Normalisation et alignement de corpus français et vietnamiens : Format et Logiciels

Brigitte Bigi, Viet-Bac Le

LIG - CNRS UMR 5217 - 220, rue de la chimie - B.P. 53 - 38 041 Grenoble Cedex 9

{Brigitte.Bigi,Viet-Bac.Le}@imag.fr

Abstract

The creation of text corpora requires a sequence of processing steps in order to constitute, normalize, and then to directly exploit it by a given application. This paper concentrates on the aspects of methodology and linguistic engineering, which serve to develop a multipurpose multilingual parallel text corpus. The original documents can come from various sources like HTML or ASCII written in different languages: French and Vietnamese. A document structuring method and some text corpus normalization tools are proposed in this paper.

Résumé

La création d'un corpus électronique exploitable par une application donnée nécessite une chaîne de traitements afin de constituer, normaliser puis exploiter ce qui, au départ, n'est qu'un ensemble de documents textuels quelconques. Cet article se concentre sur les aspects de méthodologie et d'ingénierie linguistique qui sous-tendent l'élaboration de corpus multi-lingues parallèles non dédiés à une tâche. Les documents d'origine peuvent provenir de différentes sources telles que HTML ou ASCII, de différentes langues : français et vietnamien. Une structuration de type XML et un ensemble d'outils logiciels sont proposés pour normaliser les corpus, et créer des alignements.

Mots-clés : formalisme, XML, logiciel, normalisation, corpus, multi-lingue, alignements.

1. Introduction

Depuis quelques années, l'essor de la puissance informatique a permis une grande facilité de stockage, et les corpus de documents ne cessent de croître. La plupart des pays du monde sont connectés à l'Internet qui devient ainsi une ressource intéressante pour la collecte de documents textuels. Le Wold Wide Web est devenu une des plus importantes sources d'information disponible de manière électronique, particulièrement pour les langues peu dotées. Il est gratuit, riche, important et accessible pour de nombreuses langues. Ainsi, des recherches se concentrent actuellement sur la construction de corpus de textes en collectant des pages Web (Kilgarriff, 2001) : pour une langue minoritaire donnée, pour une thématique donnée, pour des documents multi-lingues, pour des documents parallèles (Resnik, 2003).

Cet article propose un format XML et présente une batterie logicielle pour la normalisation de corpus de textes. Les outils développés, ainsi que le format de stockage proposé, se veulent volontairement très ouverts en terme d'application : statistique linguistique, modélisation du langage, recherche d'information, traduction automatique... Le logiciel peut, en effet, traiter des données en langue française ou vietnamienne, et peu de développement est nécessaire pour ajouter de nouvelles langues. Il permet aussi de traiter des corpus de très gros volumes (millions de documents), de différentes sources, avec un bon compromis temps/qualité.

Dans un premier temps, cet article présente le format XML choisi. Ensuite, les outils logiciels seront décrits pour la langue française et la langue vietnamienne. La dernière section sera consacrée à la notion d'alignement et la manière dont elle est traitée ici.

2. Description du format XML

2.1. Besoins pour nos applications

Disposer de corpus textuels sous forme électronique n'est pas une tâche simple. Les choix à faire sont nombreux et la démarche pour les réaliser difficile (Habert, 1998). Le logiciel décrit dans cet article a initialement été créé pour répondre à la problématique de la constitution de corpus textuels pour l'apprentissage de modèles statistiques du langage. En effet, la construction d'un nouveau modèle nécessite la recherche d'un corpus spécifique, ce qui implique systématiquement des redondances de traitements, de multiples copies des données, etc. La transformation au format XML permet de disposer d'une version complète, « propre » et unifiée des différents corpus disponibles, mais aussi ouverts pour tous les types de modèles de langage à créer (après sélection des documents pertinents par requête, par exemple).

La structuration XML a été conçue pour être indépendante de la langue, et offrir la possibilité de stocker plusieurs langues dans un même fichier. Pour une langue donnée, on peut également conserver plusieurs niveaux de décomposition d'une même phrase. Le format est aussi ouvert au problème de la traduction avec la possibilité d'intégrer des alignements entre les différentes décompositions.

