Feature-Based & Model-Based Semantics for English, French and German Verb Phrases

Stuart Kent

Department of Computing
University of Brighton
Sussex, UK
Email: S.J.H.Kent@bton.ac.uk

Jeremy Pitt

Department of Computing
Imperial College
London, UK
Email: jvp@doc.ic.ac.uk

Abstract

This paper considers the relative merits of using features and formal event models to characterise the semantics of English, French and German verb phrases, and considers the application of such semantics in machine translation. The feature-based approach represents the semantics in terms of feature systems, which have been widely used in computational linguistics for representing complex syntactic structures. The paper shows how a simple intuitive semantics of verb phrases may be encoded as a feature system, and how this can be used to support modular construction of automatic translation systems through feature look-up tables. This is illustrated by automated translation of English into either French or German. The paper continues to formalise the feature-based approach via a model-based, Montague semantics, which extends previous work on the semantics of English verb phrases. In so doing, repercussions of and to this framework in conducting a contrastive semantic study are considered. The model-based approach also promises to provide support for a more sophisticated approach to translation through logical proof; the paper indicates further work required for the fulfilment of this promise.

1 Introduction

This paper addresses the problem of using features and formal event models to characterise the semantics of English, French and German verb phrases, and considers the application of such semantics in machine translation.

Feature systems are widely used in theoretical and computational linguistics for syntactic representation of complex symbols. In Computer Science, they have also been used in compiling for representing so-called 'static semantics' of programs. We describe how a set of features can be used to represent the semantics of English verb phrases, as denoted by their (syntactic) tense and aspect markers. This is contrasted with how distinctions of meaning are conveyed in French and German verb phrases. We then show how translation between English/French and English/German can be implemented by a simple table look-up based on the semantic features, and how pragmatic factors (such as the presence of adverbial expressions which also carry semantic features) can influence the translation, in particular by disambiguation. The implementation demonstrates that the approach is easily mechanisable and could have a useful application in machine translation.

We continue by giving examples of how the feature-based analysis can be formalised by event models, via a Montague-style semantics for each of the languages using a common logical formalization. This adapts and formalises the well-known Vendlerian classification of verbs and verb phrases as a classification of events which is embodied in the underlying model theory of the logic; this, in turn, effectively formalises the distinctions between different features. The uniformity of the analysis between languages comes from being able to relate logically and semantically operators derived from the syntactic markers of one language with those of another. This takes the analysis to a deeper semantic level, and offers some interesting insight into the process of conducting a contrastive analysis. It also holds out the promise of establishing a unifying framework for representing the semantics and pragmatics of verb phrases in English, French and German. This would in turn support the development of a more sophisticated, logic-based implementation of the translation mechanism.

2 Background

2.1 The Investigation Space

This paper brings together the research and experimentation of three theses: that of Pitt [1991], who implemented a grammar development environment and a wide-coverage syntax and static semantics of English; Tucker [1991], who used what we call a 'static semantic' analysis to identify features which were used to implement a translation of English verb phrases into French and German; and Kent [1993], who developed a detailed, formal semantics of events in English based on a logic of intervals.

Thus our investigation space is defined by three axes: on one axis, we have implementation, on a second semantic analysis, and on the third we have the number of languages studied. The thesis of Pitt developed a parser-generation system, but applied it to just one language, English, and considered semantics in terms of Montague-style logical translation and feature systems; Kent developed a framework for deep semantic analysis, but again studied just one language, English; and Tucker did a restricted implementation and a restricted semantic analysis of verb phrases, but did so for English, French and German.

This situation is illustrated in Figure 1. This paper provides the basis for extending the scope of implementation and semantic analysis to embrace French and German.

