Finding Target Language Correspondence for Lexicalized EBMT System

Wei Wang[1]
Beijing University of Posts and Telecoms, 181#, Beijing, 100876, P.R.C ,
y9772209@bupt.edu.cn

Jin-Xia Huang
Microsoft Research, China Beijing, 100080, P.R.C
i-jxh@microsoft.com

Ming Zhou
Microsoft Research, China Beijing, 100080, P.R.C
mingzhou@microsoft.com

Chang-Ning Huang
Microsoft Research, China Beijing, 100080, P.R.C
cnhuang@microsoft.com

Abstract

This paper presents a three-phase approach to find the correspondence in Target Language (TL) sentence for a fragment of Source Language (SL) sentence in a lexicalized EBMT system. To be practical, it exploits surface information as much as possible instead of using parsers. Experiments show that, although not so perfect, it is very robust and effective. The three phases are: First, align the sentence pair at word level in order to provide anchors for phrase alignment. Second, based on the aligned anchors, find all the TL fragments which may possibly correspond to a SL fragment. And finally, using a score function, select the best TL fragment as the correspondence of a SL fragment.

1 Introduction

In an EBMT system, translation generally involves two fundamental operations: Matching and Transfer. The former is to retrieve the "closest match" for an SL (Source Language) fragment from the example database. The latter is to generate the translation in terms of the matched examples. To be specific, it is actually the process of deciding which fragment in TL (Target Language) sentence corresponds to the fragment in SL sentence. This is exactly the problem of alignment.

So far, several methods have been put forward to fulfill this kind of task. Most of them can be classified into two categories.

Structural: By methods in this category correspondences are found with the help of parsers, and the correspondences found are grammatical units. The SL sentence is first parsed with a SL parser and the TL sentence with a TL parser to get the paired parses. Then the structural correspondences are found based on the structural constraints of the paired parse trees. Kaji (1992) uses CKY parses. Watanabe (2000), Grishman (1994), and Meyers (1993) propose methods for finding the structural matching (pair of dependency trees) from dependency parses.

Grammarless: Methods in this category are used to find correspondences with co-occurrence information and geometric information rather than using parsers. Co-occurrence information is obtained through examining whether there are co-occurrences of SL fragment and TL fragment. Geometric information is to constrain the alignment space. The correspondences found by these methods are grammarless. McTait, K. (1999) obtains translation templates by firstly aligning collocations with the help of the whole corpus statistics first, and then aligning the "slots" of the templates. Cicekli, I. (2001) aligns sentence fragment through "Similarity and Difference" approach. (Wu, D., 1995a, 1995b) uses ITG to constrain alignment space to bracket and align the bilingual corpora simultaneously. (Nirenburg, S. et al, 1994) uses eight tests to select translations in "inter-language" phase.

Almost all these methods focus mainly on the "appropriateness" of the extracted correspondences. (Such as: whether the correspondences can be extracted for translation?). Once the correspondences are extracted, they are stored in the example base (EB). During translation, an example in EB will be stimulated only if there is a fragment matching the SL side of the example in the input SL sentence.

[1] This work was done while the author was visiting Microsoft Research China.

The alignment approach to be described in this paper works in a different way. It is designed for an EBMT engine integrated into the MS software localization tool to support English-Chinese translation. Figure 1 shows its role in the whole system.

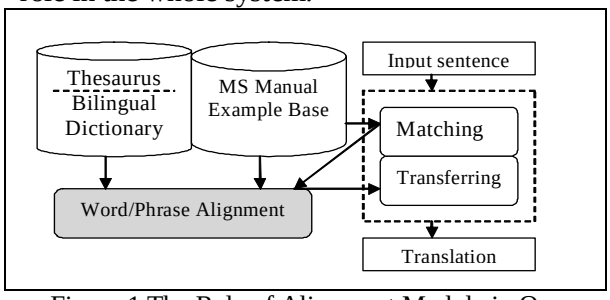

Figure 1 The Role of Alignment Module in Our EBMT system

When a SL sentence is put into the system, the "matching operation" will decompose it into fragments by looking through the EB for matched examples. And when an example is retrieved, the boundary of the matched part in SL side of the example is recorded. Then the example with the recorded information is sent to the alignment module to find the TL correspondence for the recorded SL fragment. In the end, the correspondences found are processed by the "transfer operation" to produce the final translation.

