Efficient Language Independent Generation from Lexical Conceptual Structures

Nizar Habash, Bonnie Dorr and David Traum
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742
phone: +1 (301) 405-6768
fax: +1 (301) 314-9658
{habash,bonnie,traum}@umiacs.umd.edu
WWW home page: http://umiacs.umd.edu/labs/CLIP

August 30, 2001

Abstract. This paper describes a system for generating natural-language sentences from an interlingual representation, Lexical Conceptual Structure (LCS). The system has been developed as part of a Chinese-English Machine Translation system; however, it is designed to be used for many other MT language pairs and Natural Language applications. The contributions of this work include: (1) Development of a language-independent generation system that maximizes efficiency through the use of a hybrid rule-based/statistical module; (2) Enhancements to an interlingual representation and associated algorithms for interpretation of multiply ambiguous input sentences; (3) Development of an efficient reusable language-independent linearization module with a grammar description language that can be used with other systems; (4) Improvements to an earlier algorithm for hierarchically mapping thematic roles to surface positions; (5) Development of a diagnostic tool for lexicon coverage and correctness and use of the tool for verification of English, Spanish, and Chinese lexicons. An evaluation of translation quality shows comparable performance with a commercial translation system. The generation system can also be straightforwardly extended to other languages and this is demonstrated and evaluated for Spanish.

Keywords: Generation, Machine Translation, Interlingua, Lexical Conceptual Structure, Language-Independent NLP

1. Introduction

This paper describes a system for generating natural-language sentences from an interlingual representation, Lexical Conceptual Structure (LCS). The system has been developed as part of a Chinese-English Machine Translation (MT) system; however, it is designed to be used for many other MT language pairs (e.g., Spanish and Arabic (Dorr et al., 1995)) and other natural language applications (e.g., cross-language information retrieval (Dorr et al., 2000)).

The contributions of this work include: (1) Development of a language-independent generation system that maximizes efficiency through the use of a hybrid rule-based/statistical module; (2) Enhancements to an interlingual representation and associated algorithm (Dorr, 1993b) for interpretation of multiply ambiguous input sentences; (3) Development of an efficient reusable language independent linearization module with a grammar description language that can be used with other systems; (4) Improvements to an earlier algorithm (Dorr et al., 1998) for hierarchically mapping thematic roles to surface positions; (5) Development of a diagnostic tool for lexicon coverage and correctness and use of the tool for verification of English, Spanish, and Chinese lexicons. An evaluation of translation quality shows comparable performance with a commercial translation system. The generation system can also be straightforwardly extended to other languages and this is demonstrated and evaluated for Spanish.

We will provide an overview of LCS-based MT and then describe our interlingual representation. We will then examine the generation component of our MT system in detail, followed by an evaluation of different aspects of our system.

2. Overview of LCS-based Machine Translation

One of the major challenges in natural language processing is the ability to make use of existing resources. Large differences in syntax, semantics, and ontologies of such resources create significant barriers to their usage in large-scale applications. A case in point is the wide range of "interlingual representations" used in machine translation and cross-language processing. Such representations are becoming increasingly prevalent, yet views vary widely as to what these should be composed of, varying from purely conceptual knowledge-representations, having little to do with the structure of language, to very syntactic representations, maintaining most of the idiosyncrasies of the source languages. In our generation system we make use of resources associated with two different (kinds of) interlingua structures: *Lexical Conceptual Structure* (LCS), and the *Abstract Meaning Representations* (AMR) used at USC/ISI (Langkilde and Knight, 1998a). The two representations serve different but complementary roles in the translation process. The deeper lexical-semantic expressiveness of LCS is essential for language independent lexical selection that transcends translation divergences (Dorr, 1993a). The shallower yet mixed semantic-syntactic nature of AMRs makes it easier to use directly for target-language realization.

The use of two representations in generation mirrors the use of two representations on the analysis side of the MT system, in which a parsing output is passed to a semantic-composition module; the target-language AMR is analogous to the source-language parse tree. (See Figure 1.) The Composition module takes the source-language parse tree and creates a deeper semantic representation (the LCS) using a source-language lexicon. In generation, the Decomposition module performs a reverse step that uses a target-language lexicon to create the hierarchical word and feature structure, a "parse-like" AMR. The linearization module flattens an AMR into a sequence of words. Because of the ambiguity inherent in all of the involved modules from the parser to the lexicons, multiple sequences are created. We use the statistical Extraction module of the generation system Nitrogen (Langkilde and Knight, 1998a; Langkilde and Knight, 1998b) to select among alternative outputs, using n-gram probabilities of target-language word sequences.

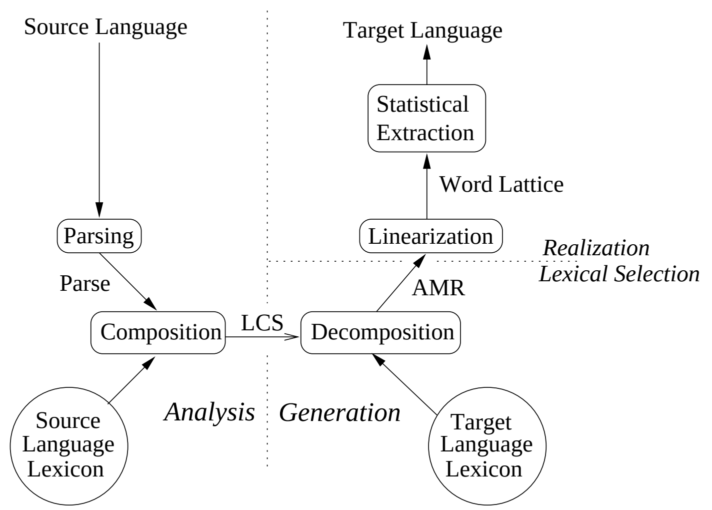

Figure 1. LCS-based Machine Translation

3. Lexical Conceptual Structure

Linguistic knowledge in the lexicon covers a wide range of information types, such as verbal subcategorization for events (e.g., that a transitive verb such as "hit" occurs with an object noun phrase), featural information (e.g., that the direct object of a verb such as "frighten" is animate), thematic information (e.g., that "John" is the agent in "John hit the ball"), and lexical-semantic information (e.g., that spatial verbs such as "throw" are conceptually distinct from verbs of possession such as "give"). By modularizing the lexicon, we treat each information type separately, thus allowing us to vary the degree of dependence on each level.

The most intricate component of lexical knowledge is the lexical-semantic information, which is encoded in the form of Lexical Conceptual Structure (LCS) as formulated by Dorr (Dorr, 1993b; Dorr, 1994) based on work by Jackendoff (Jackendoff, 1983; Jackendoff, 1990; Jackendoff, 1996). LCS is a compositional abstraction with language-independent properties that transcend structural idiosyncrasies. This representation has been used as the interlingua of several projects such as UNITRAN (Dorr, 1993a) and MILT (Dorr, 1997a).

Formally, an LCS is a directed graph with a root. Each node is associated with certain information, including a *type*, a *primitive* and a *field*. The type of an LCS node is one of *Event, State, Path, Manner, Property* or *Thing*. There are two general classes of primitives: *closed class* (also called *structural primitives*, e.g., `CAUSE, GO, BE, TO`) and *open class* primitives (also called *constants*, e.g., `john+, reduce+ed, jog+ingly`). Suffixes such as `+, +ed, +ingly` are markers of open class primitives, signaling also the type of the primitive (thing, property, event, etc.). We distinguish between the structural primitive GO and the constant `go+ingly`: the first appears in many lexical entries but the second appears only in specific lexical entries such as the one for the English verb "go". Examples of fields include `Locational`, `Possessional`, and `Identificational`. Structurally, an LCS node has zero or more LCS children. There are three ways a child node relates to its parent: as a subject (maximally one), as an argument, or as a modifier.

An LCS captures the semantics of a lexical item through a combination of semantic structure (specified by the shape of the graph and its structural primitives and fields) and semantic content (specified through constants). The semantic structure of a verb is something the verb shares with a *semantic verb class* whereas the content is specific to the verb itself. For example, all the verbs in the semantic class of "Run" verbs have the same semantic structure but vary in their semantic content (for example, run, jog, walk, zigzag, jump, roll, etc.). Semantic verb classes were initially borrowed from the classification in English Verb Classes and Alternations (EVCA) (Levin, 1993). Our *LCS Verb Database* (LVD) extends EVCA by refining the class divisions[1] and defining the underlying meaning components of each class in the LCS

[1] Levin's original database contained 192 classes, numbering between 9.1 and 57; our refined version contains 492, with more specific identifiers such as "51.3.2.a.ii".

representation. LVD also provides a relation between Levin's classes and both thematic role information and hand-tagged WordNet synset numbers. The first public release of the *LCS Verb Database* is now available for research purposes (Dorr, 2001).

Consider the sentence *John jogged to school*. This can be fully represented (except for features such as tense, telicity, etc.) as follows, roughly corresponding to 'John moved (*location*) to the school in a jogging manner':

(1)
```
(event go loc
    (thing john+)
    (path to loc
       (thing john+)
       (position at loc (thing john+) (thing school+)))
    (manner jog+ingly))
```

The lexicon entry for one sense of the English verb 'jog' and the preposition 'to' are shown in Figure 2. These entries include the root form of the word, its semantic verb class and word sense(s) from WordNet (Fellbaum, 1998) (for the verbs), and most importantly, a *Root LCS* (RLCS) which is the uninstantiated LCS corresponding to the underlying meaning of the word entry in the lexicon.

The top node in the "jog" RLCS has the structural primitive `GO` in the locational field. Its subject is marked with a "*"; star-marked nodes must be filled recursively with other lexical entries during semantic composition. The restriction on this particular LCS node is that the filler must be of type `thing`. The number '2' in that node specifies the thematic role: in this case, *theme*. The second and third child nodes are in argument positions filled with the primitives `FROM` and `TO`. The numbers '3' and '5' stand for *source* particle and *goal* particle respectively. The numbers '4' and '6' stand for *source* and *goal*. Figure 3 contains a list of variable numbers with their associated thematic roles. The second argument in the "jog" RLCS is the substructure `(to loc ...)` that unifies with the RLCS for the preposition "to". This secondary RLCS itself has a star-marked argument that must be instantiated with a `thing` such as "school".