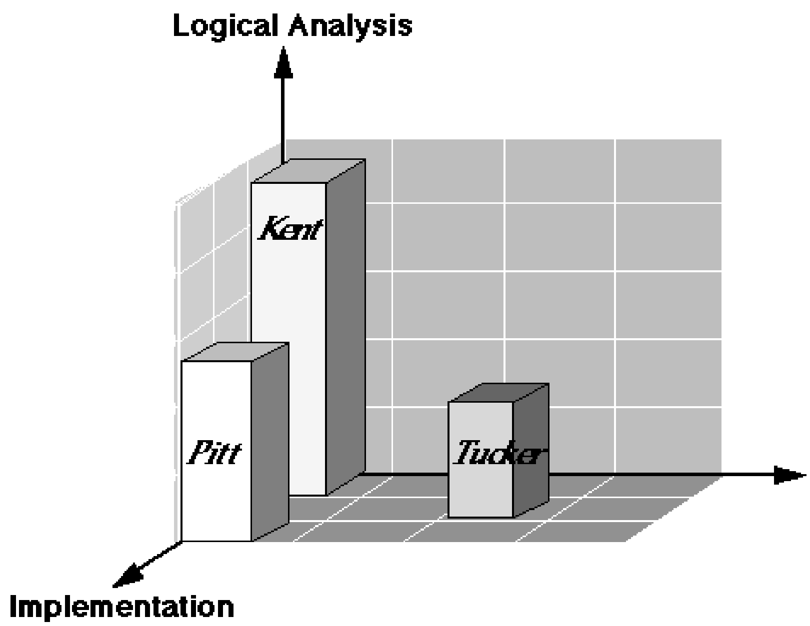

Figure 1: The Investigation Space

2.2 Machine Translation

We are particularly motivated by applications of natural language processing, such as document summarising, manual writing, and on-line information retrieval. There are two important features of these applications. These are, firstly, that very often more than one language is involved, so that automated translation between languages is essential if the application is going to have the broadest possible scope. Secondly, it is possible to have control over the input, so that the application developer can impose restrictions on that input. This is the basis of *control languages*, which originated in technical writing, and are often the basis of many practical applications of natural langauge processing. It is no longer necessary to deal with unrestricted language, and it is possible to circumvent certain linguistically interesting, but computationally difficult phenomena, and concentrate on getting the syntax, and semantics and pragmatics of the core language right. This motivates our study of language fragments, of which more below.

Most Machine Translation systems currently under development use one or other of four basic strategies. Firstly, there is the *direct approach*, which translates directly from the source language to the target language, although the general disregard for meaning makes this approach worthy only of a historical mention. Secondly, there is the *interlingua approach*, where the source text is translated into a supposedly universal language or interlingua, from which the translation in the target langauge is then generated. The search for an interlingua however remains something of a holy grail. Thirdly, there is the *transfer approach*, where the source language is converted into an abstract representation;

this is transformed into an abstract representation of the target language, from which the target language sentences are generated. Fourthly, *statistical* approaches are becoming increasingly common, although we will have no more to say about them than that we prefer logical, symbolic reasoning.

We have taken the transfer approach in our work: this is illustrated in Figure 2. The feature-based transfer approach of Tucker goes to the second level shown, and transfers (via look up tables) features of one language to features of another, and generates from these. The feature structures are given an intuitive semantics in terms of time lines, which are assumed to be universal to languages. The Kent analysis takes us to a level of abstraction where features are replaced by logical operators and transfer is performed by logical proof, or, in the absence of a proof system, by appeal to the semantics (the latter approach is taken here). The logics (one for each language) are given a formal semantics, where the semantic model is assumed to be rich enough to be universal across languages. The latter assumption underlies our belief that there is a universal logic in which all the necessary operators can be found, although we do not investigate this in any detail here.

