://www.lingua-etmachina.com/.*

27. [Sinequa, 2006] Sinequa (2006) *Corpus écrit développé par Sinequa.* http://www.technolangue.net/article.php3?id_article=79.

28. [Sonovision-Itep, 2008] Sonovision-ITEP (2008) *Le site Web Sonovision-ITEP.* http://www.sonovision-itep.fr/fr/plandusite.htm.

29. [STAR, 2009] *Transit NXT - Le système de mémoire de traduction.* http://www.star-ts.com/fr/transit_nxt_translation_memory.shtml.

30. [Sun Microsystems, Inc., 2010] *RMI - Remote Method Invocation.* http://java.sun.com/javase/technologies/core/basic/rmi/index.jsp. Visité en février 2010.

31. [Systran, 2009] Systran. (2009) *Rapport financier semestriel au 30 juin 2009.* http://www.systran.fr/.

32. [Systran, 2009] Systran. (2009) *SYSTRAN remporte une première place pour la qualité de ses traductions à la compétition internationale "WMT".* Retrieved 28 May, 2009, http://www.systran.fr/systran/nouveautes-evenements/communiques-de-presse/premiere-place-a-la-competition-internationale-WMT.

33. [Systran, 2010] *Codages attendus selon les langues.* http://www.systran.fr/support/informations-importantes/codes-de-langue. visité en février 2010.

34. [TEI, 2009] *TEI format.* http://www.tei-c.org/index.xml.

35. [TheFreeDictionary, 2010] TheFreeDictionary (2010) *Phrase.* http://fr.thefreedictionary.com/phrase.

36. [Trados, 2010] Trados (2005) *un outil commercial d'aide à la traduction. http://www.trados.com/.*

37. [Traduwiki, 2010] *Traduwiki*. http://traduwiki.org/.

38. [Translatewiki.net, 2010] *Translatewiki.net.* http://translatewiki.net/wiki/Main_Page. Visité en février 2010.

39. [Translationwiki, 2006] Translationwiki (2006) *La traduction à la Wiki*. http://www.translationwiki.com.

40. [XMLINTL, 2009] XML-INTL (2009) *Système XML-INTL*. http://www.xml-intl.com. visité en février 2009.

41. [Zweigenbaum, 2006] Zweigenbaum P. (2006) *Corpus parallèles et comparables : introduction.* 19 p. http://www.limsi.fr/~pz/p11m2r-2006/corpus-paralleles.pdf.

42. [Zwarts, 2002] Zwarts S. (2002) *MT-Evaluation.* http://web.science.mq.edu.au/~szwarts/MT-Evaluation.php, visité en mars 2010.

43. [XWiki Translation, 2010] XWiki Translation (2010*) XWiki Translation*. http://l10n.xwiki.org/xwiki/bin/view/L10N/.

44. [Wikipedia, 2007] Wikipedia (2007) *http://www.wikipedia.org*.

45. [Wiktionary, 2007] Wiktionary (2007) *Dictionnaire libre multilingue*. http://www.wiktionary.org.

46. [Yakushite.Net, 2009] *Yakushite.Net*. http://www.yakushite.net/. visité en octobre 2009.

Bibliographie personnelle

Articles

Revues d'audience internationale avec comité de rédaction

1. [Boitet et al., 2010] Boitet C., Bellynck V., Mangeot M., **Huynh C-P**., Nguyen H-T., Vo-Trung H., Zaïri A. (2010) *Multilinguïsation en contexte et de qualité de documents, sites Web et applications: aspect informatiques et linguistiques.* Revue TAL, Volume 51 – n° 2/année 2010, 35 p. (soumis)

Communications à des congrès internationaux avec comité de lecture

2. [Blanchon et al., 2009] Blanchon H., Boitet C., **Huynh C-P** (2009) *A Web Service Enabling Gradable Post-edition of Pre-translations Produced by Existing Translation Tools: Practical Use to Provide High Quality Translation of an Online Encyclopedia*. In Proceedings of MT Summit XII 2009, International Association for Machine Translation hosted by the Association for Machine Translation in the Americas, 9 p.

3. [Boitet et al., 2009] Boitet C., **Huynh C-P**, Blanchon Hervé, Nguyen Hong-Thai (2009) *A Web-oriented System to Manage the Translation of an Online Encyclopedia Using Classical MT and Deconversion from UNL*. Conférence RIVF-IEEE, Danang, Vietnam, july 2009, 8p.

4. [Huynh et al., 2008a] **Huynh C-P**, Boitet C., Fafiotte G. (2008) *Extending an On-line Parallel Corpus Management System to Handle Specific Types of Structured Documents.* Conférence SLTU, MICA, Hanoi, Vietnam, May 2008, 6 p.

5. [Huynh et al., 2008b] **Huynh C-P**, Boitet C., Blanchon H. (2008) *SECTra_w: an Online Collaborative System for Evaluating, Post-editing and Presenting MT Translation Corpora*. Conférence LREC-08, Marrakech, Morocco, May 2008, 6 p.

6. [Phan-Huy et Huynh, 2006] Phan-Huy K., **Huynh C-P**. (2006) *A system for managing and exploring ERIM Corpus.* Conférence Nationale, Dalat, Viet Nam, June 2006, 5 p.

Séminaires

7. [Huynh et Asanobu, 2009] **Huynh C-P**., Kitamoto A. (2009) *Extending and Using SECTra_w for Managing and Supporting the Translation of DSR Captions Corpus*. DSR project seminar, NII, Tokyo, Japan, 15 slides.

Rapports d'études

8. [Huynh, 2007] **Huynh C-P.** (2007) *Etude et ingénierie d'une plate-forme logicielle et linguicielle de gestion de corpus de traductions.* Rapport de Master Recherche, INPG, GETALP-LIG, Grenoble, février 2007, 80 p.

9. [Huynh, 2006] **Huynh C-P.** (2006) *Ingénierie d'une plate-forme logicielle et linguicielle pour la communication multilingue sur réseau, fondée sur ERIM et adaptée au Vietnam.* Rapport de Master, Université de Danang, Vietnam, octobre 2006, 60 p.

10. [Huynh, 2001] **Huynh C-P.** (2001) *Un système de réception et de simulation du signal radar sur ordinateur pour Danang International Airport.* Rapport de fin d'études d'ingénieur, FI-EPUD, Vietnam, juin 2001, 45 p.

Rapports de contrats de recherche

11. [Blanchon et al., 2007] Blanchon H., Esperança-Rodier E., **Huynh C-P.** (2007) *Évaluation des systèmes Reverso et Systran sur des énoncés d'aide au touriste.* Rapport du projet TRANSAT pour France Telecom, 2007, 35 p.

Rapports de stage

12. [Huynh, 2009] **Huynh C-P.** (2009) *Support informatique à l'accès et à la multilingualisation de corpus du site Web DSR.* Internship Report during 5 months in NII, Tokyo, Japan, 20 p.

Rapports techiques

13. [Huynh, 2009] **Huynh C-P.** (2009) *SECTra_w.2.1 Towards a Contributive Operating System Enhancing Corpora of Translations on the Web by Evaluation, Extension by MT, Annotating, and Post-Editing.* Guide d'utilisation, NII, Tokyo, Japan, 48 p.

14. [Huynh, 2008] **Huynh C-P** (2008) *SECTra_w.2 Guide d'utilisation pour post-éditer des corpus de traductions.* Guide d'utilisation, GETALP-LIG, Grenoble, 10 p.

15. [Huynh, 2007] **Huynh C-P.** (2007) *SECTra_w.1 Guide d'utilisation pour faire l'évaluation des systèmes de TA.* Manuel, GETALP-LIG, Grenoble 11 p.

Annexes

Liste de définitions

Nous regroupons ici la liste des définitions proposées dans cette thèse.

Définition I-0.	Un sectra est un système d'exploitation de corpus de traductions.
Définition I-1.	Une phrase est l'unité élémentaire d'un énoncé, formée de plusieurs mots ou groupes de mots et qui présente un sens complet. [TheFreeDictionary, 2010].
Définition I-2.	Un segment est l'unité de traduction de base des traducteurs humains. Il s'agit d'une phrase, d'un titre, ou d'un terme dans une nomenclature.
Définition I-3.	Un segment multilingue est une liste qui se compose de N segments monolingues équivalents en N langues.
Définition I-4.	Un segment monolingue est un segment dont le contenu est en une seule langue.
Définition I-5.	Un diagramme traductionnel est un graphe de traductions dont les nœuds sont les langues successives (par exemple Ja, En, Fr), et donc les arcs portent éventuellement des métadonnées concernant l'opération de traduction (humaine ou automatique, personne ou systèmes, caractéristiques ou paramètres).
Définition I-6.	Une unité de traduction est une partie d'un texte ou d'un document dont certains aspects sont traités globalement durant une opération de traduction.
Définition I-7.	Un texte peut être une unité de longueur variable : un page, un livre, etc. Un texte est sous forme brute, ou sous forme formatée (html, xml, etc.). En TA, un texte est un fichier d'entrée ou de sortie.
Définition I-8.	Un document est constitué par un texte et des objets satellites (images, formulaires, etc.). Un document peut donc comprendre plusieurs fichiers.
Définition I-9.	Une traduction candidate est le résultat d'un système de TA à évaluer.
Définition I-10.	Une traduction de référence est une traduction de bonne qualité utilisable pour l'évaluation de traductions candidates.
Définition I-11.	Une évaluation est dite interne si elle vise à juger la conception de systèmes (architecture linguistique et architecture computationnelle) de TA et leurs perspectives d'amélioration et d'extension à de nouvelles langues, à de nouveaux types de documents, et à de nouvelles tâches (e.g. de l'assimilation à la dissémination).
Définition I-12.	Une évaluation est dite externe si elle consiste à juger un système de traduction comme une boîte noire.
Définition I-13.	Une évaluation est dite subjective si elle fait appel à des jugements qualitatifs humains.
Définition I-14.	Une évaluation est dite objective si elle ne fait pas appel à des jugements subjectifs. Elle peut être fondie sur des références ou liée à la tâche.
Définition I-15.	Une évaluation objective est fondée sur des références si elle est réalisée par des programmes informatiques produisant des résultats numériques à partir d'un certain nombre de traductions de référence et des traductions candidates fournies en entrée.
Définition I-16.	Une évaluation objective est liée à la tâche si elle est réalisée par des programmes informatiques produisant des résultats numériques ou

	symboliques à partir des segments source, de leurs prétraductions automatiques, et de mesures liées à la tâche (temps de compréhension, temps de post-édition, temps de réalisation d'une tâche (comme une réservation) en contexte bilingue).
Définition I-17.	Un segment non textuel peut être un treillis ou un graphe de chaînes de mots ou de lemmes directement produit par un reconnaisseur de parole. Un segment non textuel peut aussi être une représentation de l'expression langagière multimodale (texte + parole + gestes).
Définition I-18.	Les corpus pour la TA sont ceux qui sont utilisables pour la traduction, incluant la traduction humaine et la traduction automatique.
Définition I-19.	Les corpus pour la traduction humaine (corpus pour la TH) sont les mémoires de traduction, les corpus de documents, etc.
Définition I-20.	Les corpus pour la traduction automatique (corpus pour la TA) ne sont pas seulement des textes, mais aussi des annotations linguistiques.
Définition I-21.	Un document est dit monolingue si tous ses segments sont dans une seule langue.
Définition I-22.	Un corpus est monolingue si tous ses documents sont monolingues pour la même langue.
Définition I-23.	Un document est dit multilingue s'il comprend des segments en plusieurs langues.
Définition I-24.	Un corpus est multilingue s'il contient des documents multilingues, ou des documents monolingues dans au moins deux langues.
Définition I-25.	Un document est dit parallèle s'il est multilingue et s'il est aligné au niveau des segments, de sorte qu'un segment est traduction de celui avec lequel il est aligné.
Définition I-26.	Un corpus est parallèle si tous ses documents sont parallèles.
Définition I-27.	Un corpus est dit comparable s'il est composé de textes comparables dans des langues différentes, non alignés au niveau des segments, mais parlant d'un même sujet, à la même époque et dans un registre comparable.
Définition I-28.	Un corpus est un corpus de phrases s'il ne contient que des phrases.
Définition I-29.	Un corpus est un corpus de textes s'il ne contient que des textes.
Définition I-30.	Un corpus est dit corpus de documents s'il contient des documents.
Définition I-31.	Un segment multilingualisé sera pour nous un segment contenant une seule langue source.
Définition I-32.	Un segment multisource contient plusieurs langues source.
Définition I-33.	Un segment "source" multimédia est un graphe avec un type associé au corpus, par exemple simple chaîne, ou treillis de segmentation, ou treillis pondéré, ou automate fini, pondéré ou non.
Définition II-1.	Un mot typographique est une suite finie de signes appartenant à un ensemble fini connu, et délimitée par des séparateurs appartenant eux-mêmes à un ensemble fini connu.
Définition II-2.	Un multisegment est un segment qui se compose de plusieurs segments dont la langue n'est pas indiquée comme source ou cible.
Définition II-3.	Un segment multilingualisé est un segment multilingue dans lequel une seule langue est considérée comme langue source.
Définition II-4.	Une page standard contient 250 mots typographiques ou environ 1400 caractères (1 page A4, double interligne, Times 12) pour les alphabets et certains syllabaires (comme les langues européennes), et 400 caractères pour les idéogrammes (comme le japonais, le chinois, et le coréen).

Définition II-5. Une page logique est une unité de traitement informatique contenant un nombre spécifique de caractères, ou de mots, ou de segments, prédéfinie par les utilisateurs (administrateurs, post-éditeurs, évaluateurs, réviseurs, etc.) ou par les développeurs.

Définition II-6. Un corpus est dit corpus de documents TMX si c'est un corpus de documents dont le texte est en format TMX.

Définition II-7. Un corpus est dit corpus de documents « multifichier » si c'est un corpus de documents dans lequel un document est constitué de plusieurs fichiers.

Définition II-8. Un métasegment est un segment contenant des variables lexicales.

Définition II-9. Un livre de phrases est un document dont les segments sont des métasegments multilingualisés et contextualisés.

Définition II-10. Le principe de proactivité consiste à préparer les données à l'avance afin de permettre aux contributeurs d'accéder à ces données sans attente.

Définition II-11. Une fonction délégable d'un système S1 est une fonction réalisable par appel à un service d'un autre système S2, qu'elle soit ou non réalisée ou réalisable par un service interne au système S1.

Définition II-12. Un segment multilingualisé contextualisé ou segment-mc est un objet formé d'un segment dans une langue source, de ses contextes d'apparition dans les documents où il est apparu, et des traductions proposées dans ces contextes, ainsi que des autres informations attachables à ce segment, par exemple un minidictionnaire construit à partir de la liste de ses lemmes, ou un ou plusieurs arbres d'analyse, ou un graphe UNL, etc.

Définition II-13. Le contexte structurel d'un segment est la position du segment source dans la hiérarchie courante (paragraphe, section…).

Définition III-1. Nous appellerons fragment toute sous-chaîne [connexe] d'un segment (source ou cible), et bi-fragment tout couple de fragments en relation (potentielle) de traduction mutuelle.

Définition III-2. Un supersegment est un morceau de texte qui contient plusieurs segments.

Définition III-3. Un infrasegment est une portion d'un segment.

Définition III-4. Un sous-document est un document contenu dans un autre. Il peut aussi être contenu dans un segment.

Définition III-5. Un sous-segment est un segment d'un sous-document.

Définition III-6. Une classe lexicale (de mots ou locutions) est une liste nommée créée par ailleurs dans le corpus; une des valeurs de cette liste viendra instancier chaque occurrence de la variable lexicale correspondante, notée $nom_de_la_classe.

Exemples du format CCM pour les corpus

Exemple du format CCM pour le corpus Survitra

```
<corpus>
<header>
   <name>Survitra</name>
   <date>2004-10-20</date>
   <domain>Restaurant</domain>
   <project>Survitra</project>
   <Nlang>3</Nlang>
   <lang>English</lang>
   <lang>French</lang>
   <lang>German</lang>
</header>
<body>
   <doc type="Ensemble_phrases" name="D_inRestaurant_1">
     <article type="phrase">
        <segment id="survitra_enfrde_01_121108.171103" type="content">
           <occurrence type ="org" lang="en">In restaurant</occurrence>
           <occurrence type ="rev" lang="en" producer="Xan" date="121108.181107"
            level="***" score="11">In a restaurant</occurrence>
           <occurrence type ="mt" lang="fr" producer="Google"
            date="121108.171103">Dans le restaurant</occurrence>
           <occurrence type ="mt" lang="fr" producer="Systran"
            date="121108.171105">Dans restaurant</occurrence>
           <occurrence type ="postedition" lang="fr" producer="hcphap"
            date="121108.181106" level="***" score="13">Au restaurant</occurrence>
           <occurrence type ="mt" lang="de" producer="Systran"
            date="121108.171105">In dem Restaurant</occurrence>
           <occurrence type ="postedition" lang="de" producer="Xan"
            date="121108.171108" level="***" score="13">Im Restaurant</occurrence>
      </segment>
   <segment id="survitra_enfrde_02_121108.171106">
  ……..
</doc>
</body>
</corpus>
```

Figure 68: Exemple du format CCM pour le corpus Survitra

Exemple du format CCM pour le corpus TRANSAT

```
<corpus>
<header>
   <name>Evaluation Corpus</name>
   <date>2007-11-22</date>
   <domain>Tourisme</domain>
   <project>Transat</project>
   <Nlang>2</Nlang>
   <lang type=”source”>English</lang>
   <lang type=”target”>French</lang>
   <othermeta type=”MT_system”>Systran</othermeta>
</header>
<body>
<doc name=”Systran_en_fr” type=“textfile“>
  <article type="touristsection">
     <segment id="btec_enfr_00_100310.211502" type="textsegment">
      <occurrence type ="source" lang="en"> A burglar broke into my room.
            </occurrence>
      <occurrence type ="MT" lang="fr" producer="Systran"> Un cambrioleur est entré
            de force dans ma pièce. </occurrence>
      <occurrence type = ”reference” version="1" lang= "fr"> Un cambrioleur est entré
            de force dans ma chambre. </occurrence>
      <occurrence type="BLEU"> 0.46 </occurrence>
      <occurrence type="NIST"> 1.12 </occurrence>
      <occurrence type="mWER"> 6 </occurrence>
      <occurrence type="fluidity" producer="Xan"> 4/5 </occurrence>
      <occurrence type="adaquation" producer="Xan"> 2/3 </occurrence>
      <occurrence type="fluidity" producer="Hervé"> 3/5 </occurrence>
      <occurrence type="adaquation" producer="Hervé"> 2/3 </occurrence>
     </segment>

   <segment id="btec_enfr_01_100310.211734">
  ……..
</doc>
</body>
</corpus>
```

Figure 69 : Exemple du format CCM pour le corpus TRANSAT.

