Integrating translation services within a structured editor

Ali Choumane
CLIPS-IMAG
&
INRIA Rhône-Alpes
ali.choumane@imag.fr

Hervé Blanchon
CLIPS-IMAG
385, rue de la Bibliothèque,
38041 Grenoble, France
herve.blanchon@imag.fr

Cécile Roisin
INRIA-RA & UPMF Grenoble
655 avenue de l'Europe,
38334 Montbonnot, France
cecile.roisin@inrialpes.fr

ABSTRACT

Fully automatic machine translation cannot produce high quality translation ; Dialog-Based Machine Translation (DBMT) is the only way to provide authors with a mean to translate source documents to languages she does not master or ever know. With such environment, the author must help the system to "understand" the document by means of an interactive disambiguation step. In this paper we study the consequences of integrating the DBMT services within a structured document editor (Amaya). The source document (named *edited document*) needs a *companion document* enriched with different data produced during the interactive translation process (question trees, answers of the author, translations). The *edited document* also needs to be enriched (annotated) in order to enable access to the question trees. The enriched *edited document* and the *companion document* have to be synchronized in case the *edited document* is further updated.

Categories and Subject Descriptors

I.2.7 [**Artificial Intelligence**]: Natural Language Processing - machine translation; I.7.2 [**Document and Text Processing**]: Document Preparation - format and notation, markup languages; H.5.2 [**Information Interfaces and Presentation**]: User Interface - graphical user interface

General Terms

Modeling, Experimentation, Design

Keywords

DBMT, interactive disambiguation, Self-Explaining Document, XML document, editing of structured documents

1. MOTIVATION

While most translation tools[1] provide an authoring environment to translators who usually have to translate a given source text to a target one (managing the two texts with alignment services for instance), we are interested in this paper in providing services for an author who wants to obtain automatic translations of her current work. This scheme has proven to be relevant if the authoring process is augmented with an interactive disambiguation step [4, 6].

In the framework of the LIDIA project [4] we investigate ways of increasing the automatic translation quality through interactive disambiguation dialogs for monolingual authors. Such an author will be able to translate her own source documents with no knowledge of the target language nor of the system itself. The quality of the target document will be high enough so that it will not need to be reviewed (the monolingual author is not able to carry out such a task). Whenever the DBMT system encounters an ambiguity it is not able to solve on its own, it prepares disambiguation questions to be asked to the author. For a unit of translation (sentence), if several ambiguities are encountered, the disambiguation questions are organized within a question tree. The information structure related to this process is called a *companion document* because it has to be associated with the source document.

The result of interactive disambiguation represents the meaning chosen by the author, it can be also useful for the readers. From that comes the idea of Self-Explaining Document (SED) [5] to enrich the *companion document* with the answers of the disambiguation.

First steps toward this goal [4, 5] have proven the benefits of the disambiguation approach for translation, but the system was not usable because it did not cover authoring needs and imposed a linear process from edition to translation. Our current goal is to enable access to the DBMT services through a real structured document editor. We claim that such an integration will benefit to the author in providing him with a simple environment. As we don't want to re-implement an authoring system from scratch, we look for an editor that provides a rich editing environment, with evolved XML editing services [7] and a WYSIWYG mode. Moreover, the editor must be expandable either directly at the source code level or through APIs. We have chosen to use Amaya [3] because it fits all these requirements for XHTML documents.