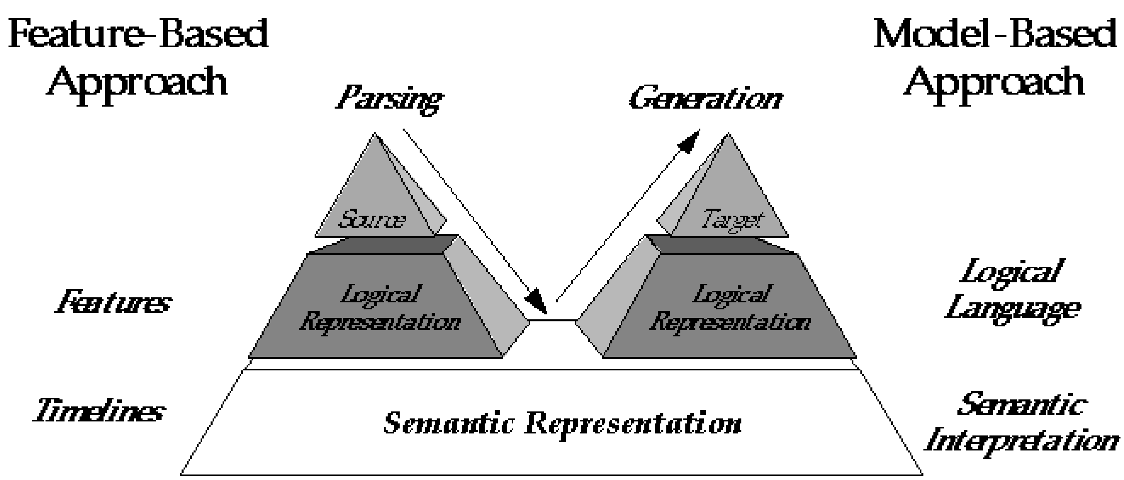

Figure 2: Machine Translation

2.3 Formal Semantics

The approach to formal semantics adopted in this paper is taken from Kent [1993]. There, a formal semantic theory, in the Montague tradition, is developed to account for the semantic behaviour (e.g. entailments) of a corpus of sentences involving tense, aspect and temporal adverbial constructions. Previous attempts have been made to construct semantic theories to account for similar behaviour. Kent [1993] builds on the theory of Lascarides [1988], which, in turn, attempts to solve problems unresolved in previous theories, notably [Dowty, 1979, Hinrichs, 1985, Taylor, 1985, Cooper, 1985, Parsons, 1985]. Lascarides classifies these theories according to the particular strategy adopted, under three headings: heterogeneous, eventual outcome and event-based. She argues the only the latter, embodied in Parsons' theory, is able to account for the *imperfective paradox*; her main contribution is to recouch the event-based strategy in a framework that provides an explanation of aspects of linguistic behaviour (notably at adveriabls) which can not be explained within the framework used by Parsons.

The key to the success of the event-based strategy is an *ontology of events*, saying how events are structured and related to one another. This, in turn, is based on a *classification* of events. Both Lascarides and, subsequently, Kent provide a formalisation of informal classifications dating back to Vendler [1967]. Both are adaptations of the classification suggested by Moens [1987]: Kent's classification is richer, to account for behaviour not handled by Lascarides. Moens also gives an informal description of an event ontology, in terms of an *aspectual network* of event classes. Here, an event is constructed by taking some basic event (assumed to belong to a particular class) and transforming it by following arcs through the network. Such a network is used to build events denoted by a linguistic expression by identifying the arcs with syntactic constructions (e.g. the progressive) and aspects of the context of utterance. Kent's, and to a limited extent Lascarides', theories may be regarded as a formalisation of this approach.

3 The Grammatical Basis

3.1 A Grammar of English

We are not committed to any particular linguistic theory, so we have been eclectic and selected elements from a variety of linguistic sources which we have found computationally expedient. However, we have particularly exploited two main works. These are firstly the grammatical theory of Generalized Phrase Structure Grammar of Gazdar *et al.*, (GKPS: [Gazdar *et al.*, 1985]), including the English GPSG they give in Appendix A (p249*ff*), and secondly the formal (descriptive) grammar of Quirk *et al.* (QGLS: [Quirk *et al.*, 1985]).

Generalized Phrase Structure Grammar, as described in GKPS, is a linguistic theory that is context-free, transformation free, and postulates only one representational level of syntax. It comprises six main components, of concern to us here are the following:

i *linguistic categories*: the constituents of syntax rules are complex symbols called linguistic categories;

ii three *universal feature conventions*: these govern the way features are distributed between linguistic categories in a parse tree;

iii *feature specification defaults*: these are default values of features unless explicitly stated otherwise;

iv *feature co-occurrence restrictions*: these state what features and values can (or cannot) appear on a linguistic categories.

The specification and computation of features is, of course, critical for evaluating the feature-based translation described in section 4.

