A Paradigm for Non-head-driven Parsing: Parameterized Message-Passing

Bonnie Dorr
Department of Computer Science
University of Maryland
College Park, MD 20742
bonnie@cs.umd.edu

Dekang Lin
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada, R3T 2N2
lindek@cs.umanitoba.ca

Jye-hoon Lee
Department of Computer Science
University of Maryland
College Park, MD 20742
jlee@cs.umd.edu

Sungki Suh
Department of Linguistics
University of Maryland
College Park, MD 20742
sksuh@wam.umd.edu

Abstract

The parsing component of previous principle-based parsers are inefficient since they tend to adopt a generate-and-test paradigm. We combine the benefits of a message-passing paradigm with the benefits of a parametric approach in the implementation of a parser that avoids overgeneration and is easily ported to multiple languages. The algorithm has been implemented in C++ and successfully tested on well-known, translationally divergent sentences. We are currently incorporating the parser into a machine translation (MT) system called PRINCITRAN.

Keywords: cross-linguistic parsing, message passing, parameterization, machine translation

Introduction

This paper presents an efficient, implemented approach to cross-linguistic parsing based on Government-Binding (GB) Theory (Chomsky, 1981; Haegeman, 1991; van Riemsdijk and Williams, 1986). One of the drawbacks to alternative GB-based parsing approaches is that they generally adopt a filter-based paradigm. These approaches typically generate all possible candidate structures of the sentence that satisfy $\overline{\text{X}}$ theory, and then subsequently apply filters in order to eliminate those structures that violate GB principles. (See, for example, (Abney, 1989; Correa, 1991; Dorr, 1991; Fong, 1991; Frank, 1990).) The current approach provides an alternative to filter-based designs which avoids these difficulties by applying principles to *descriptions* of structures without actually building the structures themselves. Our approach is similar to that of (Lin, 1993; Lin and Goebel, 1993) in that structure building is deferred until the descriptions satisfy all principles; however, the current approach differs in that it provides a parameterization mechanism along the lines of (Dorr, 1993a) that allows the system to be ported to languages other than English. We focus particularly on the problem of processing head-final languages such as Korean.

We are currently incorporating the parser into a machine translation (MT) system called PRINCITRAN.[1] In general, parsers of existing principle-based interlingual MT systems are exceedingly inefficient since they tend to adopt the filter-based paradigm. We combine the benefits of the message-passing paradigm with the benefits of the parameterized approach to build a more efficient, but easily extensible system, that will ultimately be used for MT. The algorithm has been implemented in C++ and successfully tested on well-known, translationally divergent sentences.

It is important to point out that the efficiency of the system is not simply a side effect of using an efficient programming language (i.e., C++), but that the algorithm is inherently efficient, independent of the programming language used for the implementation. The key to the efficiency of the algorithm is that it is based on a system that is already provably fast (Lin and Goebel, 1993). A formal, worst-case analysis reveals that the original CFG parsing algorithm is $O(|G|n^3)$ (Lin and Goebel, 1993), where n is the length of the input sentence and $|G|$ is the size of the grammar (the number of occurrences of non-terminal symbols in the set of grammar rules). This is an improvement of a factor of $|G|$ over standard CFG algorithms such as the Earley parser (Barton et al., 1987). This complexity measure is based on a serial simulation of parallel distributed message passing.

The extended version of the system differs only in that it includes attribute values, constraints, and movements, thus allowing a wider range of phenomena to be handled while still retaining the

[1] The name PRINCITRAN is derived from the names of two systems, UNITRAN (Dorr, 1993b) and PRINCIPAR (Lin, 1993).

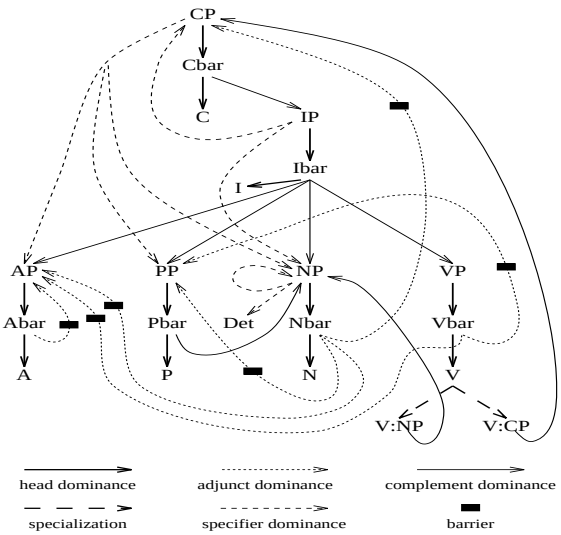

Figure 1: Network Representation of English Grammar

efficiency of the original algorithm. The complexity of this version has not yet been formally determined; however, in this paper we will provide an intuitive sense of the inherent speed of this version by describing a set of experiments that show that the average time does not grow exponentially to the sentence length.[2]

