Selectional restrictions in natural language sentence generation

Raymond Kozlowski, Kathleen F. McCoy, and K. Vijay-Shanker
Department of Computer and Information Sciences, University of Delaware
Newark, DE 19716, USA

ABSTRACT

Selectional restrictions place semantic constraints on arguments and account for the implausibility of sentences such as *Colorless green ideas slept furiously.* They have been used in natural language understanding for disambiguation and pronoun resolution. Not much attention has been paid to the relevance of selectional restrictions in generation. We discuss this issue in this paper. We also show how selectional restrictions can be naturally incorporated into our generation architecture and our notion of a lexico-grammatical resource.

Keywords: Natural language generation, Selectional restrictions, Lexical resources, Multilingual generation, Machine translation.

1. INTRODUCTION

The term *selectional restrictions* has been defined in various ways in literature. [1] defines them as "specifications of the legal combinations of senses that can co-occur," [3] as "general semantic restrictions on arguments," and [2] as "conditions on lexical insertion applied to arguments." Selectional restrictions have been used in natural language processing, but the emphasis has been on understanding, where they have been used for disambiguation and pronoun resolution. Selectional restrictions are also crucial for the generation process in order to make sure that words chosen to realize concepts are appropriate for the larger semantic context. Nevertheless, selectional restrictions have only been treated in a limited way in the generation literature. In this paper we investigate the use of selectional restrictions in generation and point out several challenges in their use that have not previously been discussed. We point out how specifying selectional restrictions within our generation architecture (described in [4], [5], and [6]) overcomes these challenges.

Selectional restrictions in understanding

Selectional restrictions have been successfully used in natural language understanding, which takes a sentence and attempts to come up with a semantic interpretation of it. Since words have multiple senses, one of the challenges in understanding is to select the proper sense for every word in the sentence. For instance, the verb *break* has several senses including fracturing, interrupting, halting, making known, etc. Those different senses place different selectional restrictions on the arguments. For instance, the selectional restriction of the fracture sense states that the theme must be a physical object while that of the interrupt sense expects a state. These selectional restrictions can be used to disambiguate the sense of the verb used in a sentence based on the type of the theme.

Selectional restrictions can also be used for pronoun resolution. For instance, in the sentence *Teresa broke it*, if we knew the sense of the verb then the selectional restriction could be used to rule out some otherwise possible referents of the pronoun.

In addition, they can be used for inferencing purposes. For instance, given the sentence *Jessica broke the ornament*, the selectional restriction on the fracture sense of *break* might allow a system to infer that *the ornament* is breakable.

When a natural language processing system employs selectional restrictions, they are generally part of the system lexicon. This is the system's repository for lexical information. Generally, a lexical entry contains a word (used in one of its senses), syntactic information about how the word might fit into a larger syntactic context, the word's selectional restrictions, and its "meaning."

In understanding, we use the lexicon to find all the possible senses of a word. Then, selectional and syntactic restrictions can be used to constrain the senses possible in the given sentence. Once the sense is identified, the word meaning can be used in building up the semantic interpretation of the input sentence.

Selectional restrictions in generation

As opposed to understanding, generation takes a semantic input and produces a sentence constituting its realization. It consists of three parts: the decomposition of the input into pieces that can be matched with the meanings of items stored in the lexicon, selecting realizations of the matched pieces by selecting the words and syntactic constraints from the lexicon whose meanings match the decomposed pieces of the input (and whose selectional restrictions are satisfied), and putting the realizations of the pieces together to form a sentence.

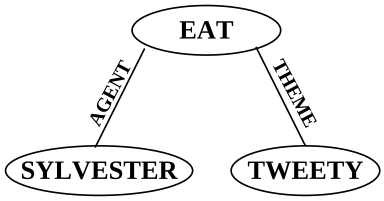

Figure 1: The semantic input underlying *Sylvester ate Tweety.*

The question we wish to consider is the nature of the selectional restrictions to be used in generation. While the type of selectional restrictions discussed in the previous subsection is very helpful from the point of view of understanding, it does not appear to be useful for generation, where the semantic input can be assumed to be well-formed. Notice that the selectional restrictions of the senses of the verb *break* were helpful in understanding because the verb (which was in the input to the understanding process) was ambiguous with respect to its meaning (the single lexical item mapped to multiple meanings) and the understander needed to know which of those meanings was correct in the given sentence. In generation, since the meaning is given, no such ambiguity exists. If the generator did test selectional restrictions like the one above, it would preclude generating sentences in cases where the semantic input was anomalous[1]. Thus, we argue, this kind of selectional restriction should *not* be tested by the generator.

