Formal Mechanisms for Capturing Regularizations

Adam Meyers[†], **Ralph Grishman**[†], **Michiko Kosaka**[‡]

[†]New York University, 719 Broadway, 7th Floor, NY, NY 10003 USA
meyers/grishman@cs.nyu.edu

[‡]Monmouth University, West Long Branch, N.J. 07764, USA
kosaka@monmouth.edu

Abstract

While initial treebanks and treebank parsers primarily involved surface analysis, recent work focuses on predicate argument (PA) structure. PA structure provides means to regularize variants (e.g., actives/passives) of sentences so that individual patterns may have better coverage (in MT, QA, IE, etc.), offsetting the sparse data problem. We encode such PA information in the GLARF framework. Our previous work discusses procedures for producing GLARF from treebanks and parsed data. This paper shows that GLARF is particularly well-suited for capturing regularization. We discuss crucial components of GLARF and demonstrate that other frameworks would require equivalent components to adequately express regularization.

1. Introduction

The past decade has seen a great deal of work on developing 'treebanks' and using treebanks to create increasingly accurate parsers. Although initial work primarily involved surface-structure analysis, much recent work has focused on predicate argument (PA) structure. PA structure can be used to capture syntactic regularizations, providing a common representation for variants (such as active and passive clauses) which convey the same semantic relationship. In this way regularizations can reduce the number of patterns required to capture a semantic relation for such applications as MT, QA, and IE. While this advantage has long been recognized, the goal of recent efforts is to combine these benefits with the high accuracy of corpus-based analyzers. To a limited degree, PA structure has been added to parse trees using function tags, labels carrying grammatical role information or semantic class information (Marcus et al., 1994; Blaheta and Charniak, 2000). However, efforts to incorporate much more PA structure information are underway.

PA structure seems to mean different things to different researchers, but all to some degree seem to address the problem of regularization – expressing noncanonical constructions in terms of canonical ones. PA structures may include any of the following elements: (1) function tags which semantically classify constituents or assign them grammatical roles; (2) labeled arcs (representing dependencies or grammatical roles) which show how constituents are related to each other; and (3) filler/gap representations, e.g., empty categories coindexed with antecedents. In our work, we focus specifically on representing regularization incorporating these sorts of mechanisms and others into one coherent framework: GLARF (Grammatical and Logical Argument Representation Framework). By focusing on the regularization aspect of PA structure, we are trying to maximize the utility of our research for a range of applications involving generalization of syntactic patterns. Regularization could be a major tool for combating the sparse data problem because a simple pattern may be able to recognize 20 or more times the number of sentence types when applied to regularized data rather than unregularized data. If regularizations are adequately exploited, statistical NLP should be able to achieve better coverage with less training data.

Our previous work (Meyers et al., 2001a; Meyers et al., 2001b) discusses our procedures for producing GLARF from treebanks and parsed sentences – we currently have a small, but growing hand-corrected GLARF corpus and have applied our GLARFing procedures to the entire Penn Treebank and thousands of computer-parsed sentences. In this paper, we show that GLARF is better suited than previous treebank frameworks for capturing regularizations. We will define three components of GLARF that are crucial to adequately representing regularizations: gap typing, index typing and arc typing. Without equivalent components, we argue that other frameworks will not be able to represent crucial details of regularizations. Until now, the broad range of issues involved in representing regularization have not been addressed in a single computational framework.

2. Regularization in Syntactic Theory

Regularization has been explored thoroughly in syntactic theories over the last half century beginning with the work of Zellig Harris, and continuing with subsequent work in most other frameworks (Transformational Grammar, Relational Grammar (RG), Case Grammar, Feature Structure frameworks, Dependency Grammar frameworks, ...).

Zellig Harris used transformations (Harris, 1968) to capture paraphrase relations between related sentence types so that, for example, a passive sentence ("Apple was acquired by Disney") or a nominalization phrase ("Disney's acquisition of Apple") could be transformed into the same simple sentence ("Disney acquired Apple"). Later work in Transformational Grammar (TG) (Chomsky, 1973; Fiengo, 1974) provided a way for one analysis to represent both the noncanonical construct and its regularization (D-structure). Thus in "e_i Apple$_j$ was acquired t_j by Disney$_i$": e and t represent canonical or logical subject and object positions and the coindexed words occur in their surface positions.[1]

[1]This analysis uses the formal mechanisms of TG of the 1970s,

The placeholders (e and t) for the gaps are called empty categories (ecs). RG and other graph-based frameworks (e.g., Feature Structure frameworks) adopted an approach in which a gap is represented by an arc rather than an empty category. Under this approach, the same constituent appears at the head of more than one arc in a graph.[2] Only one of the arcs represents the surface position of the constituent.[3]

In addition to pure lexical or grammatical alternations where a constituent has "moved" (metaphorically) from its canonical position (e.g., passive), there are some constructions in which a single constituent is assigned multiple grammatical roles. For example, in the control (or equi) construction "John$_i$ tried t$_i$ to talk to Mary", "John" is the logical subject of both "tried" and "talk", where the second relation is modeled here as an ec which is coindexed with "John". While many current linguistic frameworks handle control and related phenomena, (Mel'ĉuk, 1988) enumerates some additional argument sharing structures (represented with "lexical functions"). For example, "Carthage" is the subject of "suffered", as well as the logical object of the predicate "attack" in "Carthage suffered from Rome's attack", assuming that the nominalization "Rome's attack" is regularized to its sentential counterpart.

