Building an Annotated English-Vietnamese Parallel Corpus for Training Vietnamese-related NLPs

Dinh Dien * and Hoang Kiem **
* Faculty of Information Technology, University of Natural Sciences, VNU-HCMC
Tel : +84-8-8497-181 E-mail: ddien@saigonnet.vn
**Center of Information Technology Development, Vietnam National University of HCMC
Tel : +84-8-9317-442 E-mail: hkiem@citd.edu.vn

Abstract:

In NLP (Natural Language Processing) tasks, the highest difficulty which computers had to face with, is the built-in ambiguity of Natural Languages. To disambiguate it, formerly, they based on human-devised rules. Building such a complete rule-set is time-consuming and labor-intensive task whilst it doesn't cover all the cases. Besides, when the scale of system increases, it is very difficult to control that rule-set. So, recently, many NLP tasks have changed from rule-based approaches into corpus-based approaches with large annotated corpora. Corpus-based NLP tasks for such popular languages as English, French, etc. have been well studied with satisfactory achievements. In contrast, corpus-based NLP tasks for Vietnamese are at a deadlock due to absence of annotated training data. Furthermore, hand-annotation of even reasonably well-determined features such as part-of-speech (POS) tags has proved to be labor intensive and costly. In this paper, we present our building an annotated English-Vietnamese parallel aligned corpus named EVC to train for Vietnamese-related NLP tasks such as Word Segmentation, POS-tagger, Word Order transfer, Word Sense Disambiguation, English-to-Vietnamese Machine Translation, etc.

Keywords: Corpora, bilingual corpus, corpus annotation, machine learning, machine translations.

1. INTRODUCTION

Nowadays more and more people are interested in extracting information about language from very large annotated corpora. Such annotated corpora have been built for popular languages (e.g. Penn TreeBank for English, French, Japanese, etc.) and these corpora have been used to effectively serve such well-known NLP tasks as POS-Tagger, Phrase Chunker, Parser, Structural Transfer, Word Sense Disambiguation (WSD), etc. Unfortunately, so far, there has been no such annotated corpora available for Vietnamese NLP tasks. Furthermore, building manually annotated corpora is very expensive (e.g.: Penn Tree Bank was invested over 1 million dollars and many person-years). To overcome this drawback, we have indirectly built such an annotated corpus for Vietnamese by taking advantages of annotated English corpora available, mutual disambiguation and projection via automatic word-alignments in an English-Vietnamese parallel corpus with 5 millions words.

In this paper, we present our building an annotated English-Vietnamese parallel Corpus named EVC, a corpus consisting of over 5 million words of English and Vietnamese. This EVC has been automatically word-aligned and POS-tagged by semantic-class model and K-best fast-TBL algorithm. This annotated parallel corpus has been exploited to be training data for Vietnamese-related NLP tasks such as Vietnamese Word Segmentation[1], Vietnamese-POS-Tagger [2], English-to-Vietnamese Word Order transfer [3], Word Sense Disambiguation [4], English-to-Vietnamese Machine Translation [5], etc.

The remains of this paper will be organized as follows:

- Collecting EVC: resources of raw parallel texts, its styles, etc.
- Normalizing EVC: converting from different formats, spelling, etc. into unique ones. Sentence alignment of EVC.
- Word alignment of EVC: to automatically word-align EVC by Semantic-Class approach. Manually correcting word-alignments. Problems of Vietnamese Word Segmentations.
- Annotating EVC with POS-tags and SEM-tags: Automatically tagging by TBL method of Eric Brill [6] for English side first, then projecting the English side to Vietnamese one. Tagsets of English and Vietnamese.
- Applications of EVC: to train for Vietnamese-related NLP tasks: POS-tagger, Word Order transfer, Word Sense Disambiguation, Machine Translation, etc.
- Conclusion: limitations of current EVC and its future developments, etc.