This is not to say that no selectional restrictions should be tested in generation - indeed there are cases where selectional restrictions are necessarily tested in generation. This occurs when the generator faces choices in the realization of the semantic pieces of the generation input. While in understanding there are multiple senses possible for a single word, in generation there are multiple realizations possible for a single piece of the semantic input (e.g. a predicate). In determining which ones are appropriate in the given context, we may need to evaluate whether the lexical item fits in with the meaning of the *entire* semantic input. This would be evaluated with selectional restrictions, typically testing the type of an argument of the predicate being realized by the chosen lexical resource. Note that, at the time this choice is made, we must have access to the type information of the arguments, in order to test the selectional restrictions.

Generation starts with a semantic input. An input is commonly a hierarchical predicate/argument structure. A node in the structure contains a predicate. Subtrees, corresponding to arguments, are linked to the predicate by thematic roles. Consider the predicate/argument structure in Fig. 1. The top predicate `EAT` has two arguments, `SYLVESTER` and `TWEETY`, filling the thematic roles of `AGENT` and `THEME`, respectively.

Let us consider the realization of the top of this input: `EAT`. In German, there are at least two possible realizations of the predicate `EAT`: the verbs *essen* and *fressen*. Which one is appropriate depends on the type of the agent of the predicate (here: `SYLVESTER`). The selectional restriction of the main sense of the verb *essen* states that the agent must be a person while that for *fressen* states that the agent must be an animal[2]. Assuming that at the time of the choice we have access to the type of the agent, we can use the selectional restrictions to make that choice. Notice that testing this selectional restriction is crucial for generating an appropriate sentence: the sentence would appear anomalous if *essen* were chosen and Sylvester was a cat. The same selectional restriction, however, need not be tested to do parse disambiguation (although it may be useful in understanding for inferencing purposes and reference resolution).

Challenges for using selectional restrictions in generation

Selectional restrictions have been used in some existing generation systems ([9], [7]). In [9], lexical entries include semantic denotations that include type restrictions on arguments. These restrictions are tested when the lexical items are chosen in the realization of a semantic node.

There are a number of difficulties with existing approaches to using selectional restrictions in generation. First, there is no distinction made between selectional restrictions that must be tested and those that need not (and should not). Other systems have not made that distinction and therefore may test some selectional restrictions unnecessarily. We will make a distinction between selectional restrictions appropriate for generation and those appropriate for understanding (not to be tested in generation).

Second, there is an issue of access to the needed information. If the generator is going to choose the appropriate verb in realizing the input in Fig. 1, it must have access to the type information about the agent (at the time the verb is chosen). While it may be quite reasonable to augment the generator input with this type information in this case, there are cases which bring into question what information a generator must have available in order to make appropriate lexical choices (in the face of selectional restrictions).

[1] such as in papers discussing selectional restrictions, intentional contexts, e.g. *Joey believes teddy-bears talk*, or even ordinary speech, e.g. *A tomato can't move*

[2] Actually, the agent may be a person but then the verb has a meaning different from normal eating contained in our example input here.

As an example, consider the semantic input in Fig. 6, which underlies the sentence *Columbus went to America.* While there is no problem for this English sentence, this semantic input is ambiguous in German. In particular, the predicate `GO` might be realized as either the verb *gehen* or *fahren*, depending on the mode of motion. So, we might include a selectional restriction with the main sense of the verb *gehen* that states that the motion must be on foot, and one in *fahren* that states that the mode must be a means of transportation. Notice, however, that no information about the mode of motion is included in the input, and so it is not clear how this selectional restriction could be evaluated.

This leads us to conjecture that a sentence generator must have a limited access to a knowledge base containing information about the situation underlying the specific input to the generator in order to evaluate selectional restrictions necessary for lexical selection. A specific lexical item may be chosen on the basis of 1) its meaning (the piece of semantic content to be matched against the input), and 2) the fulfillment of its selectional restrictions (that may require looking up further information in the knowledge base associated with its realization)[3]. Other examples of selectional restrictions that illustrate the need for a knowledge base include one for the main sense of the verb *rent* which states that the compensation is monetary and one for the main sense of the German verb *schenken* (give) which expects no compensation (given as a gift). Other generation researchers who have included selectional restrictions have not considered cases such as these ([9], [7]).