Modern theta roles originated with (Gruber, 1965) and were elaborated by others (Fillmore, 1968; Jackendoff, 1983).[4] Under these approaches, a "role" is assigned to each argument of a predicate. Constituents bearing similar semantic relations to their governing predicate are assigned the same role (agent, patient, theme, . . .). Thus "the book" is the theme in both "John gave Mary the book" and "John gave the book to Mary". Unfortunately, it is difficult to enumerate the entire set of theta roles and it is sometimes difficult to decide which theta role should be assigned to a particular constituent. For example, one would have to go outside of the set {agent, experiencer, patient, theme, goal, location} to assign roles to the arguments of "surrounds" in "The circle surrounds the square" or "multiplied" in "Acme multiplied its workforce by three". The set of roles seem to grow very quickly as more verbs (and nouns and adjectives) are covered. While one can imagine a system with hundreds of such roles, such systems are hard to apply unambiguously. Furthermore, should the number of roles approach a large fraction of the size of the lexicon, the system's generality would come into question. For purposes of regularization, however, it seems that a smaller number of course-grained roles can be used effectively. We have found that RG's initial (logical) grammatical relations (GRs): subject, object, etc. provide enough distinctions to describe the verbal roles in all the alternations we have tried to model. (Additional roles are used to describe constituents of NPs, ADJPs, etc.) For example, if the bracketed constituents are the logical direct objects of "spray" in both "Mary sprayed [paint] on the wall" and "Mary sprayed the wall with [paint]", we can generalize over instances of the verb "spray". Given a specific predicate, it is assumed that the logical grammatical roles it assigns remain constant. However, unlike theta roles, logical GRs do not necessarily generalize semantic properties of a category (e.g. direct object) across predicates or across languages.[5] This aspect of RG has subsequently been adapted for various other frameworks and formalisms including LFG, HPSG, Penn Treebank II and the upcoming PropBank (http://www.cis.upenn.edu/ ace/). GLARF explicitly adopts an RG-based approach to GRs.

Crucially, the naming of roles (initial grammatical roles, theta roles, etc.) and the positing of filler/gap relations (structure sharing arcs, ecs and antecedents, etc.) are formal mechanisms for modeling regularizations while maintaining a surface analysis (e.g., without reordering words or removing structure). GLARF aims to extends these formal mechanisms, so that regularizations are properly represented – so that the surface form of the sentence is maintained and so that the relation between the surface form and the regularized form is characterized as precisely as possible.

3. Regularization in Treebanks

The Penn Treebank II (PTB) (Marcus et al., 1994) and automatic function taggers (Blaheta and Charniak, 2000) modify node labels with suffixes that they call "function tags". Penn's function tags are used both to semantically classify phrases (e.g., loc, tmp, dir) and to mark grammatical roles (e.g., sbj, lgs, clr). However, one important detail is missing: the predicates associated with the roles are not marked in the corpus. Thus a user must use patterns to identify which predicate an NP is the subject (sbj) of. While these patterns are usually trivial, they are not always trivial (particularly with lgs, which marks the logical subject of a passive). In addition, ecs (constituents of category -NONE-) indicate "missing constituents". When coindexed with node labels, these empty categories represent regularizations along the lines of the TG analyses described above. PropBank, a project currently underway at the University of Pennsylvania, will extend the Penn Treebank annotation further to include grammatical function labels that are regularized across both the sort of regularizations represented by filler/gap relations (passive, etc.) and verb alternations, e.g., "John" would be the Arg0 in both of the following sentences from the "PropBank Annotation Guidelines" (see their website): "John works hard" and "Penn works John

but would not be standard for that framework.

[2]Unfortunately, both graph theory and linguistic theory use the term "head" to mean different things. In a directed graph theory, an arc originates with a node called the tail and terminates in a node called the head. In linguistic theory, the head is a privileged constituent of a phrase which has special properties, as discussed below.

[3]We use "surface" for constituents in their canonical positions, but never to gaps. In contrast, some linguistic theories assume that certain ecs occur on the surface.

[4]The Karakas described in Panini's work more than 2000 years ago may be a precursor of theta roles.