2. DESCRIPTION OF THE SERVICE

The workflow of our application is described in the Figure 1. The author starts with editing the text that she considers for

[1] Systran system [1] for instance

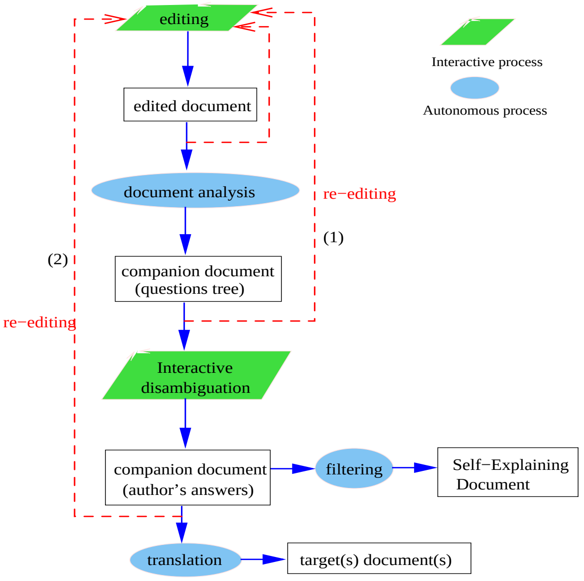


Figure 1: Functional diagram

translation. She can then request the system to launch the analysis phase, the result being a question tree attached to each ambiguous sentence. Through an interactive disambiguation phase, the author can disambiguate the document. Then, the author may request autonomous translation of the source document into document(s) in the target language(s) available. Besides translations the SED document can be produced by filtering the *companion document*. Note that re-editing is possible at any step of this process.

3. MAIN CHOICES

3.1 Documents modeling

The application must consistently handle both the source document (here the edited structure is XHTML) and the *companion document*. This is achieved by applying a transformation process from the *edited document* to the *companion document* and by enriching the *edited document* with disambiguation annotations (cf. Figure 2). The *companion document* (see Figure 3), defined through the XML schema "CDoc.xsd"[2], contains an element *paragraph* made of one or several *sentences*. The element *sentence* is the minimal unit on which the grammatical analysis is performed. One difficult point in the transformation process[3] from the *edited document* to the *companion document* was to produce the correct segmentation of *paragraph* into *sentences*.

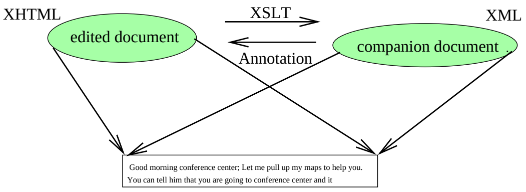
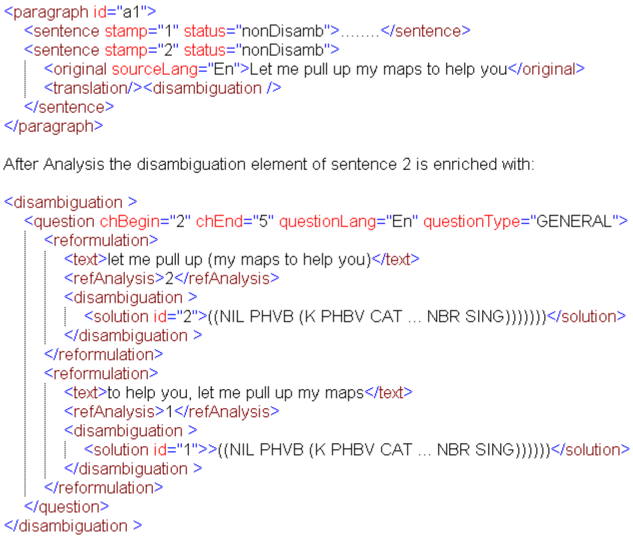

XHTML
edited document
XSLT
Annotation
XML
companion document
Good morning conference center; Let me pull up my maps to help you.
You can tell him that you are going to conference center and it

Figure 2: Sharing of document published between two different structures

A *sentence* contains the element *disambiguation*, possibly empty, depending on whether the sentence was analyzed or not. The element *disambiguation* consists of a set of questions about the sentence. More precisely, each question is about a part of the sentence (it can be the whole sentence itself), which is called the *ambiguity support*, and is identified by its begin and end characters indexes, stored as attributes *chBegin* and *chEnd* in the element *question*. The question is defined by at least two *reformulations*, associated with their corresponding analysis (grammatical solution). If the sentence is not ambiguous the *disambiguation* element only contains the analysis part (without questions). The *reformulation* can recursively lead towards one or several other *disambiguations*. Figure 3 shows the *companion document* before and after the analysis for the text : "Good morning conference center. Let me pull up my maps to help you.".