Even with a knowledge base available, the evaluation of selectional restrictions leads us to question how that knowledge base is to be accessed from a lexical entry in order to evaluate selectional restrictions. In the previous examples, the knowledge base could be accessed through the predicate realized by the resource (e.g. `GO`); the selectional restriction queries the type of an argument of that predicate (e.g. the agent). The situation is different, however, when modifier realization is considered. An example is the adjective *heavy* realizing intensification in phrases such as *heavy smoker* and *heavy drug use.* Notice that whether or not *heavy* realizes intensification depends on the predicate that it is modifying. For example, phrases such as *#heavy reader* and *#heavy affection* are not appropriate. To implement a selectional restriction on the use of *heavy*, the system would need to evaluate a selectional restriction on the predicate that is being modified (in this case, that it be an addiction). In most generation systems (e.g. [9]), the predicate being modified is not included in the lexical resource for realizing the modifier. Thus, it is difficult to see how such a selectional restriction could be stated[4].

The case of selectional restrictions in modifiers may be even more complicated. In particular, this is the case because lexical resources may realize only a thematic role (i.e. and not the predicate or the argument filling the thematic role). Such is arguably the case with the preposition *by* in by-phrases in passive sentences. The preposition itself may be regarded as realizing the `AGENT` role, with the object of the preposition realizing the agent argument. In German, there are two prepositions corresponding to *by*, each placing a different selectional restriction on the argument. The preposition *von* is used when the agent is animate; otherwise, *durch* is used. Thus, in general, at the time the preposition is chosen, in order to evaluate selectional restrictions of such a resource, the generator must have access to both the predicate being modified and the argument filling the thematic role.

In the remainder of this paper, we describe how our generation system incorporates selectional restrictions. Our solution overcomes the difficulties we have just described. First, we present a brief overview of our generation architecture, particularly its most relevant aspect for selectional restrictions, lexico-grammatical resources. Then, we address incorporating selectional restrictions into our generation architecture, including the complications discussed here.

2. OUR GENERATION ARCHITECTURE

In [4], [5], and [6], we developed a simple generation architecture. The architecture is totally driven by the input. It requires selecting a lexico-grammatical resource to realize a top portion of the input, recursively realizing arguments and modifiers, and then putting the realizations together. Central to our approach is the notion of a lexico-grammatical resource for generation, which consists of three parts: the semantics realized with appropriate selectional restrictions (which may access the knowledge base from the realized semantics), the syntactic unit (which contains lexical items and the syntactic consequences of their choice), and the mapping between semantic and syntactic constituents (made use of in the recursive realizations of subpieces and in putting realizations together).

We have developed a set of principles, motivated by semantic and syntactic completeness considerations, that dictate what the lexico-grammatical resources

[3] Of course, the syntactic constraints on the lexical items must also be evaluated.

[4] We should note that it would be possible to consider *heavy* as realizing a predicate, with the noun phrase it modifies realizing the predicate's argument. Then, the selectional restriction would again be on the argument. That, however, would require a non-hierarchical input to allow realizations such as *The heavy smoker ate Tweety.*

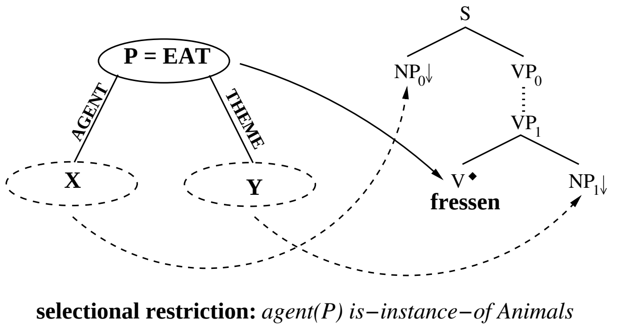

Figure 2: A lexico-grammatical resource for the predicate `EAT` realized by the German verb *fressen* (eat)