From the formal grammar of English described by QGLS, we can identify the following levels of syntactic structure, and the sorts of constituent found at that level:

i *word classes*: words can be classified into the appropriate words classes common to traditional grammar, e.g. nouns, verbs, determiners, pronouns, etc.;

ii *phrases*: sequences of words form (function as) phrases. QGLS identify noun, verb, prepositional, adjective and adverbial phrases;

iii *clause elements*: according to QGLS there are five clause elements which detmine the function of a phrase within a clause: these are S (Subject), V (Verb), O (Object), C (Complement), and A (Adverbial);

iv *clauses*: there are seven patterns of clause elements which can constitute a clause, these are SV, SVO, SVA, SVC, SVOO, SVOC, and SVOA.

v *sentences*: QGLS identify four types of sentence: declarative, interrogative (*wh*– and *inverted* (*yes–no*) questions), exclamatives and imperatives.

Each sort of constituent can be labelled by a terminal or non-terminal symbol of the syntax, which can also be conveniently identified with the functional property of the constituent. The levels and symbols can be related by context-free syntax rules, which may be thought of as a specification of how 'things' at a lower level (the right hand side of the syntax rule) can function as a 'sort of thing' at a higher level (the left hand side of the rule). For a simple example, we have the following derivation (read $\leadsto$ as "can function as"):

$$\left.\begin{array}{ccccccc} John & \leadsto & noun & \leadsto & noun\_phrase & \leadsto & S \\ walks & \leadsto & verb & \leadsto & verb\_phrase & \leadsto & V \end{array}\right\} \leadsto SV\text{-}clause \leadsto declarative \leadsto sentence$$

QGLS also give the intuitive descriptions of the conditions and exceptions on how objects can function, which are interpreted as constraints and implemented as annotations on the syntax rules.

The synthesis of the descriptive grammar of QGLS and the grammatical theory of GKPS is computationally very powerful. QGLS provides a constituent structure of English which is based on the *functional* properties of words, phrases, clauses etc., and the intuitive rules which govern the function of, use of, and relations between, words, phrases, clauses *etc.* GKPS provides the mathematical basis and insight for: implementing complex syntactic symbols as feature-value structures, which are common to unification-based grammars [Reape, 1991]; distributing features between nodes of the parse tree during the parse, via GPSG's *feature instantiation principles*, (i.e. the head-feature convention etc.); and implementing the intuitive rules of Quirk *et al* and unbounded dependency constructions as constraints on the co-occurrence features.

3.2 Formal Specification

The format of the syntax rules used to specify our grammar formally is[1]:

[1] Note here we concentrate only on feature structures: in practice we allow an array of *attributes*, also used for computing semantics, which is essential for our applications. See [Pitt and Cunningham, 1990].

$$\begin{array}{lll} X_0(F_0,\ldots) & \longrightarrow & X_1(F_1,\ldots),\ \ X_j(F_j,\ldots) \\ & & \{ \\ & & C_1(\ldots),\ \ \ldots,\ \ C_k(\ldots) \\ & & \} \end{array}$$

where each X_i, $0 \le i \le j$ is a syntactic symbol in S, each F_i, $0 \le i \le j$ is a feature structure associated with the corresponding syntactic symbol; and each C_i, $1 \le i \le k$ is an element of C, whose parameters are feature structures and which encode constraints that must be satisfied or compute F_0 from $F_1, \ldots, F_j$. Alternatives for rewriting X, each with its own constraints, are introduced by the meta-symbol '|'.

A feature system F is a 2-tuple $< N, V >$, where N is a set of feature names, and V is a set of atomic values. We need these terms and notation in the sequel:

- a *feature-value pair* is a feature name n and an associated value v, written n::v, where n is a feature name and v is either an atomic value, a variable, an undefined element $\perp$ or a feature structure;
- a *feature structure* is a collection of feature-value pairs, written $[n_1::v_1, \ldots, n_i::v_i]$;
- for a feature structure F and a feature name n, $F(n)$ is the value v if $n::v \in F$ and $\perp$ otherwise;
- for a sequence of feature names $p = n_1/n_2/\ldots/n_i$ and a feature structure F, the expression $F(p)$ represents the value given by $(\ldots(F(n_1)n_2)\ldots)n_i$.