The field `:THETA_ROLES` specifies the set of thematic roles appearing in the RLCS entry. Theta roles preceded by an underscore (`_`) are obligatory; whereas roles proceeded by a comma (`,`) are optional. Parentheses indicate that the corresponding phrases must necessarily be headed by a preposition. Sometimes the specific preposition is provided inside the parentheses. The roles are ordered in a canonical order that reflects their relative surface order: first available role is subject; second is object; etc.

```
(DEFINE-WORD
  :DEF_WORD "jog"
  :CLASS "51.3.2.a.ii"
  :THETA_ROLES "_th,src(),goal()"
  :WN_SENSE (01315785 01297547)
  :LANGUAGE ENGLISH
  :LCS
   (event go loc (* thing 2)
     ((* path from 3) loc (thing 2)
        (position at loc (thing 2) (thing 4)))
     ((* path to 5) loc (thing 2)
        (position at loc (thing 2) (thing 6)))
     (manner jog+ingly 26))
  :VAR_SPEC ((3 :optional) (5 :optional)))

(DEFINE-WORD
  :DEF_WORD "to"
  :LANGUAGE ENGLISH
  :LCS (path to loc
         (thing 2)
         (position in loc (thing 2) (* thing 6))))
```

Figure 2. Lexicon Entries for *jog* and *to*

The field `:WN_SENSE` links the entry to its corresponding WordNet synset. The Lexicon entries use WordNet 1.6 senses (Fellbaum, 1998; Miller and Fellbaum, 1991). The variable specifications (indicated here as `:VAR_SPEC`) assign the arguments headed by `FROM` and `TO` an `:optional` status. Other possible variable specifications that appear in our lexicon include `:obligatory`, `:promote`, `:demote`, `:EXT` *(external)*, `:INT` *(internal)* and `:conflated` (see (Dorr, 1993a) for more details).

The current English lexicon contains over 11000 RLCS entries such as those in Figure 2 (see also Figure 8 later). These entries correspond to different senses of over 4000 verbs. Figure 4 compares four of the nine RLCS entries for the verb "run". These entries are classified by verb class. Verb-classes are used as templates to generate the RLCS entries of verbs in the class. For example, the lexical entry for "bake" in class 26.3 would be identical to the top RLCS entry shown in Figure 4, except that node 9 would instead contain the primitive `bake+ed` rather than `run+ed`.

As described in (Dorr, 1993b), the meaning of complex phrases is captured through a *composed LCS* (CLCS). A CLCS is constructed

#	Thematic Role	Definition
0		no thematic role assigned
1	AG	agent
2	TH ,EXP ,INFO	theme or experiencer or information
3	SRC()	source preposition
4	SRC	source
5	GOAL(), PRED()	goal or pred preposition
6	GOAL	goal
7	PERC()	perceived item particle
8	PERC	perceived item
9	PRED	identificational predicate
10	LOC()	locational particle
11	LOC	locational predicate
12	POSS	possessional predicate
13	TIME()	temporal particle preceding time
14	TIME	time for TEMP field
15	MOD-POSS()	possessional particle
16	MOD-POSS	possessed item modifier
17	BEN()	beneficiary particle
18	BEN	benefactive modifier
19	INSTR()	instrumental particle
20	INSTR	instrument modifier
21	PURP()	purpose particle
22	PURP	purpose modifier or reason
23	MOD-LOC()	location particle
24	MOD-LOC	location modifier
25	MANNER()	manner
26		reserved for conflated manner
27	PROP	event or state
28	MOD-PROP	event or state
29	MOD-PRED()	identificational particle
30	MOD-PRED	property modifier
31	MOD-TIME	time modifier

Figure 3. Inventory of Thematic Roles

(or *composed*) from several RLCS entries corresponding to individual words. The composition process starts with a parsed tree of the input sentence and maps syntactic leaf nodes into RLCS entries whose argument positions are filled with other RLCS entries. For example, the two RLCS entries we have seen already can compose together with the constants for "John" and "school" to give the CLCS for the sentence: *John jogged to school*, shown in (1). The star-marked node `(* path from 3)` is optional, and is left unfilled in this case. The same RLCS could also be used to compose different CLCS representations

26.3 Verbs of Preparing

```
(event cause (* thing 1)
  (event go ident (* thing 2)
    (path toward ident (thing 2)
      (position at ident (thing 2) (property run+ed 9))))
  ((* for 17) poss (*head*) (* thing 18)))
```

Example: *John ran the store for Mary.*
Other verbs: bake boil clean cook fix fry grill iron mix prepare roast roll run wash ...

47.7.a Meander Verbs (from to)

```
(event go_ext loc (* thing 2)
  ((* path from 3) loc (thing 2) (position at loc (thing 2) (thing 4)))
  ((* path to 5) loc (thing 2) (position at loc (thing 2) (thing 6)))
  (manner run+ingly 26))
```

Example: *The river runs from the lake to the sea.*
Other verbs: crawl drop go meander plunge run sweep turn twist wander ...

47.5.1.b Swarm Verbs (Locational)

```
(event act loc (* thing 2)
   ((* position [at] 10) loc (thing 2) (thing 11))
   (manner run+ingly 26))
```

Example: *The dogs run in the forest.*
Other verbs: bustle crawl creep run swarm swim teem ...

51.3.2.a.i Run Verbs - (Locational,Theme only)

```
(event go loc (* thing 2)
  ((* path from 3) loc (thing 2) (position [at] loc (thing 2) (thing 4)))
  ((* path to 5) loc (thing 2) (position [at] loc (thing 2) (thing 6)))
  (manner run+ingly 26))
```

Example: *The horse ran into the field from the barn.*
Other verbs: climb crawl fly jog jump leap race run swim walk ...

Figure 4. RLCS entries for "run" in 4 different semantic verb classes

(in combination with other RLCS entries) to produce sentences like *John jogged from home* or *John jogged from home to school.*

A CLCS can also be decomposed on the generation side in different ways depending on the RLCS entries from the target language. Figure 5 uses a compressed graphic representation of LCS to visually compare three different decompositions in three languages of a single CLCS. The CLCS generated can be paraphrased as *John caused himself to go to the inside of a room in a forceful manner*

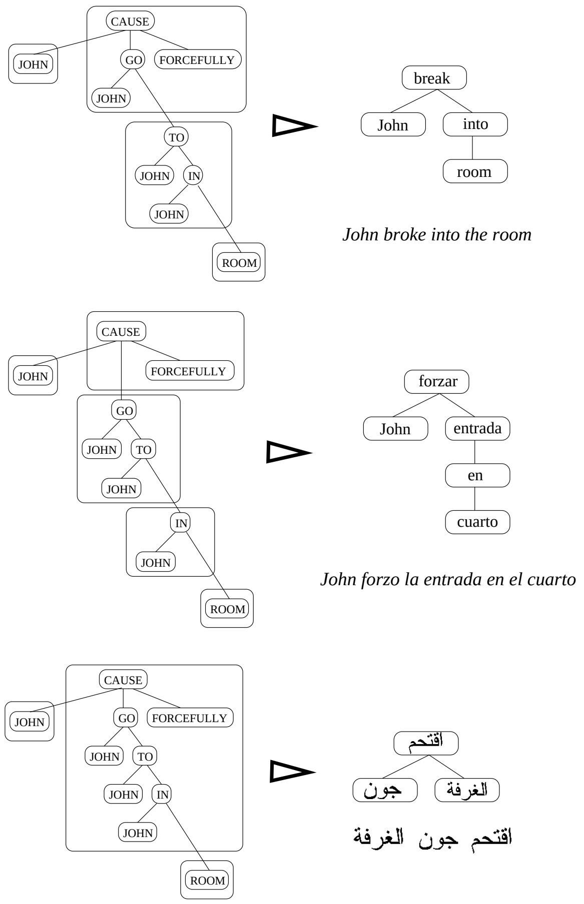

Figure 5. Different CLCS Decompositions into English, Spanish and Arabic

The input to the generation component is a text-representation of a CLCS in a format called *longhand*. It is equivalent to the form shown in (1), but makes certain information more explicit and regular (at the price of increased verbosity). The Longhand CLCS can either be a fully language-neutral interlingua representation, or one which still incorporates some aspects of the source-language interpretation process. This latter may include grammatical features on LCS nodes, but also nodes, known as *functional nodes*, which correspond to words in the source language but are not LCS-nodes themselves, serving merely as place-holders for feature information. Examples of these nodes include punctuation markers, coordinating conjunctions, grammatical aspect markers, and determiners.

An important extension of the LCS input language is the in-place representation of ambiguous sub-trees as a *possibles* node—denoted `:possibles`—which has the various possibilities represented as its own children. For example, the following structure (with some aspects elided for brevity) represents a node that could be one of three possibilities. In the second one, the root of the sub-tree is a functional node, passing its features to its child, `country+`:

```
(2) (:possibles
        (middle+ (country+ (developing+/p)))
        (functional (postposition among)
                    (country+ (developing+/p)))
        (china+ (country+ (developing+/p))))
```

It is important to point out that in our Chinese-English Translation project, sentences were not quite as simple as the examples used so far to explain the LCS approach. Figure 6 displays a CLCS from our machine translation system that was derived from the Chinese sentence in (3).

(3) 在 第 21 届 东新澳
in cardinalizer 21 session SEA-Singapore-Australia
中央银行组织 行长 研讨会 上，
central-bank-organization chief seminar at ,
中国人民银行 副 行长 殷介炎 就 “
chinese-peoples-bank deputy chief YinJieYan concerning ”
资本 大量 流入 情况 下 宏观 经济
capital large-amount influx situation beneath macro economic
政策 的 协调 ” 问题 发表 意见
policy DE agreement ” issue express opinions

At the 21st Southeast Asia-Singapore-Macao Central Bank Organization Presidents' Symposium, vice president of the People's Bank of China Yin Jieyan expressed his opinion on "coordination of macro-economic policy with a large capital inflow"

Figure 6 hides the ambiguity in the CLCS by only showing a single possibility when many occur. However, ambiguous nodes do indicate the number of the possibilities through the small black boxes under the node. For example, in Figure 6, the top node has four distinct possibilities corresponding to the verbs *issue*, *publish*, and *announce* (two instances of the latter). The number of distinct possible CLCS representations is 128. The average number of nodes per CLCS in this example is about 50. Compare these figures to those for the example in Figure 5: zero ambiguity, one CLCS, and ten nodes.