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

After Analysis the disambiguation element of sentence 2 is enriched with:

```
<disambiguation >
    <question chBegin="2" chEnd="5" questionLang="En" questionType="GENERAL">
        <reformulation>
            <text>let me pull up (my maps to help you)</text>
            <refAnalysis>2</refAnalysis>
            <disambiguation >
                <solution id="2">((NIL PHVB (K PHBV CAT ... NBR SING)))))))</solution>
            </disambiguation >
        </reformulation>
        <reformulation>
            <text>to help you, let me pull up my maps</text>
            <refAnalysis>1</refAnalysis>
            <disambiguation >
                <solution id="1">>((NIL PHVB (K PHBV CAT ... NBR SING))))))</solution>
            </disambiguation >
        </reformulation>
    </question>
</disambiguation >
```

Figure 3: Example of a *companion document*

3.2 Managing two XML structures

As shown in Figure 1, the author may incrementally update the source document while parts of it have already been disambiguated. This flexibility is an interesting feature for the author (she is not obliged to write the whole source document before translating it) but it requires a precise management of the XML structures. Indeed, inconsistencies may occur between the *edited* and *companion documents*. Thus they have somehow to be "synchronized". Any update in the *edited document* needs to be reflected into the *companion document*. Analysis and disambiguation steps have to be performed again, not on the whole document, but only on the updated sections. We have chosen to synchronize the documents at a "paragraph" level because it is the smallest element level in the *companion structure* that can be directly associated to an element of the *edited structure*. In our experiment based on XHTML source structures, we map the paragraph companion element to the p, h1, h2, h3, etc. XHTML elements.

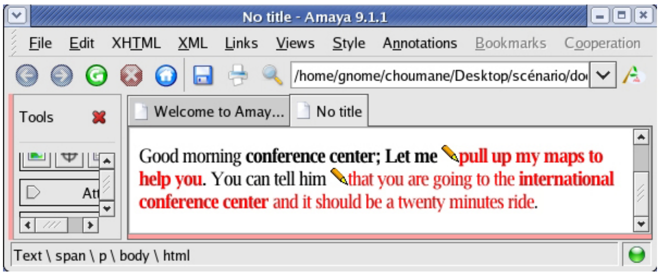
[2] http ://wam.inrialpes.fr/people/roisin/lidia/CDoc.xsd
[3] http ://wam.inrialpes.fr/people/roisin/lidia/TransToCDoc

3.3 Disambiguation interface

During the interactive disambiguation step, the system displays one or several dialog boxes with several rephrasings, of the original input. The author is requested to choose the right one [4]. Among all the available visualization techniques (graphical structures for tree, ...) we have chosen to experiment a mechanism based on structured annotations [2]. An ambiguity support can be marked with an annotation mark that shows the location of an ambiguity. Structured annotations cover our needs because they allow a recursive process where annotations can be annotated.