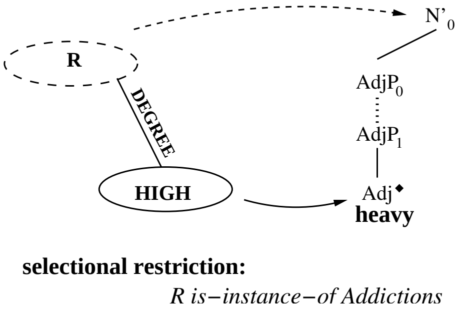

Figure 3: A modifier resource for intensification realized by the adjective *heavy*

look like. In this paper, we will simply concentrate on the semantic side and its consequences with respect to selectional restrictions. An example of a lexico-grammatical resource is given in Fig. 2. It realizes the predicate `EAT` (with `AGENT` and `THEME` thematic roles) using the German transitive verb *fressen* in the active form configuration. The semantic side (the left-hand side) includes the predicate `EAT`, the `AGENT` and `THEME` roles, and the uninstantiated agent and theme arguments (`X` and `Y`)[5], which will themselves be recursively realized.

Another kind of a lexico-grammatical resource, a modifier resource, is shown in Fig. 3. It realizes intensification using the adjective *heavy* as a pre-modifier in a noun phrase. The semantic side includes the uninstantiated predicate being modified (`R`), the thematic role `DEGREE`, and the argument `HIGH`. The inclusion of the uninstantiated predicate (motivated by the principles) enables the appropriate selectional restriction in the use of *heavy* to be stated.

[5]Arrows from the semantic side to the syntactic side provide the mapping between semantic and syntactic constituents. The syntactic side is the elementary structure of a D-Tree Substitution Grammar ([8]), anchored by the verb *fressen*. This structure includes substitution nodes NP_0 and NP_1 for the subject and direct object. This is where the realizations of the arguments are to be substituted. The mapping includes links between the uninstantiated arguments and the substitution nodes.

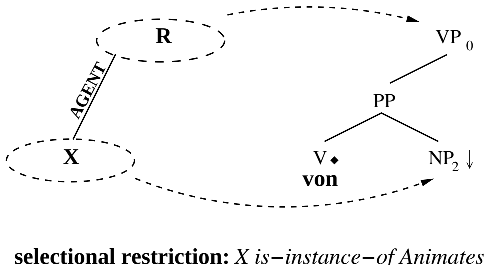

Figure 4: A modifier resource for the `AGENT` role realized by the German preposition *von* (by)

Another kind of a modifier resource, one realizing only a thematic role, is shown in Fig. 4. It realizes the `AGENT` role using a by-phrase in German, anchored by the preposition *von* and includes the uninstantiated predicate being modified (`R`) and the uninstantiated argument filling the `AGENT` role (`X`). The appropriate selectional restriction can be stated.

3. DISCUSSION: SELECTIONAL RESTRICTIONS IN OUR GENERATION ARCHITECTURE

The way we have defined our lexico-grammatical resources provides a suitable domain of locality over which selectional restrictions can be stated. This allows us to include selectional restrictions in lexico-grammatical resources and to test them as part of the matching process of the resource against the piece of the semantic input. We discuss in this section how the complications discussed in the Introduction have been overcome in our system.

What selectional restrictions are necessary in generation?

In both understanding and generation, selectional restrictions can be used to reduce ambiguity. Selectional restrictions for understanding are/should be employed in cases where a single word (phrase) maps to several different semantic meanings and semantic properties of other portions of the phrase may disambiguate this meaning (such is the case with the *break* examples). We argue that a generation system should not test such selectional restrictions. They are helpful in the understanding process because they rely on the integrity of the underlying semantic meaning.

In generation, the semantic input is given and the generation system itself is (and should be) powerless to alter that input. Thus, the assumption by the system should be that the semantic input is perfectly well-formed (and contains no semantic anomalies). There is nothing to be gained by the generation system in testing the selectional restrictions for understanding. In fact, doing so would make the generation system unable to generate anomalous sentences (such as *Col-*

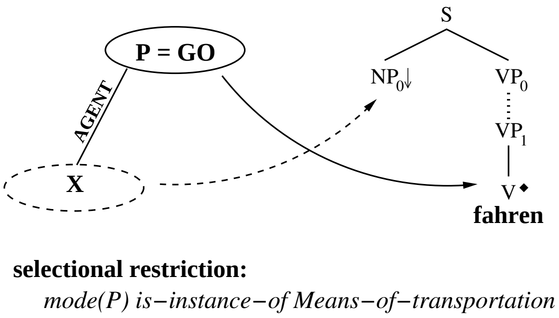

Figure 5: A resource for the predicate `GO` realized by the German verb *fahren*

orless green ideas slept furiously.). Thus, selectional restrictions for understanding are necessary when (1) a single lexical item (in a syntactic context) maps to several meanings (and thus which meaning is intended is ambiguous given just the lexical item in a syntactic configuration) and (2) there are semantic restrictions that it is possible to state on other elements in the wider context, that can differentiate the meaning.

