A Thematic Hierarchy for Efficient Generation from Lexical-Conceptual Structure

Bonnie Dorr and Nizar Habash and David Traum

UMIACS
University of Maryland
College Park, Md 20742
phone: +1 (301) 405-6768
fax: +1 (301) 314-9658
{dorr,habash,traum}@umiacs.umd.edu
WWW home page: http://umiacs.umd.edu/labs/CLIP

Abstract. This paper describes an implemented algorithm for syntactic realization of a target-language sentence from an interlingual representation called Lexical Conceptual Structure (LCS). We provide a mapping between LCS thematic roles and Abstract Meaning Representation (AMR) relations; these relations serve as input to an off-the-shelf generator (Nitrogen). There are two contributions of this work: (1) the development of a thematic hierarchy that provides ordering information for realization of arguments in their surface positions; (2) the provision of a diagnostic tool for detecting inconsistencies in an existing online LCS-based lexicon that allows us to enhance principles for thematic-role assignment.

1 Introduction

This paper describes an implemented algorithm for syntactic realization of a target-language sentence from an interlingual representation called Lexical Conceptual Structure (LCS). We provide a mapping between LCS thematic roles and Abstract Meaning Representation (AMR) relations; these relations serve as input to an off-the-shelf generator (Nitrogen). There are two contributions of this work: (1) the development of a thematic hierarchy that provides ordering information for realization of arguments in their surface positions; (2) the provision of a diagnostic tool for detecting inconsistencies in an existing online LCS-based lexicon that allows us to enhance principles for thematic-role assignment.

Several researchers have proposed different versions of thematic hierarchies (see [9, 4, 3, 13, 18, 7, 22, 21, 1, 2, 8]).[1] 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 (20 roles overall) than those of previous approaches (maximum of 8 roles). We have implemented the approach described here as part of a Chinese-to-English Machine Translation (MT) project

[1] For an excellent overview and a comparison of different thematic hierarchies see [20].

at the University of Maryland and we have used the resulting output to guide enhancements to a LCS-based database.

The next section describes our framework for mapping LCS roles to AMR relations. Section 3 introduces the thematic hierarchy used for syntactic realization of arguments. Section 4 describes the implementation, while sections 5 and 6 present the results of testing the implementation, and our conclusions.

2 Mapping LCS Roles to AMR Relations

The input to our MT system is a Chinese sentence that is parsed into a syntactic structure. This is passed to a semantic composition module which creates a corresponding LCS [10, 11, 12].[2] 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 [6] and MILT [5].

The LCS is passed to a generator which produces an output English sentence by means of two steps: lexical selection and syntactic realization. Lexical selection involves a comparison between LCS components and abstract LCS frames associated with words in an English lexicon. Syntactic realization re-casts LCS-based thematic roles as relations in an Abstract Meaning Representation (AMR), i.e., an unordered tree where the root is a concept and each child is linked by a relation.[3] An intermediate form (LCS-AMR) is produced as a by-product of this mapping between roles and relations. The AMR is as input to the Nitrogen system [15, 16, 17] which provides the mechanism needed for linearization, morphological derivation, word order and agreement.[4] (More details about Nitrogen and AMRs are given in Section 4.1.)

An example of the steps in the conversion from LCS to AMR is shown in (1) below, for the sentence *China arms the coalition*. The LCS can be roughly glossed as "China caused the coalition to go identificationally (or transform) towards being at the state of being armed." From this the agent (China) and theme (coalition) are extracted; these serve as slot-fillers in the LCS-AMR. These are then mapped into their corresponding slots in the AMR.

```
(1) LCS: (CAUSE (Thing China 1)
           (GO Ident (Thing Coalition 2)
              (TOWARD Ident (Thing Coalition 2)
                 (AT Ident (Thing Coalition 2)
```

[2] The parser and composition module were developed at the University of Maryland.

[3] We also map LCS-based modifiers, quantifiers, and other features into corresponding AMR-based components. For the purposes of this paper, we will focus specifically on realization of LCS arguments.

