Interfacing Systemic Functional Grammars with Frame Semantics

Jeremy Dohmann, Terry Patten, Ceara Chewning, Joseph Campolongo

Charles River Analytics

625 Mount Auburn Street, Cambridge MA 02138

[jdohmann,tpatten,cchewning,jcampolongo]@cra.com

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

Interfacing Systemic Functional Grammars with Frame Semantics

Abstract

As computational applications play a greater role in society, Systemic Functional Linguistics can play an important role in producing effective interactions between those applications and the people who use them. Significant progress has been made in the fields of computational representations of semantics (e.g., FrameNet: Ruppenhofer et al., 2006) and automated semantic reasoning (e.g., the Semantic Web: Berners-Lee, Hendler, Lassila, 2001), and it will be important for computational systems based on Systemic Functional Grammars (SFGs) to integrate with these semantic representations. In this paper, we present an interface between SFGs and the semantic frame representations used in artificial intelligence in general (e.g., Minsky, 1974), and in FrameNet in particular. We describe the new realization rules and how they map between the grammar and semantic frames, and how the new realization rules and semantic frames can be used by computational systems to generate and interpret sentences that are both syntactically and semantically complex.

Keywords: Frame semantics, systemic functional grammar, realization rules, computational linguistics

Word count: 5698

Halliday’s (2003) Systemic Functional Grammar (SFG) framework already contains several mechanisms that address semantic issues in language. The framework has a full set of thematic or case roles (Agent, Instrument, Time, Place, etc.) that explicitly identify the semantic role of grammatical constituents. The framework also has a layer above the grammar, which Halliday (1978, 2007) calls the *semantic stratum*, that enables arbitrary semantic and contextual labels to be associated with combinations of grammatical features. These existing mechanisms are sufficient for informal analyses, where it can be assumed that the thematic roles have associated values, such as a person’s name, or the name of a location. When building computational systems that process language, however, these types of assumptions cannot be made—actual values must be assigned to the thematic roles somehow.

The thematic or functional roles in SFG are closely related to Fillmore’s (1968) Case Grammar. Each of these roles may also be associated with semantic categories. For example, the Agents associated with certain verbs may be required to be *human*, or *animate*. The set of thematic roles and their constraints is called a frame, and Fillmore led the development of an extensive database of frames for English verbs (Ruppenhofer, et al., 2006), and related projects have created frame databases for other languages. Frames are also used in artificial intelligence where values can be assigned to particular roles or *slots* in the frame—the Agent slot can be filled by a specific name, such as “John Smith.” This paper describes minor extensions to the SFG formalism that enables SFGs to interface with frame representations, formally associating thematic roles with specific values and semantic properties.

Systemic Functional Grammars and Semantics

SFG is a constraint grammar formalism developed by Halliday (2003). It presents the structure of a grammar as arising naturally out of a hierarchy of functional options, which Halliday represents as *system networks*, structurally similar to a decision tree. SFGs share a kinship with other constraint grammar formalisms developed within AI and theoretical linguistics, such as Lexical Functional Grammar (Kaplan & Bresnan, 1982) and Functional Unification Grammar (Kay, 1985). In SFG, structural constraints called “realization rules” are attached as properties to functional classes, such as *interrogative* or *passive voice*. These constraints are expressed in terms of the functional roles. For example, the class *interrogative* may have the realization rule that the Subject precedes the Finite verb.

There is some variation in the types of realization rules used in the SFG literature. Our project has adopted a set of realization rules based on those used by the Penman system (Mann & Matthiessen, 1983):

- **Lexification**: Role = “text” (e.g., Agent_marker = “by”). Here a role is directly assigned a specific textual value in the grammar.
- **Adjacency**: Role1 ^ Role2 (e.g., Agent_marker ^ Agent). Here, a role must appear immediately after another role.
- **Partition**: Role1 | Role2 (e.g., Determiner | Head). Here a role must appear after (but not necessarily immediately after) another role.
- **Conflation**: Role1 / Role2 (e.g., Agent / Subject). Here two roles are realized by the same constituent (i.e., the constituent has two roles).
- **Expansion**: Role1(Role2) (e.g., Mood(Subject)). Here a Role has one or more subroles that form a group.

- **Preselection**: Role : feature (e.g., Subject : singular). Here a role is assigned a specific grammatical or lexical class.

The top-level representation used by SFGs is the *system network*, a taxonomy of classes. An SFG typically contains a separate system network for each constituent type (e.g., clause, noun phrase, prepositional phrase, etc.). Each class in a system network can have zero or more realizations rules that constrain how that particular class is realized. System networks can have conjunctive branches, represented using a curly brace, enabling a constituent to be classified simultaneously along several dimensions.

The example in Figure 1 illustrates a very simple system network that represents 8 basic noun phrases: There are two alternatives for the Determiner and four choices for the Noun. Note that the top-level class (noun_phrase_root) always applies, so its realization rule (requiring a Determiner immediately followed by a Noun) applies in all cases: “a dog”, “the dog”, “a cat”, “the cat”, “a snake”, “the snake”, “a rat”, “the rat”.