[5]According to the Universal Alignment Hypothesis (UAH) (Perlmutter and Postal, 1984), clusters of semantic roles may be collapsed into single initial GRs (e.g., agent and experiencer are initial subjects, themes and patients are initial direct objects, etc.). As this hypothesis holds in the overwhelming majority of cases, it is extremely useful for identifying GRs. See (Rosen, 1984) for a discussion of the exceptions.

hard". Penn has decided to number arguments from 0 to some maximum (probably 3) and use names for adjuncts and other labels. Similarly RG assigns numbers to grammatical roles: subject = 1, direct object = 2, ... GLARF differs in a number of important respects from PTB plus PropBank: (1) all GLARF constituents are marked with role labels, not just constituents of VPs and S; (2) gap typing is more extensive in GLARF - while PTB has multiple types of ec, the classification is too coarse to make the distinctions discussed in Section 6; (3) Penn has two types of index - most Penn indices are indicated with a hyphen and number (-1), but some are indicated with an equal sign (=1) instead. The latter indicate parallel arguments of gapping constructions and the former indicate filler gap coindexing. In contrast, GLARF allows more types of coindexing just for fillers and gaps. Without similar mechanisms, it is unclear how PTB+PropBank can properly represent many of regularizations described in Sections 6 through 7.

The Prague Dependency Treebank (PDT) (Hajiĉová and M. Ceplová, 2000) uses distinct representations for unregularized (Analytical) and regularized (Tectogrammatical) analyses. In their regularized structure, they fill in the gaps with copies of the fillers, which they type according to similar criteria as we use below (cf. Section 6). Their distinct representations are similar in spirit to GLARF arc types described in Section 5. The difference between PDT and GLARF that is most significant to the topic of this paper is that PDT does not distinguish between types of coindexing (except when implied by gap type). This suggests that PDT may have difficulty adequately representing the relation between a nominalization and the related sentence and also may have trouble with instances of split reference. See Section 7.

4. GLARF Feature Structures

GLARF is an extended typed feature structure formalism (Carpenter, 1992). Following common practice, we model feature structures (FSs) as single-rooted edge-labeled directed acyclic graphs. As GLARF is an extension of the Penn Treebank (PTB), we maintain some PTB structure, e.g., leaf nodes representing words bear Penn parts of speech. As in other FS frameworks, some arcs and nodes represent attributes rather than constituents: semantic and morphological features are represented as arcs labeled VOICE, ASPECT, SEM-FEATURE, etc. with atomic values Pass, Prog, TMP, etc.; arcs labeled with INDEX, EXP-INDEX, etc. are used for coindexing (details below). We refer to arc labels that dominate constituents as role labels because their names are grammatical roles (subject, object, etc.) Figures 1 through 4 are sample GLARF FSs.

4.1. GLARF Relational Arcs

Following the model of RG, our analysis of each constituent includes sets of GRs holding among its children. Each GR holds between a pair of constituents as determined by their arc labels. Thus the COMPlement Relation holds between a constituent at the head of a HEAD arc and a sister constituent at the head of a COMP arc. For each relation, one role marks the "functor" and the other marks the "non-functor". Following Categorial Grammar, TAGS and