Therefore, in our case, the alignment algorithm has no right to decide the boundary of SL fragment. The boundary is given by the matching component. This is a difference between our case and the "traditional phrase alignment".

To resolve fragment alignments of this kind of problem, we present an approach, which follows three phases: Firstly, align the sentence pair at word level in order to provide anchors for fragment alignment. Secondly, based on the aligned anchors, find all the TL fragments which may possibly correspond to the SL fragment. And finally, using a score function, select the best TL fragment as the translation (alignment).

This paper is structured in this way: Section 2 describes the first phase (word alignment) and a cascaded aligner is introduced in addition. Section 3 concerns the other two phases. And the phrase alignment results are also presented in this section. Finally, we give some conclusions.

2 Word Alignment Design

In order to supply enough anchors for phrase alignment, it is important and necessary to realize word alignment first. Word alignment has been widely studied, and many approaches have been put forward. It should resolve two problems: One is **word-similarity model.** The model is to describe the degree of possibility that two words in different languages can be translated into each other. The other is **word-distortion model**. This model is to describe the degree of the likeliness that one position in the SL sentence can be aligned to another position in the TL sentence.

2.1 Resources Available

The resources that we can exploit are:

1) E-C/C-E bilingual dictionary (79201 English entries for E-C and 45607 Chinese entries for C-E)
2) Stemming tools
3) Manually word aligned bilingual corpus of general domain (30000 sentence pairs)
4) Microsoft software manual bilingual corpus (sentence aligned) (12590 sentence pairs)
5) Chinese monolingual lexicon (Extra dictionary plus the words extracted from bilingual dictionary) Chinese monolingual lexicon (Extra dictionary plus the words extracted from bilingual dictionary) English phrase dictionary (Extracted from above bilingual dictionaries)
6) Thesaurus: English: Word Net, with 45784 classes (92609 words/phrases). Chinese: 现代汉语分类词典 (*Xian4 Dai4 Han4 Yu3 Fen1 Lei4 Ci2 Dian3*) which has 3724 classes (40289 words).

2.2 Word Aligner Based-on Cascaded Architecture

Having done several experiments to test the efficiency of different ways to combine resources and the efficiency of different topologies of aligners, we find that better word alignment results can be achieved through using the cascaded architecture, which is shown in figure 2.

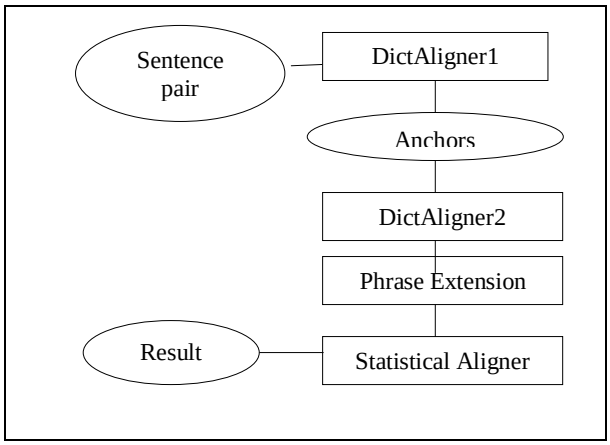

Figure 2 Cascaded Architecture for Word Alignment

precision = C / D, recall = C/ E	(1)

Where,
C is the number of correctly aligned English words in the aligner dumps.
D is the number of those English words, which have alignment in aligner dump.
E is the number of English words, which have alignment in test set.

NOTE that our evaluate metric is very strict. For the case of "many to many" alignment, if one word is not aligned correctly, all the words in one concept (word group) should be regarded as being wrong.

The sentence pairs are firstly put into the DictAligner1. The aligner uses dice coefficient Ker(1997) as the word similarity metric, and uses no distortion model because the Chinese sentences are not word segmented yet. And there is no consideration on duplicate word in this phase. DictAligner1 aims to find high confidence alignment anchors only based on a bilingual dictionary. After this, the unaligned part in the Chinese sentences will be word segmented. Then DictAligner2 will adopt a distortion model to align the remained words with a threshold. Now, there will appear the partial alignment phenomenon in the alignment result. Figure 3 shows an example. To resolve this problem, we use a phase called Phrase Extension to make up for it. That is, use monolingual dictionaries to merge the words/characters into phrase/words. Finally, we use a statistical aligner to align the words left.