The next section presents a general framework for parsing by message passing. The following section describes our implementation of GB principles as attribute-value constraints in the message-passing framework. We then present the parameterization framework, demonstrating the feasibility of handling cross-linguistic variation within the message-passing framework. A technique for automatic precompilation of parameter settings is then described. In the section on Time Test Comparisons, we compare the efficiency of the parser to that of the original CFG algorithm (Lin and Goebel, 1993) as well as Tomita's algorithm (Tomita, 1986) on a test suite of representative sentences. We conclude with a discussion about the implications of this approach and its applicability to the problem of multi-language MT; preliminary results on translationally divergent sentences in Korean and English are presented.

Message Passing Paradigm

There has been a great deal of interest in exploring new paradigms of parsing, especially non-traditional parallel architectures for natural language processing (Cottrell, 1989; Waltz and Pollack, 1985; Small, 1981; Selman and Hirst, 1985; Abney and Cole, 1985; Jones, 1987; Kempen and Vosse, 1989) (among many others). Recent work (Stevenson, 1994) provides a survey of symbolic, non-symbolic, and hybrid approaches. Stevenson's model comes the closest in design to the current principle-based message-passing model in that it uses distributed message passing as the basic underlying mechanism and it encodes GB principles directly (i.e., there are precise correspondences between functional components and linguistic principles). However, the fundamental goals of the two approaches are different: Stevenson's objective concerns the modeling of human processing behavior and producing a single parse at the end. Her system incorporates a number of psycholinguistic-based processing mechanisms for handling ambiguity and making attachment decisions.[3] Our model, on the other hand, is more concerned with efficiency issues, broad scale coverage, and cross-linguistic applicability; we produce all possible parse alternatives wherever disambiguation requires extra-sentential information.

We view Stevenson's approach as a special case of our own in that it would be possible to program our system so that it provides identical output for the same phenomena. The benefit of Stevenson's approach is that the linguistic constraints fall out from the machinery whereas our own design requires the constraints to be explicitly encoded in the processing data structures. On the other hand, a critical deficiency of Stevenson's design is that it is incapable of handling head-final languages unless the machinery undergoes a massive modification.

Our approach provides a language-independent processing mechanism that accommodates structurally different languages (e.g., head-initial vs. head-final) with equally efficient run times and it does not require major modifications to the underlying processing mechanism as each language is added. The approach extends the GB-based message-passing design proposed by (Lin, 1993; Lin and Goebel, 1993). The grammar is encoded as a network of nodes that represent grammatical categories (e.g., NP, Nbar, N) or subcategories, such as V:NP (i.e., a transitive verb that takes an NP as complement). Figure 1 depicts a portion of the grammar network for English.

There are two types of links in the network: **subsumption links** (e.g., V to V:NP) and **dominance links** (e.g., Nbar to N). A dominance link

[2] Presumably, runtime differences that are due solely to differences in efficiency of programming language will result in a curve that is similar to others, but at a different scale. We will show that, in fact, this is the result we obtained.

[3] Stevenson's system currently handles structural ambiguity, but not lexical ambiguity; her research has focused almost entirely on the question of attachment decisions rather than on issues concerning selection of lexical items.

from α to β is associated with an integer id that determines the linear order between β and other categories immediately dominated by α, and a binary attribute to specify whether β is optional or obligatory.[4]

Input sentences are parsed by passing messages in the grammar network. The nodes in the network are computing agents that communicate with each other by sending messages in the reverse direction of the links. Each node locally stores a set of items. An item is a triplet that represents a $\overline{\text{X}}$ structure α: <surface-string, attribute-values, source-messages>, where *surface-string* is an integer interval [i,j] denoting the i'th to j'th word in the input sentence; *attribute-values* specifies syntactic features of the root node (β); and *source-messages* is a set of messages that represent immediate constituents of β and from which this item is combined. Each node has a completion predicate that determines whether an item at the node is "complete," in which case the item is sent as a message to other nodes.

