Inducing Lexico-Structural Transfer Rules from Parsed Bi-texts

Benoit Lavoie, Michael White, and Tanya Korelsky
CoGenTex, Inc.
840 Hanshaw Road
Ithaca, NY 14850, USA
benoit,mike,tanya@cogentex.com

Abstract

This paper describes a novel approach to inducing lexico-structural transfer rules from parsed bi-texts using syntactic pattern matching, statistical co-occurrence and error-driven filtering. We present initial evaluation results and discuss future directions.

1 Introduction

This paper describes a novel approach to inducing transfer rules from syntactic parses of bi-texts and available bilingual dictionaries. The approach consists of inducing transfer rules using the four major steps described in more detail below: (i) aligning the nodes of the parses; (ii) generating candidate rules from these alignments; (iii) ordering candidate rules by co- occurrence; and (iv) applying error-driven filtering to select the final set of rules.

Our approach is based on lexico-structural transfer (Nasr et. al., 1997), and extends recent work reported in (Han et al., 2000) about Korean to English transfer in particular. Whereas Han et al. focus on high quality domain-specific translation using handcrafted transfer rules, in this work we instead focus on automating the acquisition of such rules.

Our approach can be considered a generalization of syntactic approaches to example-based machine translation (EBMT) such as (Nagao, 1984; Sato and Nagao, 1990; Maruyama and Watanabe, 1992). While such approaches use syntactic transfer examples during the actual transfer of source parses, our approach instead uses syntactic transfer examples to induce general transfer rules that can be compiled into a transfer dictionary for use in the actual translation process.

Our approach is similar to the recent work of (Meyers et al., 1998) where transfer rules are also derived after aligning the source and target nodes of corresponding parses. However, it also differs from (Meyers et al., 1998) in several important points. The first difference concerns the content of parses and the resulting transfer rules; in (Meyers et al., 1998), parses contain only lexical labels and syntactic roles (as arc labels), while our approach uses parses containing lexical labels, syntactic roles, and any other syntactic information provided by parsers (tense, number, person, etc.). The second difference concerns the node alignment; in (Meyers et al., 1998), the alignment of source and target nodes is designed in a way that preserves node dominancy in the source and target parses, while our approach does not have such restriction. One of the reasons for this difference is due to the different language pairs under study; (Meyers et al., 1998) deals with two languages that are closely related syntactically (Spanish and English) while we are dealing with languages that syntactically are quite divergent, Korean and English (Dorr, 1994). The third difference is in the process of identification of transfer rules candidates; in (Meyers et al., 1998), the identification is done by using the exact tree fragments in the source and target parse that are delimited by the alignment, while we use all source and target tree sub-patterns matching a subset of the parse features that satisfy a customizable set of alignment

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2001		**00-00-2001 to 00-00-2001**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Inducing Lexico-Strutural Transfer Rules from Parsed Bi-texts	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **CoGenTex Inc,840 Hanshaw Road,Ithaca,NY,14850**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**8**	

constraints and attribute constraints. The fourth third difference is in the level of abstraction of transfer rules candidates; in (Meyers et al., 1998), the source and target patterns of each transfer rule are fully lexicalized (except possibly the terminal nodes), while in our approach the nodes of transfer rules do not have to be lexicalized.

Section 2 describes our approach to transfer rules induction and its integration with data preparation and evaluation. Section 3 describes the data preparation process and resulting data. Section 4 describes the transfer induction process in detail. Section 5 describes the results of our initial evaluation. Finally, Section 6 concludes with a discussion of future directions.

2 Overall Approach

In its most general form, our approach to transfer rules induction includes three different processes, data preparation, transfer rule induction and evaluation. An overview of each process is provided below; further details are provided in subsequent sections.

The *data preparation* process creates the following resources from the bi-texts:

- A training set and a test set of source and target parses for the bi-texts, post-processed into a syntactic dependency representation.
- A baseline transfer dictionary, which may include (depending upon availability) lexical transfer rules extracted from the bi-texts using statistical methods, lexical transfer rules from existing bilingual dictionaries, and/or handcrafted lexico-structural transfer rules.