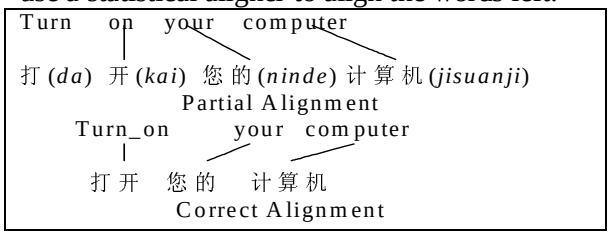

Figure 3 The Partial Alignment Problem

In this cascaded scheme, each step is to align a subset of all the words with different knowledge. And the knowledge need in each step is trained from the output of the previous step. This can make the aligner in each cascade to focus on certain problems rather than on all the problems. In fact, this is the idea of Divide and Conquer.

2.3 Word Alignment Experiment

2.3.1 Experimental Settings

We use two test sets. One is extracted from Microsoft software manual (section 2.1). This test set contains 130 sentence pairs. The other is from general domain bilingual corpus (section 2.1). it incoudes 1000 sentence pairs.
Two metrics are used and defined as (1):

2.3.2 Experiment Results

Table 1 shows the result of word alignment.
Table 1

	MS Manual (x%)		General Corpus (x%)	
	Prec.	Rec.	Prec.	Rec.
Complete	88.48	76.36	78.33	53.82
Un-complete	97.91	84.50	88.36	62.21

In this table, "Complete" means the alignment is completely right. And "Un-complete" means that the alignment is partially correct. We can see from the result that the accuracy of un-complete case is high. It means the cascaded aligner can align an English word to the correct position with high accuracy.

3 Phrase Alignment

As is mentioned (in **section 1**), there are some differences between our task of phrase alignment and that of others. We cannot use parser although we have them because of the erroneous parsing result. We have to exploit as much as possible the surface knowledge, such as length, the number of content words or number of functional words. Figure 4 shows the mechanism of our phrase alignment module.

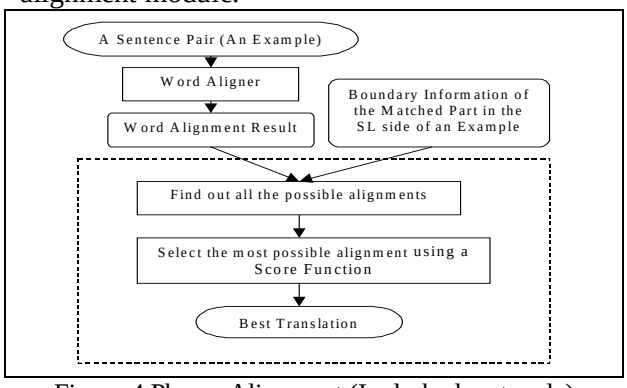

Figure 4 Phrase Alignment (In dashed rectangle)

As figure 4 shows, the fragment alignment module runs after the word alignment module and works on the base of the word alignment results of sentence pair (example). There are two phases:

1) **Find all possible candidates:**

To find all the possible TL fragments corresponding to the matched part (by the "matching operation") in the SL sentence.

2) **Select the best alignment:**

Using a **Score Function** to select the best translation out of all the possible translations

3.1 Find All Possible Candidates

If a continuous English sentence fragment always corresponds to a continuous Chinese fragment, the job would be much easier. However, it is not true. For English and Chinese, it is often the case as is shown in the following figure:

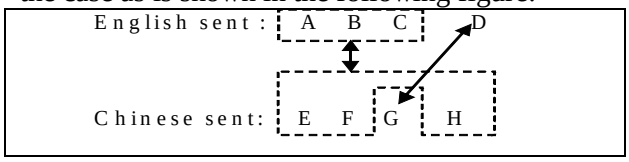

Figure 5 Non-continuous

Where, A to H represent words or word sequences

To explain what is non-continuous, we define "continuous fragment" as follows:

Continuous fragment: suppose F is a fragment in SL sentence, and G is a fragment in TL sentence. If all the words in F are aligned to G and only to G, we say F is continuous to G, and vice versa. For example, in figure 6, fragment "E F G H" is not a continuous fragment to fragment "A B C".