2. BUILDING EVC

2.1. Collecting EVC

This 5,000,000-word corpus is collected from many different resources of bilingual texts (such as books, dictionaries, corpora, etc.) in selected fields such as Science, Technology, daily conversation (see table 1). After collecting bilingual texts from different resources, this parallel corpus has been normalized in their form (text-only), tone marks (diacritics), character code of Vietnam (TCVN-3), character font (VN-Times), etc. Next, this corpus has been sentence aligned and spell-checked semi-automatically. An example of unannotated EVC is as the following:

*D02:01323: *Jet planes fly about nine miles high.*
+D02:01323: *Các phi cơ phản lực bay cao khoảng chín dặm.*

The codes at the beginning of each line above refer to the corresponding sentence in the EVC corpus. For full details of building this EVC corpus (e.g. collecting, normalizing, sentence alignment, spelling checker, etc.), please refer to [7].

2.2. Normalizing EVC

Due to the hetegenous corpus with texts in different domains and genres, we had to classify our EVC into different smaller corpora for training different domains, such as: computer, electronics, daily conversation, etc.

Remarkably, this EVC includes the SUSANNE corpus [8] – a golden corpus has been manually annotated such necessary English linguistic annotations as lemma, POS tags, chunking tags, syntactic trees, etc. This English corpus has been translated into Vietnamese by English teachers of the Foreign Language Department of University of Social Sciences and Humanity, Vietnam National University of HCM City. In this paper, this valuable annotated corpus is used as the kernel training corpus for annotating whole our EVC.

2.3. Sentence-alignment of EVC

During inputting this bilingual corpus, we have aligned sentences manually under the following format:

**D02:01323: The announcement of the royal birth was broadcast to the nation.*

+D02:01323: Lời loan báo sự ra đời của đứa con hoàng tộc đã được truyền thanh trên toàn quốc.

**D02:01324: Announcements of births, marriages and deaths appear in some newspapers.*

+D02:01324: Những thông báo về sự ra đời, cưới hỏi, tang chế xuất hiện trên một vài tờ báo.

In which, first characters are reference numbers indicating its sources and the position of sentence in texts.

If bilingual corpora are manually typed, we haven't used automatic sentential alignment, we aligned sentences by manual. If bilingual corpora are electronic media, we made use the automatic sentential alignment algorithm of Gale and Church [9] to align sentences.

2.4. Spelling Checker of EVC

After aligning sentences, we check the spell of English words and Vietnamese words automatically. Here, we have met another drawback in processing the Vietnamese word segmentation because Vietnamese words (similar to Chinese words) are not delimited by spaces [1]. However, our spelling checker is able to detect non-existent words in English or Vietnamese only. So, we must review this corpus manually. In fact, Vietnamese "word" here is only "tiếng", which is equivalent to Vietnamese "spelling word" or "morpheme" (due to features of isolated language typology).

3. WORD ALIGNMENT OF EVC

Next, this bilingual corpus has been automatically word aligned by a hybrid model combining the semantic class-based model S.K.Chang and J.S.Chang [10] with the GIZA++. In this model, the semantic classification of LLOCE of M.Arthur [11] is used. Besides, the Vietnamese word segmentation was also solved in this word-alignment [1]. An example of the word-alignment result is as in table 1 below. For full details of word alignment for this EVC, please refer to [12].

Table 1. An example of word-alignment

Jet planes fly about nine miles high.							
Các phi cơ phản lực bay cao chừng chín dặm.							
i	1	2	3	4	5	6	7
S	Jet	planes	fly	about	nine	miles	high
T	phản lực	các phi cơ	bay	chừng	chín	dặm	cao
j	2	1	3	4	5	6	4

Before describing this algorithm briefly, we have following conventions:

S stands for English sentence and T stands for Vietnamese one. We have sentence pair translated by each other is (S,T), s is the word in S, t is the word in T which is translated by s in S in context. DTs is the set of dictionary meanings for s entry, each meaning is represented by d.

W_S = { s }, set of English real words and idioms presented in S.

W_T = { t | t ∈T ∧ t ∈ VD }, set of Vietnamese possible words presented in T.

where : VD is the Vietnamese Dictionary containing Vietnamese possible words and phrases.