The *transfer induction* process induces lexico-structural transfer rules from the training set of corresponding source and target parses that, when added to the baseline transfer dictionary, produce transferred parses that are closer to the corresponding target parses. The transfer induction process has the following steps:

- Nodes of the corresponding source and target parses are aligned using the baseline transfer dictionary and some heuristics based on the similarity of part-of-speech and syntactic context.
- Transfer rule candidates are generated based on the sub-patterns that contain the corresponding aligned nodes in the source and target parses.
- The transfer rule candidates are ordered based on their likelihood ratios.
- The transfer rule candidates are filtered, one at a time, in the order of the likelihood ratios, by removing those rule candidates that do not produce an overall improvement in the accuracy of the transferred parses.

The *evaluation process* has the following steps:

- Both the baseline transfer dictionary and the induced transfer dictionary (i.e., the baseline transfer dictionary augmented with the induced transfer rules) are applied to the test set in order to produce two sets of transferred parses, the baseline set and the (hopefully) improved induced set. For each set, the differences between the transferred parses and target parses are measured, and the improvement in tree accuracy is calculated.
- After performing syntactic realization on the baseline set and the induced set of transferred parses, the differences between the resulting translated strings and the target strings are measured, and the improvement in string accuracy is calculated.
- For a subset of the translated strings, human judgments of accuracy and grammaticality are gathered, and the correlations between the manual and automatic scores are calculated, in order to assess the meaningfulness of the automatic measures.

3 Data Preparation

3.1 Parsing the Bi-texts

In our experiments to date, we have used a corpus consisting of a Korean dialog of 4183 sentences and their English human translations. We ran off-the-shelf parsers on each half of the corpus, namely the Korean parser developed by Yoon et al. (1997) and the English parser developed by Collins (1997). Neither parser was trained on our corpus.

We automatically converted the phrase structure output of the Collins parser into the syntactic dependency representation used by our syntactic realizer, RealPro (Lavoie and Rambow, 1997). This representation is based on the deep-syntactic structures (DSyntS) of Meaning-Text Theory (Mel'čuk, 1988). The important features of a DSyntS are as follows:

- a DSyntS is an unordered tree with labeled nodes and labeled arcs;
- a DSyntS is lexicalized, meaning that the nodes are labeled with lexemes (uninflected words) from the target language;
- a DSyntS is a dependency structure and not a phrase- structure structure: there are no non-terminal nodes, and all nodes are labeled with lexemes;
- a DSyntS is a syntactic representation, meaning that the arcs of the tree are labeled with syntactic relations such as SUBJECT (represented in DSyntSs as `I`), rather than conceptual or semantic relations such as AGENT;
- a DSyntS is a deep syntactic representation, meaning that only meaning-bearing lexemes are represented, and not function words.

Since the output of the Yoon parser is quite similar, with the exception of its treatment of syntactic relations, we have used its output as is. The DSyntS representations for two corresponding Korean[1] and English sentences are illustrated in Figure 1.

In examining the outputs of the two parsers on our corpus, we found that about half of the parse pairs contained incorrect dependency assignments, incomplete lemmatization or incomplete parses. To reduce the impact of such parsing errors in our initial experiments, we have primarily focused on a higher quality subset of 1763 sentence pairs that were selected according to the following criteria:

- Parse pairs where the source or target parse contained more than 10 nodes were rejected, since these usually contained more parse errors than smaller parses.
- Parse pairs where the source or target parse contained non-final punctuation were rejected; this criterion was based on our observation that in most such cases, the source or target parses contained only a fragment of the original sentence content (i.e., one or both parsers only parsed what was on one side of an intra-sentential punctuation mark).

[1]Korean is represented in romanized format in this paper.

```
(S1) {i}    {Ci-To-Reul}     {Ta-Si}  {Po-Ra}.
     this + map-accusative + again +  look-imp
(D1) {po} [class=vbma ente={ra}] (
       s1 {ci-to} [class=nnin2 ppca={reul}] (
         s1 {i} [class=ande]
       )
       s1 {ta-si} [class=adco2]
     )
(S2) Look at the map again.
(D2) look [class=verb mood=imp] (
       attr at [class=preposition] (
         ii map [class=common_noun article=def]
       )
       attr again [class=adverb]
     )
```

Figure 1: Syntactic dependency representations for corresponding Korean and English sentences

We divided this higher quality subset into training and test sets by randomly choosing 50% of the 1763 higher quality parse pairs (described in Section 3.1) for inclusion in the training set, reserving the remaining 50% for the test set. The average numbers of parse nodes in the training set and test set were respectively 6.91 and 6.11 nodes.