Clearly this approach will not work for proper nouns. It is impossible to build a system network that contains all possible person names, organization names, etc. What is needed is a mechanism that enables a grammar to refer to semantic entities that have names so that references to those entities can be realized as proper nouns.

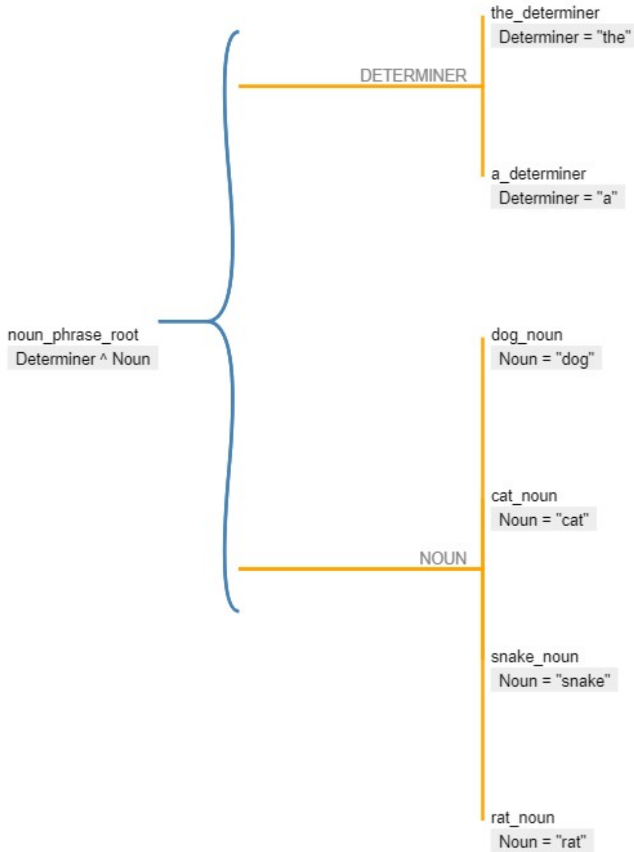

Figure 1. A simple system network illustrating classes and realization rules

Frame Semantics

Frame semantics, early forerunner to Government and Binding Theory's thematic filtering (Chomsky, 1993), was introduced by Charles Fillmore in his (1968) theory of Case Grammar. Frame semantics requires that for every verb class there is a set of mandatory roles and a set of optional roles (the frame) that describe the participants in the action described by the verb. Furthermore, specific roles may be required to be members of specific semantic categories (e.g., the Agent role may be required to be animate). These frame requirements are not arbitrary—they follow from the laws of physics and other real-world properties of the action denoted by the verb. If the frame constraints are not satisfied, the analysis does not make sense—

it is infelicitous. Fillmore's database of frames, FrameNet (Ruppenhofer, et al., 2006) explicitly specifies the frame requirements for many classes of English verbs. Figure 2 shows the mandatory roles in the "commerce buy" frame (from https://framenet.icsi.berkeley.edu/fndrupal/frameIndex). Verbs representing commercial buying (e.g., "buy," "purchase") require a Buyer and something that is being bought, the Goods. The Seller and the Money are optional semantic roles.

Core:

Buyer [Byr]	The Buyer wants the Goods and offers Money to a Seller in exchange for them. Jess BOUGHT a coat. Lee BOUGHT a textbook from Abby.
Goods [Gds]	The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction. Only one winner PURCHASED the paintings

Figure 2. The core "commerce buy" frame entry from FrameNet

The frames in FrameNet represent constraints over the participants in linguistic events via a typology of roles, similar to the way theta criteria are used in Government and Binding's and X-bar theory's case filtering (Chomsky, 1993).

In the field of artificial intelligence, frames are a ubiquitous semantic representation, lauded because of their simplicity and their semantic compositionality (e.g., Minsky, 1974). Though Minsky's conception of frames has seen application in areas far removed from linguistics (e.g. computer vision), their motivating principle is still the same: world knowledge is stored in terms of situations and expectations. Everyday events have associated models (frames) with expectations about the normal participants in that type of event. Therefore, narrowly construing the expected participants or elements of the event type enables useful assumptions and predictions to be made that greatly enhance the ability to make inferences about events of that type.

In artificial intelligence, frames are used as a knowledge structure that acts either as a template (as in FrameNet), or as specific instances of those templates. In the example shown in Figure 2, we can create an instance of the Commercial Buy frame containing specific Buyer and Seller values assigned to the specific slots in the frame:

Commercial Buy	
Buyer	"Lee"
Seller	"Abby"
Goods	"Introduction to Chemistry"

Figure 3. An instance of the Commercial Buy frame

Of course the character-strings "Lee" and "Abby" are references to specific entities that we may have information about: We may know they are human, and we may know their full names. This information can be represented with frames as well, as shown in Figure 4.