When a node receives an item, it attempts to form new items by combining it with items from other nodes. Two items $<[i_1,j_1], A_1, S_1>$ and $<[i_2,j_2], A_2, S_2>$ can be combined if:

1. their surface strings are adjacent to each other: $i_2 = j_1+1$.
2. their attribute values A_1 and A_2 are unifiable.
3. the source messages come via different links.

The result of the combination is a new item:
$<[i_1,j_2], \text{unify}(A_1, A_2), S_1 \cup S_2>$.
More specifically, the parsing algorithm consists of the following steps:

Step 1: Lexical Look-up Retrieve the lexical entries for all the words in the sentence and create a lexical item for each word sense. A lexical item is a triple: $<[i,j], av_{self}, av_{comp}>$, where [i,j] is an interval denoting the position of the word in the sentence; av_{self} is the attribute values of the word sense; and av_{comp} is the attribute values of the complements of the word sense.

Step 2: Message Passing For each lexical item $<[i,j], av_{self}, av_{comp}>$, create an initial message $<[i,j], av_{self}, \emptyset>$ and send this message to the grammar network node that represents the category or subcategory of the word sense. When the node receives the initial message, it may forward the message to other nodes or it may combine the message with other messages and send the resulting combination to other nodes.

Step 3: Build a Shared Parse Forest When all lexical items have been processed, build a shared parse forest for the input sentence by tracing the origins of the messages at the highest node (CP or IP). The structure of the parse forest is similar to (Tomita, 1986) and (Billot and Lang, 1989), but extended to include attribute values.

The parse trees representing the input sentence are retrieved from the parse forest one by one. The next section explains how constraints attached to the nodes and links in the network ensure that the parse trees satisfy all GB principles.

Implementation of Principles

GB principles are implemented as **local constraints** attached to nodes and **percolation constraints** attached to links. All items at a node must satisfy the node's local constraint. A message can be sent across a link only if it satisfies the link's percolation constraint. We will discuss three examples to illustrate the general idea of how GB principles are interpreted as local and percolation constraints. See (Lin, 1993) for more details.

$\overline{\text{X}}$ Theory

The central idea behind $\overline{\text{X}}$ theory is that a phrasal constituent has a layered structure. Every phrasal constituent is considered to have a head ($\text{X}^0 \equiv \text{X}$), which determines the properties of the phrase containing it. A phrase potentially contains a complement, resulting in a one-bar level ($\overline{\text{X}} \equiv$ Xbar) projection; it may also contain a specifier (or modifier), resulting in a double-bar level ($\overline{\overline{\text{X}}} \equiv$ XP) projection. The phrasal representation assumed in the current framework is the following:

(1) [$_{\text{XP}}$ Specifier [$_{\text{Xbar}}$ X Complement]]

We implement the relative positioning of Specifier, Complement, and Head constituents by means of dominance links as shown in Figure 1. In addition, adjuncts are associated with the Xbar level by means of an adjunct-dominance link in the grammar network.

Trace Theory

A trace represents a position from which some element has been extracted.[5] The main constraint of Trace Theory is the Subjacency Condition, which prohibits movement across "too many" barriers. (The notion of "too many" is specified on a per-language basis as we will see shortly.)

An attribute named `barrier` is used to implement this principle. A message containing the attribute value `-barrier` is used to represent an $\overline{\text{X}}$ structure containing a position out of which a

[4] For the purpose of readability, we have omitted integer id's in the graphical representation of the grammar network; link ordering is indicated instead by the starting points of links, e.g., C precedes IP under Cbar since the link leading to C is to the left of the link leading to IP.

[5] A trace is represented as t_i, where i is a unique index referring to an antecedent.

Feature	Significance
+ca	the head is a case assigner
-ca	the head is not a case assigner
+govern	the head is a governor
-govern	the head is not a governor
-cm	an NP m-commanded by the head needs case marking

Figure 2: Case Theory Attribute Values

Node	Local Constraint
P	assign +ca to every item
V	assign +ca to items with -passive
I	assign +ca to items with tense attribute

Figure 3: Constraints on Case Assignment

wh-constituent has moved, but without yet crossing a barrier. The value `+barrier` means that the movement has already crossed one barrier. Certain dominance links in the network are designated as barrier links (indicated in Figure 1 by solid rectangles). The Subjacency condition is implemented by the percolation constraints attached to the barrier links, which blocks any message with `+barrier` and changes `-barrier` to `+barrier` (i.e., it allows the message to pass through).

Case Theory

Case theory requires that every NP be assigned abstract case. The Case Filter (Chomsky, 1981) rules out sentences containing an NP with no case. Case is assigned structurally to a syntactic position governed by a case assigner. Roughly, a preposition assigns Oblique Case to a prepositional object NP; a transitive verb assigns Accusative Case to a direct object NP; and tensed Infl(ection) assigns Nominative Case to a subject NP.