The problem is how computers can recognize which t in T will be aligned with which s in S. Relying on W_T, we can solve the case resulting in the wrong definitions of words in Vietnamese sentences when we only carry out word segment relying on VD. Our algorithm is in conformity with the following steps.

3.1 Dictionary-based word alignment

We mainly calculate the similarity on morpheme between each word d in DTs with all t in W_T based on formula calculating Dice coefficient (Dice, 1945) as follows:

where: | d | and | t | : the number of morphemes in d and in t.

| d ∩ t | : the number of morphemes in the intersection of d and t.

Next, for each word pair (s, t) obtained from Descartes product (W_S x W_T), we calculate the value of DTSim(s, t) presenting the likelihood of a connection as follows :

$$DTSim(s, t) = \max Sim(d, t)$$

Examining a sample on following sentence pair:

S = "The old man goes very fast"

T = "Ông cụ đi quá nhanh"

We will have:

W_S = { the, old, man, go, very, fast }

W_T = { ông, ông cụ, cụ, đi, nhanh, quá }

Suppose that we are examining on "man",

DT(man) = { người, đàn ông, nam nhi }

So, we have:

DTSim(man, ông) = max{ Sim(người, ông), Sim (đàn ông, ông), Sim(nam nhi, ông) }=

max{(2x0)/(1+1),(2x1)/(2+1),(2x0)/(2+1)}

= 0.67

DTSim(man, ông cụ) = max{ Sim(người,ông cụ), Sim(đàn ông,ông cụ), Sim(nam nhi, ông cụ)}=max{(2x0)/(1+2),(2x1)/(2+2),(2x0)/(2+2)} =0.5

Then, we choose candidate translation pairs of greatest likelihood of connection.

3.2. Calculating the correlation between two classes of two languages

The correlation ratio of class X and class Y can be measured using the Dice coefficient as follows:

$$ClassSim(X,Y) = \frac{\sum_{a \in X} From(a,Y) + \sum_{b \in Y} To(X,b)}{|X| + |Y|}$$

Where |X|= the total number of the words in X, |Y|= the total number of the words in Y, From(a,Y) =1,if $(\exists y \in Y)(a,y) \in ALLCONN$,

= 0, otherwise

To(X,b)= 1, if $(\exists x \in X)(x,b) \in ALLCONN$,

= 0, otherwise,

ALLCONN : a list of initial connections obtained by running above dictionary-based word alignment over the bilingual corpus.

3.3. Estimating the likelihood of candidate translation pairs

A coefficient, presented by Brown establishing each connection is a probabilistic value Pr(s,t), showing translated probability of each pair (s,t) in (S,T), calculated by product of dictionary translated probability, t(s | t), and dislocated probability of words in sentences, d (i | j, l, m). However Sue J. Ker and Jason S. Chang did not agree with it completely. In their opinion, it is very difficult to estimate t(s, t) and d(i, j) exactly for all values of s, t, i, j in the formula:

$$Pr(s, t) = t(s, t) \times$$

We have the same opinion with them. We can create functions based on dictionary, word concept and position of words in sentences to limit cases to be examined and computed.

The similar concept of word pair (s, t) function:

$$ConceptSim(s,t) = \max_{s \in X, t \in Y} ClassSim(X,Y)$$

Then, combining with DTSim(s, t), we have four value of t(s, t). We have to combine with DTSim(s, t) because we are partially basing on dictionary. Besides, we can solve the case that there are many words belonging to the same class in sentences.

Table 2. Constants in word alignment.

DTSim(s, t)	ConceptSim(s, t)	
a) t1	≥ h1	≥ h2
b) t2	≥ h1	< h2
c) t3	< h1	≥ h2
d) t4	< h1	< h2

Where h1 and h2 are thresholds chosen via experimental results.