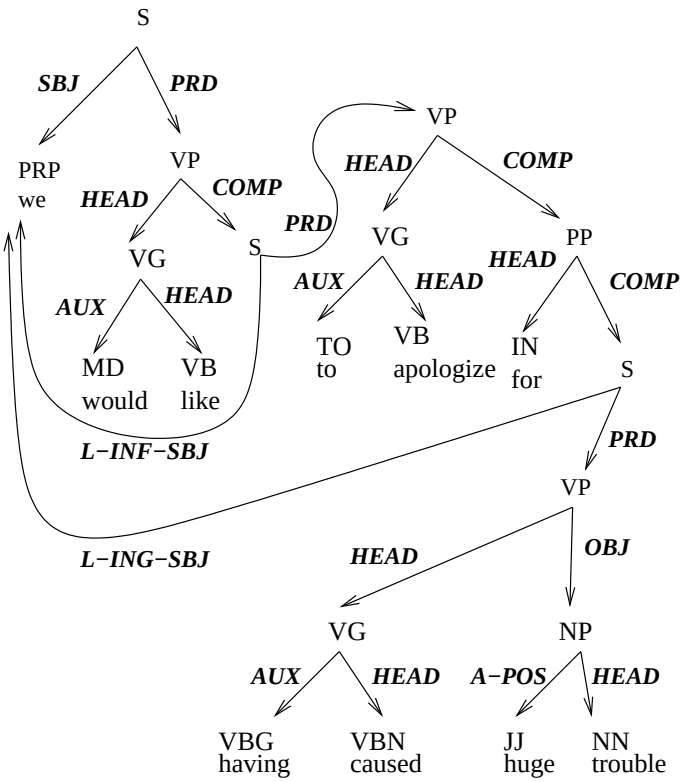

Figure 1: Control

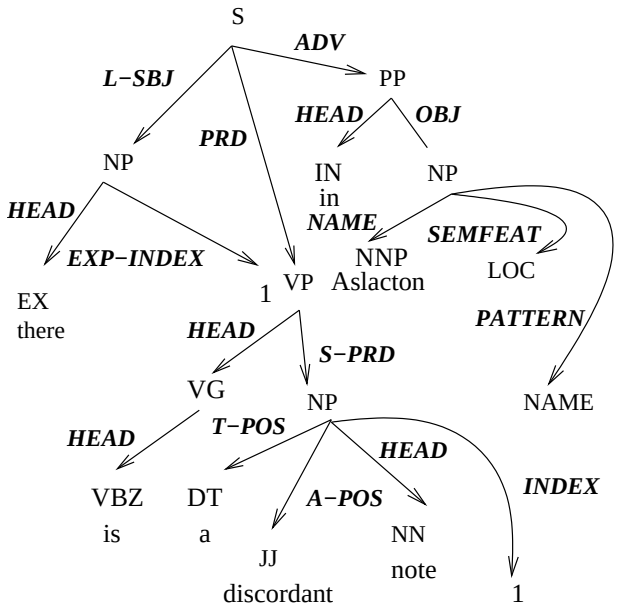

Figure 2: Pleonastic "there"

others, we separate "functor" from "head" so that they need not be the same constituent. (cf. (Corbett et al., 1993) regarding alternative definitions of "head"). We assume these definitions:

Head: The child X that determines the category of its parent XP.

Functor: Given a GR between sisters X and Y, X is the functor if X determines how X and Y combine.

Some phrases lack heads or include special sets of constituents which collectively act like a head including: con-