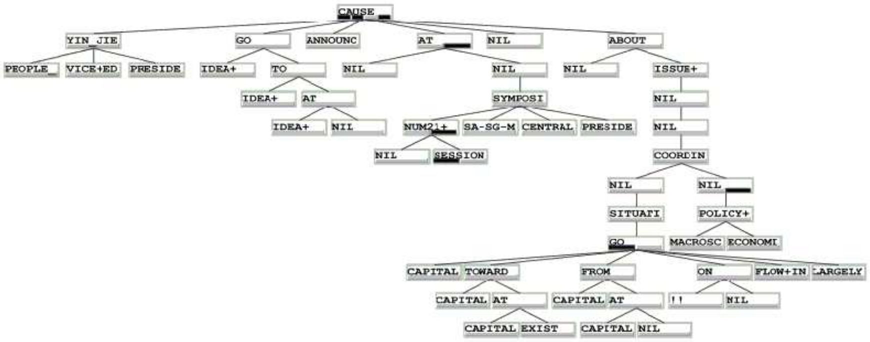

Figure 6. Large-scale CLCS

The rest of the examples in this paper will refer to the less complex CLCS for the Chinese sentence in (4).

(4) 美 单方 削减 中国 纺织品 出口 配额
US unilateral reduce China textile_product export quota

The United States unilaterally reduced the China textile export quota

The representation for this example is shown in (5) below, which roughly corresponds to "The United States caused the quota (modified by China, textile and export) to go identificationally (or transform) towards being at the state of being reduced." This LCS is presented without all the additional features, or type and function markers for sake of clarity. Also, it is actually one of eight possible LCS compositions produced by the analysis component from the input Chinese sentence.

(5)
```
(cause (united_states+)
  (go ident (quota+ (china+) (textile+) (export+))
    (to ident (quota+ (china+) (textile+) (export+))
      (at ident (quota+ (china+) (textile+) (export+))
                (reduce+ed))))
  (with instr (*HEAD*) nil)
  (unilaterally+/m))
```

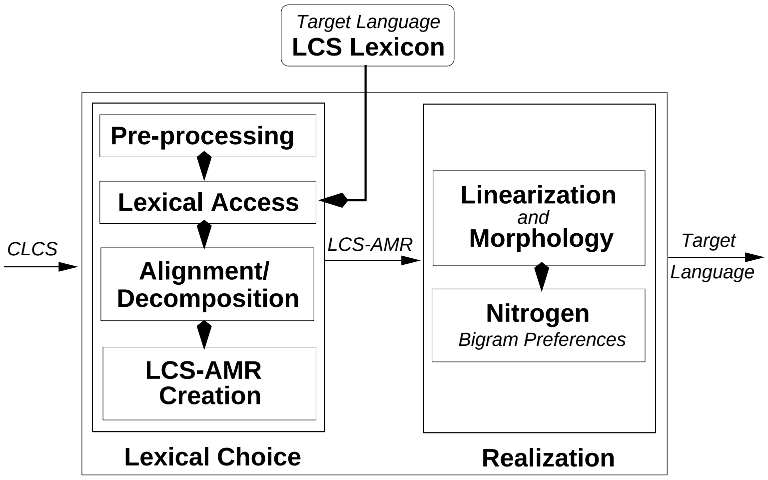

Figure 7. Generation System Architecture

4. The Generation System

The architecture of the generation system is presented in Figure 7, showing the main modules and sub-modules, and flow of information between them. In the generation process, the first phase, *Lexical Choice*, uses language-specific lexicons that relate lexical items in the target language to their LCS representation. The output of this phase is a target-language representation of the sentence in a modified form of the Abstract Meaning Representation (AMR) interlingua called LCS-AMR. The second phase, *Realization*, first handles the linearization and morphology to generate lattices of target-language sequences from the LCS-AMR and then statistically extracts preferred sequences using a bigram language model. For linearization, we use our own language independent linearization engine, Oxygen (Habash, 2000). As for the statistical extraction (and morphological generation), we use the Nitrogen generation system, from ISI (Langkilde and Knight, 1998a; Langkilde and Knight, 1998b).

4.1. Lexical Choice

The first major component, divided into four pipelined sub-modules as shown in Figure 7, transforms a CLCS structure into an LCS-AMR structure. This new representation is a modified form of the AMR interlingua that uses words and features specific to the target language, and also includes syntactic and semantic information from the LCS representation that is relevant for realization.

4.1.1. *Pre-Processing*

The pre-processing phase converts the text input format into an internal graph representation for efficient access of components (with links for parents as well as children). This phase also removes extraneous source-language features. For example, it converts the CLCS in (2) to remove the functional node and promote `country+` to be one of the possible sub-trees. This involves a top-down traversal of the tree, including some complexities when functional nodes without children (which then assign features to their parents) are direct children of possibles nodes.

4.1.2. *Lexical Access*

The lexical access phase compares the internal CLCS form to the target language lexicon, *decorating* the CLCS tree with the RLCS entries of target-language words which are likely to match sub-structures of the CLCS. The matching between a given CLCS, and the target-language lexicon is potentially a complex process, given the large amount of structural similarity between the entries of the lexicon. For example, the differences between the RLCS entries for "run" and "bake" in class 26.6 would only be distinguished by looking down 5 nodes deep from the root (c.f., Figure 4 and the discussion of verb-classes above). In a previous version of the system, we represented the lexicon in a trie structure, so that individual entries were only consulted at appropriate points in the CLCS tree-traversal. This still proved a fairly complex and inefficient procedure given the large amount of places that complex structures can be embedded (e.g., complement events). Our current approach uses a two phase process, in which RLCS entries are first located based on the distinguishing information (e.g., `run+ed vs. bake+ed`) and then placed in the appropriate matching node (`CAUSE`) for later comparison.

The lexical access process thus proceeds as follows. In an off-line lexicon processing phase, each word in the target-language lexicon is stored in a hash-table, with each entry keyed on a *designated primitive* which would be a most distinguishing node in the RLCS. Information is also kept about how deep from the root of the RLCS this primitive's node is to be found. For example, the designated primitive for the RLCS entries corresponding to class 26.3 would be `run+ed` (or `bake+ed`), and the depth would be 5. On-line decoration then proceeds in two step process, recursively examining each node in the CLCS:

(6) (i) Look for RLCS entries stored in the lexicon under the CLCS node's primitive

(ii) Store retrieved RLCS entries at the node in the CLCS that matches the root of this RLCS (follow a number of parent links from the CLCS node corresponding to the depth of the designated primitive).

Figure 8 shows some of the English entries matching the CLCS in (5). For most of these words, the designated primitive is the only node in the corresponding LCS for that entry. For *reduce*, however, `reduce+ed` is the designated primitive. When traversing the CLCS nodes in (5), this entry will be retrieved at the `reduce+ed` node in step (6)i; it will be stored at the root node of (5) in accordance with step (6)ii.

```
(:DEF_WORD "reduce"
  :CLASS "45.4.a"
  :THETA_ROLES "_ag_th,instr(with)"
  :WN_SENSE (00154752 00162871 00163072 00163532)
  :LANGUAGE ENGLISH
  :LCS (event cause (* thing 1)
        (event go ident (* thing 2)
            (path toward ident (thing 2)
                (position at ident (thing 2) (reduce+ed 9))))
        ((* position with 19) instr (*head*) (thing 20)))
  :VAR_SPEC ((1 (animate +))))

(:DEF_WORD "United States" :LCS (thing united_states+ 0))

(:DEF_WORD "China" :LCS (thing china+ 0))

(:DEF_WORD "quota" :LCS (thing quota+ 0))

(:DEF_WORD "with"
  :LCS (position with instr (thing 2) (* thing 20)))

(:DEF_WORD "unilaterally"
  :LCS (manner unilaterally+/m 0))
```

Figure 8. Lexicon entries

4.1.3. *Alignment/Decomposition*

The heart of the lexical choice phase is the *decomposition* process. In this phase, we attempt to align RLCS entries selected by the lexical access portion with parts of the CLCS, to find a covering of the CLCS graph that satisfies the "full coverage constraint" of the original algorithm described in (Dorr, 1993b). Our algorithm differs from that in (Dorr, 1993b) in its inclusion of some extensions to handle the in-place ambiguity represented by the *possibles* nodes.

The algorithm recursively checks whether CLCS nodes match corresponding RLCS nodes coming from the lexical entries retrieved and stored in the previous phase. If significant incompatibilities are found, the lexical entry is discarded. If all (obligatory) nodes in the RLCS match against nodes in the CLCS, then the rest of the CLCS is recursively checked against other lexical entries stored at the remaining unmatched CLCS nodes.

A CLCS node matches an RLCS node, if the following conditions hold:

(7) (i) The primitives are the same (or the primitive for one is a wild-card, represented as `nil`)

(ii) The types (e.g., thing, event, state, etc.) are the same (or `nil`)

(iii) The fields (e.g., identificational, possessive, locational, etc) are the same (or `nil`)

(iv) The positions (e.g., subject, argument, or modifier) are the same

(v) All obligatory children of the RLCS node have corresponding matches (recursively invoking this same definition) to children of the CLCS

Star-marked nodes in an RLCS (nodes indicated with a "*", see also discussion above) require not just a match against the corresponding CLCS node, but also a match against another lexical entry. Thus, in (5), the node (`united_states+`) must match not only with the corresponding node from the RLCS for "reduce" in Figure 8 `(* thing 1)`, but also with the RLCS for "United States", `united_states`. The result is that some CLCS nodes must match multiple RLCS nodes.