Exemple du format CCM pour le corpus ERIM

```
<corpus>
<header>
   <name>ERIM</name>
   <date>2004-10-20</date>
   <domain>Restaurant</domain>
   <project>ERIM</project>
   <Nlang>2</Nlang>
   <lang>Vietnamese</lang>
   <lang>French</lang>
</header>
<body>
<doc name=”dialog1” type=“dialog“>
  <article type="LOC">
     <segment id="erim_enfr_01_201004.161502" type="TP">
       <occurrence type ="son" lang="vi" producer="A"> ”A/TP1.wav” </occurrence>
       <occurrence type ="minitexte" lang="vi"> "" </occurrence>
       <occurrence type = ”transcription” version="1" lang= "vi"> Xin chào
             </occurrence>
       <occurrence type = ”translation” version="1" lang= "fr" producer="Sysran">
             Bonjour </occurrence>
      </segment>

    <segment id="erim_enfr_01_201004.161502">
  ……..
</doc>
</body>
</corpus>
```

Figure 70 : Exemple du format CCM pour le corpus ERIM.

Exemple du format CCM pour le corpus EOLSS

```
<corpus>
<header>
   <name>EOLSS</name>
   <date>2008-04-20</date>
   <domain>Water</domain>
   <project>EOLSS/UnescoL</project>
   <Nlang>2</Nlang>
   <lang type ="source">English</lang>
   <lang type ="target">French</lang>
</header>
<body>
   <doc type="import_file" name="document_1">
     <article type="page1">
        <segment id="eolss_enfr_01_200408" type="seg">
           <occurrence type ="org" lang="en">Ethics and Science</occurrence>
           <occurrence type ="unl" lang="pivot" producer="UNDL"
            date="250308.161103">and(science(icl>knowledge):0B.@entry,
              ethics(icl>principle):00)</occurrence>
           <occurrence type ="mt" lang="fr" producer="Systran"
            date="200408.161103">Éthique et la Science </occurrence>
           <occurrence type ="mt" lang="fr" producer="Reverso"
            date="121108.171105">Éthique et Science </occurrence>
           <occurrence type ="postedition" version="2" lang="fr" producer="Hervé"
            date="121108.181106" level="***" score="12">Éthique et Science
            </occurrence>
           <occurrence type ="postedition" version="1" lang="fr" producer="Hcphap"
            date="121108.181106" level="***" score="8">Éthique et la Science
            </occurrence>
         </segment>
          <segment id="eolss_enfr_02_200408">
      ……..
     </doc>
 </body>
</corpus>
```

Figure 71: Exemple du format CCM pour le corpus EOLSS

SECTra_w User Manual

This manual corresponds to version 2.3 (April 2010) of SECTra_w.

III.1 User Management

III.1.1 User groups

In SECTra_w, there are 4 main user groups with associated access rights.

SuperAdmin Group. This group includes users who have all permissions to work and to modify the source code of SECTra_w. Users in this group are system developers.

Admin Group. Each post-edition, evaluation, and transcription project in SECTra_w has a corresponding Admin group containing project administrators who can:

- create new users,
- define users' profiles,
- assign tasks to users,
- delete users,
- call Machine Translation systems,
- import corpora, export results, and select corpus from Translation Memory.

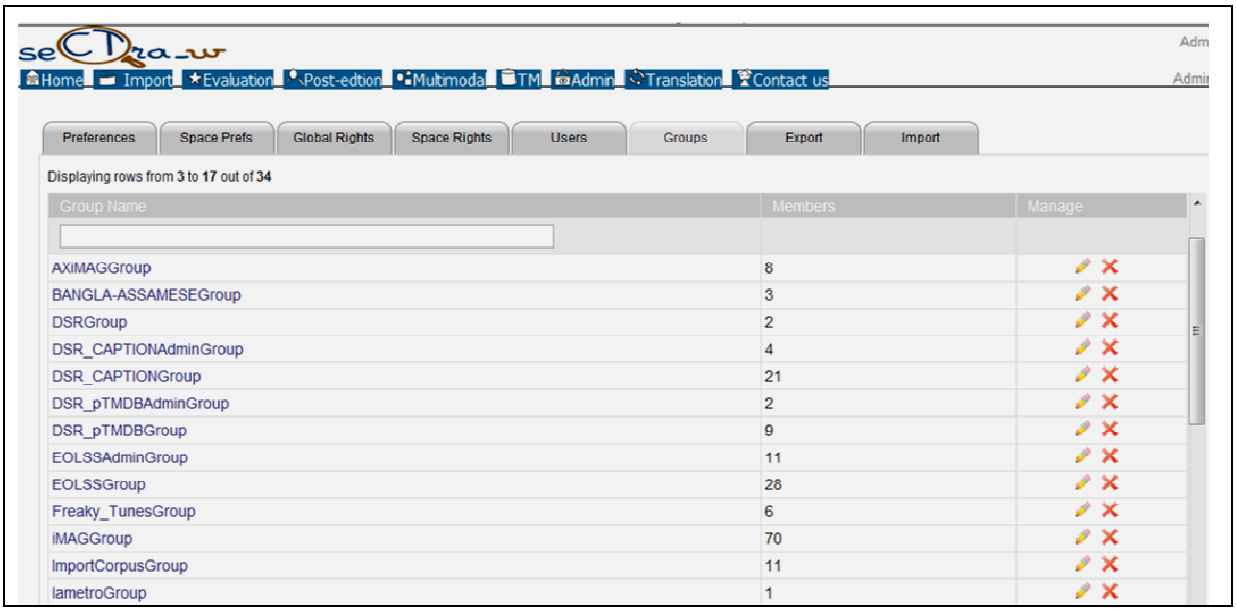


Figure 72: SECTra_w User Groups

Contributor Group. Each project has corresponding contributor groups, including evaluators, or transcribers, translators, post-editors and reviewers, who can participate to evaluate, transcribe, translate and post-edit the corpora of the project. Contributor groups and Admin group are created automatically when the corresponding project is created. The contributor

groups and admin group are named from the project name, for example EOLSSAdminGroup, EOLSSPostedionGroup, etc.

A user can only work with a project if s/he belongs to its corresponding contributor group. A user can also be a member of several contributor groups.

Guest Group: Guest users who do not log in SECTra_w can only access the home page and view certain corpora, but they can't change any data.

III.1.1.1 Login and register a new account

It is possible to view certain corpora in SECTra_w without any registration, but if a user wants to work (evaluate, postedit, transcribe the corpora) with SECTra_w, s/he must have an account. Registration asks only minimal information: first name, last name, login and password. After the registration is done and the profile assignment is set, registered users can work on the projects they are allowed to access.

In the home page of SECTra_w, in the top-right area, the **Register** link allows registering new users.

Figure 73: Registration screenshot

Once the "login" information is input, save it by clicking on the **Register** button. Now it is possible to login into SECTra_w. By clicking the "Login" link (top-right), you get the following interface:

Figure 74: Login interface

III.1.2 Change your password and/or profiles

If you want to change your password and/or profiles, you should move the mouse over the **Admin** menu. Next, choose **SECTra_w's Users**. Then, choose your nickname. Finally, change any information of your profile that you want.

Figure 75: Admin menu

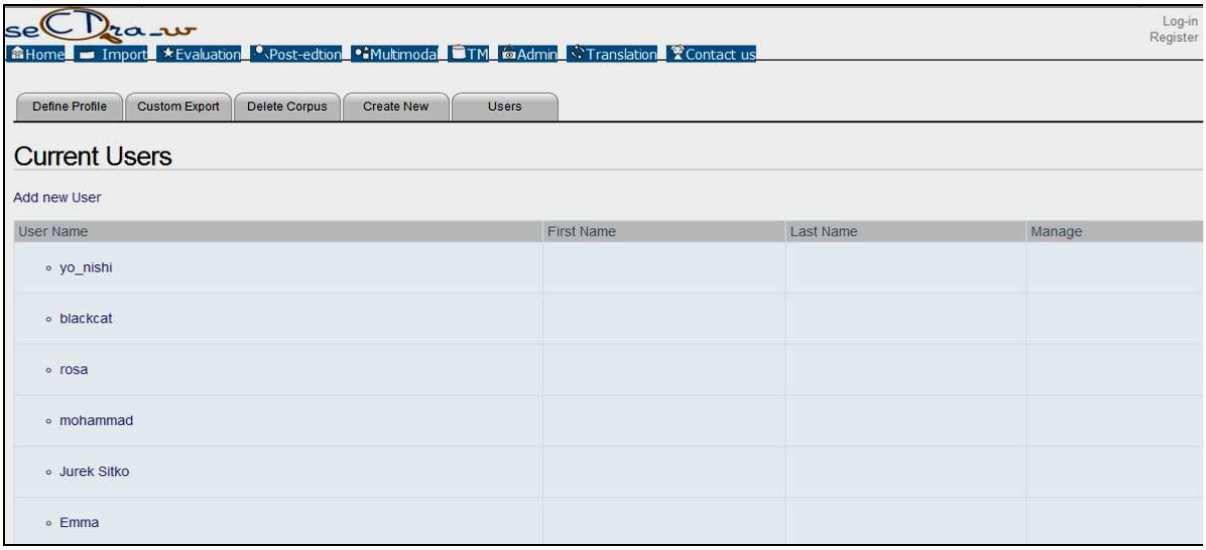


Figure 76: Users list screenshot

Figure 77: User profile screenshot

III.2 Presenting et Evaluating MT Translation Corpora

SECTra_w supports all aspects of the evaluation of MT systems, including subjective, objective, and task-related objective evaluation.

It is possible to show not only tables of figures as results of an evaluation campaign, but also the real data (source, MT outputs, references, post-edited outputs), and to make the post-edition effort sensible by transforming the trace of the edit distance computation into an intuitive presentation, much like a "revision" presentation in Word.

It is also possible to recompute n-gram-based scores by using the post-edited translations as references, and/or by adding them to the already available references.

III.2.1 Import of corpus and preparation for the evaluation campaign

III.2.1.1 Import and preparation process

If you are an administrator of an evaluation campaign, you can prepare your evaluation campaign by following steps:

1) create a campaign name,
2) import evaluation corpora,
3) assign tasks (corpus name, collections of pages, etc.) to evaluators.

In order to import an evaluation corpus, click on the **Import** menu, then choose the **Evaluation** tab.

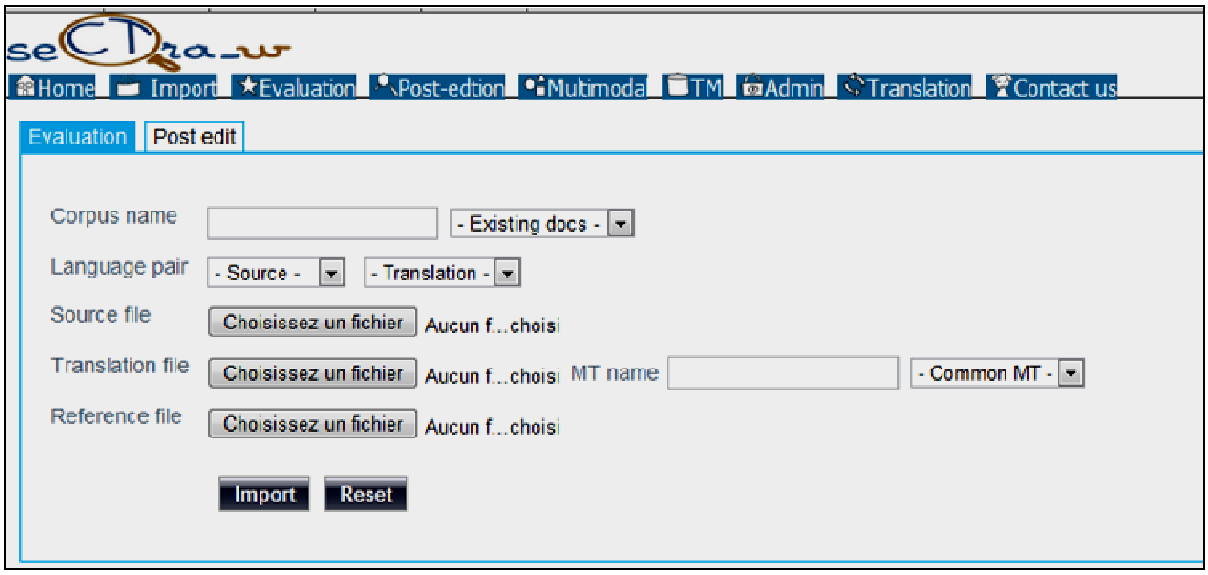


Figure 78: Import screenshot of evaluation corpus

Now start importing the evaluation corpus by:

1) entering its corpus name,
2) choosing a languages pair,
3) uploading the corresponding source file, translation file, and reference file,
4) entering an MT name (Machine Translation system to be evaluated),

5) and finally clicking the Import button.

III.2.1.2 Corpus structure

An evaluation corpus is constituted of 3 files:

- The source file contains source segments
- The translation file contains translation outputs of an MT system to be evaluated (candidate translations).
- Reference file contains translations with gold standard quality. These references are usually achieved from the human post-edition.

III.2.1.3 Corpus format

SECTra_w accepts evaluation corpora in text format and UTF-8 encoding. Each “document” in an evaluation corpus consists of a list of translation units (TU), each containing one or more segments separated by “|”. The textual syntax for importing a TU is :

id ":" texte ("|" texte)*

Example.

BTEC0032 : Est-ce que je peux vous aider ? |Qu'est-ce que vous aimeriez ?

Each TU in the translation file or in the reference file is aligned with one and only one TU in the source file by an Id. However, one source TU can be aligned with several TUs in the translation file or in the reference file (There may be several candidate translations and references for a source, for the same MT system).

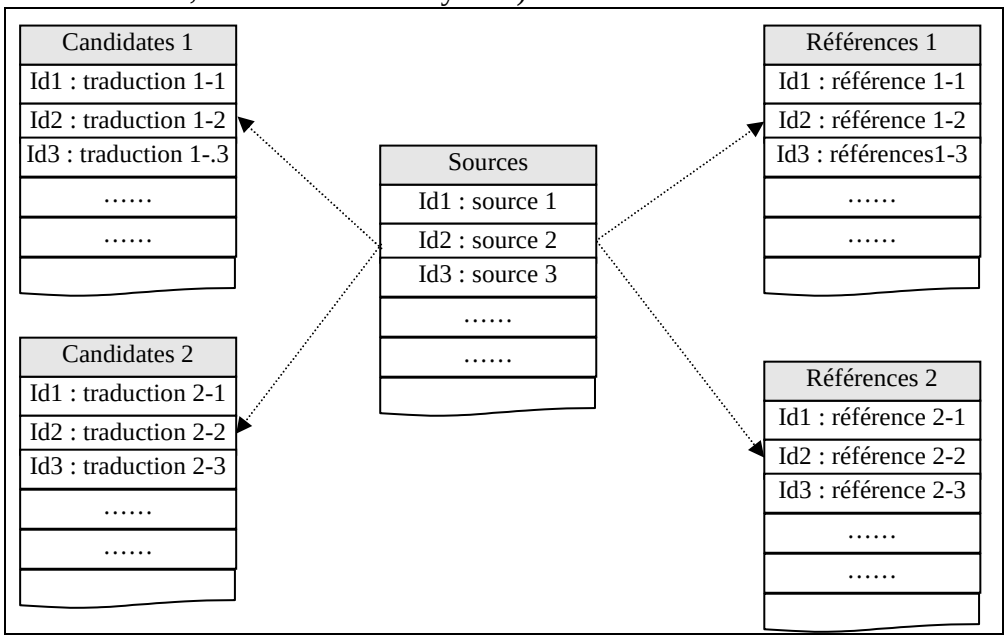


Figure 79: Structure de corpus à importer

III.2.2 Evaluating a corpus

III.2.2.1 Subjective evaluation

If you are evaluator, you should click on the “Evaluation” menu. The screen will display as follows.