3.2 Creating the Baseline Transfer Dictionary

In the general case, any available bilingual dictionaries can be combined to create the baseline transfer dictionary. These dictionaries may include lexical transfer dictionaries extracted from the bi-texts using statistical methods, existing bilingual dictionaries, or handcrafted lexico-structural transfer dictionaries. If probabilistic information is not already associated with the lexical entries, log likelihood ratios can be computed and added to these entries based on the occurrences of these lexical items in the parse pairs.

In our initial experiments, we decided to focus on the scenario where the baseline transfer dic-

```
@KOREAN:
    {po} [class=vbma] (
      s1 $X [ppca={reul}]
    )
@ENGLISH:
    look [class=verb] (
      attr at [class=preposition] (
        ii $X
      )
    )
@-2xLOG_LIKELIHOOD: 12.77
```

Figure 2: Transfer rule for English lexicalization and preposition insertion

```
@KOREAN:
    $X [class=vbma ente={ra}]
@ENGLISH:
    $X [class=verb mood=imp]
@-2xLOG_LIKELIHOOD: 33.37
```

Figure 3: Transfer rule for imperative forms

tionary is created from lexical transfer entries extracted from the bi-texts using statistical methods. To simulate this scenario, we created our baseline transfer dictionary by taking the lexico-syntactic transfer dictionary developed by Han et al. (2000) for this corpus and removing the (more general) rules that were not fully lexicalized. Starting with this purely lexical baseline transfer dictionary enabled us to examine whether these more general rules could be discovered through induction.

4 Transfer Rule Induction

The induced lexico-structural transfer rules are represented in a formalism similar to the one described in Nasr et al. (1997), and extended to also include log likelihood ratios. Figures 2 and 3 illustrate two entry samples that can be used to transfer a Korean syntactic representation for *cito-reul po-ra* to an English syntactic representation for *look at the map*. The first rule lexicalizes the English predicate and inserts the corresponding preposition while the second rule inserts the English imperative attribute. This formalism uses notation similar to the syntactic dependency notation shown in Figure 1, augmented with variable arguments prefixed with `$` characters.

4.1 Aligning the Parse Nodes

To align the nodes in the source and target parse trees, we devised a new dynamic programming alignment algorithm that performs a top-down, bidirectional beam search for the least cost mapping between these nodes. The algorithm is parameterized by the costs of (1) aligning two nodes whose lexemes are not found in the baseline transfer dictionary; (2) aligning two nodes with differing parts of speech; (3) deleting or inserting a node in the source or target tree; and (4) aligning two nodes whose relative locations differ.

To determine an appropriate part of speech cost measure, we first extracted a small set of parse pairs that could be reliably aligned using lexical matching alone, and then based the cost measure on the co-occurrence counts of the observed parts of speech pairings. The remaining costs were set by hand.

As a result of the alignment process, alignment id attributes (`aid`) are added to the nodes of the parse pairs. Some nodes may be in alignment with no other node, such as English prepositions not found in the Korean DSyntS.

4.2 Generating Rule Candidates

Candidate transfer rules are generated using three data sources:

- the training set of aligned source and target parses resulting from the alignment process;
- a set of alignment constraints which identify the subtrees of interest in the aligned source and target parses (Section 4.2.1);
- a set of attribute constraints which determine what parts of the aligned subtrees to include in the transfer rule candidates' source and target patterns (Section 4.2.2).

The alignment and attribute constraints are necessary to keep the set of candidate transfer rules manageable in size.