Subject and argument children are obligatory unless specified as optional, whereas modifiers are optional unless specified as obligatory (see Figure 2 for an example of an *optional* marking). In the RLCS for "reduce" in Figure 8, the nodes corresponding to agent and theme (numbered 1 and 2, respectively) are obligatory, while the instrument (the node numbered 19) is optional. Thus, even though there is no matching lexical entry for node 20 ("*"-marked in the RLCS for "with"), the main

RLCS for "reduce" is allowed to match, though without any realization for the instrument.

A complexity in the algorithm occurs when there are multiple possibilities in a position in a CLCS. In this case, only one of these possibilities is required to match all the corresponding RLCS nodes in order for a lexical entry to match. In the case where some of these possibilities do not match any RLCS nodes (meaning there are no target-language realizations for these constructs), these possibilities can be pruned at this stage. On the other hand, ambiguity can also be introduced at the decomposition stage, if multiple lexical entries can match a single structure.

The result of the decomposition process is a match-structure indicating the hierarchical relationship between all lexical entries which, together, cover the input CLCS.

4.1.4. *LCS-AMR Creation*

The match structure resulting from decomposition is then converted into the appropriate input format used by the Nitrogen generation system. Nitrogen's input, Abstract Meaning Representation (AMR), is a labeled directed feature graph written using the syntax for the PENMAN Sentence Plan Language (Penman 1989). A BNF structural description of an AMR is shown in (8).

(8) AMR = <concept> | (<label> / <concept> {<role> <AMR>}*)

An AMR is either a basic concept such as |run|, |john| or |quickly| or a labeled instance of a concept that is modified by a set of feature-value pairs. Features, or *roles*, can be syntactic (such as `:subject`) or semantic (such as `:agent`). The basic notation `/` is used to specify an instance of a concept in a non-ambiguous AMR.

We have extended the AMR language to accommodate the thematic roles and features provided in the CLCS representation; the resulting representation is called an LCS-AMR. To distinguish the LCS terms from those used by Nitrogen, we mark most of the new roles with the prefix `:LCS-`. Figure 9 shows the LCS-AMR corresponding to the CLCS in (5), decomposed using the lexicon entries in Figure 8.

The LCS-AMR in Figure 9 can be read as an instance of the concept |reduce| whose category is a verb and is in the active voice. The concept |reduce| has two thematic roles related to it, an agent (:LCS-AG) and a theme (:LCS-TH); and it is modified by the concept |unilaterally|. The different roles modifying |reduce| come from different origins. The :LCS-NODE value comes directly from the unique node number in the input CLCS. The category, voice and telicity are derived from features

```
(a7537 / |reduce|
    :LCS-NODE 6253520
    :LCS-VOICE ACTIVE
    :CAT V
    :TELIC +
    :LCS-AG (a7538 / |United States|
        :LCS-NODE 6278216
        :CAT N)
    :LCS-TH (a7539 / |quota|
        :LCS-NODE 6278804
        :CAT N
        :LCS-MOD-THING (a7540 / |China|
            :LCS-NODE 6108872
            :CAT N)
        :LCS-MOD-THING (a7541 / |textile|
            :LCS-NODE 6111224
            :CAT N)
        :LCS-MOD-THING (a7542 / |export|
            :LCS-NODE 6112400
             :CAT N))
    :LCS-MOD-MANNER (a7543 / |unilaterally|
        :LCS-NODE 6279392
        :CAT ADV))
```

Figure 9. LCS-AMR

of the RLCS entry for the verb |reduce| in the English lexicon. The specifications *agent* and *theme* come from the RLCS representation of the verb *reduce* in the English lexicon as well, as can be seen by the node numbers 1 and 2, in the lexicon entry in Figure 8. The role :LCS-MOD-MANNER combines the fact that the corresponding AMR had a modifier role in the CLCS and because its type is a Manner.

We have additionally extended the AMR syntax in our system by providing the ability to specify an ambiguous AMR as an *instance-less* conglomeration of different AMRs; this is achieved by means of the special role `:OR`. For example, a variant of the LCS-AMR in Figure 9 in which the root concept is three way ambiguous would appear as in (9) (details below the root omitted).

(9) `(# :OR (# / |reduce| . . . )`

```
:OR (# / |cut| . . . )
:OR (# / |decrease| . . . ))
```

4.2. Realization

The LCS-AMR representation is then passed to the realization module, which uses the Nitrogen approach to generation. The strategy used by Nitrogen is to allow over-generation of possible sequences of target-language words from the ambiguous or under-specified AMRs and then decide amongst them based on bigram frequency. The interface between the linearization module and the statistical extraction module is a word lattice of possible renderings. The Nitrogen package offers support for both subtasks, linearization and statistical extraction. Initially, we used the Nitrogen grammar to do linearization. But complexities in recasting the LCS-AMR roles as standard AMR roles as well as efficiency considerations (that will be discussed later in detail) compelled us to create our own linearization engine for writing target-language grammars, Oxygen (Habash, 2000).

In this module, we force linear order on the unordered parts of an LCS-AMR. This is done by recursively calling grammar rules that create various phrase types (NP,PP, etc.) from aspects of the LCS-AMR. The result of the linearization phase is a word lattice specifying the sequence of words that make up the resulting sentence and the points of ambiguity where different generation paths may be taken. Example (10) shows the word lattice corresponding to the LCS-AMR in Figure 9.

```
(10) (SEQ (WRD "*start-sentence*" BOS)
          (WRD "united states" NOUN)
          (WRD "unilaterally" ADJ)
          (WRD "reduced" VERB)
          (OR (WRD "the" ART)
              (WRD "a" ART)
              (WRD "an" ART))
          (WRD "china" ADJ)
          (OR (SEQ (WRD "export" ADJ)
                   (WRD "textile" ADJ))
              (SEQ (WRD "textile" ADJ)
                   (WRD "export" ADJ)))
          (WRD "quota" NOUN)
          (WRD "*end-sentence*" EOS))
```

The keyword SEQ specifies that what follows is a list of sub-lattices in their correct linear order. The keyword OR specifies the existence of disjunctive paths for generation. In the above example, the noun 'quota' is given a disjunction of all possible determiners since its definiteness is not specified. Also, the relative order of the words 'textile' and 'export' is not resolved so both ordering possibilities are inserted into the lattice.

Finally, the Nitrogen statistical extraction module evaluates the different paths represented in the word lattice and orders the different word renderings using uni- and bigram frequencies calculated based on two years of the Wall Street Journal (Langkilde and Knight, 1998b). Example (11) shows Nitrogen's ordering of the sentences extracted from the lattice in (10).

(11) united states unilaterally reduced the china textile export quota.
united states unilaterally reduced a china textile export quota.
united states unilaterally reduced the china export textile quota.
united states unilaterally reduced a china export textile quota.
united states unilaterally reduced an china textile export quota.
united states unilaterally reduced an china export textile quota.

4.2.1. *Linearization Issues*

The unordered nature of siblings under an LCS-AMR node complicates the mapping between roles and their surface positions, yielding several interesting linearization issues. In this section, we look at some of the choices made for our English realizer for ordering linguistic constituents.

4.2.1.1. *Sentential Level Argument Ordering* Sentences are realized according to the pattern in (12). That is, first subordinating conjunctions, if any, then modifiers in the temporal field (e.g., "now", "in 1978"), then the subject, then most other modifiers, the verb (with collocations if any) then spatial modifiers ("up", "down"), then the indirect object and direct object, followed by prepositional phrases and relative clauses. Nitrogen's morphology component was also used, e.g., to give tense to the head verb. In the example above, since there was no tense specified in the input LCS, past tense was used on the basis of the telicity of the verb to give "reduced" in (10),(11).[2]

(12) (SubConj ,) (TempMod)* Sub (Mod)* V (coll) (SpaceMod)* (IObj) (Obj) (PP)* (RelS)*

[2] See (Dorr and Olsen, 1996) and (Olsen et al., 2001) for a detailed study on the use of telicity for tense and aspect realization.

4.2.1.2. *Thematic Role Ordering* Given the above general shape for a sentence, there is still an issue of which thematic role should be mapped to which argument positions. This situation is complicated by the lack of one-to-one mapping between a particular thematic role and an argument position. For example, a theme can be the subject in some cases and it can be the object in others or even an oblique. Observe *cookie* in (13).

(13) (i) John ate *a cookie* (object)

(ii) *the cookie* contains chocolate (subject)

(iii) she nibbled *at a cookie* (oblique)

To solve this problem, a thematic hierarchy is used to determine the argument position of a thematic role based on its cooccurence with other thematic roles. Several researchers have proposed different versions of thematic hierarchies (see (Jackendoff, 1972; Carrier-Duncan, 1985; Bresnan and Kanerva, 1989; Kiparsky, 1985; Larson, 1988; Giorgi, 1984; Wilkins, 1988; Nishgauchi, 1984; Alsina and Mchombo, 1993; Baker, 1989; Grimshaw and Mester, 1988)).[3] Ours differs from these in that it separates arguments (e.g., agent and theme) from obliques (e.g., location and beneficiary) and provides a more complete list of thematic roles (30 roles overall, see Figure 3) than those of previous approaches (maximum of 8 roles).

The final thematic hierarchy for arguments was extracted by analyzing subcategorization information in the `:THETA_ROLES` field for all the verbs in our English lexicon.

(14)
```
special case : ag {goal src ben} th
ext > ag > instr > th > perc >  Everything Else
```

Thus, in the case where a theme occurs alone, this role is mapped to the first argument position. If a theme and an agent occur, the agent is mapped to first argument position and the theme is mapped to second argument position. When an agent and theme occur with a third role that is either a goal, a source or a beneficiary, a middle inversion is invoked on the order. The pseudo-role `ext` is used when the `:VAR_SPEC` field in the lexical entry of a verb includes an `:EXT` marker indicating that the verb violates the normal thematic hierarchy. The `ext` marker refers to an externally marked thematic role such as the perceived *John* in $John_{perc}$ *pleases* $Mary_{th}$. As for the ordering of obliques, all possible permutations are generated. For the LCS-AMR in Figure 9,

[3] For an excellent overview and a comparison of different thematic hierarchies see (Levin and Rappaport Hovav, 1996).

the thematic hierarchy is what determines that the |united states| is the subject and |quota| is the object of the verb |reduce|. A more detailed discussion is available in (Dorr et al., 1998). We will return to discuss thematic hierarchies later in this paper when evaluating English and Spanish realization.