Commercial Buy	
Buyer	Frame19
Seller	Frame35
Goods	Frame87

Frame19	
Features	PERSON, SINGULAR
Given Name	"Abigale"
Nick Name	"Abby"
Family Name	"Smith"
Date of Birth	"21 December 1988"

Frame35	
Features	PERSON, SINGULAR
Given Name	"Lee"
Family Name	"Jones"
Date of Birth	"7 April 1995"

Frame87	
Features	TEXTBOOK, INANIMATE
Title	“Introduction to Chemistry”
Author	Frame54
Published	“January 2018”

Figure 4. A more detailed frame structure

Frames, in our approach, consist of a set of *features* and set of *slots*. A semantic feature is a character string denoting a binary semantic property (a frame either has this property or it doesn’t). A slot (referred to as a “frame element” or “FE” in FrameNet) is a substructure that contains either a literal value or a reference to a subordinate frame.

Consider the FrameNet frame for “taking” verbs (from https://framenet.icsi.berkeley.edu/fndrupal/frameIndex):

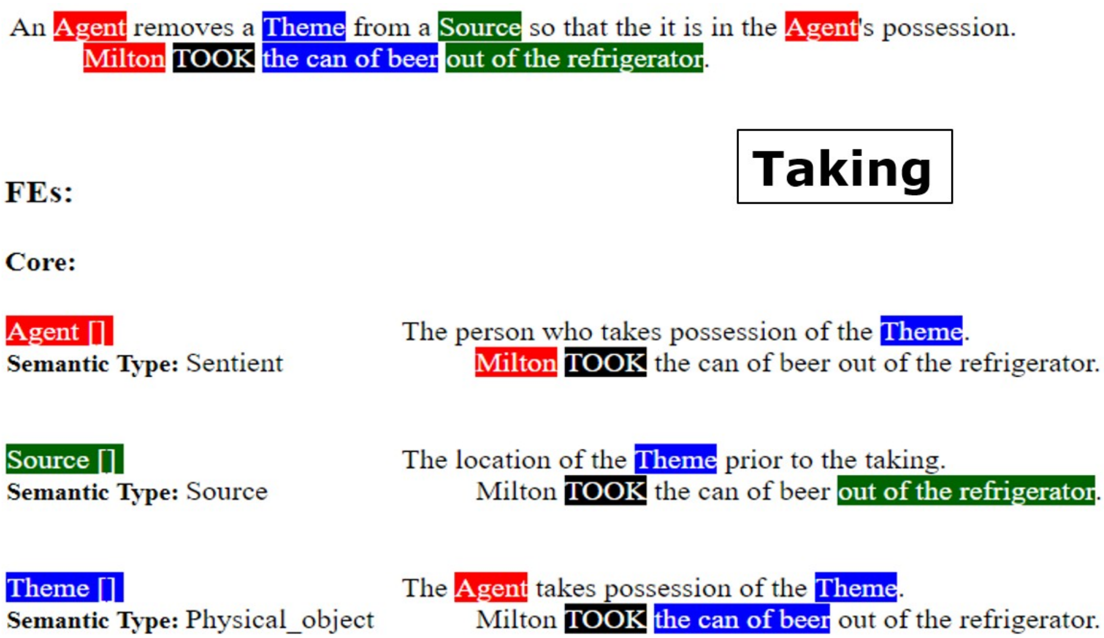

Figure 5. The “Taking” entry from FrameNet

For compactness, we can represent the general frame structure as a tree, as shown in Figure 6, rather than the traditional box structure shown in Figure 4. By convention, we write both semantic feature names and slot names in upper case.

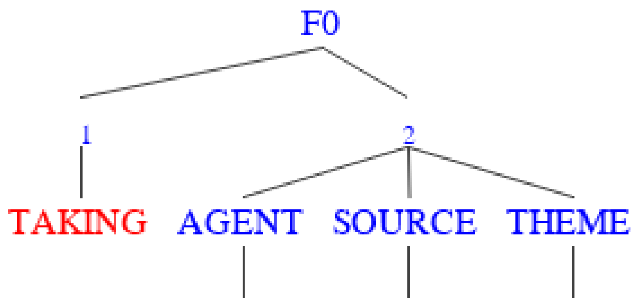

Figure 6. The general template for “taking” verbs

A particular instantiation of the frame, such as that corresponding to “Milton took the can of beer out of the refrigerator,” places specific literal values in the tree structure, as illustrated by the AGENT slot in Figure 7.