Selectional restrictions for generation, on the other hand, deal with ambiguity in the opposite direction. Such ambiguity occurs when more than one lexical entry in the lexicon contains the same semantic realization (meaning), but the words and/or syntactic contexts in which they occur are different. This ambiguity does not in itself require a selectional restriction for generation - after all, paraphrases are possible. Nevertheless, in the face of this ambiguity, a selectional restriction for generation is necessary if such a lexical item places further constraints on the semantics outside of the semantics realized by the lexical entry. Such is the case with *essen* vs. *fressen*, both realizing the predicate `EAT`. The first requires the agent to be a person while the second expects an animal. We argue that only selectional restrictions for generation should be tested in the generation process.

Knowledge base for selectional restrictions
As we pointed out in the Introduction, testing selectional restrictions may require knowledge absent from the semantic input. We argue that, in addition to the semantic input indicating what is to be realized, a generation system must have access to a knowledge base containing the underlying information about the concepts being realized by the input. Thus, for each node in the input, we have a corresponding entry in the knowledge base. Consequently, the input can actually be thought of as a subset of the knowledge base[6].

[6]We assume the knowledge base to be frame-based and consist of frames corresponding to instances and classes. Instances are linked to the classes they belong to using `instance-of` links. Classes are linked to their superclasses using `is-a` links. Thematic relationships hold between instances.

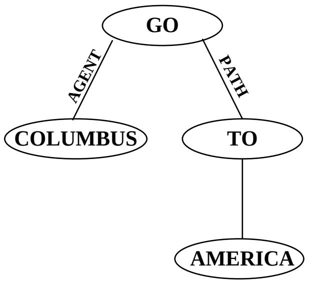

Figure 6: The semantic input underlying *Columbus went to America*

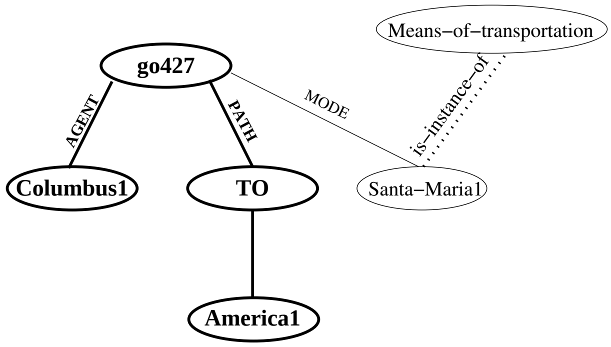

Figure 7: A knowledge base with part corresponding to the input in Fig. 6 in bold

A generation selectional restriction may consist of a query to the knowledge base that is statable given the access to the knowledge base provided by nodes unified with the semantic realization in that resource. Consider the resource for the German verb *fahren* (go) in Fig. 5. The predicate realized is labeled by the variable `P`. When the resource is matched against the input in Fig. 6, `P` is unified with the `GO` predicate in the input. That node has a corresponding instance in the knowledge base in Fig. 7, `go427` (note that the portion of the knowledge base corresponding to the input is shown in bold; the remaining part (not to be realized) corresponds to additional information about the mode). The function `mode` takes this instance and returns the instance in its `MODE` role, `Santa-Maria1`. The relation `x is-instance-of y` holds iff we can get from `x` to `y` by following one `instance-of` link and zero or more `is-a` links. `Means-of-transportation` is a class. We can see that the relation `mode(P) is-instance-of Means-of-transportation` holds in this case.

Access to the knowledge base for selectional restrictions
Recall that selectional restrictions for generation may be any knowledge base query to be evaluated given access provided to the knowledge base by unifying the semantics of a lexicon entry against the input. This seems straightforward for our previous example, but

it places conditions on what must be included on the semantic side of a modifier lexical resource.

