A Visualizing Annotation Tool for Semi-Automatically Building a Bilingual Corpus

Quoc Hung-Ngo

Faculty of Computer Science
University of Information Technology
Vietnam National University – HoChiMinh City
hungnq@uit.edu.vn

Werner Winiwarter

University of Vienna
Research Group Data Analytics and Computing
Universitätsstraße 5, 1010 Vienna, Austria
werner.winiwarter@univie.ac.at

Abstract

Bilingual corpora are critical resources for machine translation research and development since parallel corpora contain translation equivalences of various granularities. Manual annotation of word alignments is of significance to provide a gold-standard for developing and evaluating both example-based machine translation models and statistical machine translation models. The annotation process costs a lot of time and effort, especially with a corpus of millions of words. This paper presents research on using visualization for an annotation tool to build an English-Vietnamese parallel corpus, which is constructed for a Vietnamese-English machine translation system. We describe the specification of collecting data for the corpus, linguistic tagging, bilingual annotation, and the tools specifically developed for the manual annotation. An English-Vietnamese bilingual corpus of over 800,000 sentence pairs and 10,000,000 English words as well as Vietnamese words has been collected and aligned at the sentence level; and a part of this corpus containing 200 news articles was aligned manually at the word level.

Keywords: annotation tool, bilingual corpus, word alignment

1. Introduction

In natural language processing, a bilingual corpus is a valuable resource. A huge bilingual corpus is not only used to train natural language processing (NLP) tasks effectively but also to evaluate NLP systems objectively, such as chunking in bilingual text, bilingual comparison, bitext transfer, and machine translation.

In building corpora, developing tools is also as important as collecting data, aligning, and tagging linguistic information. If the corpus is built semi-automatically, it means it is tagged or corrected by annotators and by using annotation tools. Therefore, the visualization ability of an annotation tool helps annotators to review and correct the linguistic information as well as the whole document in the corpus. For this purpose, several tools have been researched and developed, such as the Yawat tool of Ulrich Germann (2008), the Cairo tool of Smith and co-authors (2000), annotation tools for parallel treebanks by Yvonne S. and Martin V. (2007), or tools for a Japanese-Chinese parallel corpus by Yujie Zhang and co-authors (2008).

For the English-Vietnamese language pair, there exist several projects for building an English-Vietnamese corpus for special purposes, such as building a bilingual corpus for word sense disambiguation by Dinh Dien(2002), and building a bilingual corpus through web

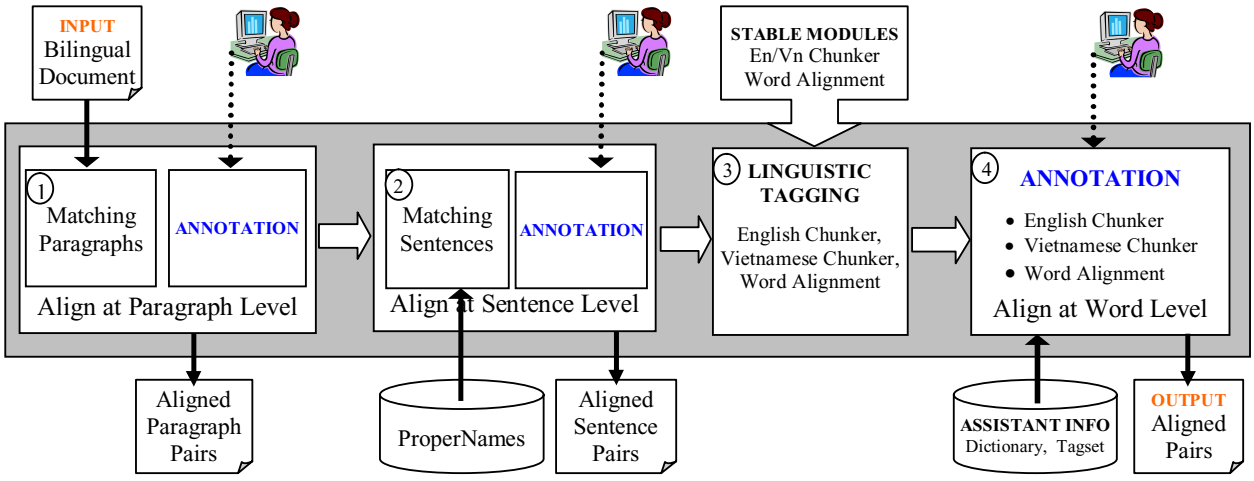

Figure 1: Overview of Building Bilingual Corpus Process

mining by Van D. B. and Bao Quoc H. (2007). However, most of these corpora are not available for download or just at the aligned sentence level.

In this paper, we describe the design of an annotation tool for building an English-Vietnamese Bilingual Corpus (EVBCorpus). More specifically, the goal is to build and annotate a large bilingual corpus which is tagged with linguistic information, such as part-of-speech, chunks, bitext alignment at the word level, and more. This bilingual corpus can then be used for the automatic training of machine translation systems.