The implementation of case theory in our system is based on the following attribute values: `ca, govern, cm` (see Figure 2). The attribute values `+ca` and `+govern` are assigned by local constraints to items representing phrases whose heads are case assigners (e.g., tensed I) and governors (e.g., V), respectively (see Figure 3). The Case Filter is then applied by checking the cooccurrence of the attributes `ca, govern,` and `cm`.

Implementation of Parameters

While the principles described in the previous section are intended to be language independent, the structure of the grammar network in figure 1 is too language-specific to be applicable to languages other than English. The most obvious flaw is that each dominance link is associated with an integer id that imposes a linear order on phrasal constituents. In this particular network, all phrasal heads precede their complements. However, in head-final languages such as Korean, the reverse order is required. In order to capture this distinction, we incorporate the parameterization approach of (Dorr, 1993a) into the message-passing framework so that the network can be automatically generated on a per-language basis.

The reason the message-passing paradigm is so well-suited to a parameterized model of language parsing is that, unlike head-driven models of parsing, the main message-passing operation is capable of combining two nodes (in any order) in the grammar network. The result is that a head-final language such as Korean is as efficiently parsed as a head-initial language such as English. What is most interesting about this approach is that model is consistent with experimental results (see, for example, (Suh, 1993)) which suggest that constituent structure is computed prior to the appearance of the head in Korean.

The remainder of this section describes our approach to parameterization of each subtheory of grammar described in the last section; we conclude with a summary of the syntactic parameter settings for English and Korean.

$\overline{\mathrm{X}}$ Theory

$\overline{\mathrm{X}}$ theory assumes that a constituent order parameter is used for specifying phrasal ordering on a per-language basis:

(2) **Constituent Order:** The relative order between the head and its complement can vary, depending on whether the language in question is (i) head-initial or (ii) head-final.

The structure above represents the relative order observed in Korean, i.e., the head-final parameter setting (ii). In English, the setting of this parameter is (i). This ordering information is encoded in the grammar network by virtue of the relative ordering of integer id's associated with network links. Other types of parameters encoded in the grammar network are those pertaining to basic categories (i.e., possible replacements for X in (1) above), pre-terminal categories (e.g., determiner), potential specifiers, and adjuncts for each basic category.

Trace Theory

In general, adjunct nodes are considered to be barriers to movement. However, Korean allows the head noun of a relative clause to be construed with the empty category across more than one intervening adjunct node (CP), as shown in the following:

(3) [$_{\mathrm{CP}}$ [$_{\mathrm{CP}}$ t_1 t_2 kyengyengha-ten] hoysa$_2$-ka manghayperi-n] Bill$_1$-un yocum uykisochimhay issta

[CP [CP managed-Rel] company-Nom is bankrupt-Rel] Bill-Top these days depressed is

'Bill, who is such a person that the company he was managing has been bankrupt, is depressed these days'

The subject NP 'Bill' is coindexed with the trace in the more deeply embedded relative clause. If we assume, following (Chomsky, 1986), that relative clause formation involves movement from an inner clause into an outer subject position, then the grammaticality of the above example suggests that the Trace theory must be parameterized so that crossing more than one barrier is allowed in Korean. Our formulation of this parametric distinction is as follows:

(4) **Barriers:** (i) only one crossing permitted; (ii) more than one crossing permitted.

In English the setting be (i); in Korean the setting would be (ii).

Case Theory

In general, it is assumed that the relation between a case assigner and a case assignee is biunique. However, this assumption rules out so-called multiple subject constructions, which are commonly used in Korean:

(5) John-i phal-i pwureciessta
-Nom arm-Nom was broken
'John is in the situation that his arm has been broken'

The grammaticality of the above example suggests that Nominative Case in Korean must be assigned by something other than tensed Infl. Thus, we parameterize Case Assignment as follows:

(6) **Case Assignment:** Accusative case is assigned by transitive V; Nominative case is assigned by (i) tensed Infl; (ii) IP predication.

In a biunique case-assignment language such as English, the setting for Nominative case assignment would be (i); in Korean, the setting would be (i) and (ii).

Summary of Parameter Settings for English and Korean

We adopt the syntactic parameters of (Dorr, 1993a) as shown in Figure 4 for English. Following this paradigm, our analysis of Korean has revealed the parameter settings shown in Figure 5. The remainder of this paper will focus on how we automatically generate the grammar networks for English and Korean from $\overline{X}$ parameter settings (i.e., Basic Categories, Pre-terminals, Constituent Order, Specifiers, and Adjunction).