[4] The Nitrogen generation system was produced at the Information Science Institute at the University of Southern California. We chose to use this system as part of our generation efforts because of its large coverage and accessibility, as well as a balance between knowledge-based and statistical approaches.

```
                              (Property armed 9)))))

LCS-AMR: (A1 / |arm<render|
             :LCS-AG (A2 / |China|  :quant sing)
             :LCS-TH (A3 / |coalition|) :quant sing)

AMR: (A1 / |arm<render|
             :arg1 (A2 / |China| :quant sing)
             :arg2 (A3 / |coalition|) :quant sing)
```

Thematic roles in LCS are represented as integers. In the example above, `1` is used to designate *Agent* and `2` is used to designate *Theme*. Each role has a unique integer.[5] LCS thematic roles are defined to reflect the role taken by the object they refer to in the sentence described by the LCS. There is no built-in information about surface realization of these objects. For example, as shown in 2, a theme can be realized as the subject or the object of a sentence or even as an object of a preposition.

(2) *The girl* walked home.
We helped *the boy*.
She nibbled at *a cookie*.

This lack of direct correspondence to syntax is intentional: the LCS is a language-independent structure and positioning of arguments in the LCS is language dependent. The following table displays some of the most commonly used thematic roles, their thematic numbers, their corresponding LCS-AMR relations, and examples of possible realizations.

(3)

Theta Role	#	Abbrev	LCS-AMR Relationn	Possible Realizations
Agent	1	ag	:LCS-AG	Argument: *John* broke the chair.
Theme	2	th	:LCS-TH	Argument: *The boy* went to school. Oblique: She nibbled at *a cookie*.
Source	3	src	:LCS-SRC	Argument: She abandoned *the scene*. Oblique: We came from *the party*.
Goal	5	goal	:LCS-GOAL	Argument: He entered *the room*. Oblique: We are going to *the party*.
Location	11	loc	:LCS-LOC	Argument: *The trees* swarmed with bees. Oblique: The bees buzzed in *the trees*.

Realization of thematic roles associated with LCS positions must be provided on a per-language basis. Such information is specified in lexical entries in terms of thematic numbers (see above) coupled with requirements for optionality (:OPTIONAL and :OBLIGATORY), internal/external positioning (:INT and :EXT), and associated prepositions (:COLLOCATIONS). We use a thematic grid as an easy-to-read shorthand to encapsulate all of this information.

[5] The exception to this is that *Experiencer* and *Theme* share the integer 2.

The grid includes the thematic roles corresponding to LCS positions in their surface-realization order. A preceding underscore or comma tells whether the thematic role is obligatory or optional, respectively. The thematic grid also includes the particle(s) associated with a particular thematic role. For example, the thematic grid `_ag_th(at),instr(with)` conveys the information that, in English, the agent in the LCS must be realized as the subject and the theme and instrument (when provided) must be realized as obliques (prepositional phrases) in the order given.

In order to create a mapping between LCS-based thematic roles and AMR relations used in the Nitrogen system, we first examined the meaning behind the roles used in these two representations. At first glance, it would appear that they are entirely incompatible: LCS roles are purely thematic with no inherent syntactic ordering information while AMR relations are a mix of syntactic and semantic roles. Moreover, the AMR relations did not allow a specific preposition to be associated with certain oblique relations.[6] If the LCS associates a specific preposition with a certain role, the thematic role must be mapped into the relation `:spatial-location` in the AMR so that a preposition may be specified. This forces all obliques into one relation which, in addition to being inappropriate in many cases (since obliques are not always "spatial" in nature), does not allow for multiple oblique relations. Thus, only one oblique may be produced per sentence.

Our solution is to redefine the mapping between thematic roles and AMR relations to that of mapping between these roles and their surface realizations. We use syntactically-defined AMR relations: `:arg1`, `:arg2`, and `:goal` which refer to argument 1 (or logical subject), argument 2 (or logical object), and goal (the only relation corresponding to the logical indirect object). As for obliques, they will be mapped to two new relations that specify a preposition and its object (`:lcs-prep` and `:lcs-prep-object`). The new relations are then linearized separately, as described in the next sections.