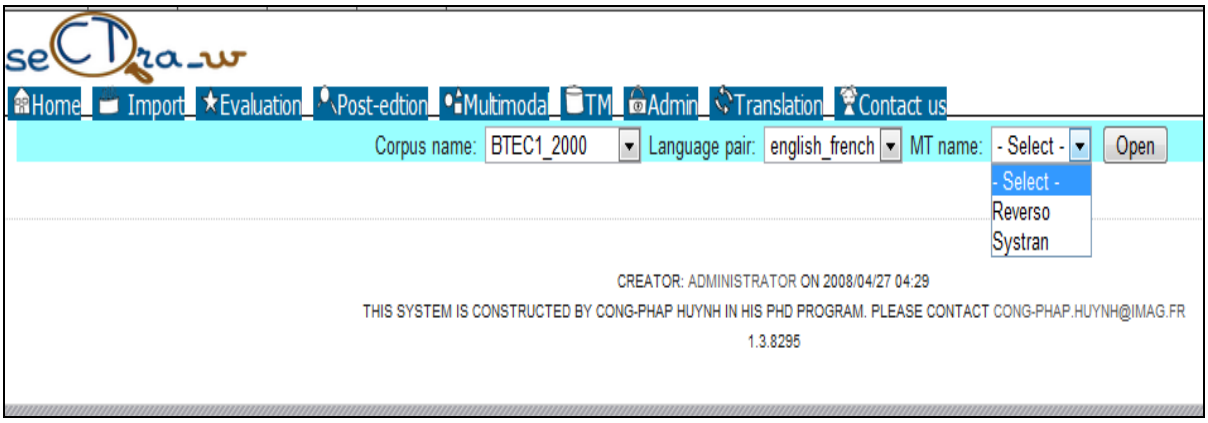


Figure 80: Choose corpus and MT system to be evaluated

After choosing a corpus name along with a language pair and a MT system to be evaluated, you can start the subjective evaluation on the interface as follows:

Figure 81: Evaluation interface

Adequation of information:	**Fluency**:
A1 : All A2 : Almost all A3 : Half A4 : Few A5 : None	F1 : written F2 : oral F3 : not acceptable

The interface for subjective evaluation generalizes slightly those classically used by judges evaluating adequacy and fluidity. The number of possible choices (presented as radio buttons) is a parameter, as well as the help strings appearing in small balloons when the cursor hovers on a button. The following other features have been included and proved useful.

- Several judges can perform evaluation at the same time on the same part of the data, which appears as a Web page of about 20 segments.
- A segment usually receives several evaluation scores from as many judges. These scores can be shown to users having enough access rights.
- A preliminary workflow tool is included, to define the judges and assign them sets of pages to evaluate.

Fluency evaluation. Indicating how the segment to be evaluated is perceived by monolingual evaluators (native speakers). Evaluators have to classify the level of language for the candidate translation by choosing a value among values indicating the fluency level. SECTra_w proposes 3 values instead of 5 in the NIST protocol standard (Flawless English, Good English, Non-native English, Disfluent English, Incomprehensible) as follows:

- (F1) perfectly understandable formulation, the style is written or oral
- (F2) acceptable formulation for the oral, possibly comprehensible with an effort
- (F3) inaccessible formulation

Adequacy evaluation. indicating the transportation of pertinent information from the source to the target. SECTra_w proposes 5 values for the adequacy level like the NIST standard protocol:

- (A1) All information is transported
- (A2) Almost all information is transported
- (A3) Half information is transported
- (A4) Few information is transported
- (A5) No information is transported

Once the evaluator clicks on a radio button for each level of adequacy (A1—A5) and fluidity (F1—F3), SECTra_w automatically saves his/her choice. When both values of adequacy and fluency for a segment have been chosen, the background color of the row containing that segment switches from pink to light sky blue.

Evaluators can move back and forth between logical pages of the corpus by clicking on *next, previous, page number* links at the top of the current page.

III.2.2.2 Objective n-gram-based scores

SECTra_w offers several n-gram-based measures (BLEU, NIST, WER, etc). After importing the evaluation corpus, administrators of the campaign will start processes to measure these n-gram-based scores. SECTra_w also allows users to integrate their own objective evaluation scripts, or to measure a selected part of corpus for their purposes.

Figure 82: Objective n-gram-based scores

In this interface, WER is named "Dw", while "Dc" is the character-based distance, and D is a linear combination of both: Dsent = a Dc + b Dw (a and b are modifiable).

In order to modify the value of ***a*** and/or ***b***, click on the red link at the top of the *Distance* column. Then SECTra_w allows you to increase or decrease the value of **a** and/or **b** within 0 and 1 (a + b =1). If we want the comparison between two segments mainly based on character-based distance, we choose a > b and conversely we choose a < b if the edit distance of two segments is mainly based on the word-based distance.

In the default presentation, the first column contains the source segments, the second column is for the MT outputs to be judged, with a radio button for each level of adequacy (A1—A5) and fluency (F1—F3). The last column contains the reference segments; the trace of the edit distance computation is shown in this column: in red, inserted strings, in overstriked blue, erased strings.

III.2.2.3 Task-related Objective Evaluation

SECTra_w allows evaluating the quality of MT systems' outputs by computing human-oriented task-related measures. It means that SECTra_w can perform the MT evaluation based on the time or the effort spent to post-edit the MT outputs.

We can perform this kind of evaluation in SECTra_w in the following steps:

1) Importing a source corpus,
2) Submitting the source corpus to several MT systems,
3) Having humans post-edit the MT outputs,
4) Calling the measure of edit distance between the MT outputs and the corresponding post-editions.

Figure 83: Post-editing interface of SECTra_w

The task-related objective evaluation interface, basing on post-editing the MT results is shown below (Figure 84).

Source	Translation (Systrans)	Distance	Post-edition
	Accept Trace Reject	D=a. Dcar+b. Dmot a:0.2, b:0.8	Accept Trace Reject
?Hamburger and stew on the right side and salad, please.	Hamburger et ragoût sur le bon côté et la salade, svp.	Dc=31,Dm=8 D= 12.6	Du Hamburger et du ragoût sur le ~~bon~~ côté droit et de la salade, ~~svp.~~ s'il vous plaît.
That fried fish, one sausage with green peas, please.	Ce poisson frit, une saucisse avec les pois , svp.	Dc=20,Dm=6 D= 8.8	Ce poisson frit, une saucisse avec ~~les~~ des ~~pois~~ petits , pois, ~~svp.~~ s'il vous plaît.
T-bone steak and sauerkraut and fried potatoes, please.	Bifteck à l'os et choucroute et pommes de terre frites, svp.	Dc=26,Dm=7 D= 10.8	Du Bifteck à l'os et de la choucroute et des pommes de terre frites, ~~svp.~~ s'il vous plaît.
Roast chicken and two slices of ham on this side and spinach, please.	Poulet rôti et deux tranches de jambon sur ces côté et épinards, svp.	Dc=21,Dm=6 D= 9	Du Poulet rôti et deux tranches de jambon sur ~~ces~~ ce côté et des épinards, ~~svp.~~ s'il vous plaît.
I'd like breakfast, please.	Je voudrais le petit déjeuner, svp.	Dc=14,Dm=4 D= 6	Je voudrais ~~le~~ un petit déjeuner, ~~svp.~~ s'il vous plaît.
Coffee, please.	Café, svp.	Dc=16,Dm=4 D= 6.4	Du café, s'il vous ~~Café,~~ plaît. ~~svp.~~

Figure 84: Edit distance and Track changes

III.2.3 Visualizing a corpus and its evaluation results

To see the subjective and objective evaluation results, we should move the mouse over the *Evaluation* menu, next click on the *View Evaluation Results* submenu**.** Then choose *Corpus name, languages pair, and MT name* as you wish.

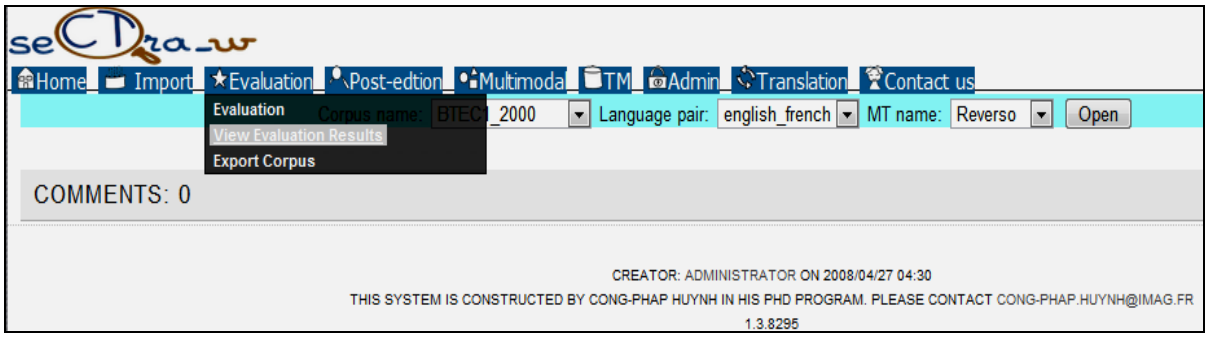
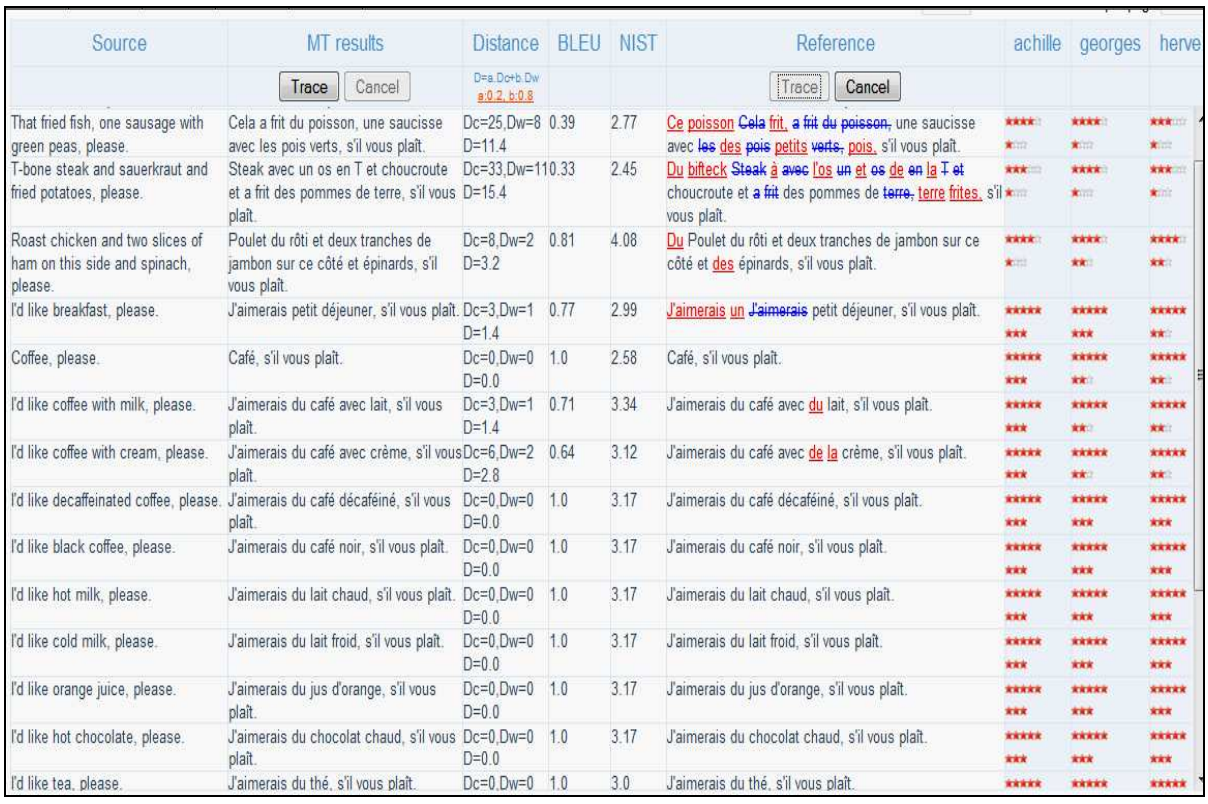


Figure 85: Choose a corpus to see its evaluation results

After choosing all parameters to view the evaluation results, we will see a visualization screen as follows:



Source	MT results	Distance	BLEU	NIST	Reference	achille	georges	herve
	Trace Cancel	D=a.Dc+b.Dw a:0.2, b:0.8			Trace Cancel			
That fried fish, one sausage with green peas, please.	Cela a frit du poisson, une saucisse avec les pois verts, s'il vous plaît.	Dc=25,Dw=8 D=11.4	0.39	2.77	Ce poisson ~~Cela~~ frit, ~~a frit du poisson,~~ une saucisse avec ~~les~~ des ~~pois~~ petits ~~verts,~~ pois, s'il vous plaît.	***** *	***** *	***** *
T-bone steak and sauerkraut and fried potatoes, please.	Steak avec un os en T et choucroute et a frit des pommes de terre, s'il vous plaît.	Dc=33,Dw=11 D=15.4	0.33	2.45	Du bifteck ~~Steak~~ à ~~avec~~ l'os ~~un~~ et ~~os~~ de ~~en~~ la ~~T~~ et choucroute et ~~a frit~~ des pommes de ~~terre,~~ terre frites, s'il vous plaît.	***** *	***** *	***** *
Roast chicken and two slices of ham on this side and spinach, please.	Poulet du rôti et deux tranches de jambon sur ce côté et épinards, s'il vous plaît.	Dc=8,Dw=2 D=3.2	0.81	4.08	Du Poulet du rôti et deux tranches de jambon sur ce côté et des épinards, s'il vous plaît.	***** *	***** **	***** **
I'd like breakfast, please.	J'aimerais petit déjeuner, s'il vous plaît.	Dc=3,Dw=1 D=1.4	0.77	2.99	J'aimerais un ~~J'aimerais~~ petit déjeuner, s'il vous plaît.	***** ***	***** ***	***** **
Coffee, please.	Café, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	2.58	Café, s'il vous plaît.	***** ***	***** **	***** **
I'd like coffee with milk, please.	J'aimerais du café avec lait, s'il vous plaît.	Dc=3,Dw=1 D=1.4	0.71	3.34	J'aimerais du café avec du lait, s'il vous plaît.	***** ***	***** **	***** **
I'd like coffee with cream, please.	J'aimerais du café avec crème, s'il vous plaît.	Dc=6,Dw=2 D=2.8	0.64	3.12	J'aimerais du café avec de la crème, s'il vous plaît.	***** ***	***** **	***** **
I'd like decaffeinated coffee, please.	J'aimerais du café décaféiné, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du café décaféiné, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like black coffee, please.	J'aimerais du café noir, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du café noir, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like hot milk, please.	J'aimerais du lait chaud, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du lait chaud, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like cold milk, please.	J'aimerais du lait froid, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du lait froid, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like orange juice, please.	J'aimerais du jus d'orange, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du jus d'orange, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like hot chocolate, please.	J'aimerais du chocolat chaud, s'il vous plaît.	Dc=0,Dw=0 D=0.0	1.0	3.17	J'aimerais du chocolat chaud, s'il vous plaît.	***** ***	***** ***	***** ***
I'd like tea, please.	J'aimerais du thé, s'il vous plaît.	Dc=0,Dw=0	1.0	3.0	J'aimerais du thé, s'il vous plaît.	*****	*****	*****

Figure 86: Visualisation of the campaign evaluation results

In this screen, we can see the results of the subjective evaluations (adequacy, fluency) of each judge, the distances between MT translations and their post-editions, the traces of minimal sequences of edit operations sufficient for transforming the MT translations into their post-editions, and the NIST and BLEU values.

The screen makes it easy to detect whether some judge has omitted his/her subjective evaluation on some segments and to confirm the results before exporting and synthesizing them.

III.2.4 Exporting a corpus and evaluation results

After finishing the evaluation, you can export the corpus along with evaluation results to files in a pre-defined format. If you are an administrator, move the mouse over the *Export* munu at the right top corner of in the *Visualization* interface, and you can see three options allowing you to download a corpus associated with objective scores.

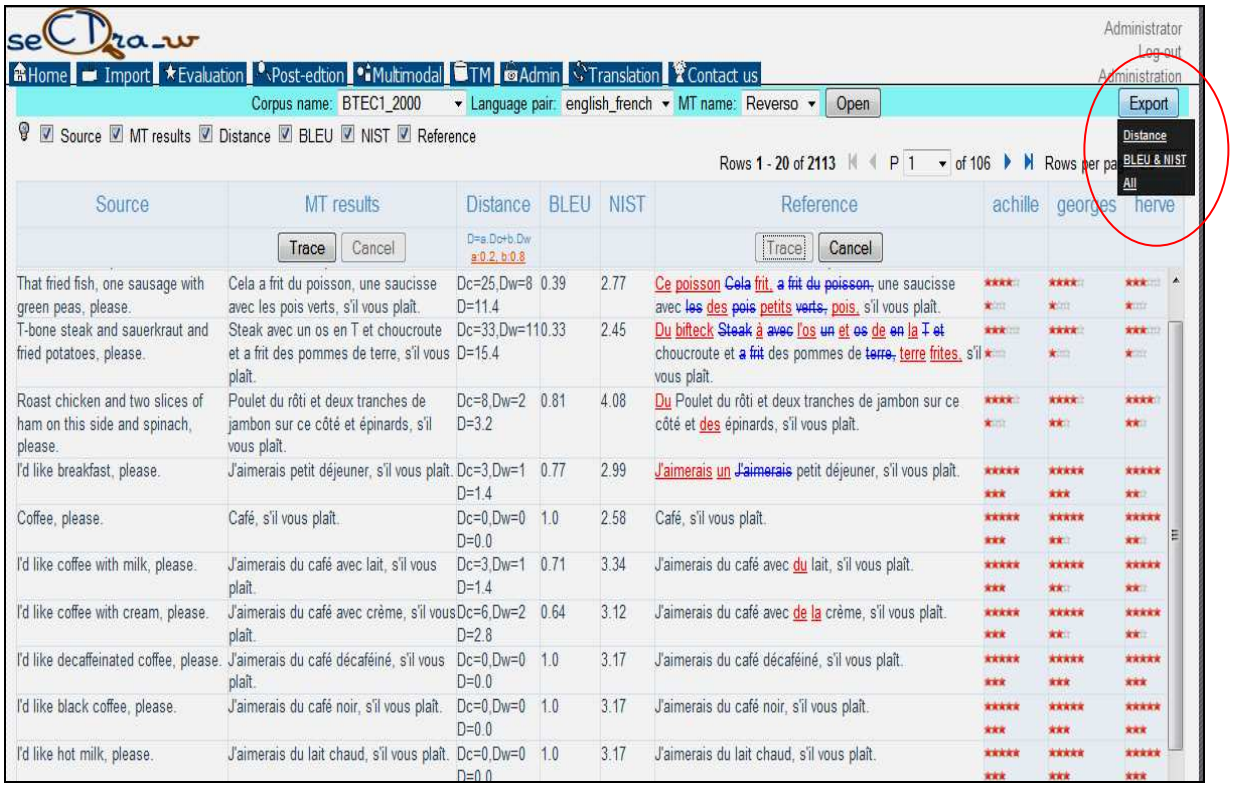


Figure 87: Corpus export fonction

III.3 Translating and Post-Editing Corpora of Translations

SECTra_w supports collaborative environment allowing the collaboration of human works on translation corpora.