3.3 A Syntax of the English Verb Phrase

According to QGLS, a V functional element (i.e. a verb phrase) comprises a sequence of one or more verbs, which may be introduced by the word `to` if it is an infinitive, and is introduced or interrupted by the word `not` if it is negated. These observations can be concretely defined by a set of left-recursive context-free rules. However, in a complex verb phrase (more than one verb), the sequence of verbs follow a strict order, and each auxiliary demands a certain morphological form in its immediate successor. So, for example, the two context-free rules:

$$\begin{array}{lll} \textit{verb\_seq(F0)} & \longrightarrow & \textit{verb\_seq(F1)},\quad \textit{not}\ ,\quad \textit{verb(F2)} \\ & | & \textit{verb\_seq(F1)},\quad \textit{verb(F2)} \end{array}$$

are annotated with the constraint:

$$\textit{internal\_verb\_agr(F1, F2, Opr)}$$

which is satisfied if the last auxiliary verb in *F1* agrees with the form of the verb in *F2*. Furthermore, this agreement determines the value of *Opr*, which is a feature-value pair that is included in *F0*. This feature records the type of the (last) auxiliary verb in *F1*, which is one of modal, progressive, perfective or passive.

The conditions for satisfying the constraint are:

F1(last/aux)	*F2(vform)*	*Opr*	*example*
modal	*base*	*modal::+*	*would take*
be	*-ing*	*PROG::+*	*was taking*
have	*-ed2*	*PERF::+*	*had taken*
be	*-ed2*	*PAS::+*	*was taken*

It is the identification of these particular features which (inter alia) are used in the translation of tense and aspect in feature-based translation. However, note that it is the very last verb in the entire sequence, i.e. the *main verb*, which determines the value of the semantic feature *Sem*.

En passant, although slightly orthogonal to the main thrust of this paper, we also note that QGLS state that the morphological form of the first verb in the verb phrase must agree in number and person with the subject of the sentence, and that the type of the 'main' (last) verb determines the type of verb complementation allowed. In the first case, this is implemented by the context-free rule and constraint for subject-verb agreement (which we assume is well known):

$$\begin{array}{lll} \textit{declarative(F0)} & \longrightarrow & \textit{subject(F1), predicate(F2)} \\ & & \{\textit{subject\_verb\_agr(F1,F2(first/vform))} \ldots\} \end{array}$$

The second case is treated similarly.

Constraints are also imposed on the *use* of a verb phrase, either finitely or non-finitely. For example, the *use* of walking the dog is acceptable in the phrase the man walking the dog if this phrase is parsed as a noun phrase, but not if it is parsed as a sentence. To implement this we have identified a set of features whose values are computed while parsing the verb phrase. These filter up the parse tree to the point where they are used to determine whether or not the *use* of the verb phrase is permissible or not. Different uses – in this case either as a predicate or as a participle – require different distributions of these features, and a single constraint (implemented by simple table look-up) determines this. For space reasons, we do not give the precise details here, but full details can be found in [Pitt, 1991].

4 Feature-Based Translation

4.1 QGLS 'Semantic' Analysis

QGLS present a detailed, but informal, 'semantics' of English verb phrases. It is based on the idea of a theoretically infinite timeline, on which is located as a continuously moving point, the present moment. Indication of past, present or future can then be conveyed firstly by *reference*, anything ahead of the present moment is the future, anything behind is the past (see Figure 3(a)); and secondly by *timespan*, so that 'present time' is defined inclusively – something is present if it has existence at the present moment, and allows for that existence to stretch into the past and/or future (see Figure3(b)).

QGLS then discuss how grammatical features such as past and present tense markers, auxiliary types such as modals, perfective and progressive, and the situation type (either