Theory	Parameter	English Setting
$\overline{X}$	Basic Categories	C I V N P A
	Pre-terminals	ADV NUM DET
	Constituent Order	I: SPEC-INITIAL HEAD-INITIAL N: SPEC-INITIAL HEAD-INITIAL C: SPEC-INITIAL HEAD-INITIAL A: HEAD-INITIAL P: HEAD-INITIAL V: HEAD-INITIAL
	Specifiers	I: NP C: NP, PP, AP N: NP (+gen), DET
	Adjunction	Ibar: PP (left), ADV (right), NP (right) Vbar: PP (right), ADV (left) Nbar: AP (left), PP (right), CP (right), NUM (left) Abar: ADV (left)
Trace	Barriers	only one crossing permitted
Case	Case Assignment	Nominative: tensed Infl; Accusative: transitive V

Figure 4: Syntactic Parameter Settings for English

Parameter Compilation Algorithm

Our algorithm for automatic precompilation of the parameter settings into a grammar network consists of dynamic generation of code that is read as data by the message-passing parsing system. The following steps are used to construct the code:

1. Use `defcategory` to define:
   - X0 nodes (using Basic Categories and Constituent Order parameters)
   - Bar1 and Bar2 nodes (projecting from X0 nodes)
   - Pre-terminal nodes (using Pre-terminals parameter)
2. Use `defeature` to define:
   - Specifier links (using Specifiers parameter)
   - Adjunct links (using Adjunction parameter)

Figure 6 shows a portion of the code that is automatically generated from the Korean parameter settings in figure 5. Figure 7 shows the network that is generated as a result of executing this algorithm using the Korean $\overline{X}$ parameter settings.

Results of Time Test Comparisons

As a broad-coverage system, PRINCITRAN is very efficient. The parsing component (PRINCIPAR) processes real world sentences of length 20 to 30 words from sources such as Wall Street Journal within a couple of seconds. The complexity of the current version of the system has not yet been formally determined. However, we have demonstrated that the efficiency of the system is not purely a result of using an efficient programming language (C++); this has been achieved by running experiments that compare the performance

Theory	Parameter	Korean Setting
$\overline{X}$	Basic Categories	C I V N P A
	Pre-terminals	ADV DEM
	Constituent Order	I: SPEC-INITIAL HEAD-FINAL N: SPEC-INITIAL HEAD-FINAL C: HEAD-FINAL A: HEAD-FINAL P: HEAD-FINAL V: HEAD-FINAL
	Specifiers	I: NP N: NP (+gen), DEM
	Adjunction	Ibar: PP (left), ADV (left), NP (left) Vbar: PP (left), ADV (left) Nbar: AP (left), PP(+gen) (left), CP (left) Abar: ADV (left)
Trace	Barriers	more than one crossing permitted
Case	Case Assignment	Nominative: tensed Infl, IP predication; Accusative: transitive V

Figure 5: Syntactic Parameter Settings for Korean

```
(defcategory I Bar0Node
      (default-atts ((cat i) -head-initial)))
(defcategory N Bar0Node
      (default-atts ((cat n) -head-initial)))
(defcategory V Bar0Node
      (default-atts ((cat v) -head-initial)))
(defcategory P Bar0Node
      (default-atts ((cat p) -head-initial)))
(defcategory Ibar Bar1Node (default-atts ((cat i))))
(defcategory Nbar Bar1Node (default-atts ((cat n))))
(defcategory Vbar Bar1Node (default-atts ((cat v))))
(defcategory Pbar Bar1Node (default-atts ((cat p))))
(defcategory IP Bar2Node (default-atts ((cat i))))
(defcategory NP Bar2Node (default-atts ((cat n))))
(defcategory VP Bar2Node (default-atts ((cat v))))
(defcategory PP Bar2Node (default-atts ((cat p))))
(defcategory DEM DEMNode (default-atts ((cat DEM))))
(defeature IP ISpec (default-atts (+spec-initial)) (NP spec))
(defeature NP NSpec (default-atts (+spec-initial))
      (NP spec (att-filter +genitive)) (DEM spec))
(defeature Nbar Nadj
      (PP adjunct left (att-filter +genitive) (+optional))
      (AP adjunct left (+optional))
      (CP adjunct left (+optional)))
(defeature Abar Aadj (ADV adjunct left (+optional)))
```

Figure 6: Automatically Generated Code for Korean Grammar Network

of the parser with two alternative CFG parsers. Since PRINCIPAR has a much broader coverage than these alternative approaches, the absolute measurements do not provide a complete picture of how these three systems compare. However, the most interesting point is that the trends of the three performance levels relative to sentence length are essentially the same. If PRINCIPAR had an **average** case complexity that was exponential relative to sentence length, but had only managed to be efficient because of the implementation language, the sentence length vs. performance curve would clearly be different from the curves for CFG parser which are known to have a **worst** case complexity that is polynomial relative to sentence length.