It offers numerous functionalities aiding contributors to accelerate, to share their works, and to communicate with each other.

III.3.1 Create a new translation project

If you are an administrator, you can create a new translation/post-edition project by moving the mouse over the Admin menu, then choosing the *Create New Project*, next entering a project name, and finally clicking on the "Create" button.

Currently, this function can be only used by the super administrator of SECTra_w because of some security issues.

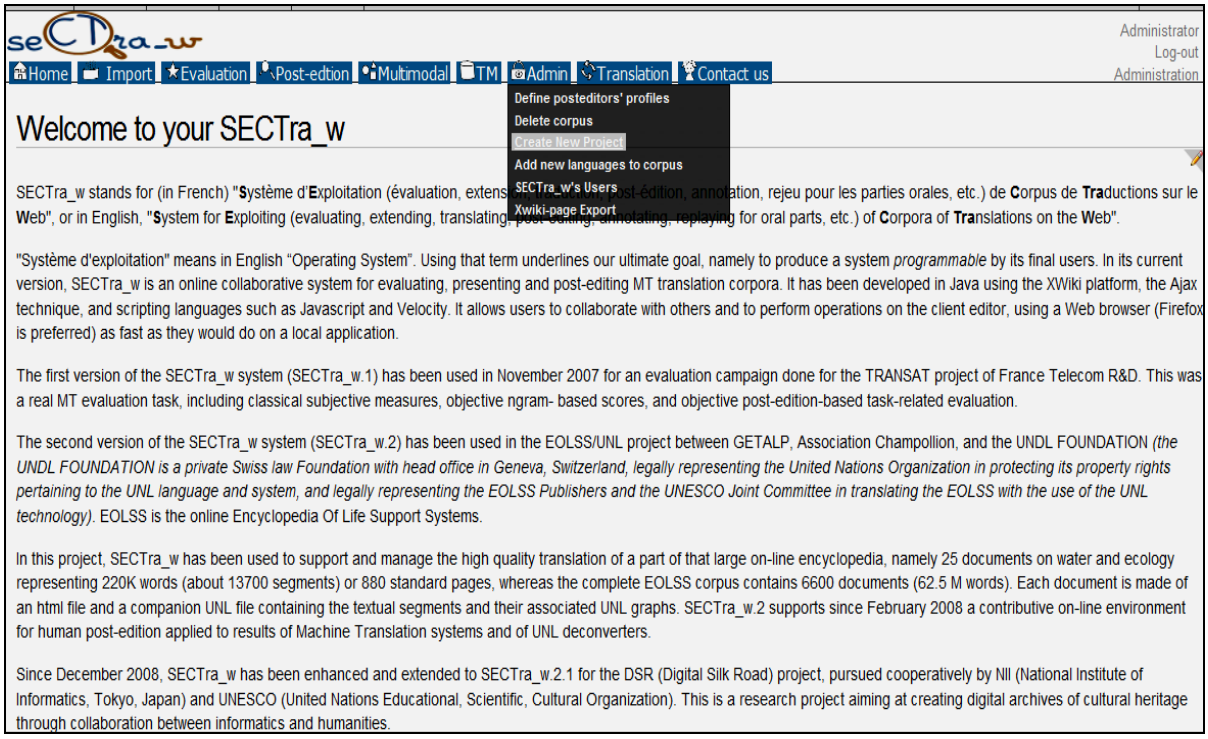


Figure 88: Admin menu

Here is the interface for creating a new project.

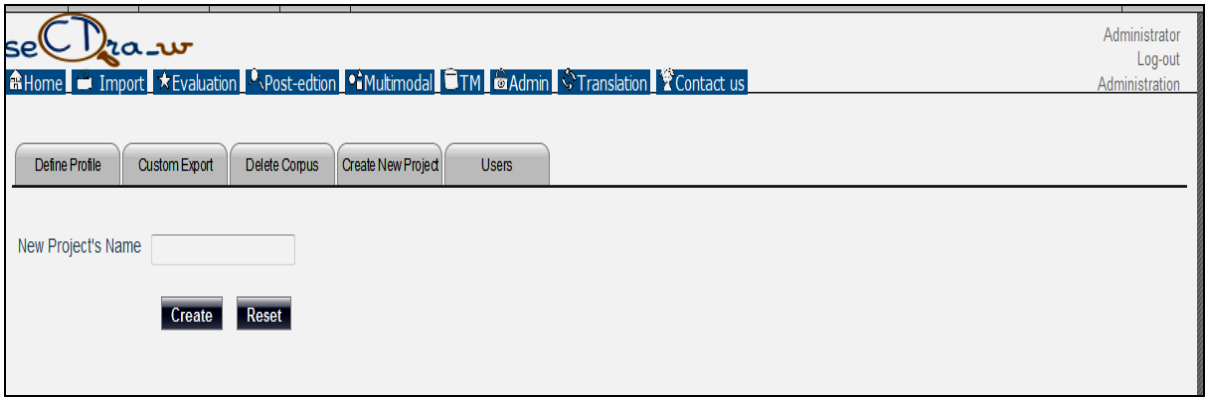


Figure 89: Interface for creating a new project

III.3.2 Configuring and defining user's profiles

If you are an administrator of the Post-editing projects, you can define profiles for post-editors in the project such as translation levels, rights, etc. by moving the mouse over the *Admin* menu and then choosing the *Define Post-editors' Profiles.*

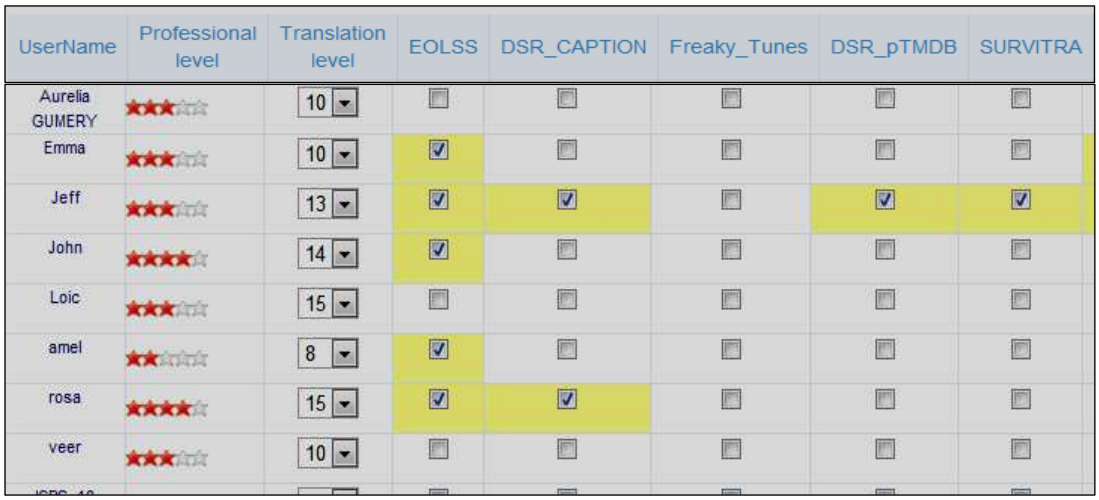


Figure 90: Interface of post-editors' profiles definition.

In the screen above, the project administrator can define for each post-editor some default values: professional level, default score, projects, languages pair(s). This is done through a form, by clicking on the appropriate number of stars, selecting the default value of the *translation score* of the contributor, and checking the checkboxes corresponding to the corpora s/he is authorized to access.

Depending on the a priori translation quality of each contributor, the default values of his score will be set differently by the administrator, but he can modify them later.

For example: MT is assigned 2 stars (**) by default, a professional translator is assigned 4 stars (****) and 15/20, a normal post-editor is assigned 3 stars (***) and 10/20.

Hint: The translation score of each contributor is used to assign a translation quality score to each segment by default, but this value can be changed on each segment by the posteditor.

For example, a '***+12/20' post-editor can say "I did a very good job on that segment", and put a 17/20 score on it, or say "I still have a problem with this segment", and give it a mere 10/20.

III.3.3 Import translation corpora for post-editing

If you are an administrator of a post-edition project, you can import translation corpora into SECTra_w for preparing the translation/post-edition work.

Before importing translation corpora into SECTra_w, you have to convert your corpora into one of the predefined formats and structures that are acceptable by SECTra_w. Here are some explanations about them.

III.3.3.1 Corpus structure

A corpus consists of many documents, and a document contains:

- necessarily, a file of source segments;
- Optionally, one or more files of translated segments (translations may be MT pretranslations by Systran, Reverso, Google, etc., or human translations).
- Optionally, one companion file (for the EOLSS corpus, these files are of .unl type, for DSR corpus there are DSR pages, etc.)

III.3.3.2 Corpus format

The format of the source and translated files can be text, xml, or excel.

For text format. A segment is represented by 1 line of text of the form:

Id: text

A segment in a source file is aligned with one or more corresponding translations in the translation files by its Id.

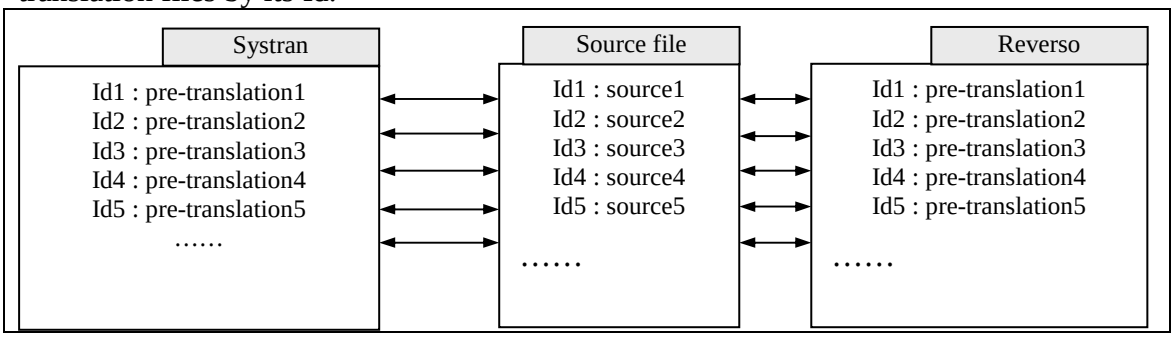


Figure 91: Aligned files of text format

For the Excel format. A segment is represented by two cells of the same row: the first cell contains an Id and the second contains a text segment.

	A	B
1	ID	En
2	E-290.9-HE01-023/V-1/0013/01	An unpleasant path for a carriage, to the south of Korla
3	E-290.9-HE01-023/V-1/0015/01	An old East Turkish man in Schinnega
4	E-290.9-HE01-023/V-1/0016/01	Cool shade in Schinnega
5	E-290.9-HE01-023/V-1/0025/01	Orderly placed canoes in Kontje
6	E-290.9-HE01-023/V-1/0026/01	A fleet of canoes on the bank of Kontje-daria
7	E-290.9-HE01-023/V-1/0031/01	Lunch at kontje-daria. The author, Chen and Kung.
8	E-290.9-HE01-023/V-1/0032/01	Loading a canoe, Kontje-daria
9	E-290.9-HE01-023/V-1/0037/01	An old poplar tree with a blue heron nest, Kontje-daria
10	E-290.9-HE01-023/V-1/0038/01	A cooking unit of kontje-daria. Li, Chia Kwei, Gagarin and two rowers
11	E-290.9-HE01-023/V-1/0051/01	A wild boar with its cubs
12	E-290.9-HE01-023/V-1/0052/01	A fleet gliding down Kontje-daria
13	E-290.9-HE01-023/V-1/0054/01	No. 56 camp of April 8
14	E-290.9-HE01-023/V-1/0057/01	Overlooking Kontje-daria from Sai-tjeke
15	E-290.9-HE01-023/V-1/0058/01	A 76-year-old Muslim shepherd from Ak-basch
16	E-290.9-HE01-023/V-1/0061/01	Khodai Kullu
17	E-290.9-HE01-023/V-1/0062/01	Departure from Sai-tjeke
18	E-290.9-HE01-023/V-1/0067/01	Patients in a Hummel tent. Dilpar.
19	E-290.9-HE01-023/V-1/0068/01	The author in his own double canoe
20	E-290.9-HE01-023/V-1/0069/01	Sattma, a hut made of reed, at Dilpar

Figure 92: Excel format

III.3.3.3 How to import

In order to import the corpus to be post-editing, we follow the steps below:

1) Click on *Import* menu,
2) Choose a corpus format (Text, Excel,…),
3) Choose corpus name,
4) Enter a document name (if the first import of document) or choose a document name in the existing documents list (if the document has been already imported and this time you want to update sources or add new translation files…),
5) Choose a language pair, attach files into corresponding fields, and finally click on Import button.

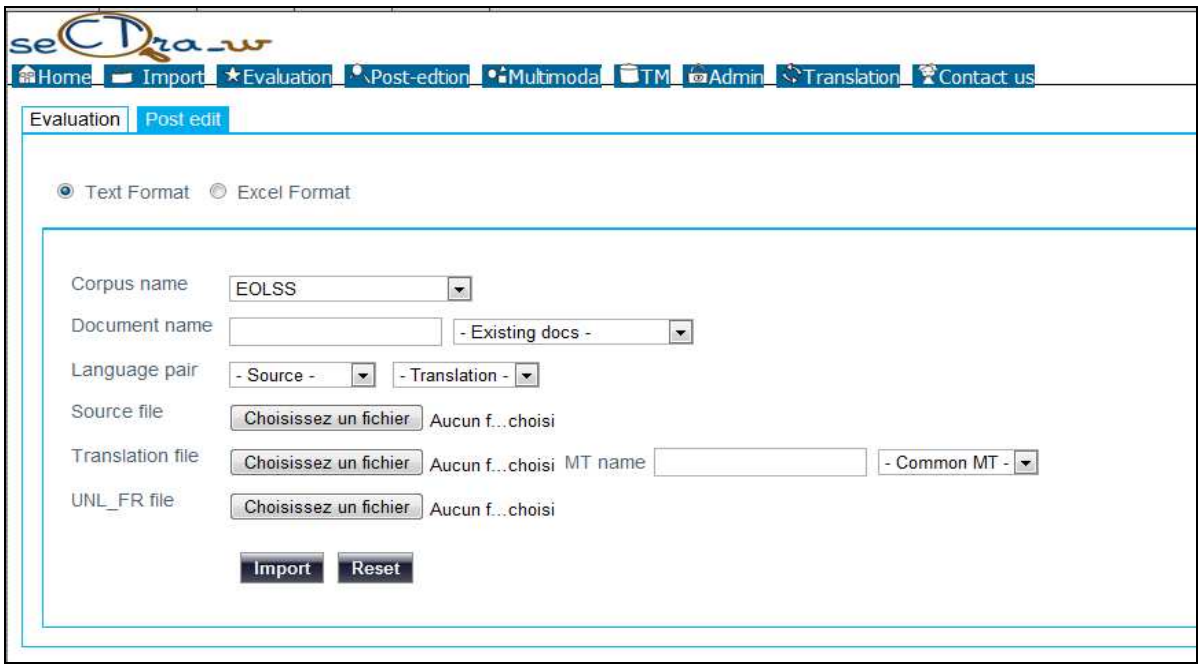


Figure 93: Import interface of post-editing corpus

III.3.4 Add (extend) new languages to existing corpus

Administrators can easily add new languages to an existing corpus in two possible ways, as follows:

1) Submit source files to MT systems such as Systran, Reverso, Google, etc. to get pretranslations files for new languages and then import these pretranslations files into SECTra_w by choosing the corresponding corpus, document name, and new target languages.
2) Open *Extend corpus* function by moving the mouse over the *Admin* menu, clicking on the *Add new languages to corpus* submenu, then choosing a corpus name, a document name, and new target language, and finally clicking on the “Start” button.