2.2. DTD XML

Il existe de nombreuses techniques de description d'un document en corpus (par exemple le CES – Corpus Encoding Standard), ou des normes pour la description de documents électroniques (par exemple le RDF – Ressource Description Frameword). Sans avoir la prétention de concurrencer les travaux en cours ou déjà aboutis, et réalisés par de grands consortiums de spécialistes du domaine, nous proposons une DTD pour la définition des corpus textuels. Il s'agit d'une solution qui présente l'avantage d'être simple et fonctionnelle. Elle est particulièrement adaptée aux corpus très volumineux, et aux corpus multi-lingues. Afin d'en faciliter la lecture et la compréhension, la figure 1 présente la DTD sous forme graphique. Les éléments sont représentés dans des cadres avec leur nom en chasse fixe et les attributs en italique. Les cadres pointillés représentent des éléments optionnels, ceux en lignes pleine, des éléments obligatoires. La suite de cette section commente cette figure et propose de petits exemples types.

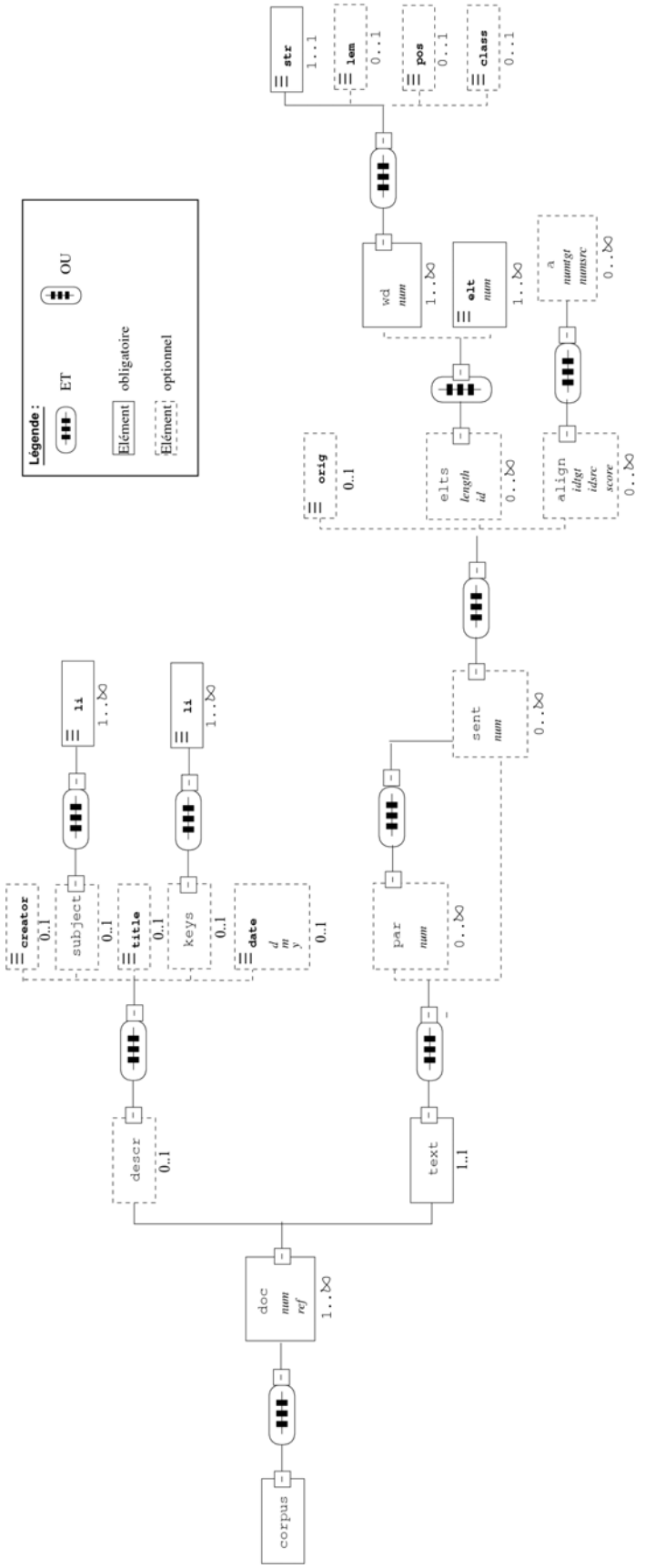

Figure 1 : DTD présentée sous forme graphique

Un corpus est constitué de documents qui comprennent une partie descriptive, et un contenu textuel. La description complète (facultative) d'un document contient :

- l'auteur ;
- le sujet, une liste de mots ou phrases ;
- un titre ;
- une date, décomposée en jour/mois/année ;
- un ensemble de mots-clés.