In this work, we use three main stages. Firstly, we collect the data from the Internet and classify it based on the type of text as well as categories. Collected data is also normalized to reduce errors and to create a unique format between two languages. Secondly, we use NLP toolkits to tag linguistic information. Finally, a tool for annotation is built to annotate and correct linguistic tags, which have been assigned before.

Figure 1 shows the process of bilingual corpus building, including three main modules: pre-processing, linguistic tagging, and bilingual annotation. In particular, the pre-processing steps include (1) matching paragraphs and (2) matching sentences. These steps also need annotation to ensure that the result of these steps are English-Vietnamese sentence pairs. These bilingual pairs are tagged linguistically by the tagging modules (3), including English chunking, Vietnamese chunking, and English-Vietnamese word alignment. The aligned source and target chunks can be corrected as chunking result, alignment result as well as Vietnamese word segmentation result at the bilingual annotation stage (4). The Vietnamese word segmentation result can be corrected at this stage because the Vietnamese chunking module includes a word segmentation module.

2. Data

The EVBCorpus consists of both original English text and its Vietnamese translations, and original Vietnamese text and its English translations. The original data is from books, fictions or short stories, law documents, and newspaper articles. The original articles were translated by skilled translators or by contribution authors, and were checked again by skilled translators. Parallel documents are also chosen and classified into categories, such as economy, entertainment, health, science, social and politics, and technology.

Each article was translated one to one at the whole article level, so we first need to align paragraph to paragraph and then sentence to sentence. At the paragraph stage, aligning is simply moving the sentences up or down and detecting the separator position between paragraphs for both articles. At the sentence stage, however, aligning is more complex and depends on the translated articles which are translated by one-by-one method or a literal meaning-based method. In many cases (as common in literature text), several sentences are merged into one sentence to create the one-by-one alignment of sentences. The details of the corpus are listed in Table 1.

Source	Document	Paragraph	Sentence	Word
En-Vn Books	15	13,980	80,323	1,375,492
En-Vn Fictions	100	192,723	590,520	6,403,511
En-Vn Laws	250	86,803	98,102	1,912,055
En-Vn News	1,000	24,523	45,531	740,534
Total	**1,365**	**318,029**	**814,476**	**10,431,592**

Table 1: Details of data sources of EVBCorpus

An important feature of the corpus is that it has been pre-processed at the basic linguistic level, namely that of words. Especially, in Vietnamese, tokens are not words, and a word can be a token or a group of tokens. Therefore, the first important step in pre-processing is a Vietnamese word segmentation which is just done to evaluate the corpus, whereas this step used for later processing is included in the Vietnamese chunking module. In our project, we use vnTokenizer of Le H. Phuong et al (2008) to segment words in Vietnamese text.

There are 10,431,592 English words and 10,298,531 Vietnamese words (containing 13,143,290 Vietnamese tokens) in our bilingual corpus (see Table 2). Vietnamese words are counted based on the result of using the vnTokenizer module on the Vietnamese text.

Based on the results shown in Table 2, it can be seen that the length of most sentences in the corpus is from 10 to 25 words, and books are the bitext type with the longest average sentences. An interesting characteristic is that there are over 4% quite long sentences which have more than 50 words per sentence, even one hundred words in several cases. Moreover, the average paragraph length is just under 5 sentences per paragraph. Books also have the

Sentence Length	~10	~20	~30	~40	~50	~60	~70	~80	~90	~100	~110	~120
En-Vn Books	9,719	14,265	10,772	5,990	3,058	1,398	657	294	183	92	54	28
En-Vn Fictions	248,699	157,588	63,117	22,587	7,828	2,608	976	400	161	86	52	34
En-Vn Laws	38,071	17,789	12,513	7,776	4,360	2,154	1,073	545	266	139	83	67
En-Vn News	9,065	12,660	7,168	2,360	686	184	34	20	9	6	3	2

Table 2: Number of English sentences for each length

highest number of sentences. We carry out these statistics to look for a sensible way of building an annotation tool at a later stage.

3. Design of Annotation Tool

To add the linguistic information to the corpus and reduce the amount of effort for annotating, we integrate the NLP modules into the annotation tool. For linguistic tagging, we tag chunks for both English and Vietnamese text. English-Vietnamese sentence pairs are also aligned word-by-word to create the connections between the two languages. The data of the corpus is stored in the HTML and SGML standard.

3.1. Standard for Data Storage

We use both the HTML and SGML standard to store and process the data. For visualization, our tool stores files of the bilingual corpus based on the HTML format (see following example). Web browsers can open and render the representation of the corpus file easily with this format. It is also easy to store and review pairs in the corpus as parallel text (see Figure 5 in Sect. 3.4). In the HTML source, tag *span* is used to define POS tags, tag *sub* is used to define chunks, and tag *sub* with class *sentence* is used to define S tags (for whole sentences).

Besides HTML format, our tool also supports to store and export the corpus files to the SGML format based on Ide's guidelines (Ide N., 1998). Moreover, as another phrase corpus, English-Vietnamese bilingual corpus files are stored in column format by our annotation tool.

An example of the visualization of the chunk result and its HTML source is shown in Figure 2.