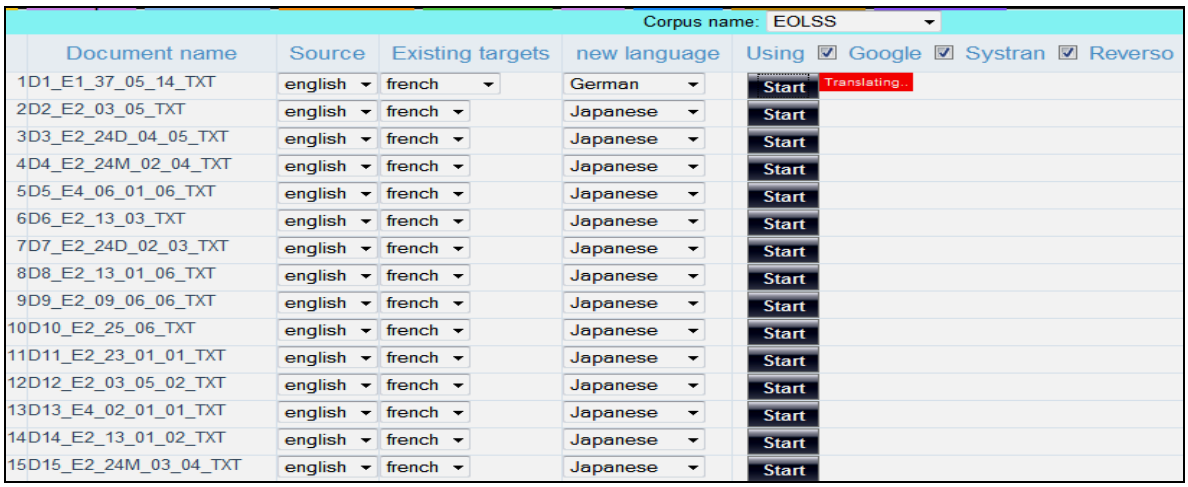


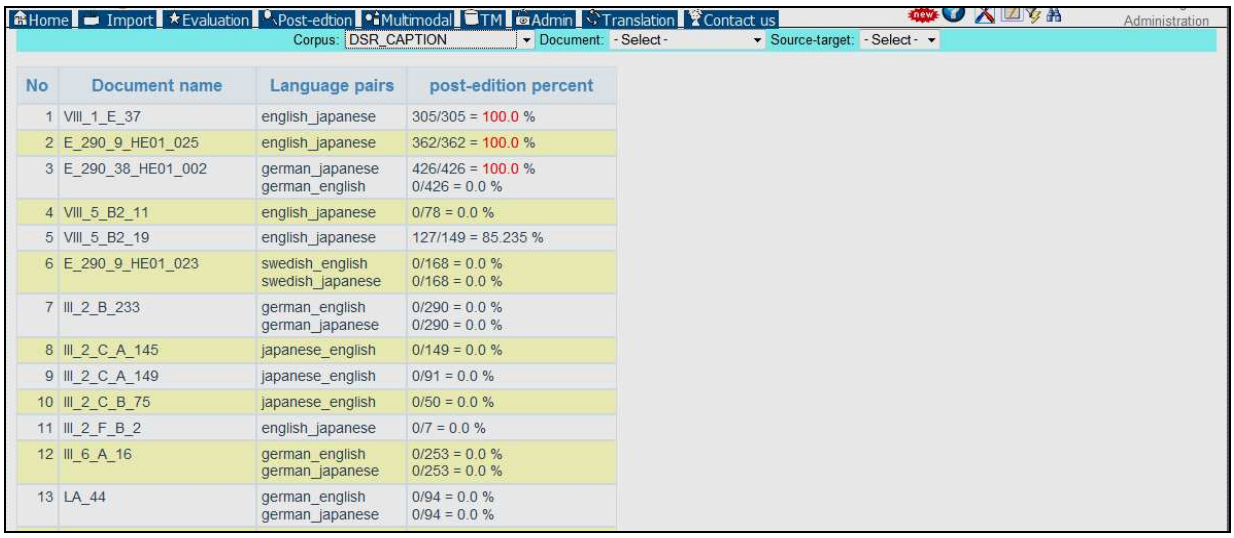
Figure 94: Calling MT systems

As you see in the screen above, as soon as the *Start* button is clicked, SECTra_w calls MT systems to translate the new segments in the document, and a red message "*Translating*" appears. This message will turn into "*Finished*" when the translation task is completed.

Figure 95: Finishing status of translation process

III.3.5 Post-editing a corpus

If you're a post-edition/translation project manager and you want to watch the progress of the translation/post-edition of your project, namely the percentage of translation/post-edition for each languages pair of a document, you click on the *Post-edition* Menu, and finally choose a relevant corpus name.



No	Document name	Language pairs	post-edition percent
1	VIII_1_E_37	english_japanese	305/305 = 100.0 %
2	E_290_9_HE01_025	english_japanese	362/362 = 100.0 %
3	E_290_38_HE01_002	german_japanese german_english	426/426 = 100.0 % 0/426 = 0.0 %
4	VIII_5_B2_11	english_japanese	0/78 = 0.0 %
5	VIII_5_B2_19	english_japanese	127/149 = 85.235 %
6	E_290_9_HE01_023	swedish_english swedish_japanese	0/168 = 0.0 % 0/168 = 0.0 %
7	III_2_B_233	german_english german_japanese	0/290 = 0.0 % 0/290 = 0.0 %
8	III_2_C_A_145	japanese_english	0/149 = 0.0 %
9	III_2_C_A_149	japanese_english	0/91 = 0.0 %
10	III_2_C_B_75	japanese_english	0/50 = 0.0 %
11	III_2_F_B_2	english_japanese	0/7 = 0.0 %
12	III_6_A_16	german_english german_japanese	0/253 = 0.0 % 0/253 = 0.0 %
13	LA_44	german_english german_japanese	0/94 = 0.0 % 0/94 = 0.0 %

Figure 96: Progress of the translation/post-edition for a project

If you're a post-editor, you should click on the *Post-edition* menu. Then you select the *Corpus name*, *Document name*, and *Language pair* (source and target languages) you wish to work on.

If you can understand more than two languages (source and target), you can choose a third language as reference language (Figure 97: Post-edition interface with reference language).

Figure 97: Post-edition interface with reference language

In this situation, you can use the two languages as source languages for your translation work. This reference language column can be shown/hidden through the corresponding checkbox at the top of the interface.

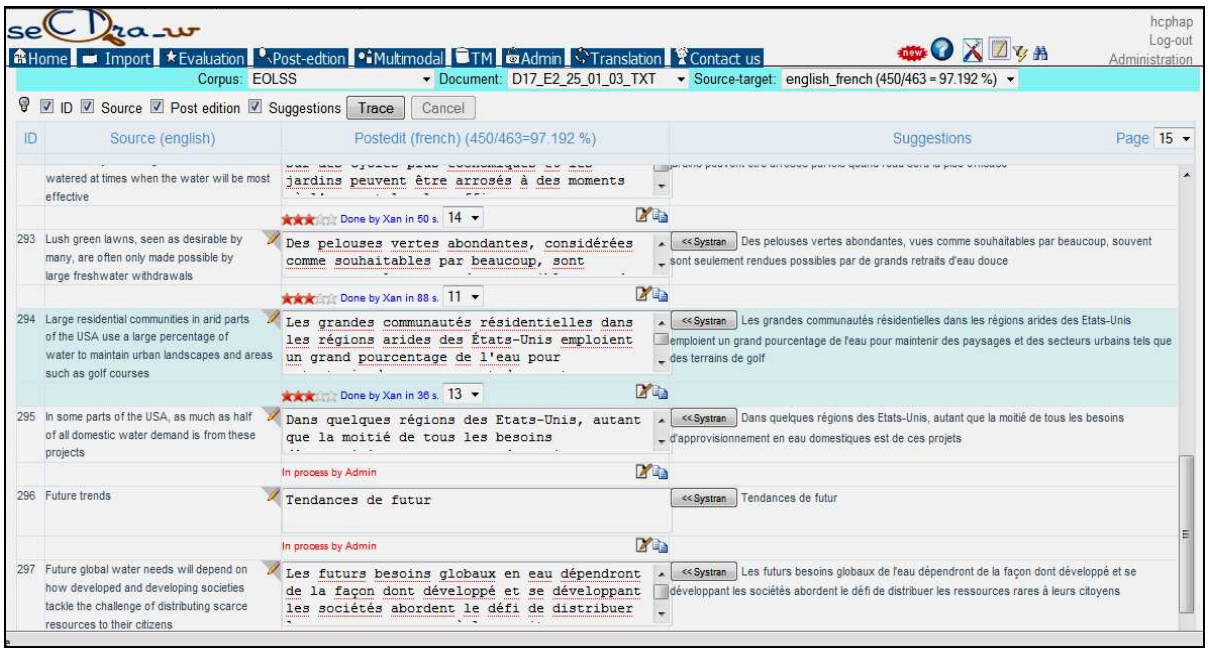


Figure 98: Post-edition interface without reference language

As you see in Figure 97 & Figure 98 above, the post-editing window is divided into three sections. The first section is fixed and contains all control information such as column names, paging navigator, show/hide table columns, etc.

The second section is dynamic and contains the corpus presentation. Post-editors can scroll data in this section using a vertical scroll bar. The content of the Web page shown is by default a logical document page equivalent to about 250 words (a standard page), but this number of words can be configured by administrators.

The third section is the dictionary area, allowing looking up meanings of source words, as well as improving and enriching the specialized lexical database. This dictionary area presents the dictionary that is chosen by the post-editor. The plan is to connect it with a pTMDB (M. Daoud) prepared for the DSR project and its languages in a contributive way and stored in a PIVAX instance.

Each segment has 3 statuses of post-editing:

1) Not yet post-edited by a human. For this case, you can see a hint “In process by yourself” in green color under the segment and 2 red stars (**).
2) Being post-edited by another user. For this case, you can see a hint “In process by XXX” in red color under the segment and you can’t post-edit this segment until its postedition has been completed.
3) Already post-edited (once or several times). In this case, you see a hint “Done by XXX” in blue color under the segment. You can modify a segment with this status. When you start modifying it, the status of this segment is changed to status 2 so that the segment becomes uneditable by other post-editors. As soon as you finish post-editing a segment and leave it, your work is saved automatically and a new version of the post-edition is created.

III.3.5.1 How to post-edit

In order to translate/post-edit efficiently and use all aides, you have to accept to work as a post-editor, but not as a reviser, that is to "post-edit".

First, you have to click on the source language part of segment. The row containing that segment enlarges, and precomputed helps appear: the dictionary area contains the entries associated to that segment, the MT pretranlations are also enlarged, and the best suggestions found in the Translation Memory are added to them.

The clock counting the post-editing/translating time on that segment starts.

You should then read the source text to understand it **before** looking at the post-edition cell, which is initialized with the “best” pretranslation or suggestions, according to some criterion[1]. If the initialization seems less usable than some other suggestion or pretranslation, you may choose another one by a simple click.

You then directly edit the text in the post-edition cell. Any modification on the segment will be automatically saved when you click the mouse out of that post-edition cell.

Beware: If you work as a reviser (reading first the target text (suggestions) and going to the source only in case of doubt or incomprehension), you won’t receive any aid.

If you see that the quality of the segment is perfect and you need not enhance it, you can click on the translation level for that segment (e.g., click one of the *** stars shown) to confirm and save the post-edition.

Optionally, you can also change the value of its score (from 1 to 20) which appears under the segment[44] (Figure 77).

III.3.5.2 Dictionary help

In the process of post-editing, you can be helped by looking at the meanings of words or terms extracted from the current source segment.

The dictionary pane is at the bottom of the post-edition editor. This pane has an absolute position, therefore you can easily find and use its content during the post-editing process.

[44] At this moment, we simply initialize each post-edition field by the first MT output computed for it (Systran, Google, or Reverso), and we cannot claim it is “the best”. There is a lot of research on evaluation of individual MT outputs, and we plan to experiment with some proposed methods later.

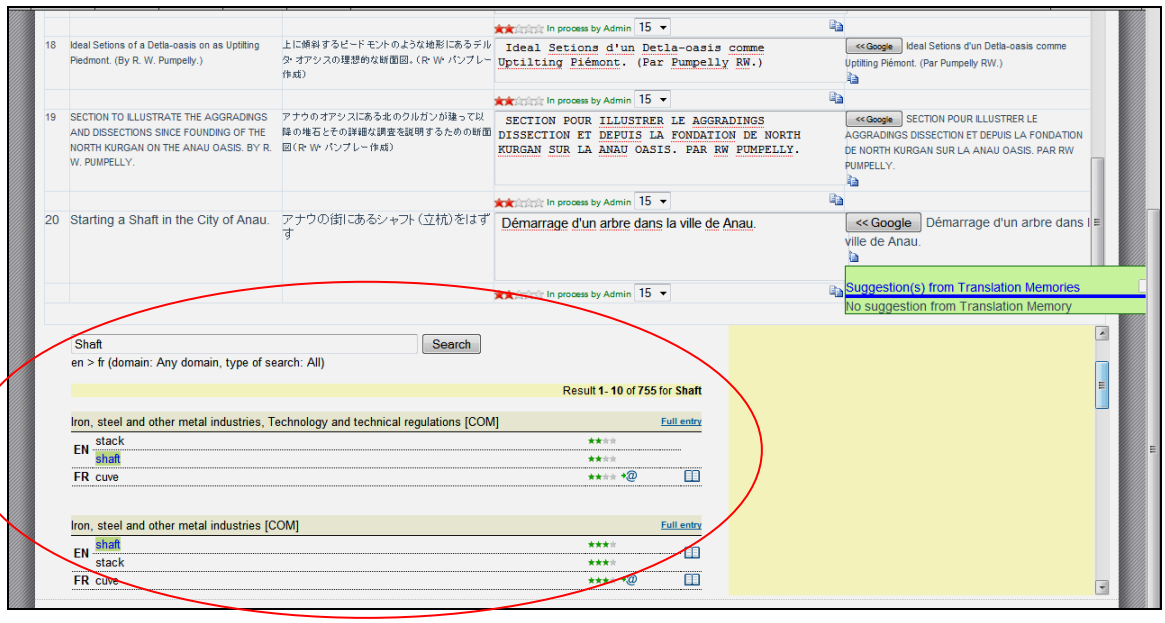


Figure 99: Lexical informations helps

III.3.5.3 Multi post-edition versions management

A new feature of SECTra_w is to manage multiple post-edited versions of a segment.

You can view all post-edited versions of a segment by clicking on the “Showolderversions” icon as shown by the screenshot below. One can also choose an older version if the current version is worse than it.

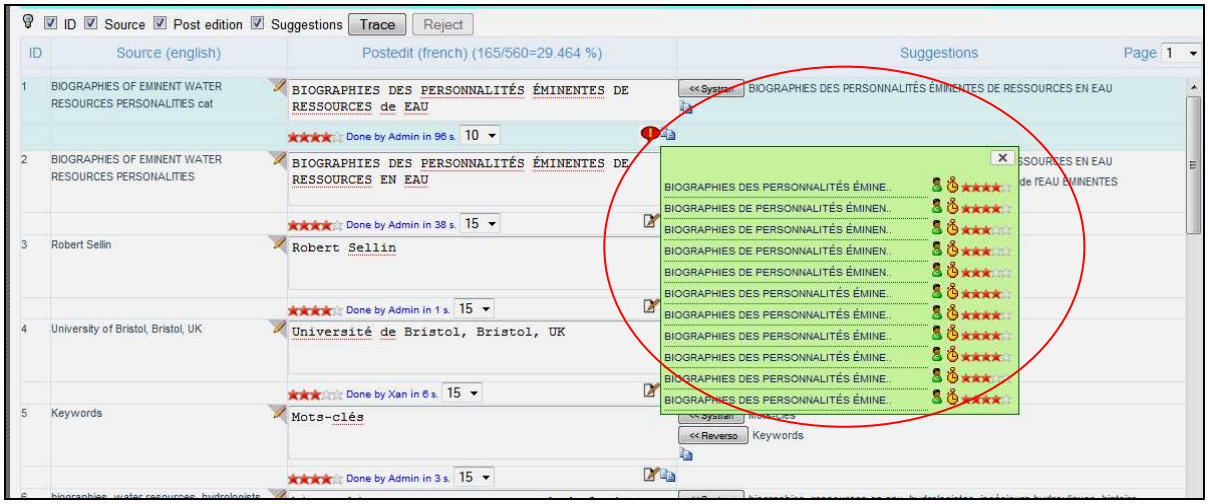


Figure 100: Multi post-edition versions for a segment

III.3.5.4 Show/hide and resize table columns

If you don’t want to see a column, you can uncheck the show/hide checkbox corresponding to that column. For example, the editor can visualize at the same time several languages, but if you just understand three of these languages, you can hide the language columns that are not necessary for you.

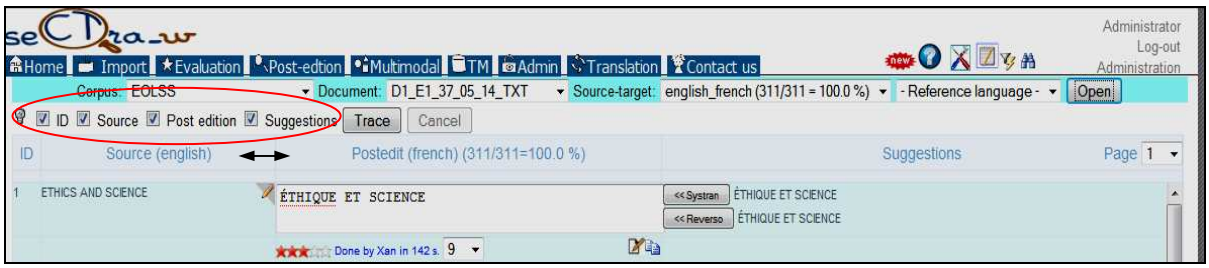


Figure 101:Show/hide and resize a column

In the current post-editing editor, the suggestions column is set two times larger than the post-edit column by default. But if you want to resize the size of a column, you can use the mouse to do it by simply pointing to a column border with the mouse pointer and dragging the border to a new location.

III.3.5.5 Giving comments and sharing translation knowledge and experiences

During the translation work, human translators need to comment and share their skills and experiences with each other. For each translation project, target language, etc., there are many different conventions, special terms, transliterations for proper names, place names, etc.