4. ANNOTATING EVC

After word-aligning the EVC, linguistic units in EVC will be annotated with linguistic tags. Nevertheless, hand-annotation of even reasonably well-determined features such as part-of-speech (POS) tags has proved to be labor intensive and costly. In our work, we suggest a solution to avoid hand-annotations for word-aligned EVC by building linguistic-taggers (POS-tagger, Chunker, SENSE-tagger, etc.) using fTBL algorithm and linguistic information of corresponding Vietnamese via its word-alignment.

Our solution is motivated by I. Dagan, I.Alon, and S.Ulrike[13]. They proposed the use of bilingual corpora to avoid hand-tagging of training data. Their premise is that "different senses of a given word often translate differently in another language (for example, *pen* in English is *stylo* in French for its *writing implement* sense, and *enclos* for its *enclosure* sense). By using a parallel aligned corpus, the translation of each occurrence of a word such as *pen* can be used to automatically determine its sense". This remark is not only true for word sense but also for POS-tag and it is more exact in such typologically different languages as English vs. Vietnamese.

In fact, POS-tag annotations of English words as well as Vietnamese words are often ambiguous but they are not often exactly the same. For example (table 3), "can" in English may be "Aux" for *ability* sense, "V" for *to make a container* sense, and "N" for *a container* sense and there is hardly existing POS-tagger which can exactly POS-tag for that word "can" in all different contexts. Nevertheless, if that "can" in English is already word-aligned with a corresponding Vietnamese word, it will be easily POS-disambiguated by Vietnamese word' s POS-tags. For example, according to POS-tagset of PennTreeBank, if "can" is aligned with "có th ", it must be *Auxiliary* (MD) ; if it is aligned with "đóng h p" it must be a *Verb*(VB), and if it is aligned with "cái h p" it must be a *Noun* (NN)

Based on this reason, we have made a POS-tagger using fTBL algorithm to bootstrap the POS-annotation results of the English POS-tagger by exploiting the POS-information of the corresponding Vietnamese words via their word-alignments in EVC. Then, we directly project POS-annotations from English side to Vietnamese via available word alignments under the model of D.Yarowsky and G.Ngai [14].

Similarly, because we have made use of the class-based word alignment, after aligning words, we determine the semantic class of each word. For example: according to the SENSE-tagset of LLOCE [11], the word "letter" has 2 senses, one is "message" (if it belongs to class G155) and the other is "alphabet" (class G148). Similarly, the word "bank" has 3 senses, one is "money" (class J104), one is "river" (class L99) and one is "line" (class J41). After aligning words, the result of semantic annotation is as table 3,4 and 5 below (*i* and *j* are positions of English and Vietnamese words). If the output of automatic-annotations above is still ambiguous, it will be manually corrected to become an annotated training data for our Vietnamese-related corpus-based NLP-tasks.

Table 3. Result of sense tagging for "letter"

i	0	1	2	3	4	5	6
S	I	write	a	*letter*	to	my	friend
T	Tôi	viết	một	bức thư	cho	của tôi	bạn
j	0	1	2	3	5	7	6
	G 280	G 190		**G 155**		G 281	C 40

Table 4. Result of sense tagging for "bank"

i	0	1	2	3
S	I	enter	the	bank
T	Tôi	đi vào		ngân hàng
j	0	1	2	3
Class	G280	M5		**J104**

Table 5. Result of POS-tagging for "can"

I	0	1	2	3	4
S	I	*can*	*can*	a	*can*
POS	PP	**MD**	**VB**	DT	**NN**
T	Tôi	có thể	đóng hộp	một	cái hộp
j	0	1	3	5	6

4.1. Algorithm for annotating EVC

The main training algorithm used in our system is the fast Transformation-based learning (or fTBL). This algorithm has been used in annotating POS-tags and Chunker-tags for EVC, extracting the transformation-rules from annotated-EVC in order to tag for new English texts in our English-Vietnamese Translation system (or EVT).

In 1995, Eric Brill [6] promoted the Transformation-Based Learning (TBL) in his doctor thesis on the base of structural linguistics of Z.S.Harris. Since its birth, heretofore, TBL algorithm has been successfully applied into most of language problems. A remarkable characteristic of TBL in comparison with other learning algorithms is intuitiveness and simplicity. Linguists can fully observe and intervene during learning and tagging process as well as its intermediate and final results. In 2001, Radu Florian and Grace Ngai [15] promoted fast-TBL (fTBL) to improve the speed of training stage of TBL noticeably without reducing its accuracy. For full details of TBL and fTBL, please refer to [6] and [15].

The fTBL algorithm for linguistic-tagger can be formalized as below:

χ : sample space, the set of language units (word/phrase). In English, it is simple to recognize the word boundary, but in Vietnamese (an isolate language), it is rather complicated and we have solved in another work [1].

C : set of language's tags c (classification). For example: N,V,A,.. in POS-tagset; HUM,ANI,NAT,... in sense-tagset, NP_B, NP_I, NP_O, ... in chunker-tagset, etc.

$S = \chi x C$: the cross-product between the sample space (word/phrase) and the classification space (tagset). It is the state space where each point is a couple (word, tag) or (phrase,tag).

π : predicate defined on S^+ space, which is on a sequence of states. This predicate π follows the human-specified templates of transformation rules. Depending on the specified linguistic-taggers, we will have different templates. For example, in the POS-tagger for English, this predicate only consists of English factors which affect the POS-tagging process, e.g.: $\bigcup_{\exists i \in [-m,+n]} Word_i$ or $\bigcup_{\exists i \in [-m,+n]} Tag_i$ or $\bigcup_{\exists i \in [-m,+n]} Word_i \wedge Tag_j$. Where, $Word_i$ and Tag_i are the word-form and the word-tag of the i^{th} word from the current word. Positive values of i mean the preceding (its left side), and negative ones mean the following (its right side). The value of i ranges within the window from $-m$ to $+n$.

A rule r defined as a couple (π, c) which consists of predicate π and tag c. Rule r is written in the form $\pi \Rightarrow c$. This means that the rule $r = (\pi, c)$ will be applied on the sample x if the predicate π is satisfied on it, whereat, x will be assigned a new tag c.

Giving a state s = (x,c) and rule r = (π, c), then the result state r(s), which is gained by applying rule r on s, is defined as:

$$r(s) = \begin{cases} s & \text{if } \pi(s)=\text{False} \\ (x, c') & \text{if } \pi(s)=\text{True} \end{cases}$$

T : set of training samples (or called *golden corpus*), which were assigned correct tags. Depending on the specified linguistic-taggers, we will have different golden corpora. In the POS-tagger and Chunker for EVC, we made use of the golden corpus SUSANNE [8]. In the linguistic-taggers for EVT, T is the annotated and revised EVC.

The score of each rule r = (π, c) is the difference between the result processed on the sample s of rule r and the initial state, in conformity with the following formula:

$$Score(r) = \sum_{s \in T} score(r(s)) - \sum_{s \in T} score(s)$$

$$score((x,c)) = \begin{cases} 1 & \text{if } c = \text{True}(x) \\ 0 & \text{if } c \neq \text{True}(x) \end{cases}$$

4.2. The training period of algorithm

Step 1: Initiating for each sample x in training set with the most suitable tag c (called *baseline tagging*). For instance, the word "can" in English has the highest part-of-speech probability as an *Auxiliary*. We call the first time corpus T_0. For English, we may make use of available powerful linguistic-taggers for English.

Step 2: Examining all transformation rules *r* influencing corpus T_k in time k^{th} and choosing a rule that has the highest Score(r) and applying this rule for corpus T_k to get new corpus T_{k+1}. We have : $T_{k+1} = r(T_k) = \{ r(s) \mid s \in T_k \}$. If there is no rule which satisfies Score(r) > β, the algorithm is stopped. β is the threshold, which is preset, and adjusted according to real demand. These rules change the linguistic-tags of words based upon the contexts they appear in. fTBL evaluates the result of applying that candidate rule by comparing the current result of linguistic-annotations with that of the golden corpus in order to choose the best one which has highest mark. These optimal rules create an ordered sequence.

Step 3: k = k+1.

Step 4: Repeat from step 2.

4.3. The executing period of algorithm fTBL

Starting with the new unannotated text, fTBL assigns an inital linguistic-tag to each word/phrase in text in a way similar to that of the training period (baseline tagging).

The sequence of optimal rules (extracted from the training period) are applied, which change the linguistic-tags based upon the contexts they appear in. These rules are applied deterministically in the order they appear in the sequence.

4.4. Projecting English POS-Tags to Vietnamese

After having English-POS-tag annotations with high precision, we proceed to directly project those POS-tag annotations from English side into Vietnamese side. Our solution is motivated by a similar work of David Yarowsky and Grace Ngai [14]. This projection is based on available word-alignments in the automatically word-aligned English-Vietnamese parallel corpus.

Nevertheless, due to typological difference between English (an inflected typology) vs. Vietnamese (an isolated typology), direct projection is not a simple 1-1 map but it may be a complex m-n map:

- Regarding grammatical meanings, English usually makes use of inflectional facilities, such as suffixes to express grammatical meanings. For example: *-s* →plural, *-ed* →past, *-ing*→continuous, *'s* → possesive case, etc. Whilst Vietnamese often makes use of function words, word order facilities. For example: "các"" "những" → plural, "đã" → past, "đang" → continuous, "của" → possesive cases, etc.
- Regarding lexicalization, some words in English must be represented by a phrase in Vietnamese and vice-versa. For example: "cow" and "ox" in English will be rephrased into two words "bò cái" (female one) and "bò đực" (male one) in Vietnamese; or "nghé" in Vietnamese will be rephrased into two words "buffalo calf" in English.

The result of projecting is as table 6 below.

In addition, tagsets of two languages are different. Due to the characteristics of each language, we must use two different tagset for POS-tagging. Regarding English, we made use of available POS-tagset of PennTreeBank. While in Vietnamese, we made use of POS-tagset in the standard Vietnamese dictionary of Hoang Phe [16] and other new tags. So, we must have an English-Vietnamese consensus tagset map (please refer to Appendix A).

Table 6. An example of English POS-tagging in parallel corpus EVC

EN	Jet	planes	fly	about	nine	miles	high
E-tag	NN	NNS	VBP	IN	CD	NNS	RB
VN	phản lực	(các) phi cơ	bay	khoảng	chín	dặm	cao
V-tag	N	N	V	IN	CD	N	R

Regarding evaluation of POS-tag projections, because so far, there has been no POS-annotated corpus available for Vietnamese, we had to manually build a small golden corpus for Vietnamese POS-tagging with approximately 1000 words for evaluating. The results of Vietnamese POS-tagging is as table 7 below:

Table 7. The result of projecting POS-tags from English side to Vietnamese in EVC.

Method	Correct tags	Incorrect Tags	Precision
Baseline tagging (use information of POS-tag in dictionary)	823	177	82.3%
Projecting from English side in EVC	946	54	94.6%

5. APPLICATIONS OF EVC

After annotating EVC, we can extract transformational rules to apply in Vietnamese-related corpus-based NLP-tasks as below. These extracted rules are intuitive rules and easy to understand by human beings. For examples:

5.1. Application in POS-Taggers

1. $((\exists i \in [-3,-1] \mid Tag_i = MD) \wedge (tag_0 = VPB)) \Rightarrow tag_0 \leftarrow VB$
2. $((tag_{-1} = TO) \wedge (tag_0 = NN)) \Rightarrow tag_0 \leftarrow VB$
3. $((\exists i \in [-2,-1] \mid Word_i = "have") \wedge (tag_0 = VBD)) \Rightarrow tag_0 \leftarrow VBN$
4. $((Word_0 = "can") \wedge (VTag_0 = MD) \wedge (tag_0 = VB)) \Rightarrow tag_0 \leftarrow MD$

The 4th rule will be understood as follows: *"if the POS-tag of current word is VB (Verb) and its word-form is "can" and its corresponding Vietnamese word-tag is MD (Modal), then the POS-tag of current word will be changed into MD"*. These learning rules have been apply in training the Vietnamese POS-Tagger [2].

5.2. Application in Sense-Taggers

$$1.\quad ((\exists i \in [+1,+3] \mid Word_i = "river") \wedge (Word_0 = "bank) \wedge (POS_0 = NN)) \Rightarrow tag_0 \leftarrow NAT$$

$$2.\quad ((SUB_0 \in HUM) \wedge (Word_0 = "enter") \wedge (POS_0 = VB) \wedge (Word_0 \in MOV) \Rightarrow OBJ_0 \leftarrow HOU$$

The 1st will be understood *"if there exist a word-form is "river" within 3 positions right after the word form "bank", the SENSE-tag of current word is changed into NAT (L99: Natural)"*.
Similarly, the 2nd rule will be understood *"if sense-tag of SUBject is HUMan and the current word-form is "enter" and its POS-tag is Verb and its is a MOTION, then its OBJect will be assign to sense-tag HOU"*. For example: in the sentence *"I enter the bank"*, the object "bank" will be assign to "financial building" (J104: money). These learning rules have been apply in training the Sense-Tagger of English-to-Vietnamese Machine Translation [4].

5.3. Application in Word-Order Transfers

$$(POS_{N_a} = Qwh) \cap (POS_{N_{a1}} = Aux) \cap (POS_{N_{a2}} = SP) \cap (POS_{N_{a3}} = VP) \Rightarrow N_{a2} - N_{a1} - N_{a3} - \text{"được không"}$$

This rules means that: *"if the interrogative sentence (Qwh) has the source syntax tree including: auxiliary verb (Aux) – subject (SP) – predicate (VP), it will be transferred into Vietnamese sentence as the following: subject – auxiliary verb – predicate and the inserted expletive "không" at the end of the sentence"*. For example: "Can you speak English ?" ⇒ "Anh có thể nói tiếng Anh được không ?". These learning rules have been apply in training the Word-Order Transfer of English-to-Vietnamese Machine Translation [3].

6. CONCLUSION

We have just presented the building an annotated English-Vietnamese parallel Corpus This 5-million word EVC has been collected from selected resources, normalized into standard format, and word-aligned by semantic class-based approach. Finally, this EVC has been POS-tagged by POS-tagging English words first and then projecting them to Vietnamese side later. The English POS-tagging is done in 2 steps: The basic tagging step is achieved through the available POS-tagger and the correction step is achieved through the fTBL learning algorithm in which the information on the corresponding Vietnamese is used through available word-alignment in the EVC.
The result of word-alignment and POS-tagging of Vietnamese in the English-Vietnamese bilingual corpus has played a meaningful role in the building of the automatic training corpus for our Vietnamese NLP tasks, such as Vietnamese POS-taggers, WSD in English-to-Vietnamese MT [5], etc. By making use of the language typology' s differences and the word-alignments in bilingual corpus for the mutual disambiguation, we are still able to improve the result of the word-alignment and other linguistic annotation.
Currently, we are improving the quality of EVC by manually correcting linguistic annotations such as: word alignment, POS-tagging, etc. We are also tagging this EVC semantic-label by using semantic class name via available word-alignment in EVC.

References

[1] Dinh Dien, Hoang Kiem, Nguyen Van Toan , "Vietnamese Word Segmentation", *Proceedings of NLPRS'01* (The 6th Natural Language Processing Pacific Rim Symposium), Tokyo, Japan, pp. 749-756, 2001.

[2] Dinh Dien and Hoang Kiem, *"POS-tagger for English-Vietnamese bilingual corpus"*, proceedings of HLT-NAACL(North American Association for Computational Linguistics)-03 Workshop "Building and Using Parallel Texts", Edmonton, Canada, pp. 88-95, 2003.

[3] Dien Dinh, Thuy Ngan, Xuan Quang, Chi Nam, "A hyrid approach to word-order transfer in the English – Vietnamese Machine Translation System", *Proceedings of the MT Summit IX*, Louisiana, USA, pp. 79-86, 2003.

[4] Dien Dinh (2002), "Building a training corpus for word sense disambiguation in the English-to-Vietnamese Machine Translation", *Proceedings of Workshop on Machine Translation in Asia, COLING-02*, Taiwan, pp. 26-32, 2002.

[5] Dien Dinh, Kiem Hoang, Eduard Hovy , "BTL: an Hybrid Model in the English – Vietnamese Machine Translation System", *Proceedings of the MT Summit IX*, Louisiana, USA, pp. 87-94, 2003.

[6] Eric Brill, "Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging". *Computational Linguistics*, 21(4), pp. 543-565, 1995.

[7] D.Dien, *Building English-Vietnamese bilingual corpus*, Master thesis in Comparative Linguistics, Literature and Linguistics Faculty of University of Social Sciences and Humanity, Vietnam National Uni. of HCM City, 2001.

[8] G. Sampson, English for the Computer: The SUSANNE Corpus and Analytic Scheme. Clarendon Press (Oxford University Press), 1995.

[9] Gale W.A and Church K.W., *A program for aligning sentences in bilingual corpora.* Proceedings of ACL, 1991.

[10] S. K. Jang and J.S. Chang, *A Class-based Approach to Word Alignment.* Computational Linguistics, 23(2), pp. 313-343, 1997.

[11] M. Arthur, Longman Lexicon Of Contemporary English (Vietnamese version by Tran Tat Thang). VN Education Publisher, 1997.

[12] D.Dien, H.Kiem, T.Ngan, X.Quang, V.Toan, Q.Hung, P.Hoi, *Word alignment in English – Vietnamese bilingual corpus*. Proceedings of EALPIIT'02, HaNoi, Vietnam, pg 3-11, 2002.

[13] I. Dagan, I.Alon, and S.Ulrike, Two languages are more informative than one. In *Proceedings of the 29th Annual ACL*, Berkeley, CA, pp.130-137, 1991.

[14] David Yarowsky and Grace Ngai, Induce, Multilingual POS Tagger and NP bracketer via projection on aligned corpora. Proceedings of NAACL-01, 2001.

[15] R. Florian, and G.Ngai, *Transformation-Based Learning in the fast lane*. Proceedings of North America ACL, 2001.

[16] H. Phe. Vietnamese Dictionary, Center of Lexicography. Da Nang Publisher, 1998.

Appendix A. English-Vietnamese consensus POS-tagset mapping table

English POS	Vietnamese POS
CC (Coordinating conjunction)	CC
CD (Cardinal number)	CD
DT (Determiner)	DT
EX (Existential)	V
FW (Foreign word)	FW
IN (Preposition)	IN
JJ (Adjective)	A
JJR (Adjective, comparative)	A
JJS (Adjective, superlative)	A
LS (List item marker)	LS
MD (Modal)	MD
NN (Noun, singular or mass)	N
NNS (Noun, plural)	N
NP (Proper noun, singular)	N
NPS (Proper noun, plural)	N
PDT (Predeterminer)	DT
POS (Possessive ending)	"của"
PP (Personal pronoun)	P
PP$ (Possessive pronoun)	"của" P
RB (Adverb)	R
RBR (Adverb, comparative)	R
RBS (Adverb, superlative)	R
RP (Particle)	RP
SYM (Symbol)	SYM
TO ("to")	-
UH (Interjection)	UH
VB (Verb, base form)	V
VBD (Verb, past tense)	V
VBG (Verb, gerund or present participle)	V
VBN (Verb, past participle)	V
VBP (Verb, non-3rd person singular present)	V
VBZ (Verb, 3rd person singular present)	V
WDT (Whdeterminer)	P
WP (Wh-pronoun)	P
WP$ (Possessive wh-pronoun)	"của" P
WRB (Wh-adverb)	R