4.2.1.3. *NP Modifier Ordering* In most cases, our input CLCS representations had little hierarchical information about multiple modifiers of a noun. Our initial, brute force solution was to generate all permutations and depend on the existing statistical extraction (in Nitrogen) to decide amongst them. This technique worked well for noun phrases of about 6 words, but was too costly for larger phrases (of which there were several examples in our test corpus). We improved both the cost of permutation generation and the fluency of the top choices by ordering adjectives within classes, inspired by the adjective ordering scheme in (Quirk et al., 1985). Our classification scheme is shown in (15). Each adjective in the target-language lexicon was assigned to one of these classes.

(15) (i) Determiner (all, few, several, some, etc.)
 (ii) Most Adjectival (important, practical, economic, etc.)
 (iii) Age (old, young, etc.)
 (iv) Color (black, red, etc.)
 (v) Participle (confusing, adjusted, convincing, decided)
 (vi) Provenance (China, southern, etc.)
 (vii) Noun (Bank_of_China, difference, memorandum, etc.)
 (viii) Denominal (nouns made into adjectives by adding -al, e.g., individual, coastal, annual, etc.)

If multiple words fall within the same group, permutations are generated for them. This situation can be seen for the LCS-AMR in Figure 9 with the ordering of the modifiers of the word |quota|: |china|, |export| and |textile|. |china| fell within the Provenance class of modifiers which gives it precedence over the other two words. |export| and |textile|, on the other hand, fell in the Noun class and therefore both permutations were passed on to the statistical component. Without this ordering, more permutations would be given to the statistical component, which, in this case, would also get a less appropriate result: "Textile china export quota" rather than "china textile export quota."

4.2.2. *Oxygen: Linearization Implementation*

The linearization module is basically an implementation of a set of rules, a grammar, that governs the relative word ordering (syntax) and word form (morphology) of an LCS-AMR in the target language. We have used three different linearization modules, each improving on problematic aspects of the previous ones. We briefly look at each of these in turn.

4.2.2.1. *Nitrogen Linearization* The Nitrogen generation system provides its own linearization module. The approach used in this module is a declarative one where a linearization engine performs on-line interpretation of a linearization grammar. The grammar is written in a special grammar description language that utilizes two basic operations: *recast* and *linearize*. A *recast* transforms an AMR into another AMR based on features of the original AMR. One example of recasting is converting an AMR with thematic roles into an AMR with surface argument position through the use of a thematic hierarchy. The second operation, *linearize*, decomposes an AMR into linearly ordered constituents, recursively applying the grammar to each. The grammar description language provides tools for defining conditions on which to make decisions to recast and/or linearize an AMR.

The advantages of this declarative approach are reusability, easy extendibility and language independence. Its main drawback is speed. Another drawback for Nitrogen's linearization grammar is a limited and inflexible grammar formalism: First, conditions of application are limited to equality of concepts or existence of roles at the top level of an AMR only. Second, recasting operations are limited to adding feature-value pairs and introducing new nodes. And, finally, there is no mechanism to perform range-unbounded or computationally complex transformations such as, for example, multiplication or division to correctly format numbers in the target language. The first two issues necessitate writing multiple rules and cascading information in order to implement complex decisions, which in turn increases the size of the grammar and furthur reduces the performance speed. The third issue is simply impossible to implement with the current formalism. A deeper look at these issues is provided in (Habash, 2000).

4.2.2.2. *Procedural Linearization* To contrast with Nitrogen's declarative approach to linearization, we look at procedural implementations of linearization grammars. In these approaches, a programming language is used to implement the rules of the grammar. The main advantages of this approach are flexibility, power and speed. Having access to the full computing power of a programming language opens a lot of possibilities for efficient implementation. It also frees the linearizer's designer from the restrictions of a limited declarative grammar by providing access to the operating system, databases, the web, etc. However, a major disadvantage of this approach is that the linguistic knowledge is coupled with the program code. This hard-coding of grammar rules can make the system rather redundant, difficult to understand and debug, non-reusable and language specific.

4.2.2.3. *Towards Improved Generation: A Hybrid Approach* After exploring both approaches in our system, we adopted a hybrid implementation (declarative/procedural) that maximizes the advantages and minimizes the disadvantages of these paradigms. The result is the linearization module Oxygen.

Oxygen uses a linearization grammar description language to write declarative grammar rules which are then compiled into a programming language for efficient performance. Oxygen contains three elements: a linearization grammar description language (OxyL), an OxyL to Lisp compiler (oxyCompile) and a run-time support library (oxyRun). Except for Nitrogen's morphological generator submodule, all of the Oxygen components were built at our Lab. Target-language linearization grammars written in OxyL are compiled off-line into Oxygen Linearizers using oxyCompile (Figure 10).

Figure 10. Oxygen Compilation Step

Oxygen Linearizers are Lisp programs that require the oxyRun library of basic functions in order to execute (Figure 11). They take AMRs as input and create word lattices that are passed to a statistical extraction unit.

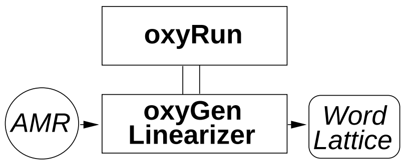

Figure 11. Oxygen Runtime Step

This implementation maximizes the advantages and minimizes the disadvantages inherent in the declarative and procedural paradigms. The separation of the linearization engine (oxyRun) from the linearization grammar (OxyL) combines in one system the best of two worlds: (1) the simplicity and focus of a declarative grammar with the power and efficiency of a procedural implementation; and (2) the efficiency of a resource-sharing implementation. Regarding this first point, the approach provides language independence and reusability since needs of the target language are only addressed in its specific OxyL grammar. Regarding the second point, the separation of language-specific code (compiled OxyL) from language-independent code (oxyRun) is efficient, especially when running multiple linearizers for different languages at the same time as in multilingual generation.

Moreover, Oxygen's linearization grammar description language, OxyL, is as powerful as a regular programming language but with a focus on linearization needs. This is accomplished through providing powerful recasting mechanisms for the most common needs of a linearization grammar and also by allowing embedding of code in a standard programming language (Lisp). This allows for efficient implementation of the more language specific realization problems (e.g., number formatting). OxyL linearization grammars are also simple, clear, concise and easily extendible. An example of the simplicity of OxyL grammars is the reduction of redundancy. For example, the handling of `:OR` ambiguities in each phrase rule (see, e.g., (9)) is hidden from the linearization grammar designer and is treated only in the compiler and support library. For a detailed presentation of OxyL's syntax, see (Habash, 2000).

Figure 12 presents a small OxyL grammar that is enough to linearize the LCS-AMR in Figure 9. In this grammar, the user-defined recast operation `&TH-order` uses the OxyL special hierarchical recast operator, `<!` to recast a small hierarchy of (agent, instrument, theme, source and goal) into subject and object positions. Rules `%S` and `%NP` linearize the different LCS-AMRs associated with specific roles. For example, `@subject` refers to the LCS-AMR paired with the role `:subject`. However, note that since `@lcs-mod-thing` matches three roles (i.e. `|china|`, `|export|` and `|quota|`, an ambiguous LCS-AMR is created and all its permutations are explored linearly. This is done at the engine level and is hidden from the user. A linearization can specify hard coded elements such as the determiners in `%NP`. The rule `:MainRule` determines which phrase-level rule to apply by considering the category, i.e. *part of speech*, of the LCS-AMR instance. This is accomplished using the automatically defined function `@CAT`, which returns the value associated with the field `:CAT` in the LCS-AMR. The sequence of `?? X -> Y -> Z` roughly corresponds to `if X then Y else Z`. The rule `:MainRule` is applied recursively until no more LCS-AMRs exist.

```
:Recast &TH-order
  (@this  <! ((:subject :object) /
        (:lcs-ag :lcs-instr :lcs-th :lcs-src :lcs-goal)))
:Rule %S
  (-> (@subject (@inst +- past) @object @lcs-mod-manner))

:Rule %NP
  (-> ((*or* "a" "an" "the") @lcs-mod-thing @inst))

:MainRule
  ((?? (&eq @cat V) -> (do %S (&TH-order @this))
    ?? (&eq @cat N) -> (do %NP)
    -> (@inst))
```

Figure 12. A Simple OxyL Grammar

The complete English Linearization grammar used in our system is much larger and more complex than the one shown in Figure 12. It includes 14 different phrase structure rules and four user-defined recast operations and it is about 300 lines of code long. The quality of the English output produced is evaluated in section 6.

5. Generation into Multiple Languages

While most of the effort has been spent on generation into English, in the context of Chinese-English translation, there has been some work using these components for generation into other languages. The main algorithms are all language independent, and retargeting the system for another languages involves only the following language-specific resources:

- Target-language LCS lexicon: a set of RLCS entries linking target language words to lexical conceptual structures, as described in Section 3.
- Target-language linearization grammar, in OxyL (see section 4.2.2).
- Word n-gram statistics for the target language, for use by lattice extractor.

In addition, the following pre-processing steps are also needed for creating a generation system for the target language:

- Hashing of target-language lexicon by "designated primitives", for on-line rapid retrieval (see section 4.1.2).
- Running oxyCompile on the linearization grammar to create an oyxGen Linearizer for the target-language (see section 4.2.2).
- Creation of a target-language n-gram database, for use by the statistical lattice extractor.

An important feature of a translation approach using an interlingua such as LCS is that the same grammar can be used for analysis and generation. Thus we already have a major component for a Chinese generation system. Likewise, large LCS lexicons also exist for other languages such as Spanish and Arabic (Dorr, 1997a).