Therefore, along with translation quality markers for each segment, such as stars for level and digits for score and stars, SECTra_w also allows human translators to comment on each segment. In order to comment on a segment, translators click on the corresponding icon (Figure 102) of that segment, then a comment text area appears, where translators can put their comments, explanations, or even some questions, etc.

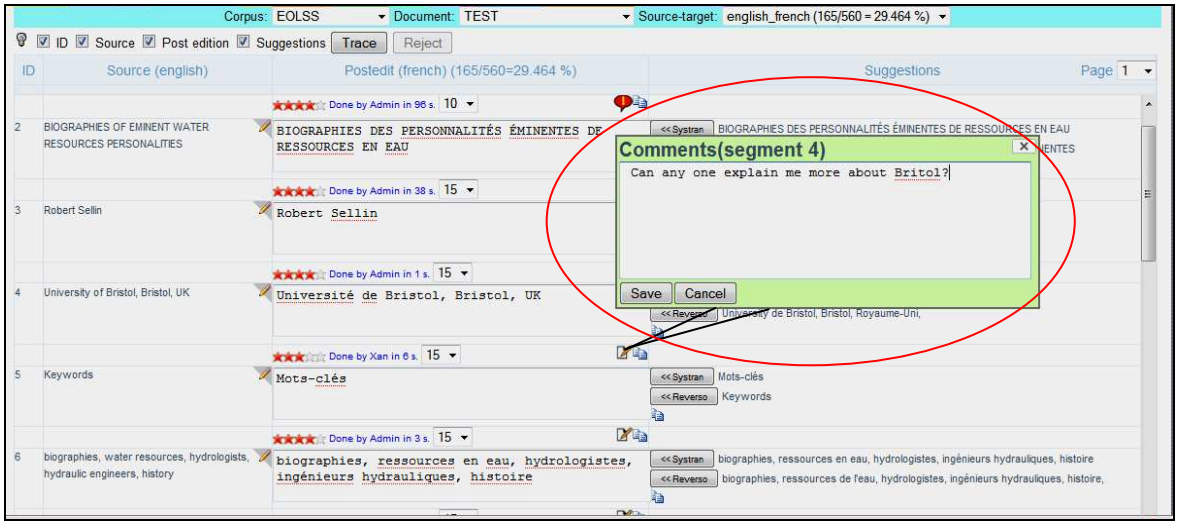


Figure 102: Comment on a segment

Human can easily see whether a segment contains comments or not because the comment icon changes.

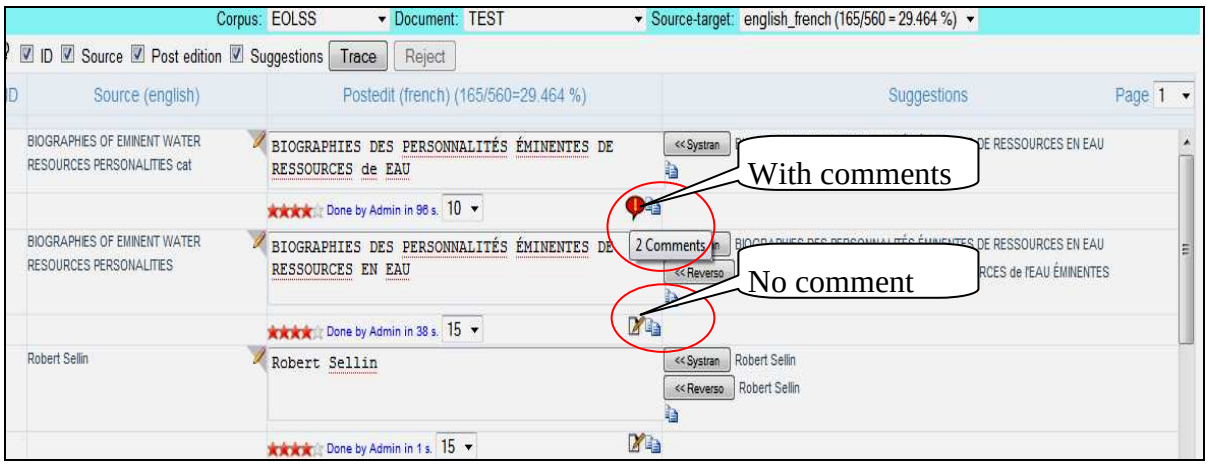


Figure 103: Difference of comment icons

There is also a global comment allowing Project Administrators and senior professional translators to propose suggestions, share experiences, etc., and other contributors to ask questions. This global comment is also like a forum where all human contributors and project Administrators can share and discuss everything concerning a project. To activate this function, click on the *Global comment* icon on the right top corner of the Post-edition interface.

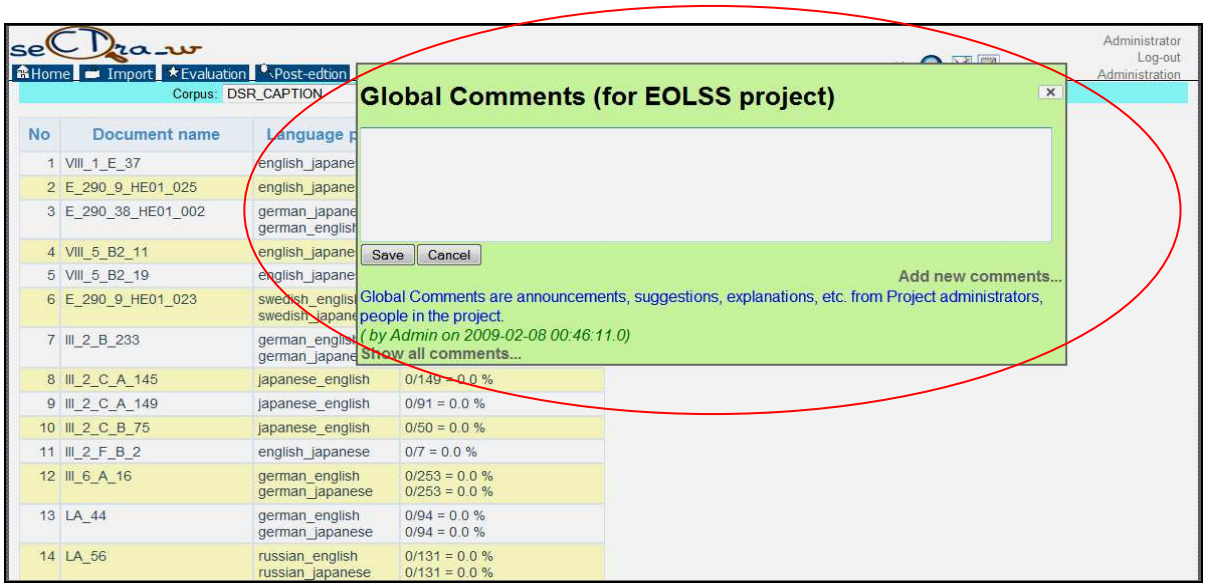


Figure 104: Global comment

This global comment can list all comments for individual segments so that project managers or senior translators can see them and reply to them. Click on *Show all comments* to show that.

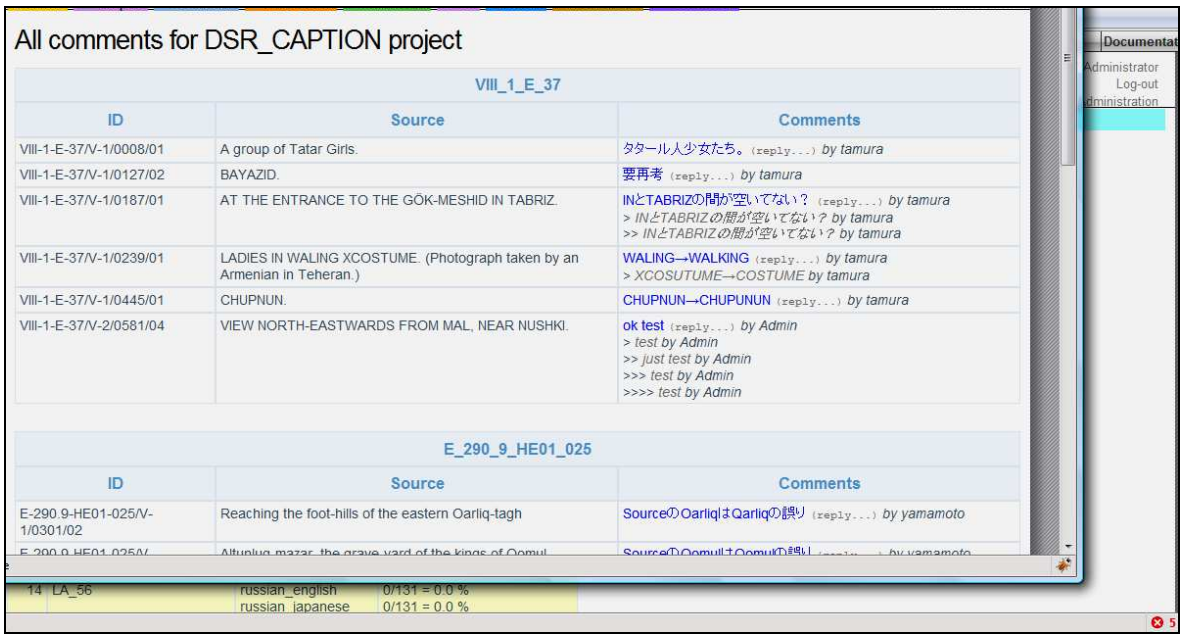


Figure 105: List all comments for individual segments

III.3.5.6 Correct source texts

Sometimes the source texts are not correct. For example, in the DSR caption corpus, the source texts have been extracted automatically by OCR software, they therefore contain a lot of errors.

However, allowing everyone to edit source texts is a dangerous task. Therefore, SECTra_w supplies a way to correct source texts for translators who have enough rights but prevents other people to do this. Project administrators can give these rights to translators by adding them into the Project admin group.

If you have the right to do that, click on the icon  at the source segment that you want to edit.

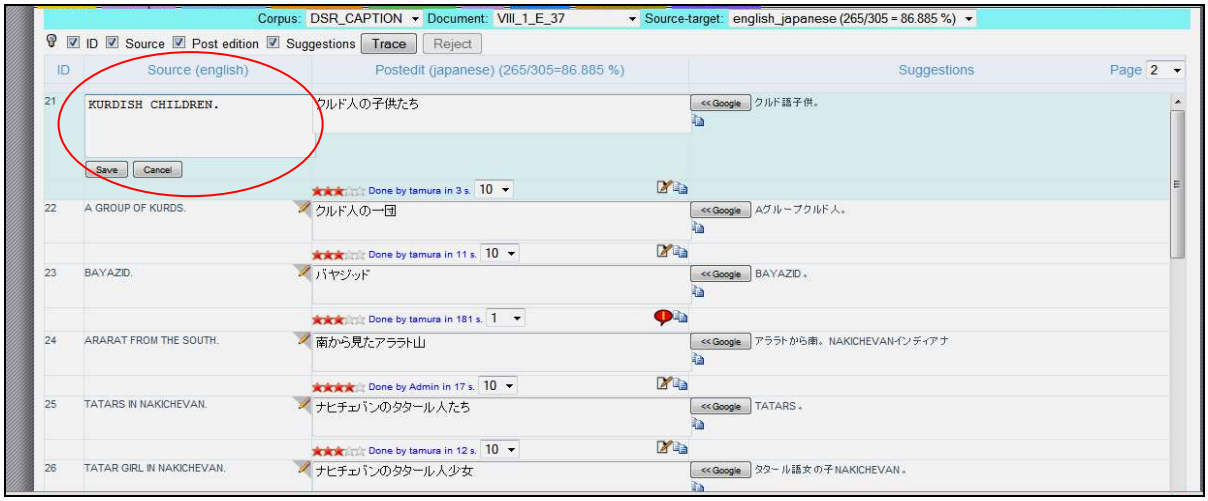


Figure 106: Correct source texts

III.3.6 Filter segments not yet post-edited

We can observe the progress of the post-edition of a document by percent ratio for each language pair. But this is not enough to know exactly which segments have not been post-edited.

SECTra_w supplies a Filter function to filter all segments that have not yet been post-edited.

To use this function, you should click on the icon at the right top of the editor window. Then choose the corpus you want to filter.

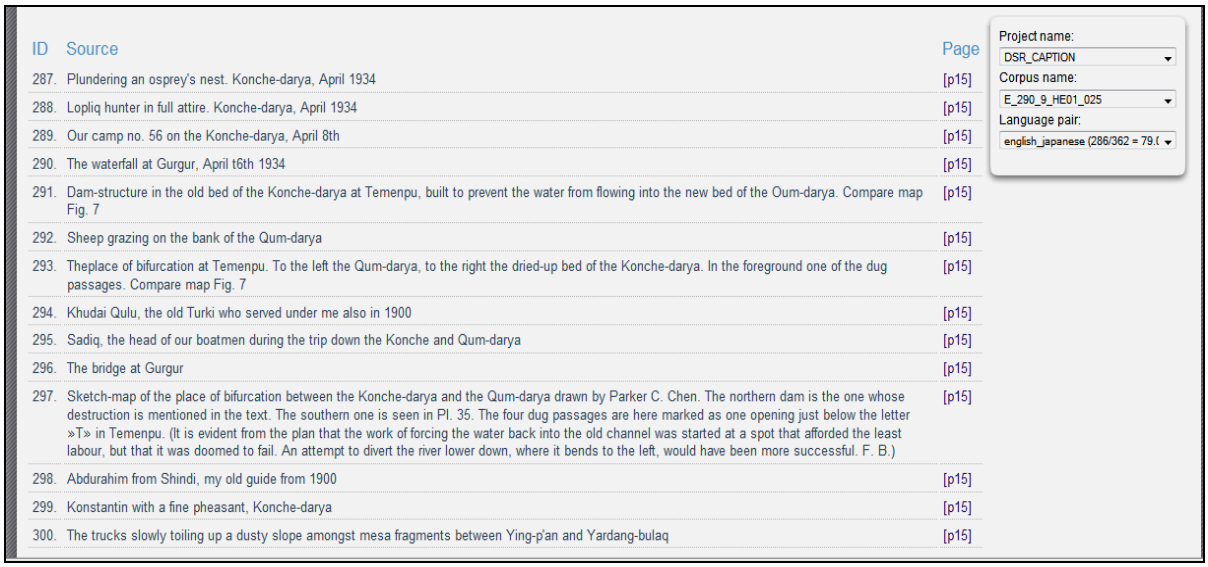


Figure 107: Filter segments not yet post-edited

All segments not yet post-edited are shown (see screen above) along with their ids and clickable links to the corresponding pages that bring you to the post-editing.

III.3.7 Find and replace a fragment/word

It is frequent that a source term appears many times in a document and is badly translated by MT systems, or even in the TM. In this case, post-editors have to modify all its translations in a whole document or in the whole corpus. To do that, SECTra_w supplies a *Find and replace* function that can find a term/word or replace a term/word by another in a page, document or corpus. To use this function, click on the icon at the right top of the editor. Then choose the domain of your find and replace operation.

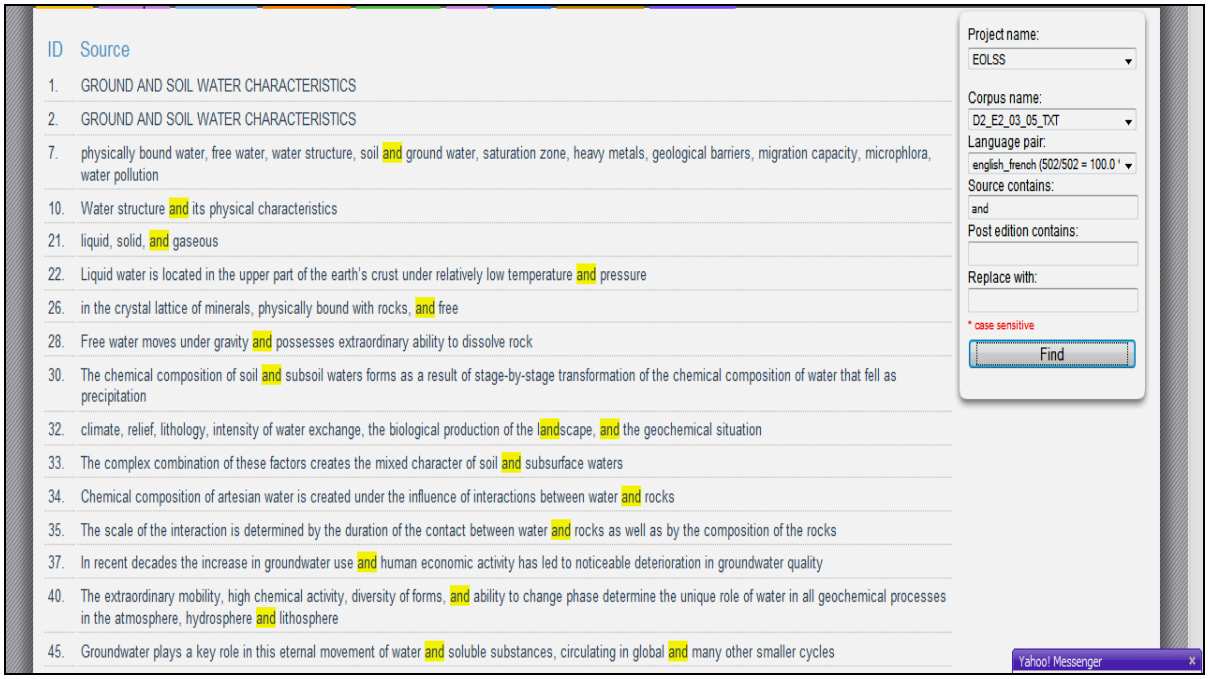


Figure 108: Find and replace function

This function allows you to replace a term by another term for each segment or all segments found.