Dans le cas d'un fichier originalement en HTML, ces informations sont recueillies dans les champs META du document.

La partie texte du document se décompose en paragraphes (optionnel) et en phrases. Au niveau de chaque phrase, on conserve la phrase originale i.e. la phrase telle qu'elle est représentée dans le document d'origine, n'ayant subie aucune modification par le logiciel. On peut aussi ajouter une ou plusieurs décompositions de cette phrase en éléments (mots, syllabes, expressions, etc). On peut également choisir de transformer cette notion d'éléments simples en mots, unité sémantique à laquelle on peut associer des informations telles que le lemme, le POS, ou la classe. On montre ci-dessous un exemple de fichier respectant la DTD.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE corpus SYSTEM "corpus.dtd">
<corpus>
   <doc num="1">
   <descr>
          <creator>Robert Bouvier</creator>
          <title>Le parler marseillais</title>
          <date d="07" m="01" y="1999"/>
   </descr>
   <text>
          <par>
          <sent num="1">
          <orig>
          La langue d'un peuple est inscrite dans sa culture ; elle en est le
véhicule naturel, en même temps que le support de sa pensée et de sa
sensibilité.
          </orig>
          <elts>
          <wd>
                 <str> la </str>
                 <pos> DETFS </pos>
                 <lem> le </lem>
          </wd>
          <wd>
                 <str> langue </str>
                 <pos> NFS </pos>
                 <lem> langue </lem>
          </wd>
          ...
          </elts>
          </sent>
          </par>
   </text>
   </doc>
</corpus>
```

Par ailleurs, on prévoit de pouvoir réaliser des alignements entre les séries d'éléments, indifféremment intra-langue ou inter-langue. Par exemple, à partir d'une phrase originale, on peut conserver plusieurs niveaux de décomposition, et spécifier les correspondances entre les différentes unités. Les petits exemples ci-dessous montrent comment on peut utiliser les alignements intra-langues. Les alignements inter-langues sont décrits en section 4.

Exemple en français :

```
<sent num="3">
   <orig> Pomme de terre </orig>
   <elts id="target" length="3">
          <elt num="1"> pomme </elt>
          <elt num="2"> de </elt>
          <elt num="3"> terre </elt>
   </elts>
   <elts id="source" length="1">
          <elt num="1"> pomme_de_terre </elt>
   </elts>
   <align idtgt="target" idsrc="source" score="1.2879">
          <a numsrc="1" numtgt="1" />
          <a numsrc="1" numtgt="2" />
          <a numsrc="1" numtgt="3" />
   </align>
</sent>
```

Exemple en vietnamien :