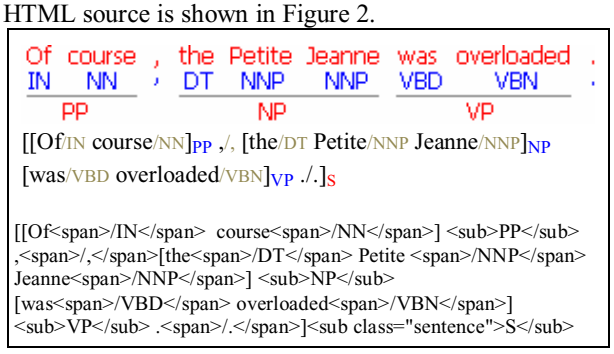

Figure 2: An example of chunking result and its HTML source

For the SGML format, the entire sentence is bracketed by tag *sentence*. Phrase structures are represented with tag *chunk*. The attribute *cat* represents the phrase symbol of a phrase. For example, the noun phrase "the Petite Jeanne" is represented as "<chunk cat="NP">the Petite Jeanne</chunk>". The next element is tag *wrd*, which is used to present words. The attribute *pos* represents the part-of-speech of a word. This is also similar to tokens in English text, however, it can be a group of tokens in Vietnamese text. The smallest element tag is *tok*. Each word in English and token in Vietnamese text is bracketed by *tok* tag.

```
<sentence id="s0"><chunk id="c0" cat="PP">
<wrd id="w0" pos="IN"><tok id="t0">Of</tok></wrd>
<wrd id="w1" pos="NN"><tok id="t1">course</tok></wrd>
</chunk><tok id="t2">,</tok><chunk id="c1" cat="NP">
<wrd id="w2" pos="DT"><tok id="t3">the</tok></wrd>
<wrd id="w3" pos="NNP"><tok id="t4">Petite</tok></wrd>
<wrd id="w4" pos="NNP"><tok id="t5">Jeanne</tok></wrd>
</chunk> <chunk id="c2" cat="VP">
<wrd id="w5" pos="VBD"><tok id="t6">was</tok></wrd>
<wrd id="w6" pos="VBN"><tok id="t7">overloaded</tok></wrd>
</chunk><tok id="t8">.</tok></sentence>
```

The encoding indicates that the translation text and its chunk tagging result is "[[Tất_nhiên/Np]$_{PP}$ [chiếc/Nc Petite_Jeanne/Np]$_{NP}$ [đã/R chở/V]$_{VP}$ [quá/T nặng/A]$_{AP}$ ./.]$_{S}$". The word alignment result in HTML format is "[1,2-1,2];[4-3];[5,6-4,5];[7,8-6,7,8,9]". It is stored in the SGML format as:

```
<links id=ls0 Xtarget="c0:c0">
   <linkw id=lw0 type=n:n Xtarget="t0,t1:t0,t1"></linkw>
   <linkw id=lw1 type=1:1 Xtarget="t3:t2"></linkw>
   <linkw id=lw2 type=n:n Xtarget="t4,t5:t3,t4"></linkw>
   <linkw id=lw3 type=n:n Xtarget="t6,t7:t5,t6,t7,t8"></linkw>
</links>
```

3.2. Linguistic Tagging

3.2.1 Chunking for English

There are several available chunking systems for English text, however, we focus on parser modules to build an aligned bilingual treebank in future. Based on Rimell's evaluation of five state-of-the-art parsers (Rimell, 2009), the Stanford parser is not the parser with the highest score. However, the Stanford parser supports both parse trees in bracket format and dependencies representation (Dan Klein et al, 2003; Marie-Catherine de Marneffe et al, 2006). We chose the Stanford parser not only for this reason but also because it is updated frequently, and to provide for the ability of our corpus for semantic tagging in future.

In our project, the full parse result of an English sentence is considered to extract phrases as chunking result for the corpus. For example, for the English sentence "*Products permitted for import, export through Vietnam's border-gates or across Vietnam's borders.*", the Stanford parser result is:

```
(S (NP (NNPS Products))
   (VP (VBD permitted)
     (PP (IN for)
        (NP (NP (NN import))
            (, ,)
            (NP (NN export))))
     (PP (PP (IN through)
         (NP (NP (NNP Vietnam) (POS 's))
             (NNS border-gates)))
         (CC or)
         (PP (IN across)
            (NP (NP (NNP Vietnam) (POS 's))
                (NNS borders)))))
     (. .))
```

Extracting chunks based on the Stanford parser result concentrates on noun and verb phrases rather than preposition phrases. The result of the extraction procedure for the example sentence is:

[Products]$_{NP}$ [permitted]$_{VP}$ [for]$_{PP}$ [import]$_{NP}$, [export]$_{NP}$ [through]$_{PP}$ [Vietnam's border-gates]$_{NP}$ [or]$_{PP}$ [across]$_{PP}$ [Vietnam's borders]$_{NP}$.