To resolve these problems of non-continuous in figure 5, we consider the following two fragment alignment cases:

1) Continuous English fragment corresponds to continuous Chinese fragment.

2) Continuous English fragment corresponds to non-continuous Chinese fragment, but the outrange Chinese words (such as G in figure 5) correspond to continuous English words (such as D in figure 5).

Algorithm 1: Find_All_Possible_Alignments_**continuous**
Input: (1) A sentence pair aligned at word level.
(2) [**a**,**b**] : Boundaries of the SL fragment to find all possible alignments in TL. (**a** and **b** are word positions in SL sentence).
Output: All possible alignments
Algorithm:

1. Find the word set (**A**) in TL sent which aligns to fragment [**a**,**b**] of SL sent based on word alignment.
2. Find the leftmost(**c**) and rightmost (**d**) sentence positions for the words in **A**. The TL fragment [**c**,**d**] is the **Minimum Possible Alignment(MinPA)**.
3. Extend **MinPA** to left and right side respectively until aligned words is met. (**e** denotes leftmost and **f** the rightmost), then the Chinese fragment (**e**,**f**) is the **Maximum Possible Alignment** (**MaxPA**).
4. Get all the continuous substrings (this continuous means no word gap) between **MinPA** and **MaxPA**, and each substring must contain **MinPA**. This set is called **AP.**
5. Return **MinPA** ∪**MaxPA** ∪ **AP**

Figure 6 Continuous Case (Case 1)

Algorithm 2: Find_All_Possible_Translations_**non-continuous**
Input: (1) A sentence pair aligned at word level.
(2) [**a**,**b**] : Fragment boundaries in SL sentence to find all possible alignments in TL sentence. (**a** and **b** are word positions in SL sent).
Output: All possible alignments
Algorithm:

1. Find the word set (**A**) in TL sent which aligns to fragment [**a**,**b** of SL sent. Find the word set (**B**) in English sent that aligns to **A**, but beyond the rang of [**a**,**b**].
2. If **B** is continuous
   - 2.1 Obtain the continuous (with no word gap) word set (**C**) containing **B.**
   - 2.2 Find all the possible alignments for **C** using **Algorithm 1**, and select the best one (**D**) using the Score Function (which will be mentioned in the next section).
   - 2.3 Find the leftmost (**i**) and rightmost (**j**) position in **A**. Remove **D** from [**i**,**j**], Then [**i**,**j**] - **D** is the **MinPA** of [**a**,**b**]
   - 2.4 Extend **MinPA** in the same way as step 3 in Algorithm 1 to get the **MaxPA**.
   - 2.5 Get all the continuous substrings (here "continuous" means no word gap). And each substring must contain **MinPA**. This set is called **AP.**
3. Return **MinPA** ∪**MaxPA** ∪ **AP**

Figure 7 Non-continuous Fragment (Case 2)

We design an algorithm (algorithm 1 in figure 6) for Case 1 and another algorithm (algorithm 2 in figure 7) for Case 2. The idea of figure 6 is: Find the minimal and maximal possible TL alignment of the SL fragment, and then return all the substrings within them. The idea of figure 7 is similar except for removing the non-continuous part (such as G in figure 5) first.

3.2 Select the Best Alignment Among All Possible Candidates

We have tested the following features (Table 3) and their combinations to get score functions.

Table 2 Factors For Score Function

Fragment Length	Both the length of the English fragment (**l**) and that (**m**) of a candidate are important cues.
Functional Words	The number of functional words (**j**) in the possibly corresponding fragments and the difference (**Δj=|j of TL – j of SL|**) are important cues.
Content Words	The number of content words (**k**) in the possibly corresponding fragments and the difference (**Δk=|k of TL – k of SL|**) are important cues.

And we have designed several score functions (SF) (Table 3) that adopt the above feature to score

Table 3 Score Functions

SF 0	Return **MinPA**	
SF 1	$P(\mathbf{m}\|\mathbf{l})$	(2)
SF 2	$P(\mathbf{m}\|\mathbf{l})P(\Delta\mathbf{k} \mid \mathbf{m}\,\mathbf{l})P(\Delta\mathbf{j} \mid \mathbf{m}\,\mathbf{l})$	(3)
SF 3	$P(\mathbf{m}\|\mathbf{l})P(\Delta\mathbf{k} \mid \mathbf{m}\,\mathbf{l}\,\mathbf{k})P(\Delta\mathbf{j} \mid \mathbf{m}\,\mathbf{l}\,\mathbf{j})$	(4)
SF 4	$P = \frac{\varepsilon P(m \mid l)}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(wordc_j \mid worde_i)$	(5)

k: the number of content words in SL sent.
j: the number of functional words in SL sent.
Other annotations see Table 3.

all the possible alignments.