In particular, note that a selectional restriction for generation of a modifier resource may have to refer to the predicate being modified. Thus, in testing such selectional restrictions, we must have access to the predicate being modified and the corresponding instance in the knowledge base. This means that the predicate being modified must be referenced on the semantic side of the lexico-grammatical resource. This is the case with our modifier resources. Consider the modifier resource for intensification realized by the adjective *heavy* in Fig. 3. The predicate being modified is labeled by the variable `R` (although the dashed oval in the figure indicates that `R` is not actually realized by this resource). In matching this resource against the input, `R` is unified with the predicate being modified. The corresponding knowledge base instance can then be tested as to whether it is an instance of addictions.

A selectional restriction for generation of a modifier resource may also refer to the argument filling the role being realized. Thus, in testing such selectional restrictions, we must have access to the argument filling the role and the corresponding instance in the knowledge base. Consider the resource for the German preposition *von* (by) in Fig. 4. The argument filling the realized `AGENT` role is labeled by the variable `X`. In matching this resource against the input, `X` is unified with the agent argument. The corresponding knowledge base instance can then be tested as to whether it is animate. A similar resource, anchored by the German preposition *durch* (by), would include the negated version of the selectional restriction in Fig. 4.

We should point out that if we treated the preposition as part of the resource for the verb anchoring the clause in the passive voice, the selectional restriction would have to be listed in all verb resources participating in the passive with a by-phrase, instead of just the two different by-phrases. Thus, we prefer to treat a by-phrase as a separate resource from the clause anchored by the verb. Notice that in this case a selectional restriction affects the generation grammar. The selectional restrictions support our particular analysis of by-phrases.

4. CONCLUSIONS

Selectional restrictions for generation are an important component of a sentence generation system. Previous work in the generation community has not carefully treated selectional restrictions in generation. In this paper, we have attempted to provide a thorough treatment of this issue. In doing so, we have divided selectional restrictions into two classes: selectional restrictions for understanding (which we argue should be tested in order to produce the correct semantic interpretation of an input sentence) and selectional restrictions for generation (which we argue must necessarily be tested in order to generate appropriate realizations of a semantic input). While information in selectional restrictions for generation may be helpful in the context of understanding (e.g. in making additional inferences or in reference resolution), we argue that selectional restrictions for understanding should *not* be tested in the generation process.

We have discussed several features of selectional restrictions for generation which apparently place some constraints on a generation architecture. These include access to a knowledge base containing additional information about the concepts to be realized by the generator. These features also place some constraints on the semantics realized by a lexico-grammatical resource (such as that the predicate being modified must be referenced on the semantic side of a lexico-grammatical resource). The architecture we have developed and the properties of the lexico-grammatical resources it employs are consistent with the features identified for selectional restrictions for generation.

6. REFERENCES

[1] James Allen. *Natural Language Understanding.* The Benjamin/Cummings Publishing Company, Inc., Redwood City, California, 1994.

[2] Noam Chomsky. *Aspects of the Theory of Syntax.* MIT Press, Cambridge, Massachusetts, 1965.

[3] Ray Jackendoff. *Semantic Structures.* MIT Press, Cambridge, MA and London, 1990.

[4] Raymond Kozlowski. *Utilizing the variety of lexico-grammatical resources in uni- and multi-lingual sentence generation.* Ph.D. dissertation proposal, Department of Computer and Information Sciences, University of Delaware, 2001.

[5] Raymond Kozlowski. Driving multilingual sentence generation with lexico-grammatical resources. In *2nd Int. Natural Language Generation Conference - Student Session*, 2002. To appear.

[6] Raymond Kozlowski. DSG/TAG: an appropriate grammatical formalism for flexible sentence generation. In *Student Research Workshop at the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. To appear.

[7] Nicolas Nicolov, Chris Mellish, and Graeme Ritchie. Sentence generation from conceptual graphs. In *3rd Int. Conf. on Conceptual Structures (ICCS'95)*, 1995.

[8] Owen Rambow, K. Vijay-Shanker, and David Weir. D-tree substitution grammars. In *Computational Linguistics*, 27(1):87-122, 2001.

[9] Manfred Stede. *Lexical semantics and knowledge representation in multilingual text generation.* Kluwer Academic Publishers, Boston, 1999.