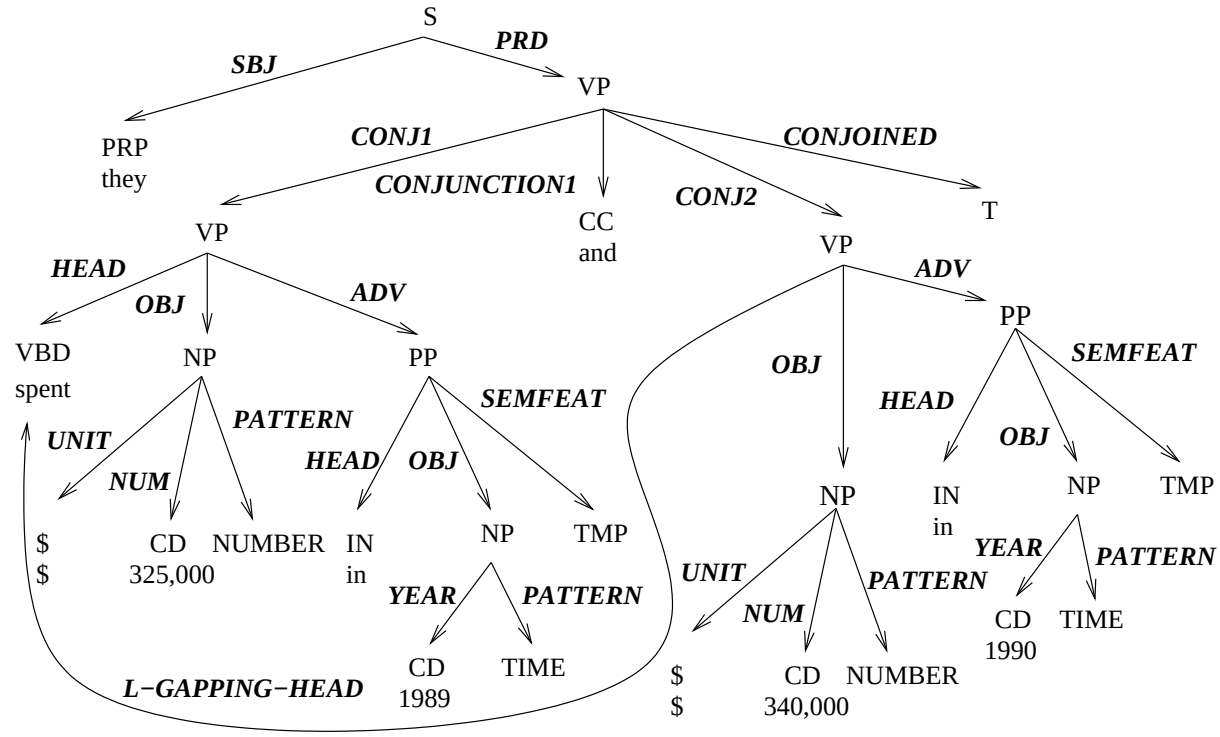

Figure 3: Gapping

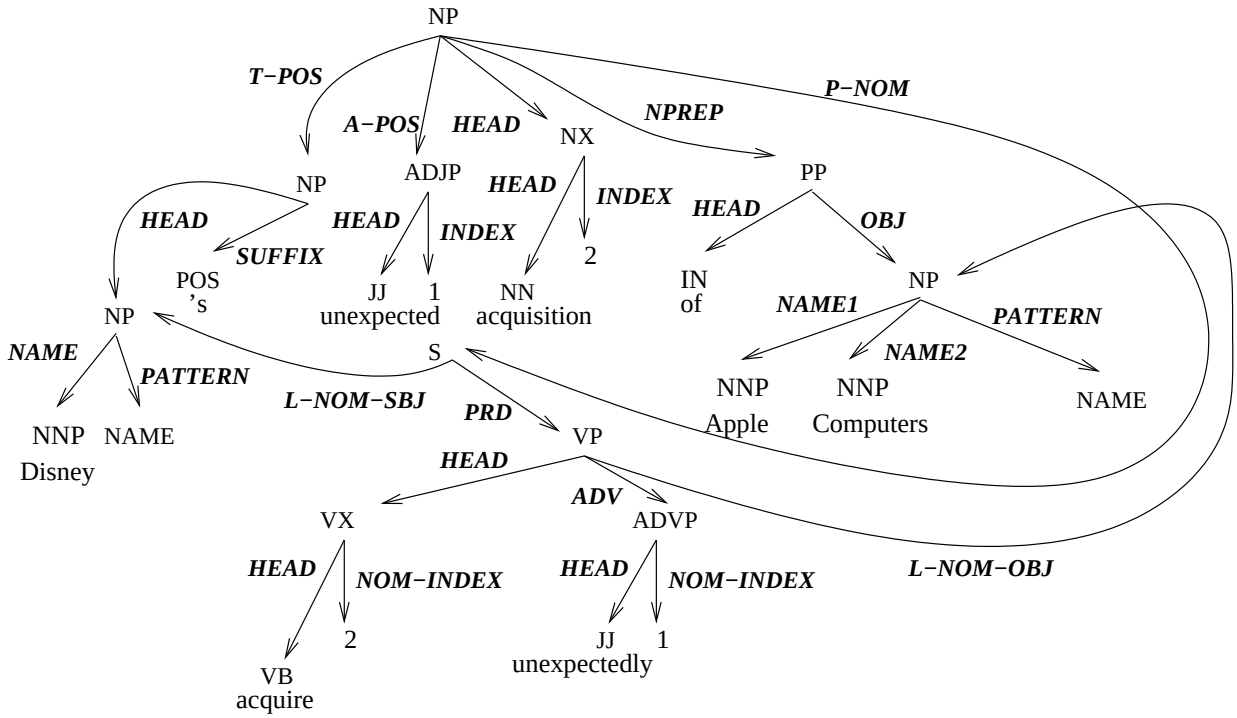

Figure 4: Paraphrase of Nominalization

joined constituents (“Fred and Mary and Sally”), named entities (“John Smith”), Time Expressions (November 20, 1962), phrases like the bracketed one in “[from 5 to 10] dollars”, etc. Table 1 lists some GRs along with the functor role and the nonfunctor role. The last two entries in the table are perhaps the most unusual: a conjunction (functor) forms one conjunct relation with each CONJunct; the combination of all name (NAME*) constituents in a name like “Dr. John Smith” collectively act like the functor in the title relation in:
(NP (Title (NNP Dr.)) (NAME1 (NNP John)) (NAME2 (NNP Smith))).
This set of assumptions is important because they make it possible to represent non-headed phrases adequately. In

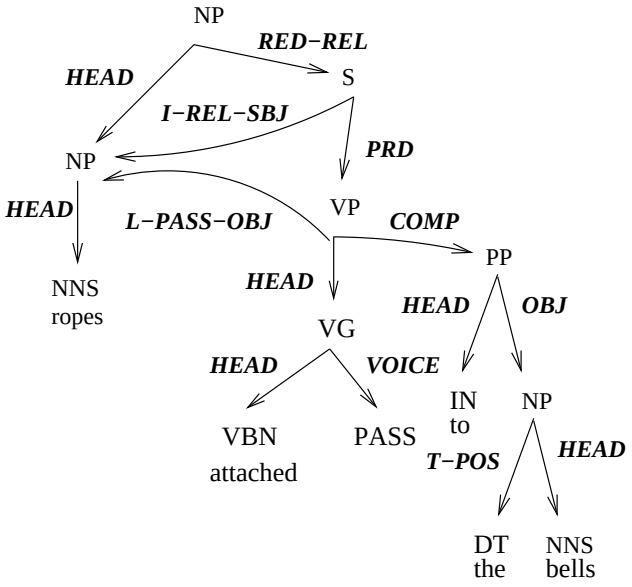

Figure 5: Reduced Relative

GR	Funct	NonFunct
SBJ	PRD	SBJ
OBJ	HEAD	OBJ
IOBJ	HEAD	IOBJ
COMP	HEAD	COMP
A-POS	HEAD	A-POS
T-POS	HEAD	T-POS
ADV	HEAD	ADV
ADV	PRD	ADV
CONJ	CONJUNCTION	CONJ
TITLE	NAME*	TITLE

Table 1: GRs, Functors and NonFunctors

particular, the explicit marking of conjuncts and conjunctions make it easy to apply metarules so that conjunction can be adapted for most pattern matching applications. Making it possible for conjunction to be handled by such metarules contributes to the regularization of text, even though we do not actually regularize a given sentence containing conjoined phrases to "look" like a set of unconjoined sentences.[6] Most PA frameworks used in NLP do not properly handle non-headed phrases. For example, the versions of Dependency Grammar typically used require that the head and functor be the same.

In our FSs (see figures), we modify these grammatical roles to make them more flexible. First of all, we number multiple instances of the same role (CONJ1, CONJ2, A-POS1, A-POS2, etc.). This makes our FSs functional (arcs with the same tail are uniquely labeled). A second modification involves the prefixes L-, S- and I- and infixes like -ING-, -INF- and -GAPPING-. The latter are crucial for our representation of regularization and will be discussed in the following two sections.

5. Arcs Representing Strata

Following (Johnson and Moss, 1993; Johnson et al., 1993), arc labels are typed to represent similar aspects of analysis as RG strata or levels of representation in TG. For this purpose, we use the prefixes: L-, S- and I-. These prefixes are used to separate the parts of GLARF FSs that represent regularized structure from those which represent unregularized structure. Furthermore they show precisely how such structures are related. Making this separation is basic to the very idea of what a regularization is and therefore essential to any formalism that is trying to capture regularization.

Unprefixed role labels mark constituents which are not subject to any regularization: (these roles are simultaneously surface and logical), e.g., in Figure 1, "we" is both the logical and surface SBJ of "would like". If prefixed with an S-, a role represents a surface relation, but not a logical one (e.g., the "surface" subject of a passive), e.g., in Figure 2, "a discordant note" is the S-PRD (surface predicate complement) of "is" (as discussed below, additional mechanisms are used to regularize it to L-SBJ). If prefixed with L-, the role represents a logical grammatical role: the role represented by the set of all regularizations that we have applied. In Figure 1, "we" is the L-SBJ (but not the surface SBJ) of both "to apologize" and "having caused".[7] Finally roles prefixed with I- (intermediate) represent some GR resulting from a regularization that is not the final one we applied. Thus in Figure 5 "ropes attached to the bells", "ropes" is I-SBJ and L-OBJ of "attached" (due to the passive). It is also the (surface and logical) head of the NP. Similarly, in "Which aliens did you say were seen?", where "which aliens" would be the I-SBJ of "were seen", its I-SBJ status is required to account for verb agreement.

6. The Filler/Gap Relation

As noted previously, there are at least two ways to represent a filler gap relation: (1) an ec coindexed with an antecedent; and (2) multiple arcs with a single head (structure sharing). Although Figures 1 through 4 use the latter method, either method is possible in GLARF. For simplicity, we will assume structure sharing.

As discussed above, the regularization of passive to active or the filling in of a "missing" subject, amounts to the strategic placement of logical and surface arcs in GLARF FSs. However, there are a number of important details that require what we call "gap typing" to adequately describe the regularizations involved. The infixes -GAPPING-, -PASS-, -ING-, etc. that are in the arc labels serve to type the gaps, i.e., they allow the user to know which regularization brought the gap into being. This is crucial to any

[6] The framework also makes it possible to express different theoretical views about the identity of a functor. For example, patterns of exceptional word order suggest that the adjective may be the functor of the A-POS relation. In English, Spanish and all other languages that we are aware of open class nouns do not vary their word order with respect to a given adjective. However, exceptional adjectives do ("president elect", "secretary general" "buen ciudado prenatal" = good prenatal care).

[7] GLARF logical relations depend on which regularizations we apply for a particular application, rather than fixed linguistic principles. Therefore, our system may use different sets of logical relations than RG.

complete regularization analysis because different regularizations have different properties.

Consider the filler/gap constructions in 1-4 ("e" = gap):

1. [Mary]$_i$ wants e_i to leave. (simple case)
2. They spent$_i$ $325,000 in 1989 and e_i $340,000 in 1990 (sloppy identity)
3. [Sally, Mary claims e$_i$, is a Martian spy]$_i$ (a cycle?)
4. [The man]$_i$ [in [whose$_i$ house]]$_j$ I slept e$_j$ (gap + obligatory pronoun coreference)

In (1) the same entity is the wanter and the leaver. In (2) (diagrammed in figure 3), "spent" and the gap represent distinct spending events. In (3) the filler of the gap is the entire sentence, with "Mary claims e$_i$" removed (otherwise the filler contains the gap it fills). In (4), the relative clause construction combines gap filling and obligatory pronoun coreference. Thus if a user knows that a logical arc represents a -GAPPING- regularization, then the gap must be treated differently for some applications, e.g., tracking events (one instance of "spent" represents two spending events, not one). Similarly, for a Question Answering program to answer "What did Mary say?" from (3), the type of the gap in (3) is important. To fill the gap without going into an infinite loop, the program would have to ignore the parenthetical clause immediately dominating the gap of type -PAREN-. And so on, for the various types of filler-gap relations. Of course for some applications, e.g., collecting verb noun pairs for selection restrictions, knowing the type of regularization may not be relevant.