We have also created a linearization component for Spanish, using a simple OxyL Spanish linearization grammar. This grammar concentrates on argument word order relative to the verb. It utilizes a thematic hierarchy mapping that is very similar to that of English. We avoided dealing with complex Spanish morphology by using the simple 'near-future' construction (*va a + INF*). One example is *alguien*$_{ag}$ *va a colocar algo*$_{th}$ *en algo*$_{goal}$ (someone$_{ag}$ will (is going to) place something$_{th}$ in something$_{goal}$). In addition to the lack of a complete phrase structure for parts of speech other than verbs, the Spanish linearization grammar doesn't handle Pro-drop or clitics. In principle, both phenomena can be handled with a recast rule that would fire after the thematic hierarchy recast. In the case of pro drop, it conjugates the verb and makes the subject null. And in the case of clitics, it adds a clitic that matches the gender and number of the object.

A similar but even less sophisticated linearization grammar was created to generate Chinese. A preliminary study showed some promising results as far as thematic hierarchy mapping. However Chinese seems to require more complex linearization rules and post-lexical selection manipulations especially for obliques.

We have not yet built an n-gram extractor for other languages. Preliminary evaluation of Spanish generation is given in Section 6.4.

6. Evaluation

The evaluation of machine translation and natural language generation systems is more of an art than a science. Evaluation of generation systems is difficult, because the ultimate criterion is translation quality, which can, itself, be difficult to judge, but, moreover, it can be hard to attribute specific deficits to the analysis phase, the lexical resources, or the generation system proper. A wide range of metrics and techniques have been developed over the last fifty years to assess 'how good' a system is. Evaluation schemas vary in their focus from addressing the system's interface to system scalability, faithfulness, space/time complexity, etc. Another dimension of variation is human versus automatic evaluation. Fully automatic evaluation, a task that is AI-complete *(i.e., encompassing all components of any system that would be deemed "intelligent")*, is the ultimate goal in the field.[4]

In (Church and Hovy, 1991), three categories of MT evaluation metrics are described: system-based, text-based and cost-based. System-based metrics count internal resources: size of lexicon, number of grammar rules, etc. These metrics are easy to measure although they are not comparable across systems. And their value is questionable since they are not necessarily related to utility.

Text-based metrics can be divided into sentence-based and comprehensibility-based. Sentence-based metrics examine the quality of single sentences out of context. These metrics include Accuracy, Fluency, Coherence, etc. Typically, subjects evaluating sentences are given a description of the metric with examples and are asked to rate the sentences on an x-point scale. These scales range from 3-point to 100-point. Comprehensibility metrics measure the comprehension or informativeness of a complete text composed of several sentences. The subjects are typically given questionnaires related to the processed text. Text-based metrics are much more related to utility than system-based metrics, but they are also much harder to measure. There are some automatic text-based evaluation metrics that measure the amount of post-editing needed for a sentence given a gold standard. These are variations on edit-distance, i.e., the number of deletions, additions or modifications measured by words or keystrokes per page or sentence. These techniques are not necessarily related to utility, however; it was recently shown that the smarter tree-based edit distance might actually correlate better to human judgement (Bangalore and Rambow, 2000).

Cost-based metrics evaluate a system on how much money/time it saves/costs per unit of text, say a page. These are secondary metrics since they depend on other metrics to evaluate how much post/pre-processing is necessary for a commercially functional system.

---

[4] For excellent surveys of machine translation evaluation metrics and techniques, see (Hovy, 1999; Hovy, 1999).

Table I. Oxygen Evaluation

	Procedural (Lisp)	Hybrid (Oxygen)	Declarative (Nitrogen)
Speed	+	**0**	-
Size	**0**	+	-
Expressiveness	+	+	-
Reusability	-	+	+
Readability/ Writability	-	+	-

6.1. Preliminary Evaluations

Different aspects of our system were evaluated in previous papers. In (Dorr et al., 1998) and also in more recent work (Habash and Dorr, 2001), the thematic hierarchy implementation proved successful and the generation was demononstrated to be a diagnostic tool for fixing the lexicon, algorithmic errors, and inconsistencies in English and Spanish output.

Another major evaluation addressed the general performance of the Oxygen module (Habash, 2000). Oxygen was evaluated based on speed of performance, size of grammar, expressiveness of the grammar description language, reusability and readability/writability. The evaluation context is provided by comparing an Oxygen linearization grammar for English to two other implementations, one procedural (using Lisp) and one declarative (using Nitrogen linearization module). The three comparable linearization grammars were used to calculate speed and size. Overall, Oxygen had the highest number of advantages and its only disadvantage, speed, ranked second to the lisp implementation (see Table I).

The generation component has also been used on a broader scale, generating thousands of simple sentences — at least one for each verb sense in the English LCS lexicon, creating sentence templates to be used in a Cross-Language information retrieval system (Dorr et al., 2000).

These previous evaluation efforts have been fairly coarse-grained and subjective. In the rest of this section, we report on both quantitative and qualitative evaluations of the system in several dimensions: Translation Quality, Coverage and Retargetability. Translation Quality can be seen as a system depth evaluation whereas Coverage is a system breadth evaluation. Retargetability focuses on the extendibility of the system to other languages.

6.2. Translation Quality Evaluation

The generation system has been used as part of a Chinese-English Translation system focusing on a corpus of 10 newspaper articles from Xinhua (Chinese People's Daily). The articles included eighty sentences that our translation system was able to parse, compose into LCS interlingua, and generate into English successfully. Although the number of sentences is small, some of them are quite complex, and represent a cross-section of the types of complex phenomena handled in a large-scale MT. To measure the translation quality of the system, we performed two human evaluations: one for Accuracy (Fidelity) and one for Fluency (Intelligibility). Both tests used a set of 25 sentences randomly selected from the 80 original Chinese sentences that completely pass our translation system. For comparison purposes, we also used a commercial Chinese-English translation system to translate these sentences: Chinese-English Systran 3.0 Professional edition. Thus, we both have absolute quality metrics and compare to state of the art translation.

The test suite is a 2x2 grid: (Accuracy, Fluency) x (ChinMT, Systran). The total number of subjects is 80, all of whom are native speakers of English. Each subject participated in only one of the four possible evaluations (e.g., ChinMT Accuracy or Systran Fluency) for all 25 sentences.[5] The evaluation was performed online using a web interface (see Figure 13).

6.2.1. *Accuracy Evaluation*

This evaluation measures the Accuracy or Fidelity of the translation system, i.e., how well a system preserves the meaning of the original text whether the target language is fluent or not. The subjects were given 25 pairs of sentences. Each pair consists of a human translation of the Chinese original and a machine translated version. Subjects were asked to rate the translation accuracy on a 5-point scale (see table II).[6]

A score of 5 is given where the content of the original sentence is fully conveyed (might need minor corrections). A score of 1 is given where

[5] To avoid order bias that can result from degradation in subject performance over time, each grid cell has two versions with different sentence display: (1 to 25) and (13 to 25, 1 to 12)

[6] Loosely based on Nagao's 7-point scale for Fidelity (Nagao, 1989)

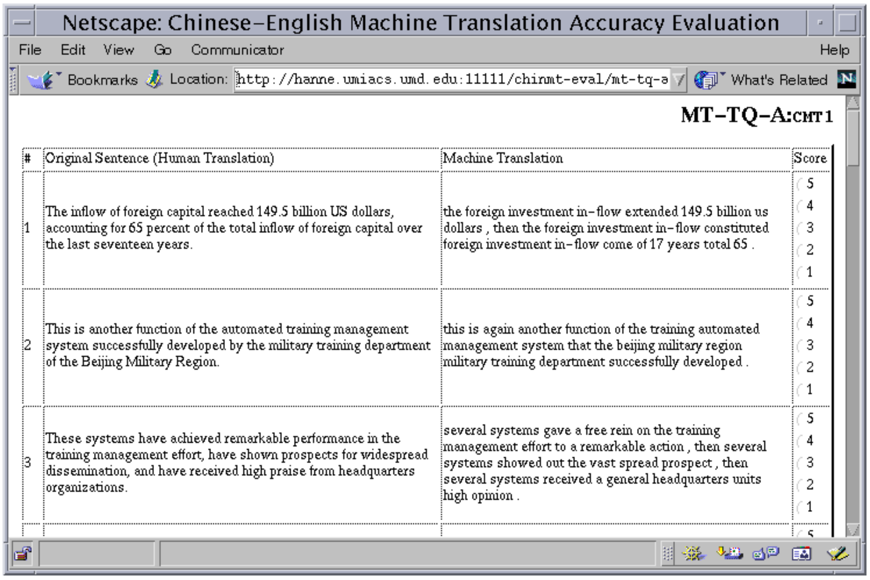

Figure 13. MT Evaluation Interface: Accuracy

Table II. Accuracy Criteria

5	contents of original sentence conveyed (might need minor corrections)
4	contents of original sentence conveyed BUT errors in word order
3	contents of original sentence generally conveyed BUT errors in relationship between phrases, tense, singular/plural, etc.
2	contents of original sentence not adequately conveyed, portions of original sentence incorrectly translated, missing modfiers
1	contents of original sentence not conveyed, missing verbs, subjects, objects, phrases or clauses

the content of the original sentence is not conveyed at all. An earlier pilot study indicated that subjects had a hard time with descriptions of the scale and preferred examples instead. Thus subjects were provided with a table containing two manually constructed examples per score to illustrate the idea behind the scoring scheme (see table III). Figure 13 displays a screen capture of the web interface showing the first three pairs of sentences in an Accuracy evaluation form.

Table III. Accuracy Scale Examples

Original Sentence (Human Translation)
The United States unilaterally reduced China's textile export quotas.

Machine Translation	**Score**
-united states reduced china's textile export quota unilaterally. -united states reduced china textile export quota unilaterally.	5
-united states cut china quota export textile unilaterally down. -united states china quota export textile cuts down unilaterally down.	4
-united states down to slash of a export textile Chinese the quotas. -some states united slash down reducingly down china textile of export ration.	3
-beautiful folk slashed porcelain export on own way. -state reduce quota.	2
-it cut. -china.	1

6.2.2. *Fluency Evaluation*

In the fluency evaluation, the subjects were given 25 machine translated sentences. The purpose of this evaluation is to measure the Fluency (or Intelligibility) of the translation system. Subjects were asked to rate the Fluency of machine translated sentences on a 5-point scale that is loosely based on Nagao's intelligibility scale metric (Nagao, 1989). The scale ranges from 5 (clear meaning, fluent sentence) to 1 (meaning absolutely unclear, sentence not fluent). Table IV details the criteria used in measuring fluency. We are aware that Fluency and Intelligibility are not the same. What we were looking for is a composed metric that includes both. Table V describes the examples given to the subjects to help them understand and use the scale. The actual evaluation input looked like the examples provided in Figure 13 without the first column.