3 Thematic Hierarchy

Once the LCS-AMR relations are identified, the generator must have access to information that establishes the relative ordering of arguments on the surface. A brute force approach to mapping relations to syntactic positions is to associate each verb instance in an AMR with its thematic grid from the associated abstract LCS frame in the lexicon. This method is expensive and inefficient as there are more than 107 distinct grids, each potentially containing several optional items that must be treated separately. Another approach is to induce an ordering among the thematic roles that mirrors their order of realization. Such an ordering may be imposed by means of a thematic hierarchy. As mentioned earlier, several researchers have proposed different versions of thematic hierarchies. An example

[6] Nitrogen tends to overgenerate by producing several possible prepositions associated with such relations.

of a simple thematic hierarchy can be the following:[7]

(4) `Agent > Theme > Location`

This means that in the case of an LCS with any two of these three roles, the roles must be realized as argument 1 and argument 2 in the order with which they appear, left to right, in the thematic hierarchy.

We constructed a more comprehensive thematic hierarchy than those proposed previously. To do this, we first extracted all the thematic grids in the verb LCS Lexicon. There were 107 distinct grids, each of which we divided into two partial grids: one for arguments and one for obliques. The relative ordering between obliques and arguments is always the same: arguments are realized closer to the verb and obliques follow. Each partial grid was then ordered topologically.

Initially, the following thematic hierarchy was found for arguments (the roles between the curly braces have equal relative order):

(5) `ag > instr > th > perc  > {goal, src, loc, poss, pred, prop}`

Several exceptions were found such as the following:

```
(6) (i) The bees buzzed in the trees.   (ii) The box contains the ball.
        Theme   >        Location            Location     >     Theme
(7) (i) They deserted the scene.  (ii) The cop fined John  40 dollars.
        Theme >      Source                Agent >   Source > Theme
```

Cases like (6) above are resolved using the lexical parameter :EXT, which is set for Location in (6)(ii). To integrate this solution, we created an intermediate LCS-AMR relation `:lcs-ext` that will replace the original thematic role. This new role is the highest on the thematic hierarchy by definition. Example (7) is the only unresolved ordering. We treat it as an exception that is addressed before everything else. Thus, the final hierarchy for arguments is as follows:

```
(8)       special case : ag src th  (in this order)
          ext > ag > instr > th > perc >  Everything Else
```

As for the ordering of obliques, the following order was established:

```
(9)       particle > mod-prop > ag > perc > th > purp > mod-loc >
           mod-pred > src > goal > mod-poss > ben
```

Note that the order of obliques is not a strict hierarchy but rather a possible topological sort. There are several interdependencies that are hard to resolve using a strict ordering. But for all possible relative orderings, the thematic hierarchy above reflects a correct realization order. Special cases to this hierarchy are found to be alternative possible realizations. For example the following two realizations are correct even though the first one, which appears in the thematic grid is not consistent with the thematic hierarchy used for the obliques:

[7] This most closely resembles the hierarchy proposed by [4].

(10) He talked about the plans to his neighbor.
`Possessed Modifier > Experiencer`
(11) He talked to his neighbor about the plans.
`Experiencer > Possessed Modifier`

4 Implementation

Assigning sentential positioning to arguments of a verb is only one task in syntactic realization. In order to complete the job of syntactic realization from the LCS-based interlingua representation, we are currently using the Nitrogen system from USC/ISI [15, 16, 17]. This system has several advantages for us: (1) Already implemented, including a large lexicon (110,000 word-senses), and grammatical and morphological rules for English; (2) Easily extensible by adding additional grammar rules; (3) Includes a statistical component to pick the most likely of possible realizations (by comparing n-grams in the sentence to a large English corpus); and (4) Variable input which can be at any of several levels including conceptual, shallow semantic, or syntactic.

Nitrogen not only handled the other tasks involved in realization, such as morphological realization and statistically picking more likely possible realizations, but also provided a formalism for writing transformation rules (as well as an implementation which executes the transformations), which allowed us to implement the thematic hierarchy discussed in the previous section. In this section, we first briefly discuss the Nitrogen system and then describe how we made use of it to implement the thematic hierarchy discussed in the previous section to realize English output sentences from LCS structures.