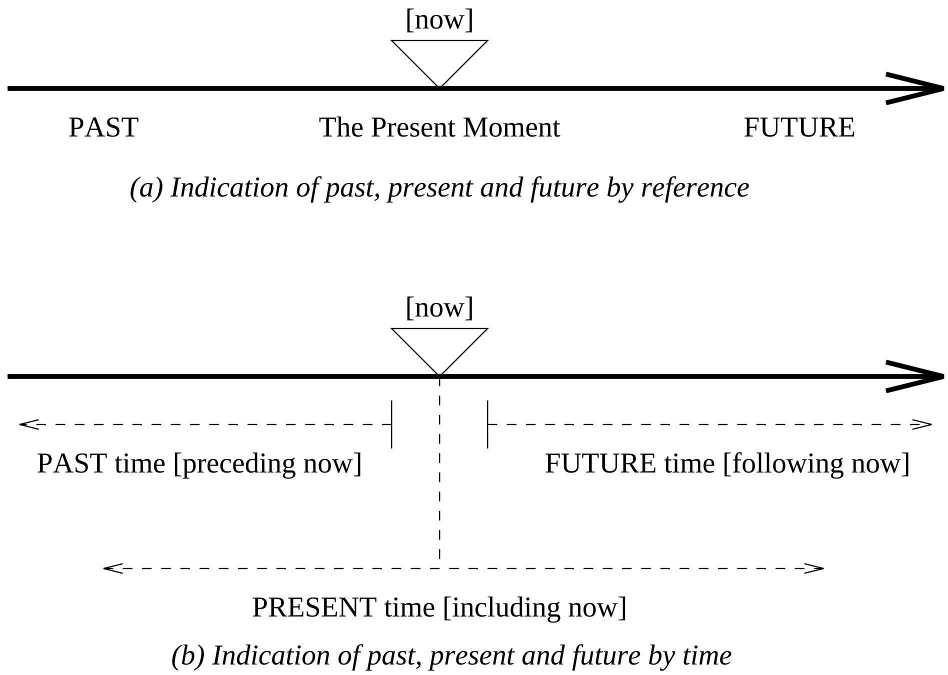

Figure 3: QGLS 'Semantics' for English Verb Phrases

stative or dynamic) of the main verb (which identifies whether the action described by the verb is a state, a habit, or an event), can be used to identify a portion of the timeline and an orientation with respect to the point of reference given by "now".

Thus, for example, the intuitive QGLS semantics of event, state and habit with the past tense can be represented by the timespan indicated on the timeline given in Figure4. Here, the meanings are located by reference to a definite time in the past (T2) and indirectly to the present moment (T1). The event past refers to a single definite event (e.g. I walked to Imperial), the state past refers to a state (e.g. I worked at Imperial), and the habitual past refers to a sequence of events (e.g. I used to walk to Imperial).

4.2 Translation by Table Look-Up

Tucker [1991] extended the QGLS 'semantic' analysis in three ways. Firstly, she formalised it by identifying 'semantic' features which related to orientation, timespan, tense, aspect, etc. in the verb phrase, and other features, especially from adverbials, which influenced the meaning. Secondly, she used these features to characterise the actual timespan(s) on a timeline denoted by a verb phrase; these were recorded in tables. Thirdly, she examined how each distinct meaning is expressed in French and German, and drew up parallel tables of distinguishing features for the meanings of the French and German verb phrases.

Following Tucker, in order to identify the meaning of a verb phrase, it is necessary to

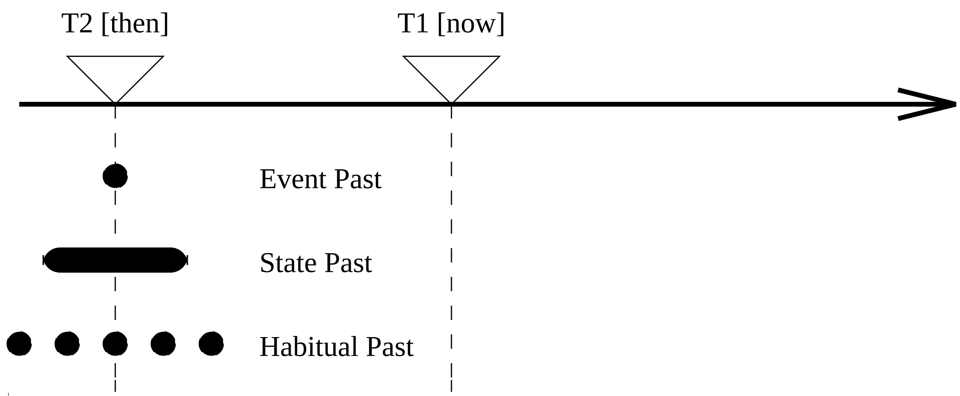

Figure 4: Event, State and Habit with the Past Tense

determine its form (tense and aspect), its type (dynamic or stative), the time of reference, the timespan, and the 'continuity' of the action or state. For the purpose of translation, it is also necessary to identify the dictionary or lexical 'meaning' of the main verb.