4.2.1 Alignment constraints

Figure 4 shows an example alignment constraint. This constraint, which matches the structural patterns of the transfer rule illustrated in Figure 2, uses the `aid` alignment attribute to indicate that

```
@KOREAN:
    $X1 [aid=$1] (
      $R1 $X2 [aid=$2]
    )
@ENGLISH:
    $Y1 [aid=$1] (
      $R2 $Y2 (
        $R3 $Y3 [aid=$2]
      )
    )
```

Figure 4: Alignment constraint

in a Korean and English parse pair, any source and target sub-trees matching this alignment constraint (where $X1 and $Y1 are aligned or have the same attribute `aid` values and where $X2 and $Y3 are aligned) can be used as a point of departure for generating transfer rule candidates. We suggest that alignment constraints such as this one can be used to define most of the possible syntactic divergences between languages (Dorr, 1994), and that only a handful of them are necessary for two given languages (we have identified 11 general alignment constraints necessary for Korean to English transfer so far).

4.2.2 Attribute constraints

Attribute constraints are used to limit the space of possible transfer rule candidates that can be generated from the sub-trees satisfying the alignment constraints. Candidate transfer rules must satisfy all of the attribute constraints. Attribute constraints can be divided into two types:

- *independent attribute constraints,* whose scope covers only one part of a candidate transfer rule and which are the same for the source and target parts;
- *concurrent attribute constraints,* whose scope extends to both the source and target parts of a candidate transfer rule.

The examples of an independent attribute constraint and of a concurrent attribute constraint are given in Figure 5 and Figure 6 respectively. As with the alignment constraints, we suggest that a relatively small number of attribute constraints is necessary to generate most of the desired rules for a given language pair.

Each node of a candidate transfer rule must have its relation attribute (relationship with its governor) specified if it is an internal node, otherwise this relation must not be specified:

e.g.

```
⊘ $X1 ( $R $X2 )
```

Figure 5: Independent attribute constraint

In a candidate transfer rule, inclusion of the lexemes of two aligned nodes must be done concurrently:

e.g.

```
$X [aid=$1]
```

and

```
$Y [aid=$1]
```

e.g.

```
⊘ [aid=$1]
```

and

```
⊘ [aid=$1]
```

Figure 6: Concurrent attribute constraint

4.3 Ordering Rule Candidates

In the next step, transfer rule candidates are ordered as follows: first, by their log likelihood ratios (Manning and Schutze, 1999: 172-175); second, any transfer rule candidates with the same log likelihood ratio are ordered by their specificity.

4.3.1 Rule ordering by log likelihood ratio

We calculate the log likelihood ratio, log λ, applied to a transfer rule candidate as indicated in Figure 7. Note that log λ is a negative value, and following (Manning and Schutze, 1999), we assign -2 log λ to the transfer rule. Note also that in the definitions of C_1, C_2, and C_{12} we are currently only considering one occurrence or co-occurrence of the source and/or target patterns per parse pair, while in general there could be more than one; in our initial experiments these definitions have sufficed.

4.3.2 Rule ordering by specificity

If two or more candidate transfer rules have the same log likelihood ratio, ties are broken by a specificity heuristic, with the result that more general rules are ordered ahead of more specific ones. The specificity of a rule is defined to be the following sum: the number of attributes found in the source and target patterns, plus 1 for each for

$$\log \lambda = \log L(C_{12}, C_1, p) + \log L(C_2 - C_{12}, N - C_1, p) - \log L(C_{12}, C_1, p_1) - \log L(C_2 - C_{12}, N - C_1, p_2)$$

where, not counting attributes `aid`,

- C_1 = number of source parses containing at least one occurrence of C's source pattern
- C_2 = number of target parses containing at least one occurrence of C's target pattern
- C_{12} = number of source and target parse pairs containing at least one co-occurrence of C's source pattern and C's target pattern satisfying the alignment constraints
- N = number of source and target parse pairs
- $P = C_2/N$;
- $P_1 = C_{12}/C_1$;
- $P_2 = (C_2 - C_{12})/(N - C_1)$;
- $L(k, n, x) = x^k(1 - x)^{n-k}$

Figure 7: Log likelihood ratios for transfer rule candidates

each lexeme attribute and for each dependency relationship. In our initial experiments, this simple heuristic has been satisfactory.