3.2.2. Chunking for Vietnamese

There are several chunking systems for Vietnamese text, such as noun phrase chunking by Le M. Nguyen et al (2008) or by Nguyen H. T. et al (2009). In our system, we use the full phrase chunker of Le M. Nguyen and Cao T. H. (2009) to chunk Vietnamese sentences. This is module SP8.4 in the VLSP project[1].

The VLSP project is a KC01.01/06-10 national project named Building Basic Resources and Tools for Vietnamese Language and Speech Processing. This project involves active research groups from universities and institutes in Vietnam and Japan, and focuses on building a corpus and toolkit for Vietnamese language processing, including word segmentation, part-of-speech tagger, chunker, and parser.

For example, the chunking result for the sentence "*Các sản phẩm được phép xuất khẩu, nhập khẩu qua cửa khẩu, biên giới Việt Nam.*" is "[Các sản_phẩm]$_{VP}$ [được]$_{VP}$ [phép]$_{NP}$ [xuất_khẩu]$_{VP}$, [nhập_khẩu qua]$_{VP}$ [cửa_khẩu]$_{NP}$, [biên_giới Việt_Nam]$_{NP}$.".

(In English: "[Products]$_{NP}$ [permitted]$_{VP}$ [for]$_{PP}$ [import]$_{NP}$, [export]$_{NP}$ [through]$_{PP}$ [Vietnam's border-gates]$_{NP}$ [or]$_{PP}$ [across]$_{PP}$ [Vietnam's borders]$_{NP}$.")

The chunking result also includes the word segmentation and the part-of-speech tagger result. These results are based on the result of word segmentation by Le H. Phuong, N. T. M. Huyen et al (2008). The tagset of chunking includes 5 tags: NP, VP, ADJP, ADVP, and PP.

3.2.3. Word Alignment in Bilingual Corpus

In a bilingual corpus, word alignment is very important because it demonstrates the connection between two languages. In our corpus, we apply a class-based word alignment approach to align words in the English-Vietnamese pairs. Our approach is based on the result of D. Dien et al (2002), to which we also contributed. This approach originates from the English-Chinese word alignment approach of Ker and Chang (1997). The class-based word alignment approach uses two layers to align words in a bilingual pair, dictionary-based alignment and semantic class-based alignment. The dictionary used for the dictionary-based stage is a general machine-readable bilingual dictionary while the dictionary used for the class-based stage is the Longman Lexicon of Contemporary English (LLOCE) dictionary, which is a type of semantic class dictionary.

Aligning words with a bilingual dictionary is estimating the distance *DTSim(s, t)* by using the meaning sets in the bilingual dictionary (*s* is a word in the source sentence and *t* is a token/word in the target sentence). Based on the collection of dictionary-based alignments, the model calculates the acquisition of pairs of mutually translatable classes (X, Y). Finally, aligning words based on classes is estimating the probability values *Pr(s,t)* based on the conceptual similarity *ClassSim(X, Y)* (*s* is a member of class *X* and *t* is a member of class *Y*) and the distortion probability *dis(i, j)* (*i* is the position of *s* in the source sentence and *j* is the position of *t* in the target sentence) (Dien Dinh et al, 2002; Ker et al, 1997). The result of the word alignment is indexed based on token positions in both sentences. For example:

English: I had rarely seen him so animated .
Vietnamese: Ít khi tôi thấy hắn sôi nổi như thế .

The word alignment result is [1-3], [3-1,2], [4-4], [5-5], [6-8,9], [7-6,7], [8-10] (visualized in Figure 3).

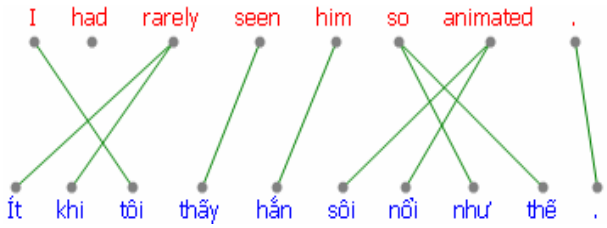

Figure 3: An example of word alignment in bilingual corpus

3.3. Word Alignment Visualization

Because of the huge value of bilingual corpora, numerous tools for the visualization and creation of word alignments have been developed. Most of them employ one of two visualization techniques. The first is to draw lines between associated words (as shown in Figure 3). The second is to use an alignment matrix (as shown in Figure 4), where the rows of the matrix correspond to the words of the sentence in one language and the columns to the words of that sentence's translation into the other language. Marks in the matrix's cells indicate whether the words represented by the row and column of the cell are linked or not.

Basically, with both visualization techniques it is easy to get an overview of the alignments at the word level, however, the drawing line technique has several advantages. For this technique, it is easy to combine the results of chunker modules and the parse trees for both

[1] http://vlsp.vietlp.org:8080/demo/

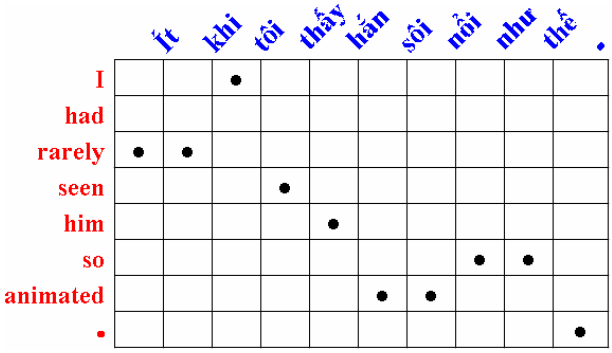

Figure 4: Visualization of word alignments with an alignment matrix

sentences (see Figure 6 in Sect. 3.4.3). It is also less space-consuming in case of lengthy sentence pairs. Because of these advantages, we use this technique in our annotation tool to demonstrate the word alignments of the English-Vietnamese sentence pairs.