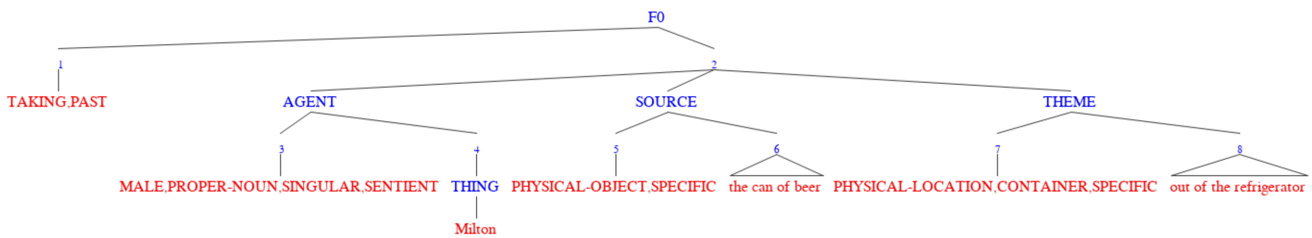

Figure 7. A particular frame instantiation

Frame Semantic and SFGs

Fillmore’s frame semantics and its implementation in FrameNet have proven to be useful resources for understanding semantic restrictions in language, and applying semantic restrictions in language processing systems. Given the close relationship between the thematic roles in Case Grammar and the functional roles in SFGs, it should be possible to interface SFGs with frame semantics. Our approach was designed with four core principles in mind:

1. Permit semantic selections on frame elements (as proposed by Fillmore’s case theory) from SFGs

2. Capture the flexibility, domain independence, and uniformity used in the many AI implementations of Minsky's frames
3. Permit semantic compositionality in a way that is faithful to Montague's work in formal compositional semantics
4. Modify the SFG formalism as little as possible

Our approach achieves these goals by adding two new semantic realization rules to the traditional SFG formalism.

Frame Features

Traditionally, frame representations do not have the type of features that we have adopted. Rather than having the feature SINGULAR, a frame would typically have a NUMBER slot with the literal value SINGULAR. Having to provide explicit slots of this type is inconvenient, and FrameNet does not provide them. Our features can therefore be viewed as convenient shortcuts that allow some common slot names to be left implicit and their values to be referred to directly.

New Realization Rules

First, interfacing frame semantics with an SFG requires that the functional roles in the grammar are tied to specific slots in a semantic frame:

1. **Unification**: Role @ SLOT (e.g., Agent @ BUYER, or Head @ NAME)

The utility of the new unification rule is most clearly illustrated when a frame slot contains the name of a person or a number. Clearly a lexicon cannot be expected to contain all possible names of people or all possible numbers. These values are inherently semantic—they can only exist outside the lexico-grammatical realm. Frames act as the portal between the semantic and lexico-grammatical realms. Unifying the role Agent, for example, with a slot containing a person's name is equivalent to lexifying that role with the same name—in both cases the realization of the role is determined.

2. **Semselection**: Role $ FEATURE (e.g., Agent $ ANIMATE, Instrument $ PHYSICAL)

The other capability required is for classes in the grammar to represent specific semantic properties of functional roles. As with the traditional *preselection* rule, a class is being selected for the role; but rather than selecting a grammatical class, a semantic class is being selected. FrameNet specifies many semantic restrictions on frame elements that can be represented directly using semselections. In Figure 5, the Agent of a "taking" event must be sentient. This is can be represented by the semselection Agent $ SENTIENT, assuming that the functional role Agent in the grammar has been unified with the frame slot Agent: Agent @ AGENT. Figure 5 also specifies that the THEME slot must be a physical object. Suppose the SFG uses a different role, such as Medium, to refer to the object being taken. In that case, the role Medium would be unified with the THEME slot (Medium @ THEME), and the semantic constraint would use the grammatical role (Medium $ PHYSICAL_OBJECT). Since unification enables any functional role name to be tied to any slot name, there is no need for the SFG to use the same labels as FrameNet (or any other semantic resource), the roles in the SFG merely need to be unified with the corresponding slot names.

These examples illustrate how semantic constraints can be placed on frames corresponding to functional roles in the grammar: Given a "taking" event, we can place semantic constraints (via semselections) on the participants of the taking event. However, it may also be necessary to place semantic constraints on the "taking" process itself. Typically, there is no slot representing the process because the frame as a whole (the "taking" frame) represents the process. So there is no explicit slot corresponding to the functional role Process in the grammar.

This means that constraining the process semantically involves matching features of the "taking" frame itself, such as the feature SIMPLE_PAST in Figure 7. This is accomplished using semselection realization rules that have no role (role-less semselections): $ SIMPLE_PAST. SFGs often represent distinctions between a mental and a verbal Process, for example. The clause network could therefore have a feature *mental_process* that has the realization rule: $ MENTAL, requiring the frame corresponding to the event being described by the clause to have the feature MENTAL.

Adding unification and semselection realization rules to the established SFG formalism enables the grammars to be integrated effectively with semantic frame representations such as FrameNet.