4. IMPLEMENTATION

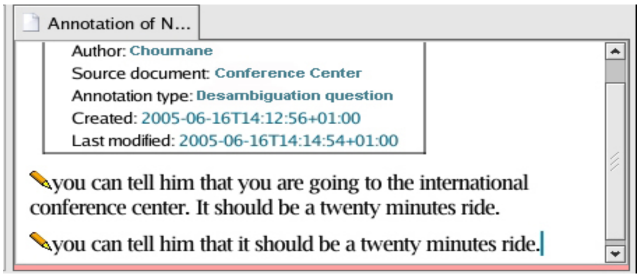
We present here an example scenario to show the current state of the application. We have extended the Amaya authoring tool so that it allows access to the disambiguation and translation services of LIDIA. After an editing phase, the author can establish a connection with the disambiguation server and request the analysis of the document. The analysis requires sentence units of text that are obtained from the *edited document* thanks to the XSLT transformation. The transformation result produces the *companion document* that is completed upon return from the analysis process. The author is informed that questions are pending by annotations added on the ambiguity support (see the annotations on Figure 4). The author can start the interactive disambiguation step by clicking the annotation. She is presented the corresponding disambiguation form (Figure 5). This form contains basic meta-data and the proposed reformulations of her text. If the chosen reformulation is ambiguous (and therefore annotated), the author can simply iterate as previously. Once the disambiguation is finished, the *edited document* has the form shown in the top part of the figure 6 where the new annotation mark indicates the new state of the text fragment. To visualize the SED, the author/reader passes the mouse over the new annotations mark and the chosen interpretations are displayed (SED rollover text shown in Figure 6).



Good morning conference center; Let me pull up my maps to help you. You can tell him that you are going to the international conference center and it should be a twenty minutes ride.

Text \ span \ p \ body \ html

Figure 4: The *edited document* after the analysis phase

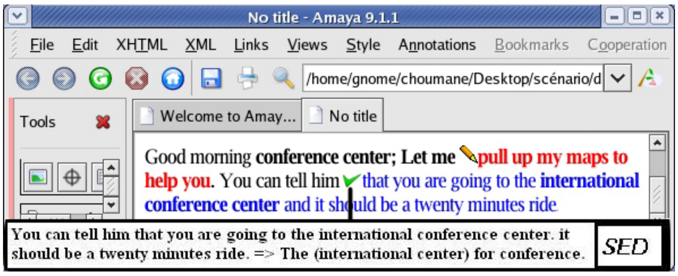


Author: Choumane
Source document: Conference Center
Annotation type: Desambiguation question
Created: 2005-06-16T14:12:56+01:00
Last modified: 2005-06-16T14:14:54+01:00

you can tell him that you are going to the international conference center. It should be a twenty minutes ride.

you can tell him that it should be a twenty minutes ride.

Figure 5: Disambiguation interface



Good morning conference center; Let me pull up my maps to help you. You can tell him that you are going to the international conference center and it should be a twenty minutes ride

You can tell him that you are going to the international conference center. it should be a twenty minutes ride. => The (international center) for conference.

Figure 6: The *edited document* after the interactive disambiguation

5. CONCLUSION

This work deals with DBMT and editing of structured documents. One of our objectives is to provide a flexible, comfortable authoring and disambiguation environment. The prototype we have realized is built upon Amaya, restricting the source document to be in XHTML format. However the *companion document* model and the access to the analysis, disambiguation and translation services are totally independent from the authoring tool, allowing their adaptation to other contexts.

We have shown in this paper how to provide the author with interaction and visualization for interactive disambiguation and SED services. Next step will be to integrate translation results in the environment and propose various accesses to the complex information recorded in the *companion* and *edited documents.*

6. ACKNOWLEDGMENTS

The authors would like to acknowledge Irène Vatton for her kind help and support in understanding the architecture of Amaya and the implementation of this application.

7. REFERENCES

[1] http ://www.systran.fr/index.html.

[2] http ://www.w3.org/2001/annotea/.

[3] http ://www.w3.org/amaya/.

[4] C. Boitet and H. Blanchon. Multilingual dialog-based machine translation for monolingual authors : the lidia project and a first mockup. *Machine Translation*, 9(2) :99–132, 1995.

[5] C. Boitet and H. Blanchon. Two steps towards self-explaining documents. *Proc. Convergence 03*, pages 032–324, December 2003.

[6] P. Langlais, G. Foster, and G. Lapalme. Unit completion for a computer-aided translation typing system. *Machine Translation*, 15(4) :267–294, 2000.

[7] V. Quint and I. Vatton. Techniques for authoring complex xml documents. *Document Engineering*, pages 115 – 123, October 2004.