4.4 Filtering Rule Candidates

Once the candidate transfer rules have been ordered, error-driven filtering is used to select those that yield improvements over the baseline transfer dictionary. The algorithm works as follows. First, in the initialization step, the set of accepted transfer rules is set to just those appearing in the baseline transfer dictionary, and the current error rate is established by applying these transfer rules to all the source structures and calculating the overall difference between the resulting transferred structures and the target parses. Then, in a single pass through the ordered list of candidates, each transfer rule candidate is tested to see if it reduces the error rate. During each iteration, the candidate transfer rule is provisionally added to the current set of accepted rules and the updated set is applied to all the source structures. If the overall difference between the transferred structures and the target parses is lower than the current error rate, then the candidate is accepted and the current error rate is updated; otherwise, the candidate is rejected and removed from the current set.

```
@KOREAN:
    {po} [class=vbma ente={ra}] (
      s1 $X [ppca={reul}]
    )
@ENGLISH:
    look [class=verb mood=imp] (
      attr at [class=preposition] (
        ii $X
      )
    )
@-2xLOG_LIKELIHOOD: 11.40
```

Figure 8: Transfer rule for English imperative with lexicalization and preposition insertion

4.5 Discussion of Induced Rules

Experimentation with the training set of 882 parse pairs described in Section 3.1 produced 12467 source and target sub-tree pairs using the alignment constraints, from which 20569 transfer rules candidate were generated and 7565 were accepted after filtering. We expect that the number of accepted rules per parse pair will decrease with larger training sets, though this remains to be verified.

The rule illustrated in Figure 3 was accepted as the 65th best transfer rule with a log likelihood ratio of 33.37, and the rule illustrated in Figure 2 was accepted as the 189th best transfer rule candidate with a log likelihood ratio of 12.77. An example of a candidate transfer rule that was not accepted is the one that combines the features of the two rules mentioned above, illustrated in Figure 8. This transfer rule candidate had a lower log likelihood ratio of 11.40; consequently, it is only considered after the two rules mentioned above, and since it provides no further improvement upon these two rules, it is filtered out.

In an informal inspection of the top 100 accepted transfer rules, we found that most of them appear to be fairly general rules that would normally be found in a general syntactic-based transfer dictionary. In looking at the remaining rules, we found that the rules tended to become increasingly corpus-specific.

5 Initial Evaluation

5.1 Results

In an initial evaluation of our approach, we applied both the baseline transfer dictionary and the induced transfer dictionary (i.e., the baseline transfer dictionary augmented with the transfer rules induced from the training set) to the test half of the 1763 higher quality parse pairs described in Section 3.1, in order to produce two sets of transferred parses, the baseline set and the induced set. For each set, we then calculated tree accuracy recall and precision measures as follows:

Tree accuracy recall The tree accuracy recall for a transferred parse and a corresponding target parse is determined the by C/Rq, where C is the total number of features (attributes, lexemes and dependency relationships) that are found in both the nodes of the transferred parse and in the corresponding nodes in the target parse, and Rq is the total number of features found in the nodes of the target parse. The correspondence between the nodes of the transferred parse and the nodes of the target parse is determined with alignment information obtained using the technique described in Section 4.1.

Tree accuracy precision The tree accuracy precision for a transferred parse and a corresponding target parse is determined the by C/Rt, where C is the total number of features (attributes, lexemes and dependency relationships) that are found in both the nodes of the transferred parse and in the corresponding nodes in the target parse, and Rt is the total number of features found in the nodes of the transferred parse.

Table 1 shows the tree accuracy results, where the f-score is equally weighted between recall and precision. The results illustrated in Table 1 indicate that the transferred parses obtained using induction were moderately more similar to the target parses than the transferred parses obtained using the baseline transfer, with about 15 percent improvement in the f-score.