4.1 The Nitrogen Generation System

As described above, the input to Nitrogen is an AMR, i.e., an unordered tree where the root is a concept and each child is linked by a relation. Each child is either an AMR itself or a terminal atom, such as a feature value. AMR relations can be either syntactic or semantic roles. Some relations specify the case of the sub-tree it heads such as `:agent`, `:patient`, `:source`, `:destination`, etc. Other relations specify certain features such as syntactic category, tense, or quantity. The following is an example AMR to represent the sentence "the boy went to school":

```
(12)  (W1 / |go<render|  :agent (W2 / |boy| :quant sing)
                         :destination (W3 / |school|) :quant sing))
```

The slash mark specifies an instance of something (the boy, the going, and the school). The symbols W1, W2, and W3 are node markers.

Nitrogen makes use of a number of heterogeneous knowledge sources, including: (1) A statistical database;[8] (2) The Sensus Ontology which constitutes

[8] This is a database of uni and bi-gram occurrences calculated based on two years of Wall Street Journal [17]

the lexical knowledge of the system;[9] (3) Morphological knowledge implemented using a morphology derivation grammar that handles both derivations and inflections [16];[10] and (4) Syntactical knowledge implemented using a grammar database that contains two types of transformation rules: linearization and recasting. We focus on this last knowledge source.

Linearization rules transform an AMR—or a part of an AMR (e.g., the object of a relation)—into a word sequence. This handles the generation of multiple surface forms. For example, the linearization rule (13) realizes two alternate word sequences (sentences). In the first, the relation `:agent` is generated as the first element in the sentence, the subject. But in the second sequence, `:agent` is realized as the object of the preposition *by* in a passive inversion of the first sentence.

```
(13) ((x1 :agent :senser) (x2 :patient :phenomenon) (x3 :rest) ->
      (s  (seq (x1 np) (x3 v-tensed) (x2 np)))
      (s  (seq (x2 np) (x3 v-passive)(wrd "by") (x1 np))))
```

Recasting rules transform an AMR into another AMR by redefining the original relations. This allows great flexibility in the level of input relations, since rules can be written to transform one structure into another. As such, it is possible to refer to the semantic level relations (e.g., `:agent` and `:patient`) as well as more syntactic relations (e.g., `:arg1` and `:arg2`). The recasting rule shown in (14) transforms an AMR with the `:time` relation having a value `future` to a similar structure in which a `:modal` relation is added and the `:time` has the value `present`.

```
(14) ((x1 :rest) (x2 (:time future)) :cut  ->
      (? (x1 (:add (:modal x2) (:time present)) ?)))
```

Both kinds of rules were used to implement the transformation of a set of LCS thematic roles into syntactic roles or positions that can be integrated with the rest of Nitrogen's realization mechanism.

4..2 Implementing the Thematic Hierarchy

The thematic hierarchy is implemented using an extension to Nitrogen's grammar that recasts LCS thematic roles into pre-existing AMR relations in an order consistent with the thematic hierarchy. In the case of arguments, three recasting rules are used to map to `:arg1`, `:arg2`, and `:goal`[11]. The ordering of the rules

[9] This is a knowledge base of 70 thousand nodes derived from several sources such as Wordnet, Longman dictionary and penman upper model [14].

[10] Nitrogen over-generates and depends on the statistical extractor to discard bad cases. For example, one morphology rule creates a plural ending *-xes* and *-xen* for all nouns ending with *-x*. This generates *boxes* and *oxen* but also **boxen* and **oxes*.

[11] `:goal` is the best match available among Nitrogen's relations for a second internal argument. It does not (necessarily) carry the intuitive semantic function of a goal relation, but is merely used to position arguments correctly.

forces Nitrogen to match first with `:arg1` then `:arg2`. If a match is found for `:goal`, then `:arg2` and `:goal` are swapped as part of the recasting. An additional rule is needed to implement the special case referred to above in (8). An example rule, the one for picking the first argument is shown below in (15). Note that the listing of options reflects the order in the thematic hierarchy.

```
(15) ((x1 :rest)
     (x2 :lcs-ext :lcs-ag :lcs-instr :lcs-th :lcs-perc :lcs-goal
         :lcs-mod-poss :lcs-mod-loc :lcs-src :lcs-mod-pred
         :lcs-loc :lcs-poss :lcs-pred :lcs-prop)
     ->  (? (x1 (:add (:arg1 x2)) ?)))
```