In (Table 3), **SF 0** only gives non-zero score to **MinPA** (Figure 6). The performance of this function is used as the Baseline. SF 1 scores candidates on the base of fragments lengths. SF 2,3 scores candidates based on content word/functional word number. SF 4 uses the language production story of SMT Model 1(Brown, 1993) plus the length factor.

3.3 Experimental Settings

All the parameters (except t(.|.) in the above formulas are estimated from the general domain manually word aligned bilingual corpus (30000 sentence pairs). And the probability t(.|.) is got from MS Manual.

The functional word list is extracted from our bilingual dictionary based on part of speech (such as PREP, etc). The size of English functional word list is 1152, and that of Chinese is 5591. The remaining words (except the functional words in the functional list) construct the content word lists.

We construct a test to examine each Score Function's performance. We randomly extracted 130 sentence pairs from the MS Manual corpus (These sentences are extracted before training, so that the test set and training set are not overlapped.) We arbitrarily annotate the boundaries (left and right side) of a fragment for each sentence pair. Thus there are 153 phrases to find alignment.

3.4 Evaluation

Evaluation Metrics are defined as (6):

precision = C / D, recall = C / E	(6)

Where *C* is the number of correct cases, and *D* is the number of cases that the candidates contain correct translation. E is the number of all test English fragment.

NOTE that the two values are **strictly** computed, that is: if and only if the alignment is completely equal to the correct answer, it will be added into C. If there is a difference (even if unimportant) between the result of the aligner and the correct answer, we will regard it as being wrong.

3.5 Phrase Alignment Results

Table 4 shows the result of phrase alignment.
It indicates, the performance of SF 1 and 2 is better than the baseline level. SF 3 and 4 are worse than baseline. Therefore we can conclude that the length factor is helpful. Both SF 2 and SF 3 both adopt the content/functional word number information. However, their results are different. This may indicate that this kind of information can be used roughly and generally. If we want to tune it fine with more complex conditional probability,

Table Result of Phrase Alignment

Score Function	Precision	Recall
Baseline (SF 0)	0.6818(90/132)	0.5882(90/153)
SF 1	0.7273(96/132)	0.6275(96/153)
SF 2	0.7045(93/132)	0.6078(93/153)
SF 3	0.6667(88/132)	0.5752(88/153)
SF 4	0.6439(85/132)	0.5556 (85/153)

more errors may come out due to the inconsistency (functional word in one language may not be functional word in another) between the two set of functional list. The worst performance of the SMT based SF (SF4) in the table shows the inclusion of unaligned word into a phrase

Using edit distance, and formula:

Precision = 1-(Del+Ins+Sub)/(word number in reference set) Where, Del is the number of deleted words, Sub is the number of inserted words, and Sub is the number of substituted words.	(7)

The accuracy for SF2 is 88.07%, and in the error cases, Sub (1.17%), Ins (3.89%), Del (6.87%).

4. Discussions

In this paper, we have presented a three-phase approach to find the TL correspondence of a SL fragment for an EBMT system.

To be practical and robust, this approach exploits surface information as much as possible. It is language independent. Experiments show its effectiveness.

Compared with the traditional phrase alignment task (section 1), our task is that the aligner must return a correspondence for the SL fragment, whose boundaries are decided by the "matching operation" (section 1) instead of the aligner itself, even if there are some unaligned words in the SL fragment and there is no structure information to utilize. Compared with other approaches to the similar task, ours not only finds correspondence not only for "continuous" SL fragment (section 3.1), but also for "non-continuous" case (section 3.1).

We have made several experiments for each phase of this approach. For the word alignment phase, we have found a better way to utilize resources and different kinds of word aligners. Experiment results show that the cascaded architecture performs well. For phrase alignment phase, we have tested 5 score functions, and found that it may be helpful to use the cues of length,

content words number and functional words number appropriately.