	Recall	Precision	F-Score
Baseline	37.77	46.81	41.18
Induction	55.35	58.20	55.82

Table 1: Tree accuracy results

5.2 Discussion

At the time of writing, the improvements in tree accuracy do not yet appear to yield appreciable improvements in realization results. While our syntactic realizer, RealPro, does produce reasonable surface strings from the target dependency trees, despite occasional errors in parsing the target strings and converting the phrase structure trees to dependency trees, it appears that the tree accuracy levels for the transferred parses will need to be higher on average before the improvements in tree accuracy become consistently visible in the realization results. At present, the following three problems represent the most important obstacles we have identified to achieving better end-to-end results:

- Since many of the test sentences require transfer rules for which there are no similar cases in the set of training sentences, it appears that the relatively small size of our corpus is a significant barrier to better results.
- Some performance problems with the current implementation have forced us to make use of a perhaps overly strict set of alignment and attribute constraints. With an improved implementation, it may be possible to find more valuable rules from the same training data.
- A more refined treatment of rule conflicts is needed in order to allow multiple rules to access overlapping contexts, while avoiding the introduction of multiple translations of the same content in certain cases.

6 Conclusion and Future Directions

In this paper we have presented a novel approach to transfer rule induction based on syntactic pattern co-occurrence in parsed bi-texts. In an initial evaluation on a relatively small corpus, we have

shown that the induced syntactic transfer rules from Korean to English lead to a modest increase in the accuracy of transferred parses when compared to the target parses. In future work, we hope to demonstrate that a combination of considering a larger set of transfer rule candiates, refining our treatment of rule conflicts, and making use of more training data will lead to further improvements in tree accuracy, and, following syntactic realization, will yield to significant improvements in end-to-end results.

Acknowledgements

We thank Richard Kittredge for helpful discussion, Daryl McCullough and Ted Caldwell for their help with evaluation, and Chung-hye Han, Martha Palmer, Joseph Rosenzweig and Fei Xia for their assistance with the handcrafted Korean-English transfer dictionary and the conversion of phrase structure parses to syntactic dependency representations. This work has been partially supported by DARPA TIDES contract no. N66001-00-C-8009.

References

Michael Collins. 1997. Three generative, lexicalised models for statistical parsing. In *Proceedings of the 35th Meeting of the Association for Computational Linguistics (ACL'97)*, Madrid, Spain.

Bonnie Dorr. 1994. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–635.

C. Han, B. Lavoie, M. Palmer, O. Rambow, R. Kittredge, T. Korelsky, N. Kim, and M. Kim. 2000. Handling structural divergences and recovering dropped arguments in a Korean-English machine translation system. In *Proceedings of the Fourth Conference on Machine Translation in the Americas (AMTA'00)*, Misin Del Sol, Mexico.

Benoit Lavoie and Owen Rambow. 1997. RealPro – a fast, portable sentence realizer. In *Proceedings of the Conference on Applied Natural Language Processing (ANLP'97)*, Washington, DC.

C. D. Manning and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

H. Maruyama and H. Watanabe. 1992. Tree cover search algorithm for example-based translation. In *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI'92)*, pages 173–184.

Y. Matsumoto, H. Hishimoto, and T. Utsuro. 1993. Structural matching of parallel texts. In *Proceedings of the 31st Annual Meetings of the Association for Computational Linguistics (ACL'93)*, pages 23–30.

Igor Mel'čuk. 1988. *Dependency Syntax*. State University of New York Press, Albany, NY.

A. Meyers, R. Yangarber, R. Grishman, C. Macleod, and A. Moreno-Sandoval. 1998. Deriving transfer rules from dominance-preserving alignments. In *Proceedings of COLING-ACL'98*, pages 843–847.

Makoto Nagao. 1984. A framework of a mechanical translation between Japenese and English by analogy principle. In A. Elithorn and R. Banerji, editors, *Artificial and Human Intelligence*. NATO Publications.

Alexis Nasr, Owen Rambow, Martha Palmer, and Joseph Rosenzweig. 1997. Enriching lexical transfer with cross-linguistic semantic features. In *Proceedings of the Interlingua Workshop at the MT Summit*, San Diego, California.

S. Sato and M. Nagao. 1990. Toward memory-based translation. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, pages 247–252.

Fei Xia and Martha Palmer. 2001. Converting dependency structures to phrase structures. In *Notes of the First Human Language Technology Conference*, San Diego, California.

J. Yoon, S. Kim, and M. Song. 1997. New parsing method using global association table. In *Proceedings of the 5th International Workshop on Parsing Technology*.