The two CFG parsers used for comparison are: a C implementation of Tomita's parser by Mark Hopkins (University of Wisconsin-Milwaukee, 1993). The test sentences are from (Tomita, 1986). There are 40 of them. The sentence lengths vary from 1 to 34 words with an average of 15.18. Both CFG parsers use the Grammar III in (Tomita, 1986, p.172–6), which contains 220 rules, and small lexicon containing only the words that appear in the test sentences. The lexicon in PRINCIPAR, on the other hand, contains about 90,000 entries extracted from machine readable dictionaries.

Tomita's parser runs about 10 times faster than PRINCIPAR; Lin and Goebel runs about twice as fast.[6] To make the parsing time vs. sentence length distribution of these three parsers more comparable, we normalized the curves; the parsing time of each of the CFG parses was multiplied by a constant so that they would have the same average time as PRINCIPAR. The adjusted timings are plotted in Figure 8. These results show that PRINCIPAR compares quite well with both CFG parsers.

Implications for Machine Translation

Our ultimate objective is to incorporate the parameterized parser into an interlingual MT system. The current framework is well-suited to an interlingual design since the linking rules between the syntactic representations given above and the underlying lexical-semantic representation are well-defined (Dorr, 1993b). We adopt the Lexical Conceptual Structure (LCS) of Dorr's work and use a parameter-setting approach to handle well known, translationally divergent sentences (Dorr, 1990).

A parametric approach to mapping between the interlingua and the syntactic structure in English, Spanish, and German is described by (Dorr, 1990; Dorr, 1993b). We present analogous examples here for English and Korean:[7]

[6] This is much slower than the result reported in (Lin and Goebel, 1993). This is because we are using a simulated version of the original system (i.e., PRINCIPAR without attribute values in the grammar and lexicon). The overhead for operations such as memory allocation/deallocation is higher in the simulated version, despite the fact that attribute values are not used.

[7] The examples presented here are not necessarily geared toward demonstrating the full capability of the parser, which handles many types of syntactic phenomena including complex movement types. (See

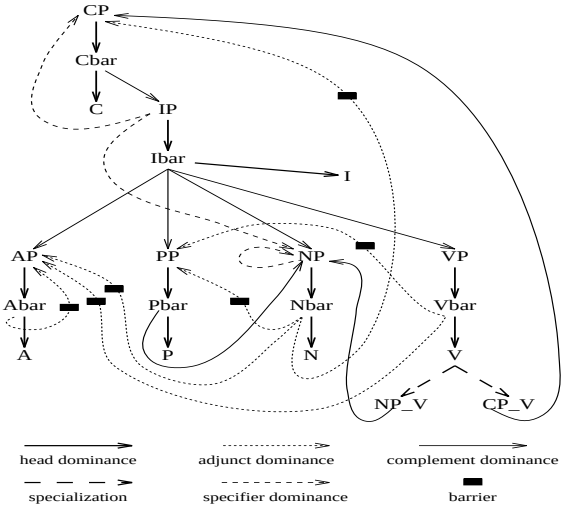

Figure 7: Network Representation of Korean Grammar

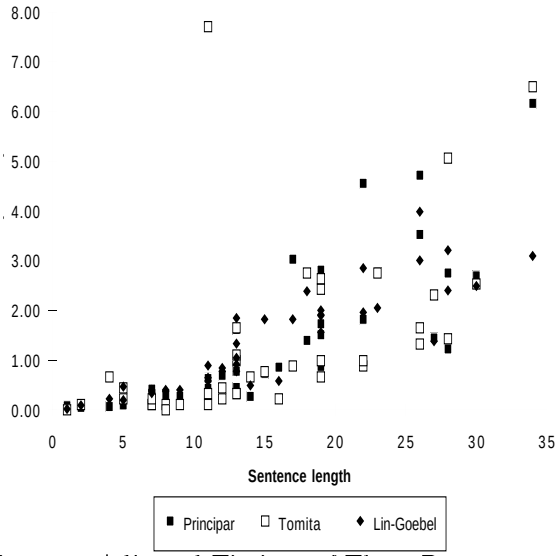

Figure 8: Adjusted Timings of Three Parsers

(7) **Structural Divergence:**
E: John married Sally
K: John-i Sally-wa kyelhonhayssta
-Nom -with married
'John married with Sally'