III.3.8 View documents and translation context

The post-edition interface can be accessed either directly, or by viewing an enriched Html form of the translated document, selecting a passage, and asking to post-edit it (somewhat like in Google's translation interface). The Html form is updated when changes are made, so that the current version in the target language is visible.

This function supplies the alignment between corresponding segments in the source and translated document. This feature helps post-editors easily comparing and observing the post-edition results. When we move the mouse over a segment in the source document or the translated document, that segment is highlighted as well as its aligned segment in the other document. However, usually the length of two documents is not the same, therefore the active segment and its aligned segment are not displayed in parallel and sometimes its aligned segment is blotted out by the windows. In order to make a segment and its aligned segment display in parallel, we can resize the column until the two documents spread out at the same vertical length.

To edit a passage, click on the segment that you want to edit. SECTRa_w will open or switch to the post-editing window at an appropriate position.

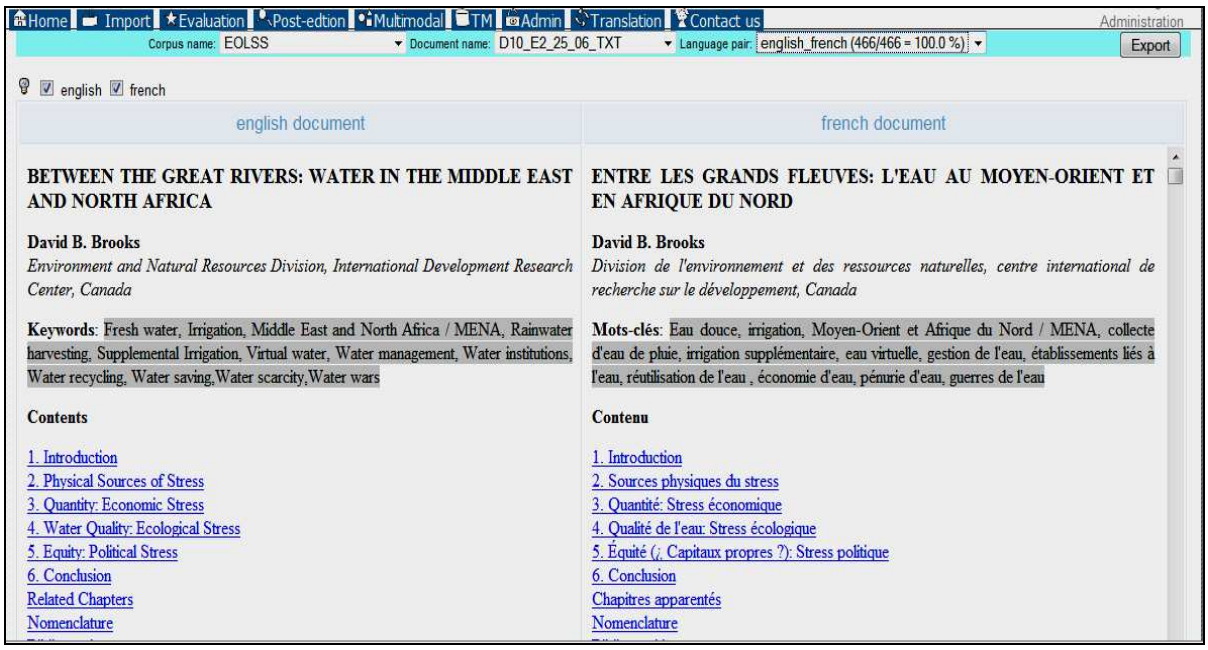


Figure 109: View source and target documents in parallel

You can turn off the source or target document column if you want.

This function can help contributors access the original segment in context.

That is useful for cases like the DSR and OMNIA corpora, where each document has an original image and an html form obtained by OCR, with some percentage of errors. Seeing the original image helps to detect these errors. The original segments can then be connected by the contributors, and the translation process, whether automated or not, can restart from the corrected segments.

For these corpora, the Id of each segment is linked to the image part of the Web page from which the segment is extracted. To see that image page, just click on the Id on the left of the segment.

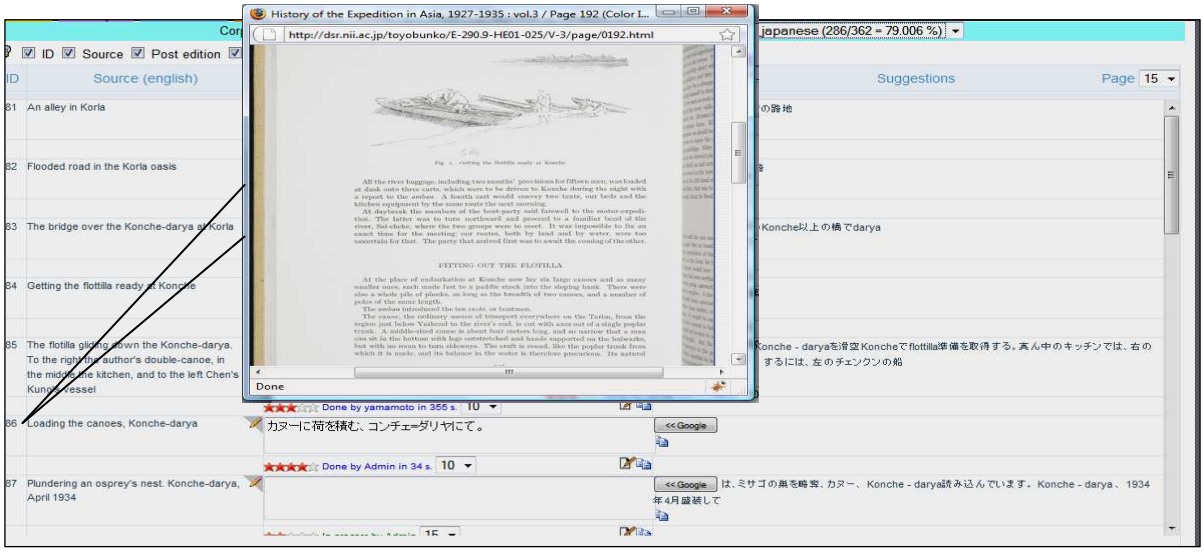


Figure 110: See the context of a caption segment

III.3.9 Statistics table and post-edition effort

This function allows you to observe the progress of post-edition for a project. You can know who did what and how many segments, words, hours, etc. have been post-edited and spent by post-editors. Of course, the times clocked can differ considerably from the time actually spent, and must be considered at best indicative.

If some contributors are paid, the measure used should be based on words, as in the translation profession:

number of words in the original text for a translation or for the initial post-edition performed by the same person (in 1 or more stretches),

word edit distance, for revision performed by the same person on a post-edited translation produced by another person.

Corpus: Statistic

	JEANPL	WALLY	MATHIEU	CATHERINENATALIZIA	PLANAS	ADMIN	HCPHAP	BOITET	NICOLASSIMON	ROSA	XAN	ETIE
Document name	D1_E1_37_05_14_TXT											
#segments post-edited	271	9	27	2	1	40	2	0	0	0	0	
#words post-edited	6137	84	388	43	4	222	7	0	0	0	0	
#chars post-edited	40341	531	2744	314	25	1594	39	0	0	0	0	
average words/segment	22.645	9.333	14.37	21.5	4.0	5.55	3.5	0	0	0	0	
time post-edited	111154(s)	111(s)	1031(s)	29(s)	0(s)	1855(s)	171(s)	0(s)	0(s)	0(s)	0(s)	0
Average time/segment	410.162(s)	12.333(s)	38.185(s)	14.5(s)	0.0(s)	46.375(s)	85.5(s)	0	0	0	0	
Document name	D2_E2_03_05_TXT											
#segments post-edited	232	40	0	0	0	11	5	85	132	0	0	
#words post-edited	4065	858	0	0	0	82	34	1555	1675	0	0	
#chars post-edited	26295	5314	0	0	0	555	222	9607	11004	0	0	
average words/segment	17.521	21.45	0	0	0	7.454	6.8	18.294	12.689	0	0	
time post-edited	43638(s)	5751(s)	0(s)	0(s)	0(s)	1607(s)	365(s)	4489(s)	4002(s)	0(s)	0(s)	0
Average time/segment	188.094(s)	143.775(s)	0	0	0	146.09(s)	73.0(s)	52.811(s)	30.318(s)	0	0	
Document name	D3_E2_24D_04_05_TXT											
#segments post-edited	0	20	7	0	0	0	0	52	37	6	5	

Figure 111: Statistic table showing post-edition effort

We will enhance this function so that we can see the Ids of segments along with their quality levels (* to *****) and scores.

Moreover, clicking on the Trace button in the Post-edition interface shows all changes between MT outputs and their post-editions. The computation of our mixed edit distance uses 3 operations, insertion I, deletion D and exchange X, at the character and word levels, but the visualization is done at word level, replacing Xyz by DyIz. This feature is very helpful for anyone to get an idea of the post-edition effort. For Project administrators, it is a way to detect differences in the work done by different persons.

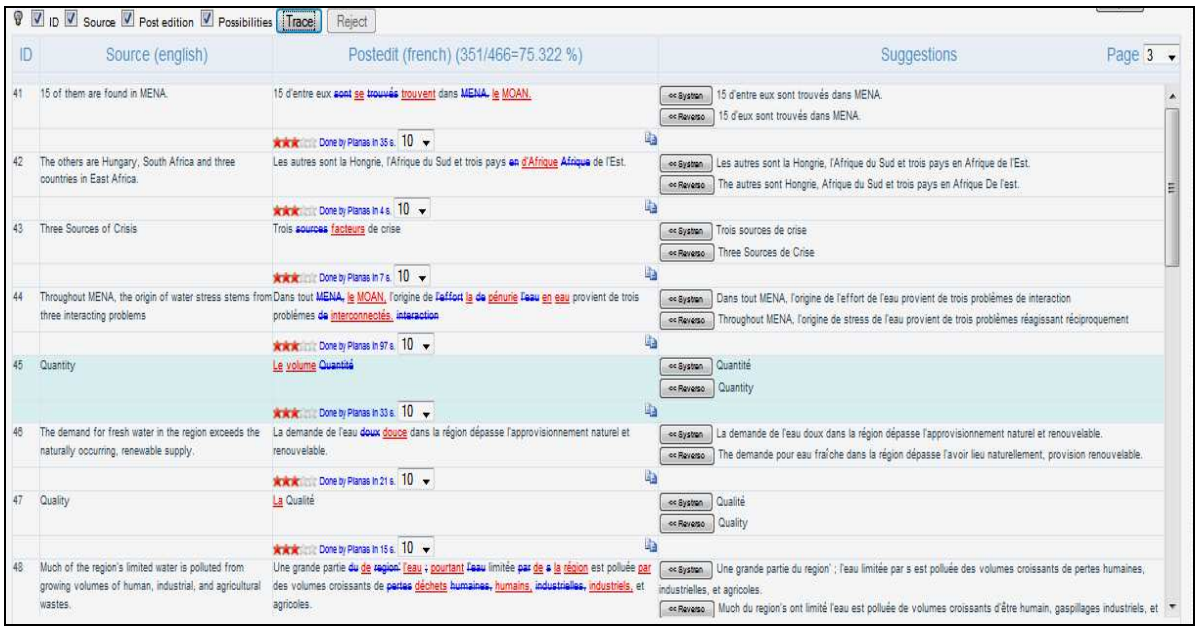


Figure 112: Trace of post-edition obtained by reconstituting edit operations

III.3.10 Exporting translation corpora

If you're an administrator, you can export or download a result corpus whenever you want by moving the mouse over the *Post-edition* menu, clicking on the *Export corpus* menu, and choosing *Corpus name*.

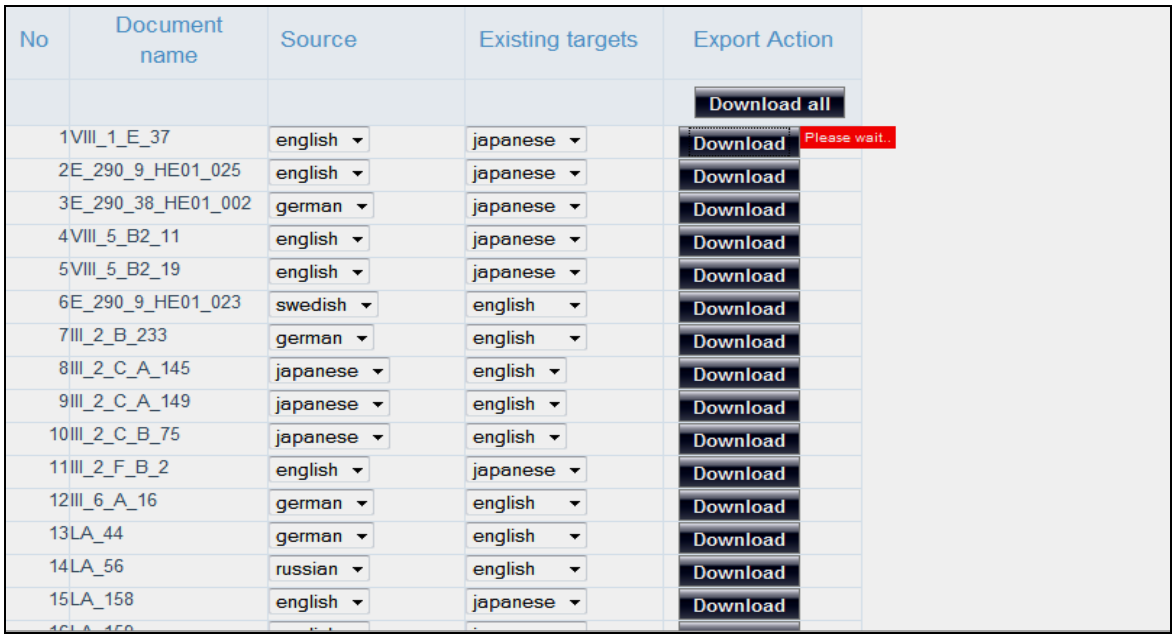


Figure 113: Downloading function interface

You can choose a document to be downloaded or download all documents by clicking on the *Download* or *Download All* button. Then, you need to wait a short time for SECTra_w to produce the result corpus. As soon as it finishes producing the result, SECTra_w puts up a dialogue for saving or opening documents.

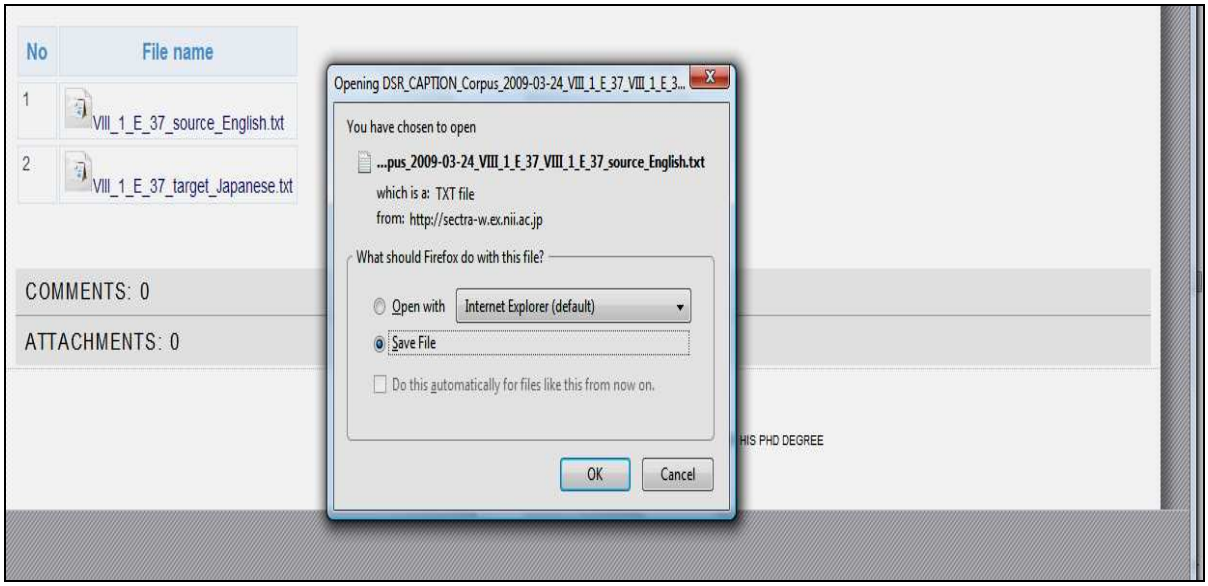


Figure 114: Saving or opening document

Especially, for downloading .unl file and HTML files including source and target HTML files like the case of the EOLSS project, open the function *View parallel documents* (Figure 109), click on the "Export" button, then follow the instructions to export the translation corpus.