The thematic hierarchy for the obliques is implemented differently for two reasons. First, each oblique must be identified by the existence of two relations: the actual thematic role (e.g., lcs-goal or lcs-src) and its corresponding particle relation (e.g., lcs-goal-part or lcs-src-part). Second, the linearization rules for obliques must be associated with each specific preposition. Therefore, there are two sets of rules associated with the realization of obliques. First, there are linearization rules to create the correct sequence of :lcs-prep and :lcs-prep-obj for every possible preposition. And secondly, there are recasting rules to transform the thematic roles and thematic particle roles into :lcs-prep and :lcs-prep-obj. These rules are ordered to reflect the thematic hierarchy of obliques. In (16) we present the two rules that realize a source using the particle "from".

```
(16) ; Linearize with the preposition "from"
     ((x1 (:lcs-prep |from|)) (x2 :lcs-prep-obj) (x3 :rest) ->
       (s (seq (x3 s) (wrd "from") (x2 np))))

   ; Recasting goal and goal particle
     ((x1 :lcs-goal-part) (x2 :lcs-goal) (x3 :rest) ->
      (? (x3 (:add (:lcs-prep x1) (:lcs-prep-obj x2)) ?)))
```

5 Results

The implementation of the thematic hierarchy was tested using a set of 100 randomly selected sentences from a set of 550 examples sentences that are associated with the LCS-based verb lexicon (to exemplify the realization of particular verb classes based on [19]). These sentences were then semi-automatically converted into the LCS-AMR representation. Full realization was performed, i.e., conversion to AMR and Nitrogen's morphological realization. Sample test sentences are given in (17), along with final generation results.

```
(17) (A1 / |place| :LCS-AG (A2 / |he|) :LCS-TH (A3 / |book|)
                 :LCS-GOAL-PART |on| :LCS-GOAL (A5 / |table|))

   he placed the book on the table .
```

```
-----------------------------------------------------
(A850 / |hear| :LCS-TH (A851 / |he|)
           :LCS-PERC-PART |about| :LCS-PERC (A853 / |murder|))

he heard about the murder .
-----------------------------------------------------
```

Out of 100 sentences, only one problematic argument assignment was found:

```
(18) (A1333 / |wink|
           :LCS-TH (A1334 / |she|)
           :LCS-INSTR (A1335 / |eye| :MOD (A1336 / |her|))
           :LCS-PERC-PART |at| :LCS-PERC (A1338 / |him|))
```

This AMR returned the sentence **her eyes wink she at him* instead of the expected *she winked her eyes at him*. This case revealed an error in the LCS-based lexicon: In all other instances where instrument and theme co-occurred, instrument was higher in the hierarchy. So, the theme (originally the experiencer) must be forced to be external by setting the lexical parameter `:EXT` (or changing the role assignments, e.g, from theme to agent).

The use of the generator as a diagnostic tool has aided detection of other types of inconsistencies in the LCS-based lexicon. This has allowed us to enhance principles for thematic-role assignment. For example, in an earlier version of the LCS lexicon the following classes of verbs were distinguished by their thematic grids:

(19) (i) `_th_loc`: bound, bracket, ..., hug, skirt, surround, ring
(ii) `_loc_th`: contain, enclose

However, our experimentation with the generator revealed that there would be no principled way to assign reversed roles to objects in sentences such as the following:

(20) (i) The fence (th) surrounded the house (loc)
(ii) The fence (loc) enclosed the house (th)

Thus, we collapsed the two classes of verbs *contain* and *enclose* into a single class associated with the grid `_loc_th`.

6 Conclusion

The small test described in the previous section shows that the thematic hierarchy implementation has good coverage over a large sample of the Levin verb classes. Our approach is efficient in that it accesses a single thematic hierarchy rather than individual ordering specifications for linearization of arguments and obliques. Moreover, the approach allows sentences to be produced in a fashion that mirrors that of parsing, with thematic roles corresponding to D-structure positions. Finally, we have used the output resulting from the generator as a diagnostic tool for detecting inconsistencies in an existing LCS-based lexicon and, consequently, enhancing this online resource.

Our future work will involve testing the system on additional data, as well as completely automating the process of generation from Lexical Conceptual structures. Our goal is to produce preliminary results on deployment of an end-to-end Chinese to English machine translation system by the fall of 1998. Further work will focus on other aspects of the generation process, such as improving the performance on grammatical features and modifiers.

Acknowledgments

This work has been supported, in part, by DOD Contract MDA904-96-C-1250. The first author is also supported by Army Research Laboratory contract DAAL01-97-C-0042, NSF PFF IRI-9629108 and 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 thank members of the CLIP lab for helpful conversations, particularly David Clark, Scott Thomas and Mari Olsen. We would also like to thank Kevin Knight and Irene Langkilde for making the Nitrogen system available and help with understanding the Nitrogen grammar formalism.

References

1. A. Alsina and S.A. Mchombo. Object Asymmetries and the Chichewa Applicative Construction. In S.A. Mchombo, editor, *Aspects of Automated Natural Language Generation*, pages 1–46. CSLI Publications, Center for the Study of Language and Information, Stanford, CA, 1993.
2. C.L. Baker. *English Syntax*. The MIT Press, Cambridge, MA, 1989.
3. J. Bresnan and J. Kanerva. Locative Inversion in Chichewa: A Case Study of Factorization in Grammar. *Linguistic Inquiry*, 20:1–50, 1989.
4. J. Carrier-Duncan. Linking of Thematic Roles in Derivational Word Formation. *Linguistic Inquiry*, 16:1–34, 1985.
5. Bonnie J. Dorr. Large-Scale Acquisition of LCS-Based Lexicons for Foreign Language Tutoring. In *Proceedings of the ACL Fifth Conference on Applied Natural Language Processing (ANLP)*, pages 139–146, Washington, DC, 1997.
6. Bonnie J. Dorr, James Hendler, Scott Blanksteen, and Barrie Migdalof. Use of Lexical Conceptual Structure for Intelligent Tutoring. Technical Report UMIACS TR 93-108, CS TR 3161, University of Maryland, 1993.
7. A. Giorgi. Toward a Theory of Long Distance Anaphors: A GB Approach. *The Linguistic Review*, 3:307–361, 1984.
8. J. Grimshaw and A. Mester. Light Verbs and Theta-Marking. *Linguistic Inquiry*, 19:205–232, 1988.
9. Ray Jackendoff. Grammatical Relations and Functional Structure. In *Semantic Interpretation in Generative Grammar*. The MIT Press, Cambridge, MA, 1972.
10. Ray Jackendoff. *Semantics and Cognition*. The MIT Press, Cambridge, MA, 1983.

11. Ray Jackendoff. *Semantic Structures.* The MIT Press, Cambridge, MA, 1990.
12. Ray Jackendoff. The Proper Treatment of Measuring Out, Telicity, and Perhaps Even Quantification in English. *Natural Language and Linguistic Theory*, 14:305–354, 1996.
13. P. Kiparsky. Morphology and Grammatical Relations. unpublished ms., Stanford University, 1985.
14. Kevin Knight and Vasileios Hatzivassiloglou. Two-Level, Many-Paths Generation. In *Proceedings of ACL-91*, pages 143–151, 1991b.
15. Irene Langkilde and Kevin Knight. Generating Word Lattices from Abstract Meaning Representation. Technical report, Information Science Institute, University of Southern California, 1998.
16. Irene Langkilde and Kevin Knight. Generation that Exploits Corpus-Based Statistical Knowledge. In *Proceedings of COLING-ACL '98*, pages 704–710, 1998.
17. Irene Langkilde and Kevin Knight. The Practical Value of N-Grams in Generation. In *International Natural Language Generation Workshop*, 1998.
18. R.K. Larson. On the Double Object Construction. *Linguistic Inquiry*, 19:335–391, 1989.
19. Beth Levin. *English Verb Classes and Alternations: A Preliminary Investigation.* University of Chicago Press, Chicago, IL, 1993.
20. Beth Levin and Malka Rappaport-Hovav. From Lexical Semantics to Argument Realization. Technical report, Northwestern University, 1996.
21. T. Nishgauchi. Control and the Thematic Domain. *Language*, 60:215–260, 1984.
22. W. Wilkins, editor. *Syntax and Semantics 21: Thematic Relations.* Academic Press, San Diego, CA, 1988.

This article was processed using the LaTeX macro package with LLNCS style