(8) **Conflational Divergence:**
E: John helped Bill
K: John-i Bill-eykey towum-ul cwuessta
-Nom -Dative help-Acc gave
'John gave help to Bill'

(9) **Categorial Divergence:**
E: John is fond of music
K: John-un umak-ul coahanta
-Nom music-Acc like
'It is John (who) likes music'

We ran the parameterized parser on both the English and Korean sentences shown here. The results shown in Figure 9 were obtained from running the program on a Sparcstation ELC. In general, the times demonstrate a speedup of 2 to 3 orders of magnitude over previous principle-based parsers on analogous examples such as those given in (Dorr, 1993b). Even more significant is the negligible difference in processing time between the two languages, despite radical differences in structure, particularly with respect to head-complement positioning. This is an improvement over previous parameterized approaches in which cross-linguistic divergences frequently induced timing discrepancies of 1-2 orders of magnitude due to the head-initial bias that underlies most parsing designs.

A preliminary investigation has indicated that the message-passing paradigm is useful for generation as well as parsing, thus providing a suitable framework for bidirectional translation. The algorithm for generation is similar to that of parsing in that both construct a syntactic parse tree over an unstructured or partially structured set of lexical items. The difference is characterized as follows: in parsing, the inputs are sequences of words and the output is a structure produced by combining two adjacent trees into a single tree at each processing step;[8] in generation, the inputs are a set of unordered words with dependency relationships derived from the interlingua (LCS). The generation algorithm must produce

	Parse	Time
E:	[CP [Cbar [IP [NP [Nbar [N John]]] [Ibar [VP [Vbar [V_NP married] [NP [Nbar [N Sally]]]]]]]]]	.15 sec.
K:	[CP [Cbar [IP [NP [Nbar [N John-i]]] [Ibar [VP [Vbar [PP [Pbar [NP [Nbar [N Sally]]] [P wa]]] [V_PP kyelhonhayssta]]]]]]]	.12 sec.
E:	[CP [Cbar [IP [NP [Nbar [N John]]] [Ibar [VP [Vbar [V_NP helped] [NP [Nbar [N Bill]]]]]]]]]	.10 sec.
K:	[CP [Cbar [IP [NP [Nbar [N John-i]]] [Ibar [VP [Vbar [PP [Pbar [NP [Nbar [N Bill]]] [P eykey]]] [NP [Nbar [N towum-ul]]] [V_PP_NP cwuessta]]]]]]]	.19 sec.
E:	[CP [Cbar [IP [NP [Nbar [N John-un]]] [Ibar [VP [Vbar [V_AP is] [AP [Abar [A fond] [PP [Pbar [P of] [NP [Nbar [N music]]]]]]]]]]]]	.12 sec.
K:	[CP [NP [0] [Nbar [N John-un]]] [Cbar [IP t [0] [Ibar [VP [Vbar [NP [Nbar [N umak-ul]]] [V_NP coahanta]]]]]]]	.07 sec.

Figure 9: Parameterized Parsing of English and Korean Divergence Examples

(Lin, 1993) for more details.) Rather, these examples are intended to illustrate that the parser is able to handle translationally contrastive sentences equally efficiently.

[8] The LCS composition routine described in (Dorr, 1992) derives the interlingua from the resulting syntactic representation.

structures that satisfy the same set of principles and constraints as the parsing algorithm.

Future Work and Conclusions

Three areas of future work are relevant to the current framework: (1) scaling up the Korean dictionary, which currently has only a handful of entries for testing purposes;[9] (2) the installation of a Kimmo-based processor for handling Korean morphology;[10] and (3) the incorporation of non-structural parameterization (i.e., parameters not pertaining to $\overline{X}$ theory such as Barriers and Case Assignment).

Summarizing, we have shown that the parametric message-passing design is an efficient and portable approach to parsing. We have automated the process of grammar-network construction, and have demonstrated that the system handles well-known, translationally divergent sentences. We expect that the current framework is suitable for bidirectional, interlingual MT since the message-passing paradigm may be used for generation as well as parsing.