The Relationship between Constituents and Frames

An important issue is the relationship between constituents in a parse tree on one hand, and the frames in the corresponding frame hierarchy on the other. Our approach is to assume a close relationship between these two hierarchies. In general, for every constituent in a parse tree we assume a parallel frame in the frame hierarchy that provides the semantics for that constituent. Note that SFG parse trees are much flatter than Chomskyan parses, so there are fewer constituents in an SFG parse. Furthermore, some lexical nodes in the parse tree may not have a corresponding frame because they do not represent any semantic content. For example, some prepositions (e.g., "by" as an agent marker or "to" as a location marker) merely indicate where semantic content belongs in a frame, but do not represent any semantic content themselves. A close relationship between grammatical and semantic structure facilitates the compositionality that is required to address complex semantics. By permitting frames to point to subordinate frames, we allow our semantics to mirror our syntax by storing semantic information as arbitrarily deep in a tree-like structure as it needs to be, projected up only to the frames (and corresponding constituents) where such information is relevant.

Just as Fillmore's vision was that frames are cognitively stored as a network general templates, easily accessible, evoked, and completed according to the exigencies of a situation, our framework permits descriptions of great generality to be instantiated to fully realized descriptions of great specificity.

Our system permits full generality over the literal values, the features, or the slot labels. This is in line with traditional approaches to frame semantics in AI where flexibility is key to adequate representation. This capability is particularly important for a practical, computational SFG tool, which may be required to work with a wide variety of lexical categories or with idiosyncratic, application-specific grammars.

Frame Semantics and Halliday's Semantic Stratum

Halliday's semantic stratum (Halliday, 1978; 2007) is a layer above the grammar that represents the relationship between the social context on one hand, and the grammar on the other. Preselections attached to features in the semantic stratum constrain the options available in the grammar. This makes it possible to represent register variation, so specific information will be realized differently in different social contexts. However, the semantic stratum, as described in the literature, does not support the representation of information about specific individuals, specific events, or specific values (e.g., numbers). Our frame semantics is intended to complement Halliday's semantic stratum by providing the means to represent this type of information about individual entities and events.

As described in the literature, the semantic stratum preselects grammatical classes. However, the semantic stratum has no way of preselecting classes for specific constituents because it does not have access to the constituent structure assembled by the grammar. For example, the semantic stratum cannot preselect a specific honorific for the Head of the Agent of an embedded clause because it has no knowledge of grammar, or what structures the grammatical stratum is working with. The semantic stratum can specify that professors should receive a specific honorific, for example, but it cannot know which specific grammatical constituents are professors. Frame semantics provides the ability to specify information about specific individuals and specific events, and the parallel constituent/frame structure makes it easy to keep track of which semantic properties apply to which grammatical constituents.

Consider the sentence “John took the hammer” and the “taking” frame shown in Figure 5. We want to represent that John is SENTIENT and the hammer is a PHYSICAL_OBJECT. We would need a preselection from the semantic stratum that encodes that the main clause has a sentient Agent and physical Object. How would we represent, in this fashion, a sentence such as “X hates that Y likes that Z feels sleepy”, where we want to enforce that X and Y have personhood and that Z has animacy? We would need to explicitly have, in the semantic stratum, a preselection for a clause of type “clause where agent has personhood with subordinate clause where agent has personhood with subordinate clause where agent has animacy.” This is impractical and would not address arbitrarily deeply nested clauses.

The frame semantic approach permits a grammar to enforce constraints locally, based on frames associated with specific constituents. Furthermore, by taking a compositional approach using a frame hierarchy, the top level frame can be agnostic regarding the constraints that are applied to subordinate frames.

While frames represent the semantic information associated with a clause, for example, the semantic stratum can determine which semantic information is realized and which is left implicit. Some registers may require spelling out all the details, while others may omit as much information as possible. If the semantic stratum preselects the features for an agentless passive clause, for example, the Agent is not be realized even if it is explicit in the frame. Conversely, if the Agent is not known, and therefore does not appear in the frame, an agentless passive is the only possible realization.

The Benefits of Interfacing SFGs with Frame Semantics

Interfacing SFGs with semantic representations in general—and with frame semantics in particular—has important practical benefits. Language instruction applications and computational applications of SFGs benefit from Halliday's semantic stratum because it explains how the social context influences lexico-grammar. However, grammar is similarly influenced by the semantics of individual entities and events, and being able to address these semantic issues has similar value for language instruction and computational applications. This section will discuss several important semantic applications and how they are addressed by frame semantics.

Thematic Filtering

Thematic filtering, broadly speaking, means ruling out a sentence as ungrammatical because its arguments have inconsistent semantic features associated with them (e.g. agency, number, person). This sort of filtering can be applied to prevent sentences such as: "They likes him" (Subject/Verb agreement), "The table reconsidered" (sentient Actor required for mental verbs) or "Bill commanded Jim to clean" where Jim is, in fact, Bill's superior. It is up to the practitioner how much semantics and pragmatics should be embedded in a grammar, but frame semantics facilitates whatever amount is desired.