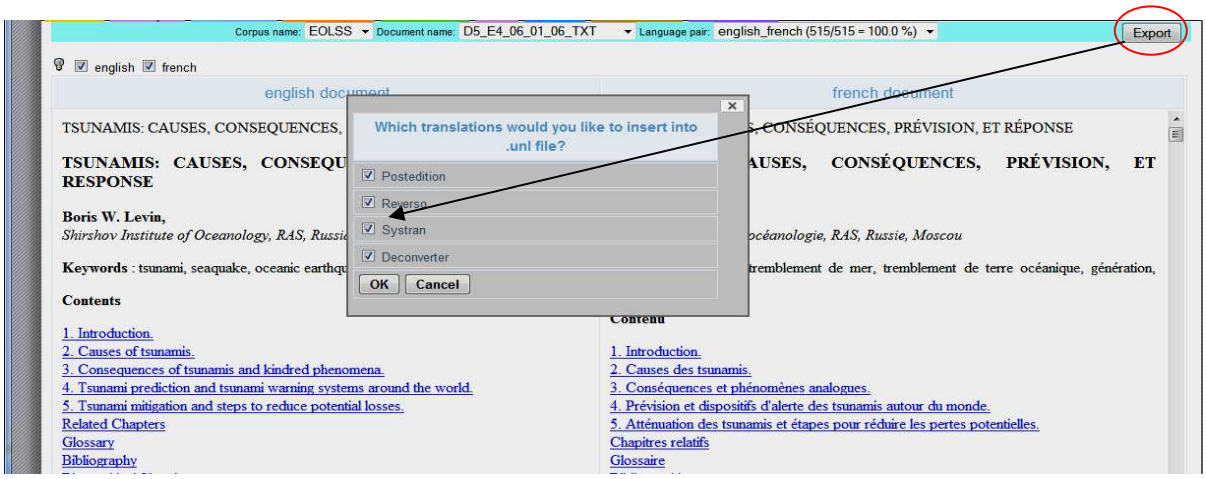


Figure 115: Downloading an .unl file

III.4 Annotating and replaying multimodal corpora

SECTra_w can support bilingual corpora such as the ERIM corpus of interpreted spoken bilingual dialogues (between French and Chinese, Vietnamese, Hindi and Tamil). For that type of corpus, users need a system permitting a wiki-like usage, to enable the study, distribution, and collaborative improvement and annotation of corpora.

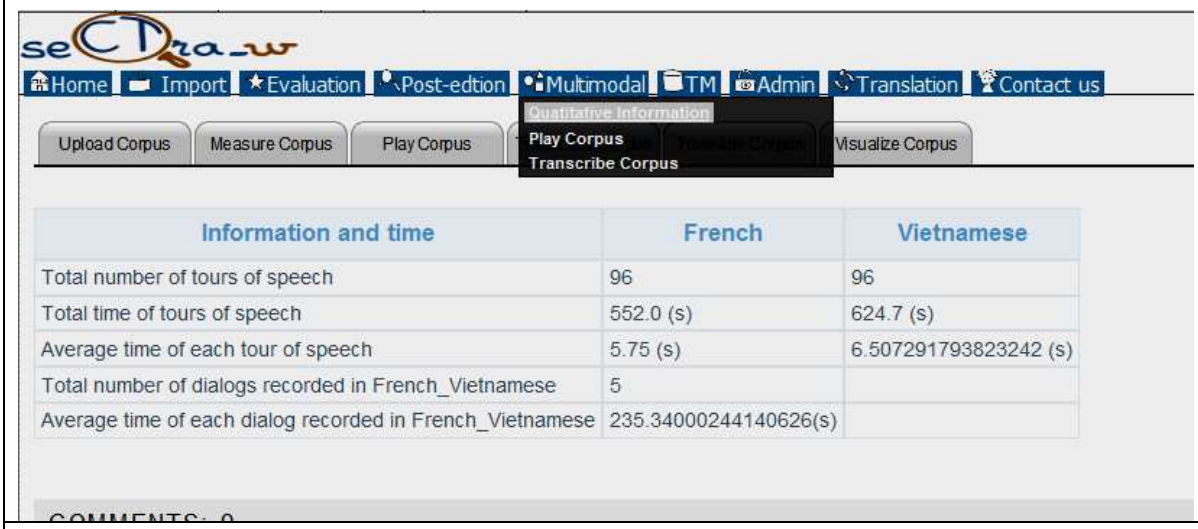
SECTra_w allows to manage not only textual forms of various types for written corpora (raw, or preprocessed by word or chunk segmentation, punctuation suppression, or syntactic annotation), but also, for SLT (spoken language translation), the primary audio form, aligned with various transcriptions and annotations.

III.4.1 Corpus study and measurement

It is possible to "replay" one or more dialogues, filtering them by language and/or interlocutor, and showing the associated short texts and transcriptions on demand.

There are functions to compute and show quantitative information about the corpora it contains.

To show the information for the part of the languages pairs (French-Vietnamese, French-English, etc.) dialogues in the ERIM corpus, move the mouse over *Multimodal* menu, and choose *Quantitative Information.*



Information and time	French	Vietnamese
Total number of tours of speech	96	96
Total time of tours of speech	552.0 (s)	624.7 (s)
Average time of each tour of speech	5.75 (s)	6.507291793823242 (s)
Total number of dialogs recorded in French_Vietnamese	5	
Average time of each dialog recorded in French_Vietnamese	235.34000244140626(s)	

Figure 116: Quantitative information of French-Vietnamese dialogues in ERIM corpus

III.4.2 Transcription and annotation

Many kinds of annotations can be considered: transcriptions, translations of the transcriptions, comments of different types, and all annotations possible on texts, at various levels (morphology, syntax, semantics, pragmatics). For the moment, SECTra_w only supports the first 2 types.

The transcription environment inherits from the editing and replay environments. Each speech turn is repeated on demand, or with a certain frequency, while the user types the transcription, until s/he user goes to the next one (or another one).

Because speech recognition (ASR) is not yet freely available for the language pairs tackled so far, and in any case is bad over telephone lines or Voice/IP, we did not yet include the possibility to call an ASR for automatic pre-transcription.

As far as the translation of transcriptions is concerned, we simply reuse what has been developed for MT evaluation.

External MT systems may be called if available, and the post-edition interface is adapted to the structure of the dialogues.

The disposition and width of the columns can be changed according to the needs. For example, to compare the French transcriptions and translations of transcriptions, it is useful to place the corresponding columns near to one another.

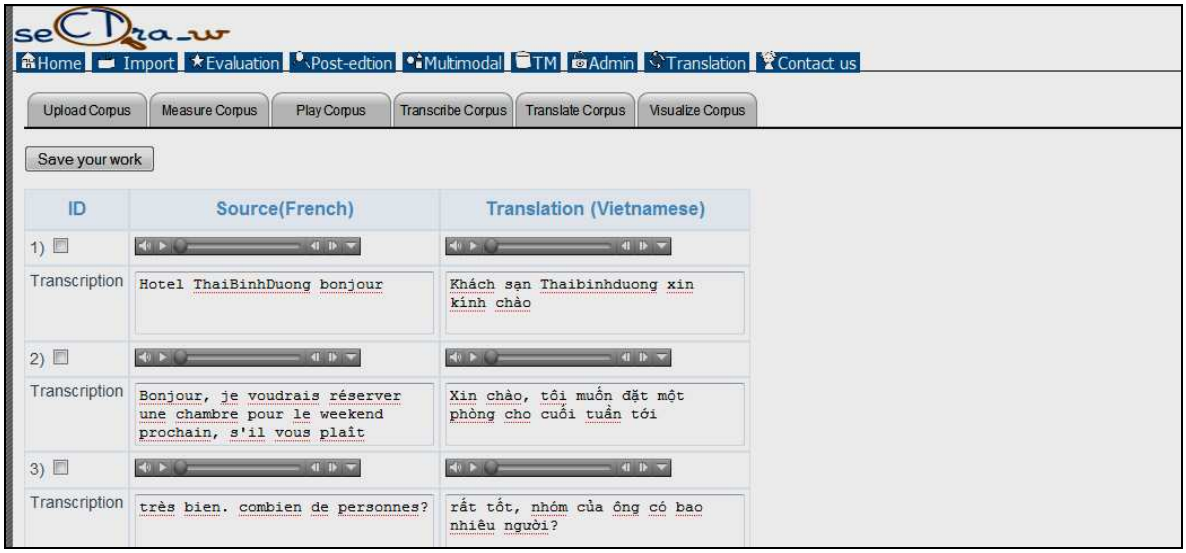


Figure 117: Transcription of ERIM corpus

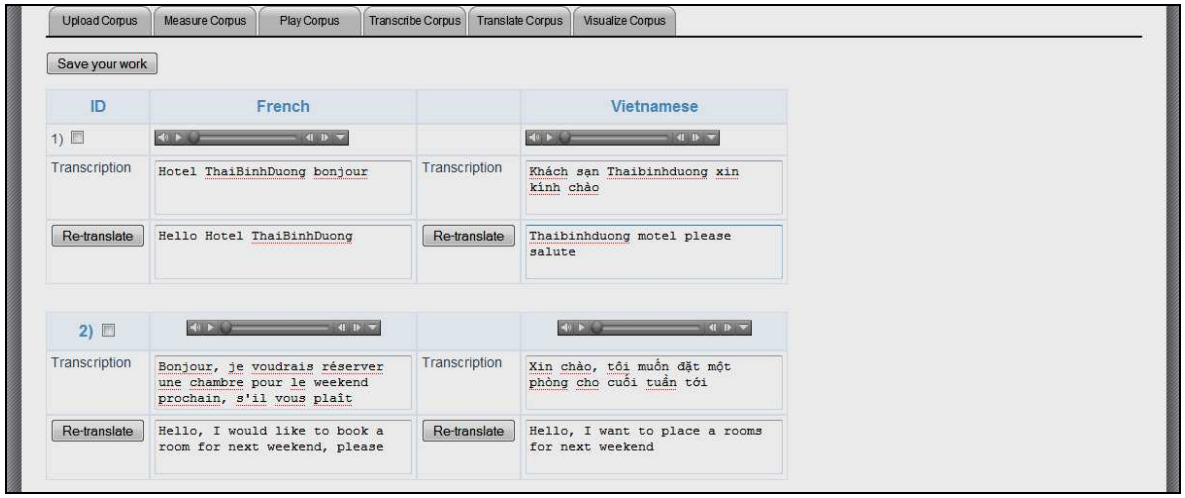


Figure 118: Translation of transcription for the purpose of alignment

III.4.3 Replaying spoken bilingual corpora

SECTra_w allows us to listen to entire dialogues multimedia corpora (such as ERIM) in one or two languages. This function helps verify a multimedia corpus easily. To play the ERIM

corpus, move mouse over the *Multimodal* menu, choose play corpus, choose a dialogue, and choose the language (s) to listen to.

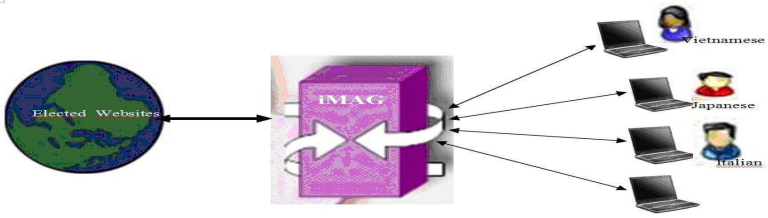
Figure 119: Playing a dialogue in ERIM corpus

III.5 iMAG-SECTra_w

An iMAG is a website used as a *gateway* allowing a multilingual access to one (in general) or several *elected* websites. The name "iMAG" stands for *interactive Multilingual Access Gateway*.

Apparently, an iMAG is similar to existing well-known translation gateways such as Google Translate, Systranet, BabelFish, Reverso, etc. The first essential difference is that an iMAG is only used for *elected* websites. This allows the iMAG to manage the multilingualization of certain websites better than existing translation gateways. With an iMAG, one can enhance the quality of translated pages, starting from raw output of general-purpose and free MT servers, usually of low quality and often understandable unless one understands enough of the source language.

Another difference is that, even when a translation gateway like Google Translate allows readers to propose better translations, these contributions are not reflected in the translated Web page, and are not used in later editions of the same translated page. They are probably used (after validation) to recompile the MT system later, but they are not remembered as such (probably because of the potentially gigantic size of a translation memory "for the whole Web" and of the low chance to find exact matches of postedited sentences in the millions of page translation requests coming to the MT server every day).



With an iMAG, we can enhance the quality of translated pages by post-editig automatic translations in two ways: (1) post-editing segment by segment (directly in the reading context, as with Google Translate), or (2) post-editing the whole page by using SECTra_w.

You can access to the iMAG via the link : http://eolss.imag.fr/xwiki/bin/view/imag/home.

The home page of the iMAG contains a list of elected websites.

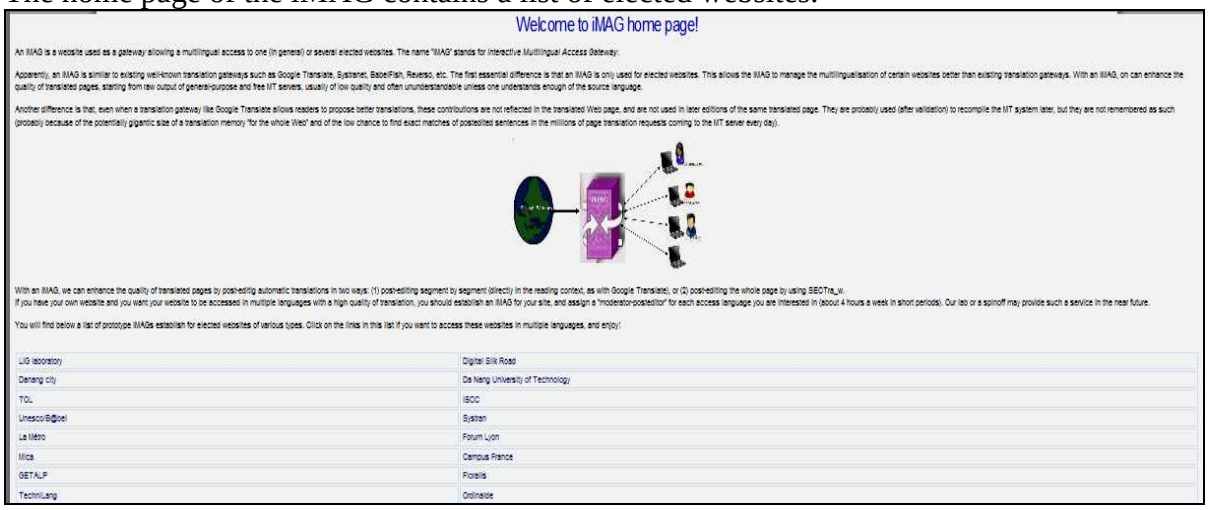


Figure 120: iMAG's home page

From this list, you can choose one from this list to view. For example, here is the iMAG for the LIG lab website.

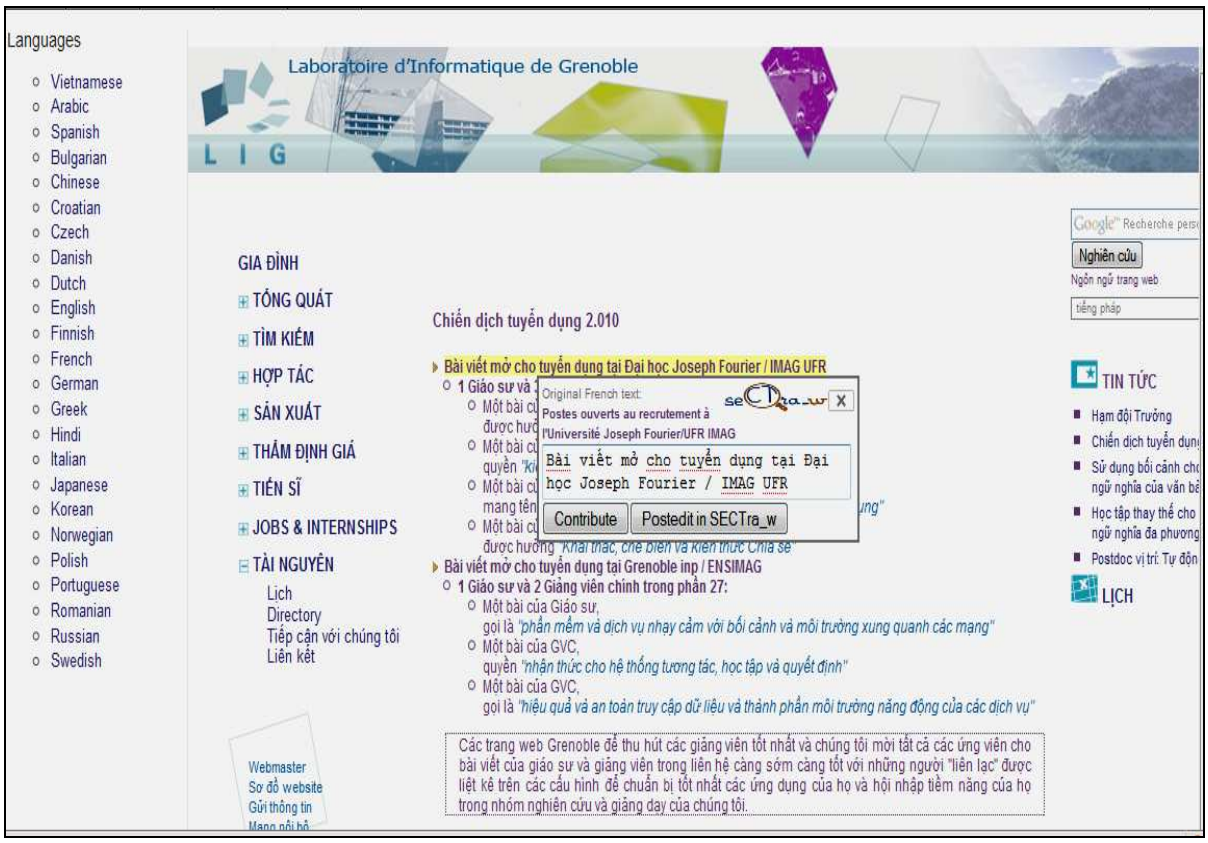


Figure 121: iMAG screenshot

To enhance a translation, you can move the mouse over the translation, and a popup window is displayed. From this popup menu, you can propose your translation. Your proposition will be then sent to SECTra_w in order to update the corresponding dedicated translation memory.

You can post-edit the whole page in SECTra_w in terms of a temporary SECTra_w document by click on the button *Postedit in SECTra_w* (Figure 121).