Acknowledgements

Dekang Lin was supported by Natural Sciences and Engineering Research Council of Canada grant OGP121338. Bonnie Dorr and her students, Jye-hoon Lee and Sungki Suh, have been partially supported by the Army Research Office under contract DAAL03-91-C-0034 through Batelle Corporation, by the National Science Foundation under grant IRI-9120788 and NYI IRI-9357731, and by the Army Research Institute under contract MDA-903-92-R-0035 through Microanalysis and Design, Inc.

References

Abney, S. (1989). A computational model of human parsing. *Journal of Psycholinguistic Research*, 18(1):129–144.

Abney, S. and Cole, J. (1985). A government-binding parser. In *Proceedings of NELS, No. 16*, pages 1–17.

Barton, E. G., Berwick, R. C., and Ristad, E. S. (1987). *Computational Complexity and Natural Language*. MIT Press, Cambridge, MA.

Billot, S. and Lang, B. (1989). The structure of shared forests in ambiguous parsing. In *Proceedings of ACL–89*, pages 143–151.

Chomsky, N. (1981). *Lectures on Government and Binding*. Foris Publications, Cinnaminson, USA.

[9] Our English dictionary has 90K entries, constructed automatically by applying a set of conversion routines to OALD entries. We have begun negotiations with the LDC for the acquisition of a Korean MRD for which we intend to construct similar routines.

[10] The English dictionary used by the message-passing system contains all morphological derivatives of every word. This approach would be impractical for Korean since the morphology is significantly richer.

Chomsky, N. (1986). *Knowledge of Language: Its Nature, Origin and Use*. MIT Press, Cambridge, MA.

Correa, N. (1991). Empty categories, chains, and parsing. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 83–121. Kluwer Academic Publishers.

Cottrell, G. (1989). *A Connectionist Approach to Word Sense Disambiguation*. Morgan Kaufmann, Los Altos, CA.

Dorr, B. (1990). Solving thematic divergences in machine translation. In *Proceedings of ACL–90*, pages 127–134, University of Pittsburgh, Pittsburgh, PA.

Dorr, B. (1991). Principle-based parsing for machine translation. In Berwick, R., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 153–184. Kluwer Academic Publishers.

Dorr, B. (1992). The use of lexical semantics in interlingual machine translation. *Machine Translation*, 7(3):135–193.

Dorr, B. (1993a). Interlingual machine translation: a parameterized approach. *Artificial Intelligence*, 63(1&2):429–492.

Dorr, B. (1993b). *Machine Translation: A View from the Lexicon*. MIT Press, Cambridge, MA.

Fong, S. (1991). The computational implementation of principle-based parsers. In Berwick, B., Abney, S., and Tenny, C., editors, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 65–82. Kluwer Academic Publishers.

Frank, R. (1990). Licensing and tree adjoining grammar in gb parsing. In *Proceedings of ACL–90*, pages 111–118, University of Pittsburgh, Pittsburgh, PA.

Haegeman, L. (1991). *Introduction to Government and Binding Theory*. Basil Blackwell Ltd.

Jones, M. (1987). Feedback as a coindexing mechanism in connectionist architectures. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, pages 602–610.

Kempen, G. and Vosse, T. (1989). Incremental syntactic tree formation in human sentence processing: A cognitive architecture based on activation decay and simulated annealing. *Connection Science*, 1(3):273–290.

Lin, D. (1993). Principle-based parsing without overgeneration. In *Proceedings of ACL–93*, pages 112–120, Columbus, Ohio.

Lin, D. and Goebel, R. (1993). Context-free grammar parsing by message passing. In *Proceedings of PACLING–93*, Vancouver, BC.

Selman, B. and Hirst, G. (1985). A rule-based connectionist parsing system. In *Proceedings of the Seventh Annual Conference of the Cognitive Science Society*, pages 212–219.

Small, S. (1981). *Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding*. PhD thesis, University of Maryland, College Park, MD.

Stevenson, S. (1994). *A Competitive Attachment Model for Resolving Syntactic Ambiguities in Natural Language Parsing*. PhD thesis, University of Maryland, College Park, MD.

Suh, S. (1993). How to process constituent structure in head final languages: The case of korean. In *Proceedings of Chicago Linguistic Society, No. 29*.

Tomita, M. (1986). *Efficient Parsing for Natural Language*. Kluwer Academic Publishers, Norwell, Massachusetts.

van Riemsdijk, H. and Williams, E. (1986). *Introduction to the Theory of Grammar. Current Studies in Linguistics*. The MIT Press, Cambridge, Massachusetts.

Waltz, D. and Pollack, J. (1985). Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9:51–74.