Referring to Non-Grammatical Information

Typically, a system network for a clause may contain a class *singular_subject* with realization rules such as Subject : singular and Finite : singular, which, together with corresponding cases for plural_subject and mass_subject, enforce agreement between the Subject and the Finite verb. However, how can it be determined whether a noun is a mass noun? In some cases, this is lexical (e.g., “sheep,” “shrimp”), but in other cases it is semantic (e.g., un-named liquids). In the semantic case, getting the number issue right requires being able to refer to the properties of the individual entities.

Using World Knowledge

For applications such as language instruction, or building natural-language interfaces for computational systems, it is often useful to apply semantic restrictions of the form addressed by FrameNet. In practical applications, it is important that language makes sense as well as being grammatical and socially appropriate. Frame semantics is ideal for this purpose because the slots in a frame—and the restrictions on those slots—provide the specific semantic knowledge necessary to make the necessary decisions, without requiring arbitrary knowledge and arbitrary reasoning.

The semantic frames themselves, however, are external to the linguistic realm and may have arbitrarily sophisticated reasoning behind them. For example, a semantic deduction system may deduce that any frame with the semantic feature HUMAN should also have the features ANIMATE and SENTIENT. Our approach is agnostic regarding which slots and features frames contain, and our approach allows frames to contain slots and features that are not used for linguistic purposes, which enables the frames to be used for other purposes in an application.

Using Proper Nouns

In applications such as language instruction or computational systems, it is useful to represent explicitly how proper nouns influence language. In general, proper nouns cannot be stored in a lexicon because new proper nouns can be created at any time—they must originate outside the linguistic realm. Frame slots can contain the names of arbitrary entities or values, so unifying a grammatical role with a frame slot is a convenient way to import a proper noun into the linguistic realm, or to export a proper noun to the semantic realm.

The properties of the semantic entities are important as well. An entity’s number and gender are required when choosing pronouns in English, for example, and these properties are easily represented as frame features.

Generating Text with SFGs and Frame Semantics

One important application for SFGs is generating text that realizes a given semantic representation. In this case, a semantic frame, or collection of frames, is the input to the generation process. Suppose the goal is to generate text for the frames shown in Figure 8.

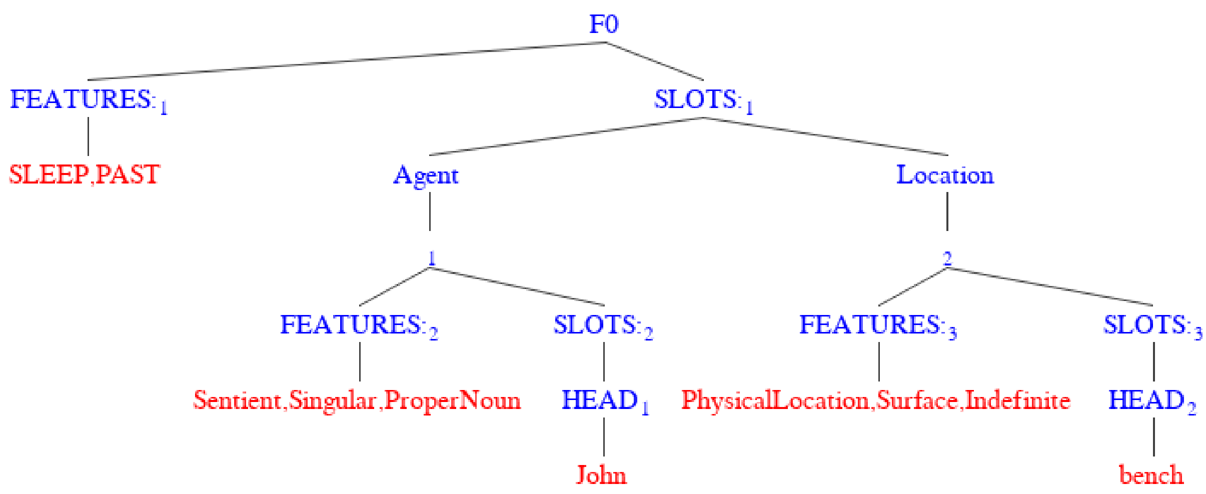

Figure 8. Example input to a text-generation process

In this case, frame semantics provides several pieces of semantic information that are critical to generating the appropriate text. First, the top-level features indicate the verb class and tense information necessary to realize the main verb. SLEEP here represents a verb class, not a literal realization; it could also have been represented as a Roget number, for example. Exactly which member of the verb class is used to realize the verb will depend on preselections from the semantic stratum reflecting the social context. In any case, all members of this verb class require a SENTIENT Agent, which the frame confirms. Second, the AGENT frame provides the name of the Agent, which may be needed if the Agent is not realized as a pronoun. The grammar accesses this information by unifying the Agent role in the grammar with the AGENT slot in the frame (Agent @ AGENT). It also provides the gender and number of the AGENT in case it does need to be realized by a pronoun, information that is accessed through semselections on the Agent role (Agent $ SINGULAR). Whether the Agent is pronominalized is a purely a linguistic issue independent of the semantic representation. Third, the LOCATION frame provides the semantic information that a bench is a SURFACE, which is key to choosing the preposition “on” in the grammar (Location $ SURFACE). It also indicates that the location is indefinite, which is key to choosing the correct determiner (Thing $ INDEFINITE). The end result is that sentences such as "John slept on a bench" or "He snoozed on a bench" can be generated depending on further constraints from the semantic stratum.