Table IV. Fluency Criteria

5	clear meaning, good grammar, terminology and sentence structure
4	clear meaning BUT bad grammar, bad terminology or bad sentence structure
3	meaning graspable BUT ambiguities due to bad grammar, bad terminology or bad sentence structure
2	meaning unclear BUT inferable
1	meaning absolutely unclear

Table V. Fluency Scale Examples

Machine Translation	**Score**
-the united states unilaterally reduced china's textile export quotas. -the united states unilaterally reduced china textile export quotas.	5
-united states cutted china export textile ration lonely. -united states reduce down china quota textile export.	4
-united states reduce an quotas export textiling of the porcelain for the only busy a decision. -a chinese ration united states cut it down.	3
-states united unilateral cut an china textile speaks ration downwardly down. -cause states go quotas to reduced.	2
-beautiful folk remedy partage china exportation filament on own shaving. -alone cut it up rations alone.	1

6.2.3. *Translation Quality Evaluation Results*

The results of the evaluation are presented in table VI. The number in each cell represents the average score given by all subjects on all sentences for each evaluation. ChinMT did slightly better than Systran but the difference is statistically insignificant. Overall, the scores given show an average performance for both systems, glossed as follows: for Accuracy, *contents of original sentence generally conveyed BUT er-*

rors in relationship between constituents (cf Table II) and for Fluency, *meaning graspable BUT ambiguities exist* (cf Table IV).

Our system was able to perform as well as a commercial system that took many person-years to develop. Systran 3.0 Professional Edition Chinese-English MT system is the result of an estimated 20 person-years of work.[7] It utilizes a large lexicon of 150,000 root stems, 6,000 expressions, 1-2,000 Cantonese terms, 2500 Names, a 300,000 word safety net lexicon (CETA dictionary) and an optional 2K military terms. With this coverage, the system's strength is in military, computer science, and electronics domains.

As for our system, it was developed over 6 person-years. The English LCS lexicon includes about 12,000 entries, of which 9,500 are verbs and 900 are prepositions. The remaining 1,200 are nouns and adjectives, which may be dynamically generated based on specific domain needs.

Since our system is interlingual, all of its resources are readily extendible for use with other languages for both Analysis and Generation. A case in point is a previous project for Language Tutoring using LCS resources was retargeted from Arabic to Spanish in 1/6th the time it took to build the original project (Dorr, 1997b).

6.2.4. *Analysis of Translation Quality Results*

For the most part, the Nitrogen strategy of over-generating translation hypotheses coupled with selection according to bigram likelyhoods (Langkilde and Knight, 1998a), works very well. There are some difficulties that can be seen as responsible for the *average* scores received. One major issue is that, especially with the bigram language model's bias for shorter sentences, fluency is given preference over translation accuracy. Thus, if there is some material that is considered optional (e.g., by the decomposition process), and there are lattice entries both with and without this information, the extractor will tend to pick the path without this information. While this technique is also very successful at picking out more fluent, terse formulations (e.g., "John went to the bank" rather than "John went to at the bank", or "convincing proof" rather than "proof having convincingness"), further work is needed to assess the right ratio of terseness vs informativeness. Also, bigrams are obviously inadequate for capturing long-distance dependencies, and so, if things like agreement are not carefully controlled in the symbolic component, they will be incorrect in some cases.

[7] This estimate is computed based on information provided through personal communication with Mr Dale Bostad from NAIC (National Air Intelligence Center), the agency that sponsored the development of this product.

Table VI. Chinese-English Translation Quality Results

	LCS-based MT	Systran 3.0 Professional
Accuracy	3.08	3.01
Fluency	3.15	3.12

Table VII. CLCS Test Corpus Examples

Class	**Example**
2	someone_{ag} wanted something_{th} (to do $\text{something}_{th})_{prop}$
10.5	someone_{ag} stole something_{th} from something_{src} for something_{ben}
22.1.C	someone_{ag} mixed something_{th} into something_{goal}
29.1.B	someone_{th} considered something_{perc} (to be $\text{someproperty}_{pred})_{mod-pred}$
45.2.A	someone_{ag} folded something_{th} with something_{inst}
55.1.C	someone_{th} continued (to do $\text{something}_{th})_{prop}$

6.3. Coverage Evaluation

For this evaluation, a test corpus of 453 simple CLCS representations corresponding to all LVD classes was constructed semi-automatically.[8] The size of the test corpus guarantees large-scale coverage over verb behavior and thematic role combinations, which is exhaustive for our purpose. The CLCS representations were constructed by randomly selecting an LCS verb entry from each class from the English verb class and filling all its argument positions with simple noun phrases (e.g. *something*$_{th}$, *someone*$_{ag}$, etc.) or simple subordinate clauses (e.g. *(to do something)*$_{prop}$, *(to be someproperty)*$_{mod-prop}$, etc.) Table VII shows some sample English sentences corresponding to the CLCS representations in the test corpus.

For this evaluation, statistical extraction was disabled to evaluate the whole range of possible outputs generated by the system. For example, each of the two subclasses defining the dative alternations for the verb *send* are expected to generate *both* alternations (i.e. *John sent*

[8] Currently, the number of classes in LVD is 492. But at the time of conducting this evaluation, there were only 453 classes.

a book to Paul and *John sent Paul a book*). Out of 453 input CLCS representations, 25 failed the lexical selection process due to problems with lexicon entries. In the remaining cases, the lexical selection process appropriately generated multiple sentences for each CLCS. All of these correctly corresponded to various related alternations of the main verb. However, there were also cases of overgeneration resulting from preposition under-specification, which is inconsequential to our evaluation (e.g. *go (to,toward,towards,to at,etc.) somewhere*). The average number of sentences generated per class was 4.

6.3.1. *Coverage Evaluation Criteria*

The results of generation were passed to a speaker of English who was asked to mark sentences as being acceptable or not acceptable on three criteria: (1) argument generation, (2) prepositional phrase generation, and (3) word order. Acceptable argument generation is defined as the generation of all arguments of the verb whether pure arguments or obliques. Acceptable prepositional phrase generation is defined as the generation of good proposition choices such as goal prepositions versus source prepositions with an oblique goal and the generation of a prepositional object. Finally, acceptable word order is word order that reflects the correct relation of the arguments to the verb.

6.3.2. *Coverage Evaluation Results*

Table VIII displays the results of this evaluation. The first row represents the number and ratio of classes that generated no correct output for each error criterion. Some classes generated both correct and incorrect outputs. These are counted as correct with the assumption that given a good statistical extractor, the correct answer would rank highest. The second row is an estimate of the percentage of unsuccessful generation of verb senses, where the raw class results are weighted by the number of verbs in each class. On average each class contains 21 verbs, but since some classes have more verbs in them than others, this second line seems a more appropriate measure to evaluate coverage over the full lexicon (estimating actual verbs covered rather than verb classes). Another useful metric might be to normalize based on the probability of occurrence of verbs, giving more weight to frequently occurring verbs. But this is a much more complicated task because it requires a corpus that tags verb senses with their appropriate LCS structures.

The results of this evaluation are quite encouraging in that they show a high percentage of coverage over the LCS lexicon. Argument errors and word-order errors were due to incorrect lexical entries. For example, in the case of word-order errors, specific realization information such

Table VIII. Coverage Evaluation

N = 428	Argument Error	Preposition Error	Word Order Error
Class-based	6 (1%)	53 (12%)	5 (1%)
Verb-based	1%	9%	3%

as :EXT was missing from some entries. This problem appeared in three subclasses of class 41.3.1 (Simple Verbs of Dressing: don, doff and wear). In our lexicon, *clothes*, the object for all three verbs, is considered the theme and the subject of the sentence is the goal, source and location respectively. Fixing these cases is a matter of adding the appropriate piece of information in the lexicon. Preposition errors are more severe in that complete entries for some prepositions were not found in the lexicon. These errors will be fixed once the proper entries have been added. The generation system has thus been quite helpful as a diagnostic tool for determining errors and inconsistencies in the Lexicon.

6.4. Retargetability

Finally, we examine the generation system's language independence. For this evaluation task we used as input the same corpus of simple CLCS entries developed for the coverage evaluation presented in the previous section, however we replaced the English generation system with the Spanish one described in Section 5.

For the purposes of this evaluation, statistical extraction was disabled because we do not have a Nitrogen bigram model for Spanish and because we wanted to examine the range of alternations produced.

The results of the generation were passed to a speaker of Spanish to evaluate in a similar manner to the evaluation done for coverage. One extra criterion in this evaluation is a check on sense generation correctness, i.e., whether this Spanish verb is a proper translation of the English verb given the argument structure presented in the verb class.

As in the case of the English generation results presented in the previous section, some of the Spanish sentences failed the lexical selection process due to problems with lexicon entries. However, there were many more sentences that were produced, which should not have been generated in Spanish. In theory, the lexical selection process limits the number of choices using the LCS entry of the Spanish verbs. But

Table IX. Retargetability Evaluation

N = 254	Argument Error	Preposition Error	Word Order Error
Class-based	15 (6%)	85 (33%)	4 (2%)
Verb-based	10%	44%	0%

that process is only as good as the lexicon entries. In cases where a bad sense is generated, the sentence involved is dropped from the evaluation. Most failures in Spanish generation are due to missing verb entries (29% of all input classes). Erroneous lexicon entries were responsible for another 10% of generation failures. And an additional 5% of classes were dropped out of the evaluation because there was no correct sense output. As a result only 254 out of 453 classes (56%) have been evaluated on argument, preposition and word-order correctness.

Table IX displays the results of this evaluation. The first row represents the number and ratio of classes that generated no correct output for each error criterion. The second row represents the same ratios including class verb count as weights.