3.4. Bilingual Annotation Process

As shown in Figure 1, there are three annotation stages in whole process, including matching paragraphs, matching sentences, and aligning words.

3.4.1. Matching Paragraphs and Sentences

In our system, before annotating for paragraph alignment, we use the Edit Distance algorithm to match sentences and split them into paragraphs by using the endline symbols of paragraphs in source document or target document. The string edit distance algorithm is sometimes known as Levenshtein distance. A very comprehensive and accessible explanation of the Levenshtein algorithm is available on the web at http://www.merriampark.com/ld.htm. The Levenshtein algorithm measures the edit distance where edit distance is defined as the number of insertions, deletions, or substitutions required to make the two strings match. A score of zero represents a perfect match. This algorithm has been applied to match names in English and Arabic by Freeman and co-authors (2006).

For matching paragraphs in both documents, it is essentially the matching of the sequence of sentences in these documents. This process is implemented by matching two strings where each sentence is represented by an element in the string. In our system, these elements are featured by merging a number of proper names and several special signs (such as question marks, exclamation marks, quotation marks, and so on).

With two strings, string s of size m and string t of size n, the algorithm has O(nm) time and space complexity. A matrix is constructed with n rows and m columns. The function $e(s_i,t_j)$ where s_i is a character in the string s, and t_j is a character in string t returns the value 0 if the two characters are equal and the value 1 otherwise. The algorithm extracts matched sub-sequences in both strings and then inserts zero values into the two strings so that they have equal length.

For example, string s is 003100210, representing the source document encoded with 9 sentences and sentence 3, 4, 7, and 8 having 3, 1, 2, and 1 proper names. Similarly, string t is 0030102100, representing 10 sentences in the target document with sentence 3, 5, 7, and 8 having 3, 1, 2, and 1 proper names. Our algorithm based on the Edit Distance algorithm tries to insert the value 0 into both strings and match characters as much as possible. The result in this example is 00301002100 with the length of 11 sentences. This result is decoded with two blank sentences which are inserted into s after sentence 3 and sentence 9.

3.4.2. Annotation for Sentence Alignment

The first stage of building a bilingual corpus is a bitext alignment, which aligns paragraph by paragraph and then sentence by sentence. Firstly, documents are manually segmented into chapters. These chapters are segmented into paragraphs by endline symbols. Basically, paragraphs in both languages are ordered as a sequence and there is rarely a change in order among paragraphs between a document pair. However, the merging and splitting of paragraphs occurs more frequently. In the next stage, paragraphs and sentences in two parallel documents are automatically aligned by the Levenshtein Edit Distance algorithm based on the number of proper names in each sentence. Finally, automatically aligned paragraph pairs are reviewed and corrected by annotators by using our tool.

For visualization, our tool simply shows paragraph pairs in each row (see Figure 5). Therefore, if the alignment of the previous pair is incorrect, the following pairs are incorrect, too. In addition, paragraph pairs with incorrect alignment have usually differences in paragraph length. In contrast, paragraph pairs with correct alignment are quite similar. Therefore, while scrolling through chapters and documents, annotators can identify the differences quickly and concentrate on correcting them. Our tool also supports to drag and drop paragraph items on paragraphs in order to merge paragraphs and to cut a paragraph into smaller paragraphs at the end of a particular position by pressing a hotkey.

3.4.3. Annotation for Word Alignment

Based on the results of the English chunking module, the Vietnamese chunking module, and the word alignment module in step 3 of the process (see Figure 1 with an explanation in the Section 3.2) , the parallel sentence pairs are linked together at the chunk level (see Figure 6).

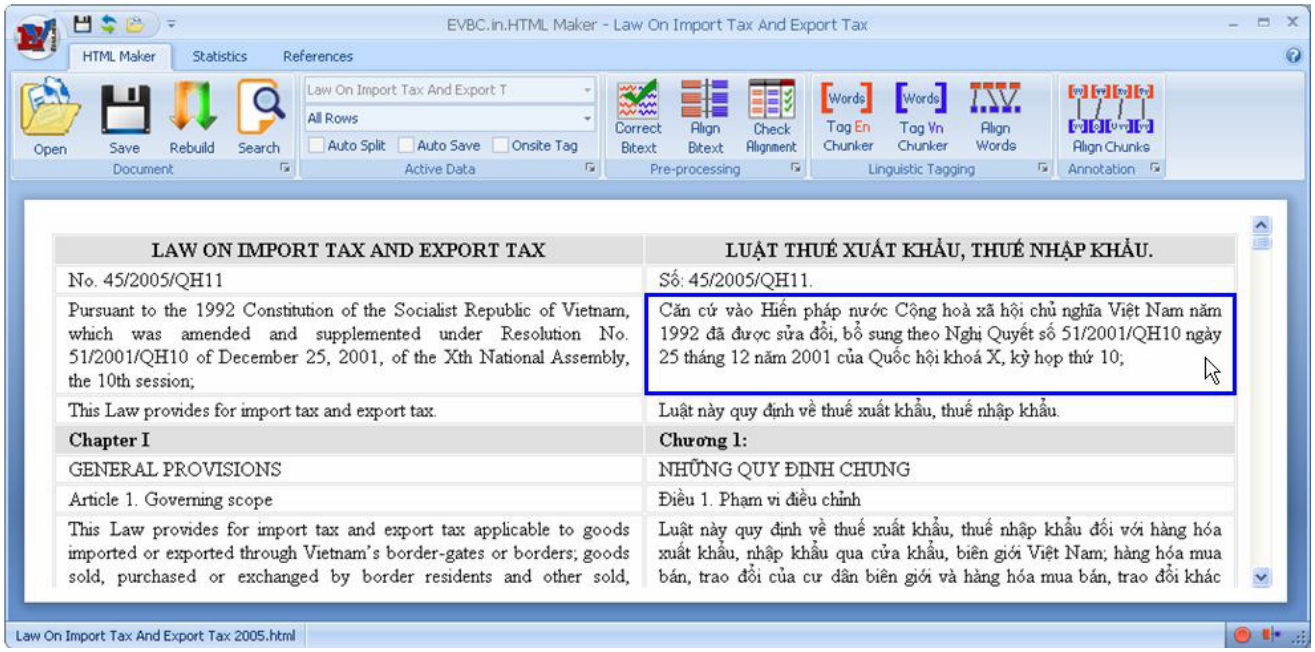

Figure 5: Drag and droppable interface of the tool for manual paragraph alignment annotation

With the visualization provided by our tool, annotators review whole phrase structures of English and Vietnamese sentences. They can compare the English chunking result with the Vietnamese result and correct them in both sentences.

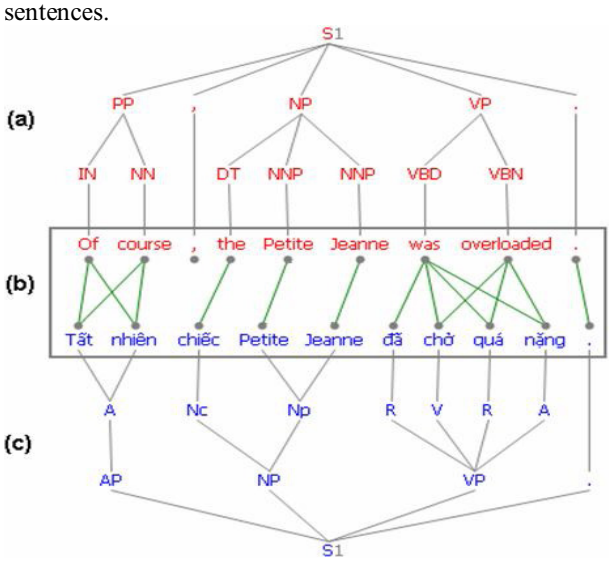

Figure 6: Combine English chunking (a), Vietnamese chunking(c), and word alignment (b)

Moreover, mistakes regarding word segmentation for Vietnamese, POS tagging for English and Vietnamese, and English-Vietnamese word alignment can be detected and corrected by drag, drop, and edit label operations (actions) of our tool. Based on drag and drop on labels and tags, annotators can change the results of the tagging modules visually, quickly, and effectively.

Different from paragraph alignment, which is based on chapter or document level, the word and chunk alignment is based on paragraph level with 2 to less than 5 sentences for each paragraph on average (as shown in Table 2). With the linguistic information including word/token, POS tag, chunking tag and word alignment, each sentence pair can be presented in one screen page. For long and complex sentences, annotators can scroll the horizontal scrollbar to view and correct the hidden part.

3.5. Details of Annotation Tool

In general, annotators have a good knowledge of linguistics, however, they have limitations in understanding formats for NLP corpora, which are normally used to process on computers. Moreover, for building a valuable corpus, the amount of annotation is very huge. Therefore, our goal is to develop a tool for annotating a corpus visually, quickly, and effectively at the alignment level of sentences, words, and chunks.

Drag and drop actions are mainly a convenient feature of the annotation tool. It allows annotators to drag a node (a word), a part of tree (a phrase), or multi-selected parts, and drop the item(s) on another node of the other tree to create alignments. For convenience purposes in annotating lengthy sentences, our tool also supports to grip the whole view and move it horizontally or vertically instead of clicking on the scrollbars. The parse trees can be expanded or collapsed to see the full details of sentences, or just an overview, or a part of long sentence pairs. Aside from mouse control, hotkeys are set up for the annotation tool. These hotkeys help annotators to navigate among

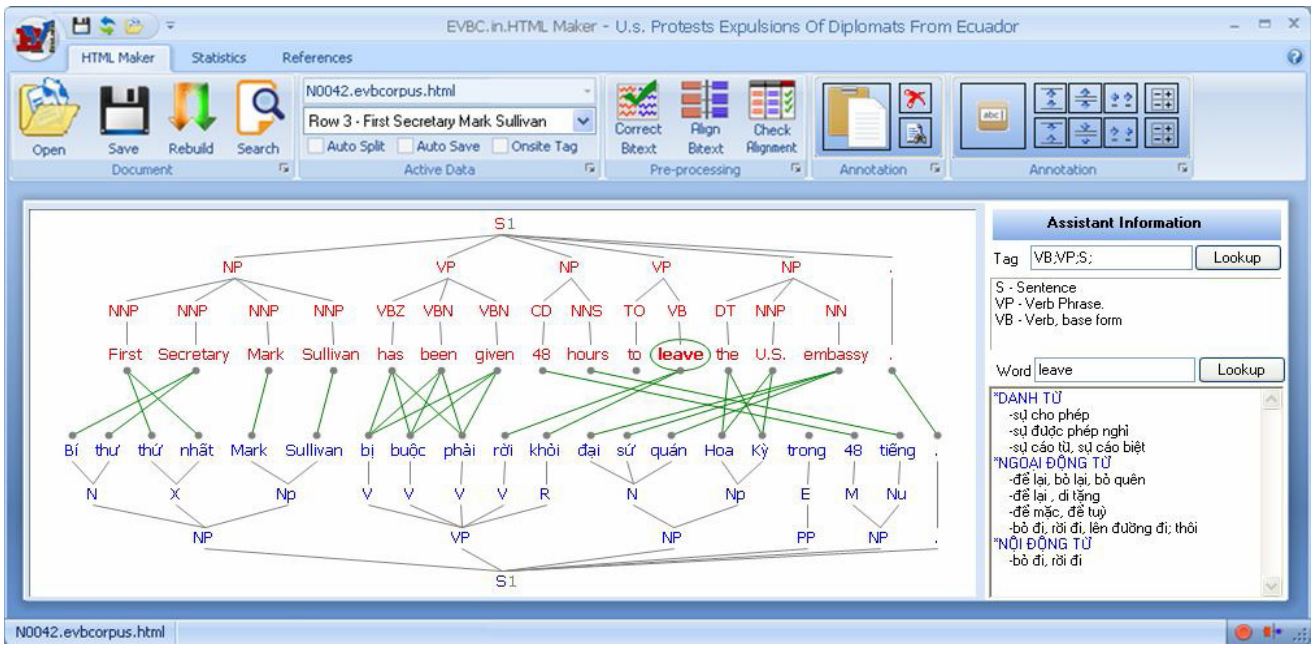

Figure 7: Overview of annotation tool for manual word/chunk alignment annotation

pairs, or to make/remove alignments.

Moreover, linguistic assistant information is shown following the annotator's actions. This assistant system accesses dictionaries to look up and show the meaning of the current word at the cursor (see Figure 7). Our annotation tool also supports both sure alignments and possible alignments which are two types of alignments.

4. Results and Analysis

4.1. Bilingual Corpus

From four resources, we built an English-Vietnamese bilingual corpus with over 800,000 sentence pairs and 10,000,000 words. This corpus is tagged with chunker labels for both English and Vietnamese, and aligned at word level. We also developed an annotation toolkit by integrating NLP modules for tagging, and a drag and droppable interface module for annotating. Our overall process illuminates four main steps of building a parallel corpus: (1) collect data and align bitext at the paragraph level; (2) align bitext at the sentence level, (3) linguistic analysis and tagging; (4) annotate and correct corpus with toolkits.

As a main result of the project, we built an English-Vietnamese bilingual corpus with 1,217 documents, over eight hundred sentences, and over ten million words from four resources: books, literal novels, law documents, and news articles. As mentioned in Section 2.1, all of these documents are collected and aligned as chapter-to-chapter (for books, novels, and laws), or article-to-article (for news articles) at first. Next, they are semi-automatically separated to align at the paragraph level, and at the sentence level at last. However, we still keep the context of paragraphs and sentences, which is very useful for other tasks in several machine translation models, such as document classification before translating or detecting the context of words in documents. A part of this corpus and the annotation tool are published at http://code.google.com/p/evbcorpus/ .

4.2. Annotation Process

The annotation process costs a lot of time and effort, especially with a corpus of over 10 million words for each language. In our evaluation, we annotated 200 news articles with 6,723 sentence pairs, and 116,246 English words (125,762 Vietnamese words and 164,447 Vietnamese tokens), as shown in Table 3.

	English	Vietnamese
Files	200	200
Sentences	6,723	6,723
Words	116,246	125,762
Tokens	116,246	164,447
Sure Alignments	70,238	70,238
Possible Alignments	88,964	88,964
Words in Alignments	90,581	121,271
Tokens in Alignments	90,581	151,905

Table 3: Details of Aligned EVBCorpus at word level

In this evaluation, the data is tagged and aligned automatically at the word level between English and Vietnamese and we just focus on the set of alignments and amount of annotation rather than evaluate the quality of the linguistic tagging modules. The number of alignments in 200 news articles is 89,222 alignments, which are aligned automatically by the word alignment module (as mentioned in Section 2.3.2) and checked and linked manually by annotators.

Alignments are annotated with both sure alignments *S* and possible alignments *P*, with $S \subseteq P$. These two types of alignment are annotated to evaluate the alignment models by the Alignment Error Rates (AER) according to the specifications described by Och and Ney (2003). In 200 annotated news articles, there are 70,238 sure alignments, accounting for 78% of possible alignments (as shown in Table 3). These alignments mainly come from nouns, verbs, adverbs, and adjectives which are meaningful words in sentences. On the other hand, the 22% remaining possible alignments are mainly from prepositions in both English words and Vietnamese words.

5. Conclusion

In this paper we introduced a design of a visualizing method for word alignment annotation and a complete workflow to build an English-Vietnamese bilingual corpus: from collecting data, tagging chunks, aligning words in bilingual text, and developing an annotation tool for bilingual corpora. We showed that the size of our corpus with 200 English-Vietnamese aligned news article pairs at the word level is a valuable contribution to build a high quality corpus in the future. We pointed out that linguistic information tagging based on our procedure, including tagging and annotation, so far, stops at the chunk level.

6. References

Dan Klein and Christopher D. Manning (2003). Accurate Unlexicalized Parsing. Proceedings of the 41st Meeting of the Association for Computational Linguistics, pp. 423-430.

Dien Dinh, (2002). Building a training corpus for word sense disambiguation in the English-to-Vietnamese Machine Translation. In Proceedings of Workshop on Machine Translation in Asia, pp. 26-32.

Dien Dinh, Kiem H., Ngan N. L. T., Quang X., Toan N. V., Quoc Hung N., and Hoi P. P. (2002). Word alignment in English – Vietnamese bilingual corpus. Proceedings of EALPIIT'02, HaNoi, Vietnam, pp. 3-11.

Freeman, Andrew T., Sherri L. Condon and Christopher M. Ackerman (2006). Cross Linguistic Name Matching in English and Arabic: A "One to Many Mapping" Extension of the Levenshtein Edit Distance Algorithm, HLT-NAACL, New York, NY.

Ide, N. (1998). Corpus Encoding Standard: SGML Guidelines for Encoding Linguistic Corpora. Proceedings of the First International Language Resources and Evaluation Conference, Granada, Spain, pp. 463 – 470.

Ker, Sue J. and Jason S. Chang (1997). A class-based approach to word alignment. Computational Linguistics, 23(2):313–343.

Germann, Ulrich (2008). Yawat: Yet Another Word Alignment Tool. Proceedings of the ACL-HLT 2008.

Le M. Nguyen and Cao, T. H. (2008), Constructing a Vietnamese Chunking System. Proceedings of the 4th National Symposium on Research, Development and Application of Information and Communication Technology, Science and Technics Publishing House, pp. 249-257.

Le M. Nguyen, Huong T. Nguyen, Phuong Thai Nguyen, Tu Bao Ho and Akira Shimaz (2009). An Empirical Study of Vietnamese Noun Phrase Chunking with Discriminative Sequence Models. The 7th Workshop on Asian Language Resources (in conjunction with ACL-IJCNLP).

Le H. Phuong, N. T. M. Huyen, R. Azim, H. T. Vinh (2008). A hybrid approach to word segmentation of Vietnamese texts. Proceedings of the 2nd International Conference on Language and Automata Theory and Applications, LATA 2008, Springer LNCS 5196, Tarragona, Spain.

Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning (2006). Generating Typed Dependency Parses from Phrase Structure Parses. LREC 2006.

Nguyen Huong Thao, Nguyen Phuong Thai, Nguyen Le Minh, and Ha Quang Thuy (2009). Vietnamese Noun Phrase Chunking based on Conditional Random Fields. Proceedings of the First International Conference on Knowledge and Systems Engineering (KSE 2009).

Rimell, L., Clark S., and Steedman M. (2009). Un-bounded dependency recovery for parser evaluation. Proceedings EMNLP, pp. 813–821.

Smith, Noah A. and Michael E. Jahr (2000). Cairo: An alignment visualization tool. Second International Conference on Linguistic Resources and Evaluation (LREC-2000).

Van B. Dang, Bao Quoc Ho (2007). Automatic Construction of English-Vietnamese Parallel Corpus through Web Mining. Research, Innovation and Vision for the Future (RIVF), IEEE International Conference. pp. 261-266.

Yujie Zhang, Zhulong Wang, Kiyotaka Uchimoto, Qing Ma, Hitoshi Isahara (2008). Word Alignment Annotation in a Japanese-Chinese Parallel Corpus. LREC 2008.

Yvonne Samuelsson and Martin Volk (2007). Alignment tools for parallel treebanks. Proceedings of GLDV Frühjahrstagung 2007.