Parsing Text with SFGs and Frame Semantics

Another important application for SFGs and frame semantics is the parsing of textual documents. In this case, the frames are a convenient output of the meaning derived from the text. When parsing, the frames must be *constructed* based on the unifications and semselections in the grammar. Initially, the set of grammatical features is inferred based on the lexifications and other realization rules associated with those features. Now that we have added unifications and semselections to those features, the corresponding frames can be constructed and filled. Consider an example sentence: "John's belongings were put out by the curb." An SFG will parse this as an agentless passive clause with a Patient and a Location. At the top level, the word "put" matches a lexification associated with the past-tense *put* verb, which semselects MOVE and PAST, resulting in those semantic features being inserted into the top frame (F0). Similarly, John's belongings will be parsed as a possessive nominal group with the Owner role being unified with the OWNER slot in the frame, which results in the proper noun "John" being inserted into that slot in the frame. Note that matching proper nouns requires having special lexifications that can match (unknown) proper nouns. For this purpose, we use lexification of the form Agent = <person>, which will match any person name. In practice, this is accomplished by having a name recognizer place XML-style <person> tags around any person names in the text.

Importantly, the resulting semantic representation (shown in Figure 9) makes it explicit that there is a semantic AGENT associated with this event, but the contents of the AGENT slot are unknown.

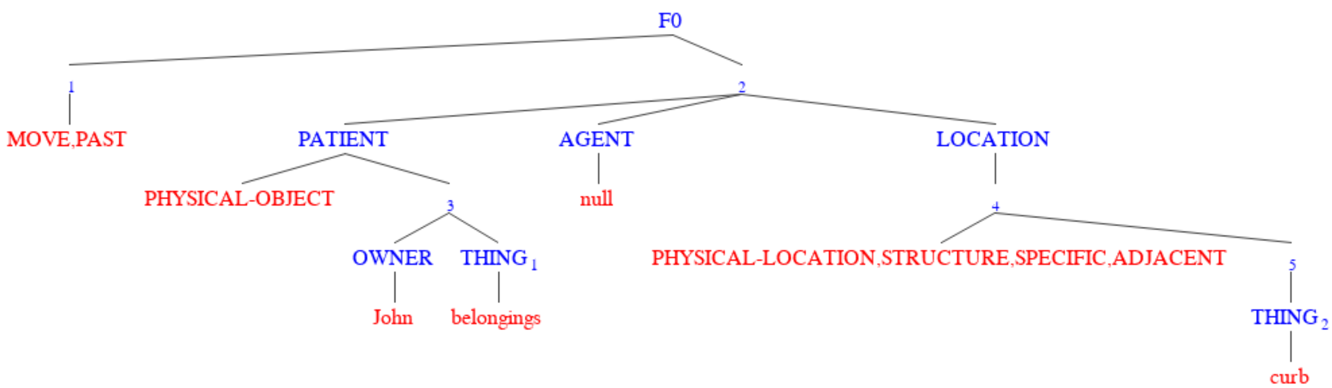

Figure 9. Frames derived from an SFG parse

Comparison to Other Approaches to Semantics

The approach to frame semantics described in this paper is compositional because it involves a hierarchy of frames, describing specific entities and events, that correspond to the major constituents in a syntactic structure. Semselection realization rules attached to classes in the grammar are therefore able to check any grammatically-relevant semantic properties of any of the specific entities and events. However, semselection rules can only check to see if a feature is present in a frame—there is no way to check the number of features that is present or to perform arithmetic over such values. In logical terms, this means that our frame semantics is equivalent to propositional logic, and is less expressive than the predicate logic often used as a semantic representation. This is important because propositional logic can be processed efficiently, while predicate logic is computationally intractable.

For example, number agreement in our approach involves matching the semantic number with grammatical number by having a feature, such as *singular_subject*, semselect the semantic number (Subject $ SINGULAR), and having *singular_subject* also preselect the number of the Finite verb (Finite : singular). In contrast, Functional Unification Grammar (FUG) allows arithmetic constrains of the form Subject.number = X, Finite.number = X, which handles the plural and mass number cases as well as the singular case, but these more general rules result in less efficient computation.

The SFG Toolkit

We are developing a computational implementation of SFGs, the *SFG Toolkit*, that includes the ability to define a semantic stratum, and the frame mechanisms described in this paper. One of the most useful and interesting characteristics of the *SFG Toolkit* is that the same SFG can be used for both generation and parsing. In the case of generation, a set of frames specifying semantic information is taken as input, and grammatical and situationally appropriate text (determined by the semantic stratum) is generated as output. In the case of parsing, a semantic frame hierarchy is the output of the process, with one frame being generated for each entity or event in the textual input. Since the frames provide semantics for each of these entities, the output frames can then be used by external applications.