Some gaps may be posited that have no fillers. However, they still may be useful for generalizing patterns. For example, virtually any transitive verb may drop its object when used habitually, e.g., "Babe Ruth really hits", or generically as in "That breed of dog bites". In these cases, one could assume a gap as the object of the verb, representing a generic NP. Assuming this gap would mean that a pattern (in information extraction, machine translation, etc.) for matching transitive verbs would match these cases as well. This is a larger issue for languages like Japanese, because arguments of verbs are dropped more frequently. For example, in

見ました。 ⟶ ***e1 e2 saw***

the subject (e1) and object (e2) are omitted. The resulting sentence means something like "someone saw something".[8] For these cases, the structure sharing analysis of gaps would not work and we would have to assume some sort of ec. The type of the gap would be on the arc dominating the ec.

7. Index Types

In Figures 2 and 4, we make use of index typing. Generically, GLARF has a mechanism for coindexing such that each INDEX arc has a numeric value and two constituents with the same number at the head of their INDEX arc are coindexed. We would use pairs of INDEX arcs for coindexing between coreferential pronouns or ecs (in alternative analyses to the structure sharing cases given above) and antecedents. However, there are other uses of coindexing which require subtypes of INDEX arcs and unless a formalism allows alternative modes of INDEXing, these phenomena cannot be properly represented.

Figure 2 models an instance of a pleonastic "there" construction. The EXP-INDEX of "there" is equal to the index of "a discordant note". "There" is at the head of a L-SBJ arc and "a discordant note" is at the head of an S-PRD arc. Under this analysis "a discordant note" is the predicate complement on the surface, but is the logical subject of the sentence. This sort of analysis is needed for all instances of grammatical pronouns like pleonastic "it" and "there" in English. A similar mechanism would be needed to cover clitic doubling in some languages, including Spanish. Some examples follow:

5. It$_i$ is hard [to understand plutification]$_i$
6. There$_i$ are [three people]$_i$ in the room
7. Le$_i$ gustan los libros a Jorge$_i$

The index type is necessary in order to distinguish different types of coindexing. In particular, pleonastic/antecedent chains are special in that the whole chain only gets a single logical role – the dummy pronoun is interpreted like a gap.

Figure 4 represents the paraphrase of (8) as (9):

8. Disney's unexpected acquisition of Apple Computers
9. Disney acquired Apple Computers unexpectedly

Index typing features prominently in this analysis and a formalism could not represent this paraphrase adequately without a similar feature. The (Harris-style) paraphrase is represented by the subgraph at the head of the P-NOM arc (a special arc representing nominal paraphrase). The NOM-INDEX arcs represent the relation between corresponding non-identical words in the NP and its paraphrase: acquired/acquisition and quick/quickly. Structure sharing is used to indicate identity (For simplicity, we will not assume tense in the regularized value of P-NOM). We have implemented such regularizations in previous work and will soon incorporate them into our GLARF procedures. This combines all of the devices above and adds a new non-relational arc type P-NOM. P-NOM is a paraphrase arc of type nominalization.

Additional index types would be needed if we extend GLARF to cover coreference among NPs and pronouns. While standard pronoun/antecedent coreference marks two NPs that refer to the same entity, other types of coreference are possible. In (10), 'they" is coreferential with John+Mary. If "John" and "Mary" had INDEXes of 1 and 2, "they" would have 1 and 2 as values of PART-INDEX1 and PART-INDEX2.

10. "John$_i$ asked Mary$_j$ if they$_{i,j}$ could leave together."

[8] It is sometimes difficult to decide whether optional elements are gaps or just optional. We solve this problem by only assuming that missing subjects, objects, direct objects and sentential complements leave gaps when omitted.

8. Implementation

As described in (Meyers et al., 2001a; Meyers et al., 2001b), we use a cascade of hand-coded transformations to convert text in Penn Treebank II format into GLARF. The input includes both human-produced treebank data and parser output. Our results on test data (hand-corrected GLARF trees for 65 sentences for which we did not tune our procedures) currently range from about 74.4% precision/76.3% recall for parser output to about 89.0% precision/89.7% recall for treebank data.

In our automatic procedures, the output of each transformation N is the input to transformation $N + 1$. As we research the details of capturing regularizations in GLARF, we add corresponding transformations to our system. Currently, the transformations do the following: (1) correct part of speech errors; (2) add verb groups; (3) disambiguate conjunction scope; (4) interpret function tags as combinations of grammatical roles and semantic features; (5) identify surface grammatical roles based on context in the tree; (6) identify antecedents of all unindexed empty categories, while identifying logical, surface and intermediate grammatical roles, and (7) add structure to represent additional regularizations. See (Meyers et al., 2001a; Meyers et al., 2001b) for more detail. All these procedures are based on tree structure, general syntactic knowledge and lexical clues derived from the COMLEX Syntax lexicon of English (Macleod et al., 1998a) and NOMLEX dictionaries (Macleod et al., 1998b).

Future implementation of nominalization regularizations will be based on previously implemented procedures described in (Meyers et al., 1998). We plan to extend the NOMLEX dictionary by automatic means, and then add nominalization regularizations to our GLARF procedures. Future work involving verb alternations will use rules that draw on a variety of sources including PropBank (website listed above), the LCS database (http://www.umiacs.umd.edu/~bonnie/verbs-English.lcs) and the online version of the verb lists from (Levin, 1993). (http://www.umich.edu/~archive/linguistics/texts/indices/evca93.index). We plan to use the procedures on raw parsed data for some applications. In addition, we plan to create a GLARF treebank using the GLARF procedures as a first step. We will correct the output of those procedures manually using ANNOTATE (Brants and Plaehn, 2000), the annotatation tool originally created for the Negra corpus, which we have adapted for working with GLARF.