```
<sent num="4">
   <orig> Việt Nam </orig>
   <elts id="syll" length="2">
          <elt num="2"> Việt </elt>
          <elt num="3"> Nam </elt>
   </elts>
   <elts id="word" length="1">
          <elt num="1"> Việt_Nam </elt>
   </elts>
   <align idtgt="syll" idsrc="word" score="2.2626">
          <a numsrc="1" numtgt="1" />
          <a numsrc="1" numtgt="2" />
   </align>
</sent>
```

3. Description du logiciel

3.1. Généralités

Afin de disposer de données textuelles exploitables, il est indispensable de les rendre homogène. Une fois le format choisi, la question du développement logiciel se pose. Cette section est axée sur les aspects de méthodologie et d'ingénierie linguistique qui sous-tendent l'élaboration de tels corpus. Dans ce domaine, les options sont nombreuses, et il est difficile, voire impossible, de faire « les choix exacts » dans la mesure où ils résultent de chaque situation. Dans un but de généricité, l'objectif est de rester le plus indépendant possible de l'application et de la langue visée. C'est pourquoi, après avoir déterminé les problèmes et traitements nécessaires pour l'obtention et le traitement de données textuelles dans différentes langues et différentes tâches, nous avons choisi un développement sous la forme de modules très spécialisés qui s'appliquent séquentiellement sur le texte. Cette solution offre de nombreux avantages, notamment la rapidité et la facilité de développement et d'utilisation,

ainsi que la simplicité lorsqu'il faut ajouter/modifier/enlever de nouveaux modules. Les modules souffrent donc des contraintes suivantes : ils doivent être facilement portables d'une langue à l'autre, et d'une tâche à l'autre afin de limiter le coût de construction d'un logiciel complet. Sous forme modulaire, il est en effet possible pour une tâche donnée, d'une part d'hériter des outils de traitement généraux, et d'autre part d'adapter rapidement et spécifiquement les autres outils. De la même manière, le « portage » vers une nouvelle langue permettra l'héritage des modules qui sont possiblement communs aux deux langues, et l'adaptation des modules qui sont intrinsèques à chaque langue. Pour réaliser cette approche, la langue « support » est le français, puis le logiciel a été ouvert au vietnamien. Les modules ont été développés et validés en environnement Linux. Ils sont écrits de manière claire et standard en langage *gawk* ; il est donc possible d'utiliser le logiciel sous tout environnement disposant de ce langage. Le logiciel est distribué librement sous licence GPL afin d'en facilité la diffusion et la modification.

3.2. Modules pour la langue française

3.2.1. Au niveau du document

- `Html2Text` : extraction du contenu et de la description d'un fichier HTML, conversion dans un format texte spécifique
- `ASCII2Text` : conversion d'un texte ASCII dans un format texte spécifique
- `Text2XML` : conversion d'un format texte spécifique vers un XML valide

3.2.2. Au niveau des éléments

- `SentOrig2Elt` : découpage en éléments de la phrase originale (aux espaces)
- `ReplaceInElt` : remplace certains éléments par d'autres éléments selon une liste donnée (par exemple remplace « titi » par « toto »)
- `ReplaceElt` : remplace les éléments « ² » par « carrés », « °c » par « degrés celcius », « ° » par « degrés », « % » par « pour_cent », et les virgules des nombres

3.2.3. Au niveau des mots

- `Elt2Word` : segmente les éléments en mots selon un dictionnaire
- `Stick` : agglutine certains mots dans des expressions selon un dictionnaire
- `Lower` : mise en minuscule des lettres accentuées. Elle se fait par l'intermédiaire d'un dictionnaire (fichier double colonne) des correspondances majuscules/minuscules
- `Num2Letter` : conversion des nombres par leur équivalent textuel (comme par exemple « 2 » qui devient « deux »)
- `Remove` : supprime des éléments, mots, lemmes ou POS
- `PosTagger` : ajoute le POS des mots. Ce programme appelle LIA_TAG[1].

[1] http://www.lia.univ-avignon.fr/chercheurs/bechet/download_fred.html

3.2.4. Programmes gawk supplémentaires

- `XML2Sent` : transforme le fichier XML en un fichier de type une phrase par ligne, avec la possibilité de sélectionner les phrases selon différents critères
- `XML2WordCounts` : génère la liste des mots et leur nombre d'occurrences
- `Counts2Vocab` : à partir du fichier des occurrences, génère une liste de mots selon différents critères
- `Counts2Zipf` : à partir du fichier des occurrences, génère un fichier 4 colonnes : mot, nombre d'occurrences, rang, valeur de Zipf

3.2.5. Scripts gawk supplémentaires

- `HTML2XML` : script texte ou graphique (avec zenity) permettant de transformer un fichier HTML (local ou sur le web) en un fichier XML, en utilisant tous les modules des point 1, 2 et 3.
- `ASCII2XML` : idem à partir d'un fichier ASCII (uniquement fichier local)

3.3. Modules du logiciel vietnamien

Depuis le 17è siècle, la langue vietnamienne dispose d'un alphabet romanisé. Dans cette écriture latinisée, le *Quốc ngữ*, les mots figurent comme une succession de monosyllabes écrites séparément, comme dans la phrase suivante : *Trung tâm nghiên cứu quốc tế MICA được thành lập vào năm 2001*.

Compte tenu de l'architecture choisie, l'ouverture du logiciel à la langue vietnamienne n'a nécessité que peu de travail de développement. Le module de conversion des nombres `Num2LetterVN` a été entièrement ré-écrit. Egalement, le module de transformation des caractères spéciaux `ReplaceEltVN` a dû être ré-écrit ; il remplace les éléments « ² » par « bình phương », « °C » par « *độ C* », « ° » par "*độ*", « % » par « *phần trăm* », et les virgules des nombres par « *phẩy* »… De plus, le module `Text2XML` dispose désormais d'une option permettant la conversion des caractères en unicode.

L'ajout de la langue vietnamienne a également impliqué la création de fichiers de données utiles aux traitements, comme pour la langue française, c'est-à-dire une liste des mots du vocabulaire, et une liste des mots composés.

4. Alignements

L'alignement des corpus parallèles au niveau des mots trouve ses applications dans des tâches telles que la traduction automatique ou encore la construction de ressources lexicales bi- ou multi-lingues (Veronis, 2000). De nombreuses méthodes permettent de réaliser cet alignement, et des logiciels déjà très performants sont disponibles. Dans la mesure où nos recherches ne portent pas directement sur l'alignement, nous avons choisi de ne pas re-développer un outils, mais d'en adapter un à nos formats de données. Nous avons choisi d'utiliser `hunalign`[2], écrit en C++. Nous pouvons donc aligner, phrase à phrase, deux documents XML, l'un étant la traduction de l'autre. Dans la suite, nous montrons un extrait du

[2] http://mokk.bme.hu/resources/hunalign

résultat de l'alignement en français-vietnamien obtenu avec les deux versions du site internet du centre MICA[3].

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE corpus SYSTEM "corpus.dtd">
<corpus>
<doc>
<text>
<sent>
   <orig>
   Les partenaires :
   </orig>
   <elts id="FR">
          <wd>
                 <str> les </str>
                 <pos> MOTINC </pos>
                 <lem> les </lem>
          </wd>
          <wd>
                 <str> partenaires </str>
                 <pos> NMP </pos>
                 <lem> partenaire </lem>
          </wd>
   </elts>
   <elts id="VN">
          <elt> các </elt>
          <elt> đối </elt>
          <elt> tác </elt>
   </elts>
   <align idtgt="VN" idsrc="FR" score="1.5" />
</sent>
</text>
</doc>
</corpus>
```

5. Conclusion/Perspective

Dans cet article, nous avons proposé un schéma XML pour stocker des corpus. Nous avons présenté un logiciel complet de normalisation de corpus pour la langue française, et avons montré qu'il est très rapide de l'adapter à une nouvelle langue : le vietnamien. Nous avons également proposé un format pour l'alignement de données multi-lingues et une solution qui permet un alignement en phrases. Par la suite, un alignement en mots pourra également être intégré, en encapsulant par exemple le logiciel `moses`[4].

Dans un avenir proche, nous projetons l'ajout de nouvelles langues telles que le khmer qui nécessitera le développement d'un module spécifique de segmentation en mots. Nous projetons aussi de développer des modules pour importer et exporter des fichiers au format utilisé par les logiciels de traduction `GIZA++/moses` (Koehn, 2007), comme l'exemple décrit ci-après :

```
# Sentence pair (3) source length 2 target length 3 alignement score : 7.5676
au Vietnam
NULL ({}) ở ({ 1 }) Việt ({ 2 }) Nam ({ 2 })
```

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

[4] http://www.statmt.org/moses/

A plus long terme, nous pensons au développement d'outils graphiques pour l'interrogation, la visualisation, et la modification des corpus.

Références

Habert B., Fabre C., Issac F. (1998). *De l'écrit au numérique : constituer, normaliser, exploiter les corpus électroniques*. Paris, InterEditions, Masson, Informatiques.

Kilgarriff A., Grefenslette G. (2001). Web as Corpus. In *Proc. Corpus Linguistics*, Lancaster, UK.

Koehn P., Hoang H, Birch A., Callison-Burch C., Federico M., Bertoldi N., Cowan B., Shen W., Moran C., Zens R., Dyer C., Bojar O., Constantin A., Herbst E. (2007). Moses: Open Source Toolkit for Statistical Machine Translation. *Annual Meeting of the Association for Computational Linguistics* (ACL), demonstration session. Prague, Czech Republic.

Resnik P., Smith N.A. (2003). The Web as parallel corpus. *Computational Linguistics*, Vol. 29, Issue 3, Special issue on web as corpus, pages 349-380.

Véronis J. (2000). Alignement de corpus multilingue. In J.-M. Pierrel (ed.) *Ingénierie des langues*. Hermès, Paris, chap. 6, p. 151-171.