We have implemented several grammars using the SFG Toolkit, some of which have been used for both generation and parsing. The unification and semselection realization rules have proven to be a convenient and effective approach to interface SFGs with frame semantics. While we have not yet computationally integrated our SFG Toolkit with the full FrameNet database, the frames we used are inspired by, and consistent with, FrameNet frames.

Future Work

Integrating Frame Semantics with the Semantic Stratum

There are several interesting directions that could follow from this research, some theoretical, and some computational. On the theoretical side, additional research is required to fully integrate frame semantics with Halliday's semantic stratum. Currently we treat these two important mechanisms as independent sources of guidance for the grammar. Choices in the grammar can be resolved based on preselections from the semantic stratum, or from semantic constraints from the frames. Both mechanisms interact with the grammatical stratum, but they do not interact directly with each other. In general, there may be interdependencies between the social context and semantics. It may therefore be necessary for the semantic stratum to reference the semantic frames as well as the grammar.

Extending Frame Semantics Computationally

One desirable extension of our *SFG Toolkit* would be a computational integration of our SFG Toolkit with the FrameNet database. This would enable our grammars to benefit from the extensive frame knowledge for specific verb classes already available in FrameNet.

Another direction for future work is the application of a semantic reasoner to frame semantics. Work on the Semantic Web project (Berners-Lee, Hendler, & Lassila, 2001) has produced semantic reasoners that can perform deduction and some other types of inference highly efficiently. These reasoners are specifically designed to reason over classes and class properties. Currently, if the grammar needs to check if a human Agent is ANIMATE, the ANIMATE feature must be explicitly in the frame for every person so the semselection (Agent $ ANIMATE) matches the frame features. An alternative is to implement the frames within a semantic reasoning system so that features such as ANIMATE can be inferred automatically. The reasoner can automatically deduce that any HUMAN entity is also ANIMATE, so the fact that a person is ANIMATE can be left implicit, to be inferred automatically by the semantic reasoner. Semantic reasoners can perform many types of inference that could be useful, including inferring that water should be treated as a mass noun because it is a liquid.

Conclusions

This paper has described a parsimonious and straightforward approach to integrating SFGs with frame semantics. This work has both theoretical and practical significance. Relatively little work has been done on how to address semantic issues in systemic functional linguistics. We have shown that adding just two new realization rules—unification and semselection—enables complex semantic issues to be addressed in a manner compatible with existing SFG conventions and notations. Furthermore, this approach does not require an idiosyncratic semantic theory, but rather integrates a well-established semantic theory that has been widely used in both linguistics and artificial intelligence: frames. The approach to semantics described here is completely compatible with, and complementary to, Halliday’s semantic stratum. While the semantic stratum addresses socio-contextual issues, frame semantics addresses ontological issues that are outside the social and grammatical realms.

The work described here is of great practical importance because it enables SFG practitioners, including language instructors and translators, to address semantics within the SFG framework. It is also enables computational systems that use SFGs to address semantic issues when generating or parsing language. This work was motivated by, and has now been integrated into, the computational *SFG Toolkit*.

References

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. *Scientific American*, 284(5), Pages 28-37.

Chomsky, N. (1993). *Lectures on government and binding: The Pisa lectures* (No. 9). Walter de Gruyter.

Fillmore, C. (1968) The Case for Case. In Bach and Harms (eds.) *Universals in Linguistic Theory*. Holt Rinehart and Winston.

Halliday, M. A. K. (1978). *Language as Social Semiotic.* London: Arnold.

Halliday, M. A. K. (2003). *On Language and Linguistics.* Vol. 3 in the Collected Works of M.A.K. Halliday. Webster, J. (ed.). New York: Continuum.

Halliday, M. A. K. (2007). *Language and Society*. Vol. 10 in the Collected Works of M.A.K. Halliday. Webster, J. (ed.). New York: Continuum.

Kaplan, R. M., & Bresnan, J. (1982). Lexical-functional grammar: A formal system for grammatical representation. *Formal Issues in Lexical-Functional Grammar*, (47). Pages 29-130.

Kay, M. (1984, July). Functional unification grammar: A formalism for machine translation. In *Proceedings of the 10th International Conference on Computational Linguistics.* Pages 75-78.

Mann, W., & Matthiessen, C. (1983). Nigel: A Systemic Grammar for Text Generation (No. ISI/RR-83-105). University of Southern California, Information Sciences Institute.

Minsky, M. (1974) A Framework for Representing Knowledge. MIT Artificial Intelligence Memo No. 306.

Ruppenhofer, J., Ellsworth, M., Petruck, M., Johnson, C., and Scheffczyk, J. (2006). FrameNet II: Extended theory and practice. Retrieved from: http://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf