Knowledge-Based NLP

LÁSZLÓ KÁLMÁN[♡]
LÁSZLÓ BALÁZS[♡]
MIKLÓS ERDÉLYI-SZABÓ[♡]
CSABA SZEPESVÁRI[♡]
VIKTOR TRÓN[♣]
KÁROLY VARASDI[♢]
DÁNIEL VARGA[♡]

[♡] *Mindmaker Ltd., Konkoly Th. M. út 29–33. H-1121 Budapest, Hungary.*
E-mail: {kalman,bazsi,mszabo,szepes,daniel}@mindmaker.hu

[♣] *Computational Linguistics Department, Universität des Saarlandes, 66041 Saarbrücken, Germany. E-mail: tron@CoLi.uni-sb.de*

[♢] *Research Institute for Linguistics of the Hungarian Academy of Sciences, P.O.Box 701/518, H-1399 Budapest, Hungary.*
E-mail: varasdi@nytud.hu

ABSTRACT: In this paper we report on Mindmaker Ltd.'s (Budapest) efforts of creating a novel type of NLP system, based on the full integration of a construction-based grammar with a higher-order reasoning system. Construction grammar allows us not to attribute strictly hierarchical structures to natural language utterances, and to control the understanding and production processes by declarative, not exclusively grammatical knowledge rather than separate parsing and generation algorithms.

KEYWORDS: Natural language processing, knowledge-based, construction grammar.

1 Introduction

Natural language processing (**NLP**) systems come in three varieties in terms of their architecture (cf. [2]): **sequential**, **interactive** (or **interleaved**) and **integrated**.[1] A sequential architecture is like a pipeline or an assembly line, i.e., the phases of natural language generation (NLG for short) or understanding (NLU for short) follow each other in a predefined order. For example, Fedder's [4] PENGUIN NLG system relies on a four-step sequence leading from an input by the dialogue manager to the text output. Interactive (or interleaved) architectures are much more complex, but also more difficult to manage, since they rely on interaction and feed-back between the various modules (for an example, see [10]). Finally, integrated models are characterized by a uniform representation format to which one single kind of procedure is applied again and again in an iterative fashion (for example, 'select target' → 'apply rule' → 'repair').

Ever since the beginning of the 90's, several authors have emphasised the need for an integrated approach to NLP. For example, [9] also claim speech repairs and other psychological evidence corroborate this view. Their system, called GLINDA, is a good example of the integrated view. Our approach is similar to theirs in some respects, but while GLINDA is rule-based, ours is **knowledge-based**.

By a knowledge-based system we mean one in which a significant part of the model's behaviour is described using an explicit declarative knowledge base stored by the system (i.e., attributed to the simulated agent). We believe 'knowledge-based' implies 'integrated', and more: just like in an integrated system, the very same procedure is repeated over and over, so the system's performance cannot be due to abilities stored in the form of algorithms applied one after the other, only to a stock of information stored in it. But we think of that stock of information as declarative knowledge rather than a rule system.

In particular, the behaviour of the system in a given situation is the result of the interaction between its current goal, the input data, and its knowledge about the relationship of the phonological build-up (henceforth, **form**) and the meaning characteristics of the utterances of a given language.

The present paper is an account of work that is mostly in progress, but that has already made steps towards such a full integration of knowledge representation (including world and linguistic knowledge) into NLP. Our aim is to explain how we plan to achieve this goal. Our approach is characterized by the following key features:

Key features of our approach

1. *Knowledge-based control*
   The control of the understanding/production process itself is not sharply distinguished from linguistic knowledge. That is, understanding/producing a linguistically articulated form with a concomitant meaning is **one unified process**. In our system, the knowledge necessary for controlling the understanding/

[1] The research reported here was partially funded by Hungarian State grant No. IKTA-00187/2000.

production processes is not procedural knowledge stored in the form of algorithms, but inseparable from 'grammar proper'.

2. *Goal-drivenness*
   Both understanding and production are **goal-driven**: the process aims at finding an answer to a question the simulated agent is 'interested in' (i.e., producing an utterance or establishing what the relevant aspect of the meaning of an input utterance is). In particular, instead of positing a universal endpoint that every instance of the understanding/production process must reach (e.g., building a parse tree, a formula etc.), in our system such a process is more like a gradual refinement and has no natural endpoint.
3. *Homogeneous knowledge representation*
   Linguistic knowledge and real-world knowledge are represented in the same format and treated in the same way. In particular, we do not make a sharp distinction between grammatically relevant ('phonological') and surface ('phonetic') distinctions in forms and, likewise, we make no distinction between grammatically relevant ('grammatical') and language-external ('encyclopedic' or 'contextual') aspects of meaning. This means that facts about perception and articulation that are related to language (though not to 'grammar proper', maybe) can be mixed freely with grammatical imformation, and the same applies to facts about real-world knowledge relevant to (a particular instance of) language use.

These key features are, at the same time, the main factors distinguishing our view from other existing approaches to NLP. We know of no similar endeavours in the field of natural language processing at all.

In what follows, we will start from the linguistic motivations and considerations (**Section 2**), then we will sketch the system architecture that we have designed and partially implemented (**Section 3**). Finally, we will elaborate on the difficulties and perspectives inherent in our approach (**Section 4**).

2 Linguistic considerations

2.1 Grammatical versus extragrammatical knowledge

According to the structuralist tradition dating back to de Saussure's [11] work, which also underlies generative grammar (i.e., all current theories of grammar since [1]) and modern computational approaches to NLP, linguistic signs constitute an autonomous system. According to this tradition, a linguistic sign is the association of grammatically relevant formal (phonological, syntactic etc.) properties with grammatically relevant semantic properties. But the very concept of 'grammatically relevant' must be re-examined and extended under an integrated view of NLP. In particular, facts related to the perception and articulation of linguistic forms and to real-world knowledge connected to meanings cannot be excluded from the concept of 'grammatically relevant' whenever they play a role in understanding and generation. As a consequence, we have to integrate grammatical knowledge with all these ‘extragrammatical’ types of knowledge in such a way that they can play their role in our simulation of the understanding/generation processes just like ‘grammar proper’ does.

Note that linguists themselves disagree on what exactly should belong to the ‘grammatical’ part of our linguistic knowledge. The standard generativist view has been challenged by authors like [3]. In phonology, the borderline between ‘phonological’ and ‘phonetic’ categories has been called into question by many (cf. [6]), and what should count as the ‘grammatically relevant’ aspect of meanings is entirely unclear. It is usually understood as ‘whatever is (can be) relevant from the linguistic point of view’, but that is far too obscure.[2]

2.2 Consequences for the grammar

Given the above arguments (or, rather, the ideological stance explained above), we will now start from the assumption that linguistic knowledge is not to be stated or described as a separate rule system that determines what ‘grammatical’ or ‘acceptable’ strings are, but as a system of **associations between form types and meaning types**, which we will call **signs** or **constructions** from now on (thus, we use roughly the same terminology as [8], [5] or [7]). If the association of a form type with a meaning type is at least partially unpredictable or unexpected, it has to be considered a construction on its own. In addition, we will assume that constructions may refer to ‘extragrammatical’ (or even ‘extralinguistic’) information related to both the formal and the semantic aspects of signs. What we might call ‘grammar proper’ within our homogeneous knowledge representation is information on how the signs can be used and combined. The actual processes that make use of them are under the simultaneous (and often competing) control of two different types of reasoning, namely, reasoning about forms and reasoning about meanings.

Let me now have a closer look at ‘grammar proper’. As we have just said, it consists of constructions, i.e., associations between form types and meaning types. As a consequence, what traditional grammar calls the lexicon is actually part of the grammar under the constructionist view, because every lexical item is accompanied by such an association. But, in addition to the constructions, we need to encode arbitrary **restrictions on their combination possibilities**. For example, consider:

---

[2] For one example of the discussion of this concept, see [12], who actually defends it. We are not going into a detailed criticism here.

EXAMPLE 2.1
Selection properties of *stop*

a. *Stop complaining!*
b. #*Stop to complain!* (out in the intended meaning)
c. **Stop complain!*
etc.

As these examples show, it just so happens that the lexical item *stop* can only be combined with a verb phrase in the gerund form (or at least it means something else otherwise). We will describe this in a positive way, namely, the construction of *stop* and the construction 'verb plus gerund verb phrase' are **connected** to each other.

It is easy to see why *stop* has a construction: it is entirely unpredictable why just this form is associated with this meaning. It is also obvious why 'verb plus gerund verb phrase' must be a separate construction: its form is (partially) unpredictable from its meaning, as witnessed by the fact that there other types of forms with the same meaning (e.g., 'verb plus to-infinitive', which is connected with verbs like *cease*).

Moreover, the grammar must specify that the connection between *stop* and 'verb plus gerund verb phrase' links *stop* to the main verb in the latter (rather than, say, the main verb of the gerund verb phrase). Therefore, we must think of connections as **labelled** with information on what to identify with what.

Finally, certain groups of connections constitute a **partition** over a domain of alternative constructions, and therefore they are mutually exclusive. For example, the construction of nominal descriptions may be connected to a number of article constructions by mutually exclusive connections. We call such systems of connections **dimensions**, e.g., which article we choose is a dimension of nominal descriptions.

As should be clear from the above, constructions are unlike the familiar rewrite rules of syntax. Production rule systems generate strictly hierarchical **constituent structures** for natural language expressions, in which every subexpression (except the entire utterance) belongs to (is immediately subordinated to or 'dominated by') exactly one constituent. Moreover, a context free rule system generates a structure in which the subexpressions immediately dominated by one and the same constituent are linearly ordered by immediate precedence. Nothing like these constraints is inherent in the constructionist approach. Instead, we only require that all formal and semantic aspects of an utterance be justified by some construction, but those constructions need not constitute a hierarchical structure, they may overlap, and need not refer to adjacent subexpressions.

For example, consider the following Hungarian sentence:

EXAMPLE 2.2
Hungarian example

1	2	3
Jánost	*Mari*	*szereti.*
John-ACC	Mary-NOM	loves-DEFOBJ

'It's Mary who loves John'

In order to justify the form and the meaning of this sentence, at least the following constructions are to be posited for it:

EXAMPLE 2.3
Constructions in (2.2)

1. '1' (*Jánost*) is the topic of the utterance, while the rest is the comment. This is why it is in front of the rest (form type), and has a consequence on the situations in which the utterance can be used successfully (meaning type);
2. '2' (*Mari*) is the focus of the utterance, the remainder is presuppositional. This has a consequence on the form of the sentence (this word must immediately precede the finite verb) and the possible uses of the utterance and its interpretation: its effect is roughly the same as the cleft construction in the English translation (*It's Mary who...*);
3. '2' (*Mari*) and '3' (*szereti*) stand in a subject/verb relation, as witnessed by subject agreement on the verb;
4. '1' (*Jánost*) and '3' (*szereti*) stand in a direct object/verb relation, as witnessed by object agreement on the verb (in terms of definiteness);
5. '2' (*Mari*) and '1 + 3' (*Jánost ... szereti*) stand in a subject/predicate relation, the semantic relevance of which is obvious.

The non-separation of grammatical regularities from the lexicon, inherent in the constructionist view of grammar, is motivated by the fact that the transition from the unpredictable form/meaning associations exhibited by words or morphemes, on the one hand, and by more or less 'abstract' syntactic or even rhetorical configurations, on the other, is arguably gradual. Many idioms (like *spill the beans*) exhibit regular syntactic behaviour, and many of them are so open-ended that they could as well be thought of as syntactic 'rules'. For example, consider resultatives:[3] it is not obvious whether the pattern of the predicates *hammer the metal flat* or *dance the floor shiny* is to be considered an idiom or a syntactic rule of English (cf. [7]).

The view of grammar explained above allows us, among other things, to posit constructions that play a role in the understanding, but not the generation of an utterance,

[3] In the so-called resultative construction of English, the argument that follows the direct object in a predicate corresponds to a property of the individual referred to by the direct object that it acquires through the process expressed by the main verb.

or the other way round. In particular, **formal clues** (i.e., perceptual features) that support our choices when understanding can be formulated as constructions (although they do not play a role in generation), and the same applies to **meaning clues**, i.e., semantic generalizations that help finding the appropriate construction to use during generation, but play no role in understanding. For example, nothing prevents us from partitioning phrase constructions into 'one-word' and 'multi-word phrases' (or even 'two-word phrases') if this helps us discriminate between phrase types during understanding. Similarly, it seems plausible to partition the types of referring expressions into 'pronominal', 'proper-noun' and 'description' phrases during generation. Such knowledge, which is usually claimed to belong to the 'heuristics' of understanding/ generation, is extremely difficult to incorporate into grammar in traditional linguistic theories.

In our system, however, we do not have separate 'production' vs. 'understanding' constructions explicitly. Instead, a mechanism which we need anyway takes care of this distinction. Remember that connections between constructions may constitute dimensions, in which case they are mutually exclusive, so a choice has to be made between them. Now, each dimension is coupled with a **discriminatory predicate**, i.e., one which one must be able to answer in order to make a choice within the given dimension. For example, if a dimension corresponds to a choice between 'one-word phrase' and 'multi-word phrase', then the discriminatory predicate will be 'the number of words in the phrase is x'. Trying to choose between alternative connections in this dimension only make sense if the value of x is known at the given moment, so we simply cannot use this dimension when in generating mode.

We have to make one more remark here concerning the approach we have taken to grammar. Most approaches to natural language grammar are hopelessly word- or head-oriented. For example, they explain the selection properties of *stop* (cf. (2.1)) as a 'subcategorization' property of this verb, i.e., they say *stop* 'requires' or 'selects' a gerund. That is, they account for structures like *stop complaining* in an inside-out manner, by holding a word in it (namely, *stop*, which acts as the 'head') responsible for their build-up. Under the constructionist view, on the other hand, such structures are described 'holistically', as it were: it is not *stop* that 'wants' a gerund, but the entire construction both 'wants' a verb (which, incidentally, can be *stop*) plus a gerund. Our choice between the head-based and the construction-based approach is not just a matter of taste. Only the latter can be used for building the kind of multiple simultaneous structures that we aim at (cf. 2.3).

There are also strictly linguistic considerations that favour the 'holistic' approach. First of all, linguistic constructs often have properties that do not stem from their alleged 'head' (whichever word in them one considers the 'head'). For example, the prepositional phrases *at the station* and *at stations* differ in the noun phrase argument of the preposition (which is not considered the 'head' in any mainstream theory), yet this difference influences the possible interpretation of a predicate like *wait* (*wait at stations* can only be interpreted habitually due to the prepositional phrase in it).

Similarly, noun phrases may have (syntactically and semantically relevant) properties mainly due to their determiner (e.g., definiteness, quantification), and others mainly due to their nominal 'head' (e.g., number, animateness). Even purely selectional properties may originate from non-heads, e.g., the preposition of a phrase that fulfils a locative function is selected by the noun in it: *live in a town*, but *be at the university*.

Second, the 'holistic' view allows us to state generalizations about entire structures independently of what concrete lexical elements they contain. For example, one can say that resultatives are obligatorily transitive in English, even if the lexical verb in them is not:

EXAMPLE 2.4
Transitivity in resultatives

We danced the floor shiny.

It is obvious that *the floor* acts as the direct object of *dance* just for the sake of the resultative construction. The head-based approach must assume the existence of a separate transitive verb *dance* (in addition to the normal, intransitive one) which, at the same time, has an obligatory resultative argument. This should be true, in addition to *dance*, for a virtually infinite number of verbs that potentially participate in resultative constructs. The construction-based view says that the resultative construction includes the verb, the direct object and the resultative phrase at the same time, and it only imposes semantic constraints on these components. Therefore, it predicts the more or less productive use of this construction (within the limits of the semantic constraints), independently of whether the verb is an existing lexical entry at all.

3 Architecture of the system

In this section, we will explain the main components and mechanisms through which we plan to implement a construction-based NLU and NLP engine in the near future. We will start with the core of the process and explain how it uses information originating from various sources.

3.1 The basic understanding/generation process

The most striking aspect of our understanding/generation module is that it performs essentially the same steps in both modes of operation, namely, it iterates through a cycle to be explained below. Both understanding and generation aim at constructing what we call an **utterance representation** (**UR**), which encodes (partial knowledge about) both the form and the meaning of the utterance we are concentrating upon. As a matter of course, it says more about form when in understanding mode, and more about meaning before generation starts, but nothing prevents us from fixing parts of the meaning before understanding (e.g., because we have fixed expectations) or parts of the form before generation (e.g., because we are playing a game or writing a poem, which implies using a fixed type of form). So the initial step of the understanding/ generation process consists in constructing an UR representing the input of the module (an input utterance form plus the logical representation of expectations, or a logical formula corresponding to the communicative goal, plus the eventual requirements on the output utterance form). Then the following cycle starts:

The understanding/generation cycle

1. **Evaluation.** The utterance representation must be examined and a decision must be brought as to what to do next (go further or return).
2. **Selection.** This consists in selecting constructions to be used (or reused, if we have selected them already). The selection is based upon the compatibility/ incompatibility of a given utterance representation with the constructions from among which we must choose.

The utterance representation contains construction **instances** (constructions themselves are like classes, and they live in the grammar; we sometimes say ‘construction’ for ‘construction instance’ if no ambiguity arises). The construction instances in it are to be combined in ways that are legitimate according to the grammar (as we have explained in **Section 2**).

The **evaluation** step decides whether the UR can be used as it is. We do not use the term ‘finished UR’, because what counts as a usable UR depends on the current **goals** of the system. So this is a chiefly goal-driven step. The typical questions the system tries to answer in this process are ‘Does this answer my question?’ and ‘Is this already pronounceable?’. If the evaluator is not happy with the UR constructed so far, it will point at the deficiencies (missing information) of the UR, in the form of dimensions (see **Section 2.2**) in which choices remain to be made, and passes that information over to the next step.

The **selection** step of the basic cycle corresponds to various steps of more traditional understanding/generation algorithms. In particular, it covers **lexical lookup** and the decision which rewrite rule to use when starting to express a given meaning. The variety of phenomena that selection covers is due to the fact that this step is always driven by the character of the information available at the given moment of constructing the utterance representation. For example, initially, most of the data at our disposal are about the input string in understanding mode and the logical representation of a communicative intention in generation mode (the latter will be referred to as the **content plan** or **CP** below); therefore, selection is driven by those data at the outset (although, as we have mentioned earlier, other types of data are not excluded). Later on, the previous evaluation step decides what to concentrate on during selection (as we will see, the evaluation step keeps a record of those parts of the UR which need to be further elaborated on).

In particular, in our example (2.2), the understanding step initially has access to the prosody/intonation contour of the sentence, so we must choose from among those utterance-level constructions which refer to prosody. Eventually we find the ‘topic/ comment’ construction, and we construct an UR containing (an instance of) just that construction, with 'topic' bound to the first word (*Jánost*), and 'comment' to the rest (*Mari szereti*). At later stages the same selection step will examine the comment part more closely, which involves lexical lookup: we have to check if a special comment-initial word is present (those constitute a closed class), and what the finite verb is (the finite verb may be preceded by the focus, in this case, the subject *Mari*).

Selection effectively means a choice between available constructions to incorporate into the UR. Given the limited amount of information focussed upon, the number of choices locally available is usually low (in our example, the utterance-level prosodic patterns are not very numerous), except, of course, when a lexical choice is to be made (looking up a word or a meaning). Another choice to be made is whether to create a new instance of the construction chosen or use one of its instances already present in the UR, if any. The general strategy is to give preference to reusing construction instances already in the UR rather than introducing new ones. In our example, when trying to decide what the subject of the main verb is, we could create a new construction instance representing it (subject pronouns can be dropped in Hungarian), but we decide to reuse the construction instance corresponding to the nominative noun phrase *Mari* for this purpose.

The selection step is one that can be made efficient using stochastic methods (by weighting the possible choices etc.), but we have not started that type of improvement yet.

Selection involves the calculation of the new form and meaning that belong to the 'new' UR (i.e., the UR after the necessary selection steps have been made). It is this part of the algorithm that effectively decides which option to choose. This is the only point where backtracking may be needed: if no contradiction-free new UR can be built, then another option must be picked. But after selection has been completed, there is no return to earlier points. Establishing coherence involves reasoning on both pieces of form description and meaning descriptions, because it combines such information.

3.2 Communication with the knowledge base

As we have pointed out in the **Introduction**, the representation of knowledge is uniform in our system. That is, one and the same **higher-order predicate logic** formalism is used for describing forms and meanings alike. The actual reasoning performed on pieces of knowledge may differ from one domain to the other, but that is entirely an internal affair of the knowledge base (which detects pieces of information that require special types of reasoning, and acts accordingly). The knowledge base provides services to the systems that connect to it by performing memory access and reasoning in terms of the incoming query; those external systems (in our case, the NLP system) consider the knowledge base a black box.

The main understanding/generation cycle that we have explained above is also driven by queries addressed to the knowledge base. In this section, we will explain what those queries are like. Breaking down the understanding/generation problem into the problems of selection and evaluation sheds a new light on how linguistic knowledge can be used, in particular together with extralinguistic pieces of knowledge.

3.2.1 Queries for selection

As we have stated in connection with our basic principles in the **Introduction**, one of the key features of our system is that 'superficial' clues are first-class pieces of linguistic knowledge, not to be separated from 'hardcore' grammatical knowledge. Accordingly, if the previous evaluation step has fulfilled its function satisfactorily, then we have all the necessary information at our disposal when selection has to take place. (As we have mentioned earlier, the entire NLU/NLG process starts with a trivial evaluation step, which picks those choice points in the construction network that can be decided on the basis of the scarce data we have at our disposal in the beginning.) Therefore, decision will be carried out by choosing one of a rather limited set of (in the ideal case, mutually exclusive) options. The options may be ordered in terms of preference, too, so that multiple possibilities do not give rise to dilemmas.

For example, the trivial evaluation step that starts an NLG process is sensitive to a small set of distinguished utterance operators (like 'assert', 'inquire' and the like), which must appear in any content plan (i.e., the representation of any communicative intention), and picks those choice points that can be decided on the basis of which operator characterizes the current CP. In any case, the queries used in this step simply inquire the knowledge base about the compatibility of two formulae, namely, that of the current UR, and that of the option we are about to decide upon. In more complex examples, e.g., the choice of the type of a referring expression in NLG, one has to ask queries like 'Is the referent a unique male person in the context of utterance?'; 'Do we know a name for this thing that the audience probably also knows?'; and so on. These propositions originate from the constructions to which a connection should be established, and the resulting UR will be contradictory unless these can be proven by the knowledge base.

That is, selection may require quite complex reasoning from the knowledge base. The compatibility of the content of a construction with the content of the UR into which we want to incorporate it is sometimes difficult to prove. For example, proving that a pronoun like *us* can be used successfully in order to refer to one's company, department, family, the people surrounding one, and so on, may require not only a fine-grained representation of the context, but also a logical theory of what people are permanently associated with a person. This is certainly not grammatical or even linguistic knowledge, yet such queries must be answered very frequently (whenever pronouns are used).

3.2.2 Queries for evaluation

Evaluating an UR involves the following elements:

1. The 'evaluator' must detect whether the content of the UR (i.e., the semantic and formal characterization in it) meets the current goal. The current goal is either answering a question on the basis of an input utterance (understanding) or expressing a content plan using an output utterance (generation). We could call this step **evaluation proper**.
2. if the UR does not meet the current goal, then the problems in it that are to be solved next must be selected. That is, it must be decided which construction instances in the UR need further elaboration in order to better meet the requirements set by the current goal. So this step can be called **problem selection**.

The step called evaluation proper consists in posing the current goal as a query to the mini-database constituted by the content of the UR. For example, the evaluator may try and produce an output string out of the formal requirements embodied in the content of the UR (see the following section on reasoning about forms), or ask a question like 'What does the speaker say (s)he is interested in?'. In either case, we expect the content of the UR to provide an unknown piece of information (which we can think of as a free variable to be equated with at least one constant expression as a result of the inquiry).

The second step of problem selection uses the result of evaluation (and, of course, the content of the UR) as its fundamental data. If the answer to the evaluator's question contains free variables that could not be bound to anything, the problem selector starts hunting for locations in the UR where the answer can possibly be found. It picks those construction instances from which the free variable(s) originate: the content of the UR is the conjunction of the contents of the individual constructions that it contains, with certain variables in them collapsed, so each variable in the content of the UR must have one or more source. Moreover, the problem selector must find those dimensions of the selected constructions that possibly provide values for the unbound variables in question. (As we have explained in connection with example (2.1), connections to other constructions specify what entities to identify in the connected constructions, i.e., they bind variables together.)

For example, consider a situation in which the current goal is to find out what the user is willing to say about the location of a certain book. The question might have been asked by the computer or simply emerged from previous conversation. As a matter of course, a simple answer referring to a location constitutes the simplest case (which, incidentally, may constitute a serious challenge to many other NLU systems because of its elliptical character). But maybe the answer is something like *I gave it to Mary*, and then a more thorough analysis is required. From the formal clues at our disposal (which we have to start from in any case), we will eventually construct a UR in which the book in question is identified as the direct object of *gave*, and we have already looked up the meaning of the main verb. So the piece of information that the object in question has been transferred is available through reasoning together with the fact that the goal of the transfer is denoted by the prepositional phrase *to Mary*. So it is this prepositional phrase, more closely, the noun phrase it contains, is to be focussed on in the next step. For the purposes of this particular goal, the identity of the subject can be ignored altogether, especially if time limitations apply.[4] For obvious reasons, the situation is quite different if the input utterance is *I got it from Mary*. (Perhaps it does very natural as an answer to 'Where is the book?', but it may well happen that you say this without explicitly saying *I have it* beforehand.)

Similarly, in a generation task, certain substrings of the output string being described by the UR may correspond to free variables, which can be bound through connections betweeen construction instances. The problem selector can find those constructions (and, more closely, those dimensions in them) that can possibly constrain the variables in question, and can instruct the next step to focus on them.

3.3 Reasoning about forms

Reasoning concerning the formal properties of a UR is quite simple. The two operations that we perform are

1. Establish whether a formula describing a (class of) forms is self-consistent;
2. Produce a string (essentially, an arbitrary string) satisfying such a formula.

The formulae that we use for describing classes of forms are pretty rudimentary: they define a partial ordering over possibly overlapping continuous domains (substrings), some of which may be associated with actual strings. Since this kind of formulae cannot refer to any kind of recursion at all, it is easy to see that the two operations above can be performed by finite-state automata.

The treatment of recursion is thus not reflected in the way we reason about forms, and this keeps this part of the machinery simple. Recursion and iteration are treated by the multiple instantiation of constructions within the same UR. For example, there may be an instance of a sentence construction (representing, e.g., a sentential complement) within a UR that itself corresponds to a sentence. Moreover, we can also instantiate the same type of constructions several times in the case of iteration (e.g., in the case of accumulated modifiers). These possibilities give at least context free power to the system, and most probably much more, since we can use the formulae in the content of the UR for controlling what constructions to instantiate.

3.4 Level of implementation

Simplified algorithms for the selection and evaluation steps have been implemented. However, their control is deterministic for the time being, and all searches are performed linearly. The knowledge base module has been operative for more than a year

[4] The treatment of time limitations is part of the independently developed reasoning system that we rely on.

now, but the ontology and the substantive data in it are being entered gradually. The principles for implementing partially ordered contexts in it have been established but not fully implemented. We have a first version of the dialog management module, which creates content plans and evaluates contents of URs. We have a working version of reasoning about forms, but its efficiency needs further improvement. Finally, our stock of constructions is rather rudimentary for the time being: it is just sufficient for testing purposes on a small demo in which the computer inquires about the user's favourite occupations.

4 Conclusion

Our aim in this project is to develop a flexible, robust and scalable NLP system the linguistic and real-world knowledge of which is easily extendible (potentially using machine learning techniques). The constructionist view of grammar, which posits separate constructions for all idiosyncratic form/meaning associations in a language, but no hierarchical structure or rule system, is a natural choice for implementing such a system. The constructions and the connections between them are purely declarative statements of pieces of linguistic knowledge, which are easy to incorporate into the overall knowledge representation framework, and can be used for encoding the heuristic elements of understanding and production processes. Thus these processes are not controlled by specific algorithms, but by general strategies on how an incomplete utterance representation can be enriched using linguistic knowledge.

References

[1] Noam Chomsky. *Syntactic Structures*. Mouton, The Hague, 1957.

[2] Koenraad De Smedt, Helmut Horacek, and Michael Zock. Architectures for natural language generation: Problems and perspectives. In *Trends in Natural Language Generation: An Artificial Intelligence Perspective*, pages 17–46. Springer-Verlag, Germany, 1996.

[3] William Diver. The theory. In E. Contini-Morava and B. Goldberg, editors, *Meaning as explanation: Advances in sign-oriented linguistic Theory*, pages 45–113. Mouton de Gruyter, Berlin/New York, 1995.

[4] Lee Fedder. *Generating natural language text from the output of an application program*. PhD thesis, Queen's College and Computer Laboratory, Cambridge, 1991.

[5] Charles J. Fillmore and Paul Kay. *Construction Grammar Coursebook*. Reading materials for Linguistics X20, University of California, Berkeley CA, 1993.

[6] Edward Flemming. Scalar and categorical phenomena in a unified model of phonetics and phonology. *Phonology*, 18(1):7–44, 2001.

[7] Adele E. Goldberg. *Constructions: A Construction Grammar Approach to Argument Structure*. The University of Chicago Press, Chicago, 1995.

[8] Charles F. Hocket. *A Course in Modern Linguistics*. Macmillan, New York, 1958.

[9] Mark Kantrowitz and Joseph Bates. Integrated natural language generation systems. In R. Dale, E. Hovy, D. Rösner, and O. Stock, editors, *Aspects of Automated Natural Language Generation*, number 587 in LNAI. Springer-Verlag, 1992.

[10] Gerard Kempen and Edward Hoenkamp. An incremental procedural grammar for sentence formulation. *Cognitive Science*, 11:201–258, 1987.

[11] Ferdinand de Saussure. *Course in General Linguistics*. McGraw-Hill, New York, 1966.
[12] Anna Wierzbicka. *The Semantics of Grammar*. John Benjamins, Amsterdam, 1988.

Received 21 March 2002.