We feel that it is difficult to achieve even higher accuracy by only modifying the two models (Word-Similarity model and Word-Distortion model in section 2) of one single aligner. We can view the word alignment task as classification: Better classifiers can be produced by classifier assembles. We believe that, with "Aligner Assembles" and voting approaches, even better results can be achieved. For fragment alignment, the difficulty is to decide the inclusion of those unaligned words (especially functional words). Other features such as word combination biases (such as "的"(*de)* may stick to the words before it, and "在"(*zai*) may stick to "之上"(on | *zhishang*)) can also be helpful.

References

Brown, P.F., (1998) *A Statistical Approach to Language Translation*, In COLING-88, vol. 1, pp.71-76, 1988.

Brown, P.F., Della Pietra, S.A., Della Pietra V.J., and Mercer, R.L. (1993) *The mathematics of Statistical Machine Translation: Parameter Estimation*, Computational Linguistics, 19(2), pp.263-311.

Cicekli, I., and Altay, H. (2001), *Learning Translation Templates from Bilingual Translation Examples*, Applied Intelligence, Vol. 15, No. 1, pp: 57-76.

Grishman, R., (1994) *Iterative Alignment of Syntactic Structures for a Bilingual Corpus*, Proc. of 2nd Workshop For Very Large Corpora, pp. 57-68.

Huang, J., Choi, K. (2000) *Chinese-Korean Word Alignment Based on Linguistic Comparson*, ACL2000.

Ido, D., Church, K.W. and Gale, W.A. (1994) *Robust bilingual word alignment for machine-aided translation*, In Proceedings of 4th conference on Applied Natural Language Processing (ANLP-94), pp.34-40, 1994.

Kaji, H., Kida, Y., and Morimoto, Y., (1992), *Learning Translation Templates from Bilingual Texts*, Proc. of Coling 92, pp. 672-678.

Ker, S., Chang, J. (1997), *A Class-based Approach to Word Alignment*, Computational Linguistics (1997, Vol. 23, Num. 2, pp. 313-343).

McTait, K., Trujillo, A., (1999) *A Language-Neutral Sparse-Data Algorithm For Extracting Translation Patterns*, TMI-99, Tenth International Conference on Theoretical And Methodological Issues in Machine Translation.

Meyers, A., Yanharber, R., and Grishman, R., (1996) *Alignment of Shared Forests for Bilingual Corpora*, Proc. of the 16th of Coling, pp. 460-465.

Nirenburg, S., Beale, S., and Domashnev, C. (1994) *A Full-text Experiment in Example-Based Machine Translation*, in new methods in language processing, Manchester, England.

Och, F.J., Tillmann, C., and Ney, H. (1999) *Improved Alignment Models for Statistical Machine Translation*. In Proc. of the Joint SIGDAT Conf. on Empirical Methods in Natural Language Processing and Very Large Corpora, pp.20-28, 1999.

F. Pascale, (1995) *A Pattern Matching Method for Finding Noun and Proper Noun Translations from Noisy Parallel Corpora*, In Computational Linguistics, 21(4), pp. 236-233.

Shin, H., Young, S. and Choi, K., (1996) *Bilingual knowledge acquisition from Kerean-English Parallel corpus using alignment method*, In Proceedings of the 15th International Conference on Computational Linguistics, 1996.

Vogel, S., Ney, H. and Tillmann, C., (1996) *HMM-based Word Alignment in Statistical Translation.* In COLING-96, pp. 836-841, 1996.

Wang, Y. and Waibel, A. (1997) *Modeling with Structures in Statistical Machine Translation*. COLING-ACL 1997.

Watanabe, H., Kurohashi, S., Aramaki, E., (2000) *Finding Structural Correspondences from Bilingual Parsed Corpus for Corpus-based Translation*, Proc. of COLING-2000.

Wu, D., (1995a) *An Algorithm For Simultaneously Bracketing Parallel Texts by Aligning Words.* ACL-95: 33rd Annual Meeting of the Assoc. for Computational Linguistics, 244-251. Cambridge, MA: Jun. 1995.

Wu, D., (1995b) *Grammarless Extraction of Phrasal Translation Examples from Parallel Texts*. TMI-95, Sixth International Conference on Theoretical and Methodological Issues in Machine Translation, v2, 354-372. Leuven, Belgium: Jul. 1995.