The Spanish output is not as clean as the English output: it has more overgeneration, more failures, and a higher error rate (except for word order errors). Argument errors are due to lexicon entries that were incorrect or missing. Most of preposition errors were due to incorrect overgeneration resulting from extra incorrect entries which were added to the lexicon automatically and were not manually checked.

A recent analysis of the Spanish lexicon indicates that 160 out of 453 semantic verb classes (about 35%) require re-verification for inconsistencies that resulted during the process of porting the classes from English to Spanish. (See (Dorr, 1997a) for more details of the porting process.) However, the focus of this evaluation was not on the quality or coverage of Spanish in our system. It was on the ease of extendibility of the system to another language. And given this criterion, this evaluation is quite positive since the amount of work that was needed was minimal: the Spanish lexicon already existed for analysis purposes and the OxyL grammar was created in a short period of time. Of course improving the quality of the system will need more work on both frontiers: the lexicon and the linearization grammar. There will also be a role to play in statistical extraction of best generated sentence, especially for cases of overgeneration that included both good and bad results.

7. Conclusions and Future Work

We have presented a system for Natural Language generation from Lexical Conceptual Structures, including situating the generation system within a larger machine translation effort, as well as evaluation of some key components of the results. The system has been used both to generate very long, complex, multiply ambiguous sentences (outputs of Chinese to English Translations), as well as thousands of simple sentence templates (spanning the whole of the English verb and preposition lexicons). Evaluation of the quality and correctness of both modes has been carried out, showing comparable translation quality with a commercial translation system. The generation system can also be straightforwardly extended to other languages, given appropriate target-language specific resources (lexicon and grammar), and this has been demonstrated and evaluated for Spanish.

As well as its utility for generating target-language sentences, the generation system also provides a crucial step in the development cycle for analysis and lexicon resources. Changes to a current lexicon, both extensions and corrections, which might be done either manually or using an automatic acquisition method can be evaluated based on how they will affect generation of sentences into that language. This has been a valuable diagnostic tool for discovering both specific errors and lacunae in lexicon coverage.

The biggest remaining step is a more careful evaluation of different sub-systems and preference strategies to more efficiently process very ambiguous and complex inputs, without substantially sacrificing translation quality. Also a current research topic is how to combine other metrics coming from various points in the generation process with the bigram statistics, to result in better overall outputs.

Acknowledgements

This work has been supported, in part, by DOD Contract MDA904-96-C-1250 and NSF PFF/PECASE Award IRI-962910. The second author is also supported by Army Research Laboratory contract DAAL01-97-C-0042, Logos Corporation, NSF CNRS INT-9314583, DARPA/ITO Contract N66001-97-C-8540, and Alfred P. Sloan Research Fellowship Award BR3336. We would like to thanks Clara Cabezas and Gina Levow from the Computational Linguistics and Information Processing Lab (CLIP) and Yi Ching Su from the Linguistics Department for their help evaluating the system. We would also like to thank other members of the CLIP lab for helpful conversations, particularly David Clark, Scott Thomas and Mari Olsen, Philip Resnik, and Amy Weinberg. And finally, We would like to thank Kevin Knight and Irene Langkilde for making the Nitrogen system available and help with understanding the Nitrogen grammar formalism.

References

Alsina, A. and S. Mchombo: 1993, 'Object Asymmetries and the Chichewa Applicative Construction'. In: S. Mchombo (ed.): *Aspects of Automated Natural Language Generation*. Stanford, CA: CSLI Publications, Center for the Study of Language and Information, pp. 1–46.

Baker, C.: 1989, *English Syntax*. Cambridge, MA: The MIT Press.

Bangalore, S. and O. Rambow: 2000, 'Corpus-Based Lexical Choice in Natural Language Generation'. In: *Proceedings of the ACL*. Hong Kong.

Bresnan, J. and J. Kanerva: 1989, 'Locative Inversion in Chichewa: A Case Study of Factorization in Grammar'. *Linguistic Inquiry* **20**, 1–50.

Carrier-Duncan, J.: 1985, 'Linking of Thematic Roles in Derivational Word Formation'. *Linguistic Inquiry* **16**, 1–34.

Church, K. and E. Hovy: 1991, 'Good Applications for Crummy Machine Translation'. In: *Proceedings of Evaluation Workshop of the 1991 ACL, Berkeley*.

Dorr, B. J.: 1993a, 'Interlingual Machine Translation: A Parameterized Approach'. *Artificial Intelligence* **63**(1 & 2), 429–492.

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

Dorr, B. J.: 1994, 'Machine Translation Divergences: A Formal Description and Proposed Solution'. *Computational Linguistics* **20**(4), 597–633.

Dorr, B. J.: 1997a, 'Large-Scale Acquisition of LCS-Based Lexicons for Foreign Language Tutoring'. In: *Proceedings of the ACL Fifth Conference on Applied Natural Language Processing (ANLP)*. Washington, DC, pp. 139–146.

Dorr, B. J.: 1997b, 'Large-Scale Dictionary Construction for Foreign Language Tutoring and Interlingual Machine Translation'. *Machine Translation* **12**(4), 271–322.

Dorr, B. J.: 2001, 'LCS Verb Database'. Technical Report Online Software Database, University of Maryland, College Park, MD. http://www.umiacs.umd.edu/˜bonnie/LCS_Database_Docmentation.html.

Dorr, B. J., J. Garman, and A. Weinberg: 1995, 'From Syntactic Encodings to Thematic Roles: Building Lexical Entries for Interlingual MT'. *Machine Translation* **9**, 221–250.

Dorr, B. J., N. Habash, and D. Traum: 1998, 'A Thematic Hierarchy for Efficient Generation from Lexical-Conceptal Structure'. In: *Proceedings of the Third Conference of the Association for Machine Translation in the Americas, AMTA-98, in Lecture Notes in Artificial Intelligence, 1529*. Langhorne, PA, pp. 333–343.

Dorr, B. J., G.-A. Levow, and D. Lin: 2000, 'Chinese-English Machine Translation and Information Retrieval: Building a Conceptual Verb Hierarchy'. In: *Proceedings of the Fourth Conference of the Association for Machine Translation in the Americas (AMTA), Cuernavaca, Mexico*. pp. 1–12.

Dorr, B. J. and M. B. Olsen: 1996, 'Multilingual Generation: The Role of Telicity in Lexical Choice and Syntactic Realization'. *Machine Translation* **11**(1–3), 37–74.

Fellbaum, C.: 1998, *WordNet: An Electronic Lexical Database*. MIT Press. Further information: http://www.cogsci.princeton.edu/~wn [2000, September 7].

Giorgi, A.: 1984, 'Toward a Theory of Long Distance Anaphors: A GB Approach'. *The Linguistic Review* **3**, 307–361.

Grimshaw, J. and A. Mester: 1988, 'Light Verbs and Theta-Marking'. *Linguistic Inquiry* **19**, 205–232.

Habash, N.: 2000, 'oxyGen: A Language Independent Linearization Engine'. In: *Fourth Conference of the Association for Machine Translation in the Americas, AMTA-2000*. Cuernavaca, Mexico.

Habash, N. and B. Dorr: 2001, 'Large-Scale Language Independent Generation Using Thematic Hierarchies'. In: *Proceedings of MT Summit VIII, Santiago de Compostella, Spain.*

Hovy, E.: 1999, 'Toward Finely Differentiated Evaluation Metrics for Machine Translation'. In: *Proceedings of the EAGLES Workshop on Standards and Evaluation*. Pisa, Italy.

Jackendoff, R.: 1972, 'Grammatical Relations and Functional Structure'. In: *Semantic Interpretation in Generative Grammar*. Cambridge, MA: The MIT Press.

Jackendoff, R.: 1983, *Semantics and Cognition*. Cambridge, MA: The MIT Press.

Jackendoff, R.: 1990, *Semantic Structures*. Cambridge, MA: The MIT Press.

Jackendoff, R.: 1996, 'The Proper Treatment of Measuring Out, Telicity, and Perhaps Even Quantification in English'. *Natural Language and Linguistic Theory* **14**, 305–354.

Kiparsky, P.: 1985, 'Morphology and Grammatical Relations'. unpublished ms., Stanford University.

Langkilde, I. and K. Knight: 1998a, 'Generation that Exploits Corpus-Based Statistical Knowledge'. In: *ACL/COLING 98, Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics (joint with the 17th International Conference on Computational Linguistics)*. Montreal, Canada, pp. 704–710.

Langkilde, I. and K. Knight: 1998b, 'The Practical Value of N-Grams in Generation'. In: *International Natural Language Generation Workshop*.

Larson, R.: 1988, 'On the Double Object Construction'. *Linguistic Inquiry* **19**, 335–391.

Levin, B.: 1993, *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago, IL: University of Chicago Press.

Levin, B. and M. Rappaport Hovav: 1996, 'From Lexical Semantics to Argument Realization'. Technical report, Nortwestern University. http://www.ling.nwu.edu/~beth/pubs.html.

Miller, G. A. and C. Fellbaum: 1991, 'Semantic Networks of English'. *Lexical and Conceptual Semantics* pp. 197–229.

Nagao, M.: 1989, 'Two Years After the MT SUMMIT'. In: *MT SUMMIT II*. pp. 117–122, Deutsche Gesellschaft fuer Dokumentation e.V. (DGD).

Nishgauchi, T.: 1984, 'Control and the Thematic Domain'. *Language* **60**, 215–260.

Olsen, M. B., D. Traum, C. van Ess-Dykema, and A. Weinberg: 2001, 'Implicit Cues for Explicit Generation: Using Telicity as a Cue for Tense Structure in a Chinese to English MT System'. In: *Proceedings of MT Summit VIII, Santiago de Compostella, Spain.*

Quirk, R., S. Greenbaum, G. Leech, and J. Svartvik: 1985, *A Comprehensive Grammar of the English Language*. Longman.

Wilkins, W.: 1988, 'Thematic Structure and Reflexivization'. In: W. Wilkins (ed.): *Syntax and Semantics 21: Thematic Relations*. San Diego, CA: Academic Press.