9. Concluding Remarks

We have described important aspects of regularization that need to be modeled in any framework, particularly if that framework is to be used for a wide variety of NLP applications. We have enumerated the formal mechanisms that GLARF uses to model these features. We use arc types primarily to differentiate between different relational strata (logical vs. surface roles). Most generalizations over patterns will operate primarily on logical arcs, e.g., acquiring selectional cooccurrence patterns. Gap typing is required, particularly for natural language understanding applications like question answering and event tracking where the structure of each event is especially important. Finally, index types are important in order to clearly mark how coindexed items are related: are they coreferential; are they partially coreferential; is one a grammatical dummy that "stands in" for the other; is one a paraphrase of the other; etc. We believe that it is particularly important for a formalism to be expandable to cover not-previously-handled regularizations, especially common ones like nominalization. These are all important representational issues for PA analyses, particularly if one hopes to use PA structure to adequately capture paraphrase. GLARF seems particularly well-suited for this endeavor.

Acknowledgements

This research was supported by the Defense Advanced Research Projects Agency under Grant N66001-00-1-8917 from the Space and Naval Warfare Systems Center, San Diego and by the National Science Foundation under Grant IIS-0081962.

10. References

D. Blaheta and E. Charniak. 2000. Assigning Function Tags to Parsed Text. In *NACL2000*.

T. Brants and O. Plaehn. 2000. Interactive corpus annotation. In *LREC2000*, pages 453–459.

B. Carpenter. 1992. *The Logic of Typed Features*. Cambridge Univ. Press, New York.

N. Chomsky. 1973. Conditions on Transformations. In S. R. Anderson and P. Kiparsky, eds., *A Festschrift for Morris Halle*. Holt, Reinhart and Winston, New York.

G. Corbett, N. Fraser, and S. McGlashan. 1993. *Heads in Grammatical Theory*. Cambridge Univ. Press, Cambridge.

R. Fiengo. 1974. *Semantic Conditions on Surface Structure*. Ph.D. thesis, MIT.

C. Fillmore. 1968. The Case for Case. In E. Bach and R. T. Harms, eds., *Universals in Linguistic Theory*. Holt, Rinehart and Winston, New York.

J. Gruber. 1965. *Studies in Lexical Relations*. Ph.D. thesis, MIT. Reproduced by Indiana Univ. Linguistics Club, January 1970.

E. Hajiĉová and M. Ceplová. 2000. Deletions and Their Reconstruction in Tectogrammatical Syntactic Tagging of Very Large Corpora. In *Coling2000*.

Z. Harris. 1968. *Mathematical Structures of Language*. Wiley-Interscience, New York.

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

D. Johnson and L. Moss. 1993. Some Formal Properties of Stratified Feature Grammars. *Annals of Mathematics and Artificial Intelligence*.

D. Johnson, A. Meyers, and L. Moss. 1993. A Unification-Based Parser for Relational Grammar. In *ACL93*.

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

C. Macleod, R. Grishman, and A. Meyers. 1998a. COMLEX Syntax. *Computers and the Humanities*, 31(6):459–481.

C. Macleod, R. Grishman, A. Meyers, L. Barrett, and R. Reeves. 1998b. Nomlex: A lexicon of nominalizations. In *Proceedings of Euralex98*.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *1994 ARPA Human Language Technology Workshop*.

I. Mel'ĉuk. 1988. *Dependency Syntax: Theory and Practice*. State Univ. Press of New York, Albany.

A. Meyers, C. Macleod, R. Yangarber, R. Grishman, L. Barrett, and R. Reeves. 1998. Using NOMLEX to Produce Nominalization Patterns for Information Extraction. In *Coling-ACL98 workshop Proceedings: the Computational Treatment of Nominals*.

A. Meyers, R. Grishman, M. Kosaka, and S. Zhao. 2001a. Covering Treebanks with GLARF. In *ACL/EACL 2001 Workshop on Sharing Tools and Resources for Research and Education*.

A. Meyers, M. Kosaka, S. Sekine, R. Grishman, and S. Zhao. 2001b. Parsing and GLARFing. In *RANLP-2001*, Tzigov Chark, Bulgaria.

D. Perlmutter and P. Postal. 1984. The 1-Advancement Exclusiveness Law. In David. M. Perlmutter and Carol G. Rosen, eds., *Studies in Relational Grammar 2*. The Univ. of Chicago Press, Chicago.

C. Rosen. 1984. The Interface between Semantic Roles and Initial Grammatical Relations. In D. Perlmutter and C. Rosen, eds., *Studies in Relational Grammar 2*. The Univ. of Chicago Press, Chicago.