As it turns out, the form, type, time of reference and 'meaning' of the verb can be determined from the verb phrase itself. The time span and continuity are determined in part by the form and type of the verb phrase and in part by any temporal constraints imposed by adverbial phrases in the sentence.

The complete analysis is given in [Tucker, 1991], and for space reasons cannot be reproduced here. To give a flavour of the analysis though, consider Figure 5. This gives the timeline for the sentence Max had been writing his book for several years. The pointer marked 'now' is the time of reference, the pointer marked 'T1' is the time of orientation, and the pointer marked 'T?' is an unspecified time when Max started writing the book. The timespan 'T?' to 'T1' is time of several years during which the action of Max writing his book occurred.

Above the timeline in this figure, we can see the features which are used to characterise this timespan. Here, the time reference (*Timeref*) is past tense from the inflection on have; we get *PERF::+* and *PROG::+* from the internal verb agreement constraint, discussed above, being satisifed for both had been and been writing. The situation type (*Type*) of the verb write is dynamic, and its semantic value is *write_* (e.g. it's logical translation in the style of Montague [Montague, 1974]). The feature *recur* indicates if the action described by the verb phrase is habitual (recurrent) or not: in this sentence it is undefined, which combined with the value of the *Type* feature enables us to infer that this verb phrase is describing an event. Finally, the *ext* feature indicates if the action described by the verb phrase continues up to and includes the time of reference. In this sentence it can be either, hence its value is {+,-}, and, as we shall shortly see, this has significance for the translation.

These features can then be used to index the corresponding features that the translated French or German verb phrase should have.

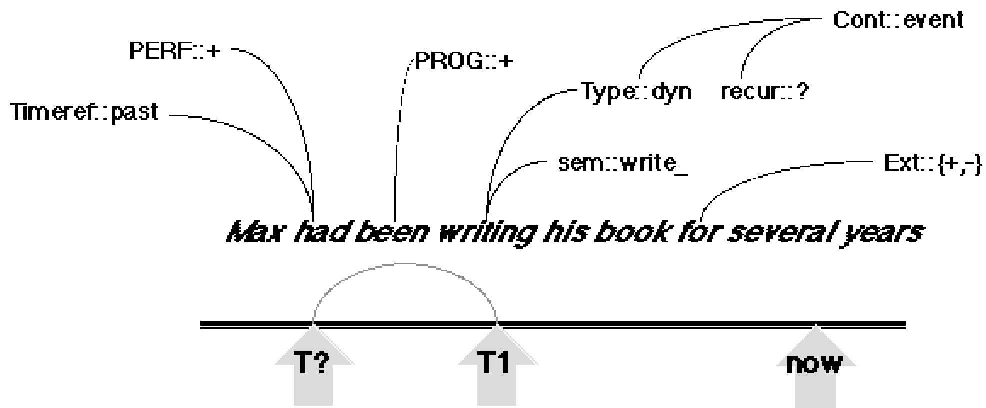

Figure 5: Features formalising QGLS 'Semantics': An Example

4.3 Automatic Translator Generation

Pitt [1991] developed a natural language grammar development environment, similar in purpose to the ALVEY NL tools [Grover *et al.*, 1992]. This comprises formal specification languages for e.g. morphologies and syntaxes, which can be automatically processed by generic tools, to produce customised morphological and syntactical analysers. Our grammar development environment comprises a (logic) grammar formalism, a parser-generator based on YACC [Johnson, 1974], a Prolog LALR(1) parser shell, and a module for standard operations on feature structures (unification, application, etc. [Reape, 1991]). A parser is automatically generated as follows:

i the grammar is specified as a triple comprising a context-free syntax S, a feature system F, and a set of constraints and semantic actions C;

ii the logic grammar specification is processed by the parser-generator to produce parsing tables and a compiled form of the grammar;

iii the tables and compiled grammar are used by the parser shell to analyse sentences of the language described by the grammar.

Our motivation for building a grammar development environment was *modularity*. Most programming languages nowadays come with standard support for inter-process communication, enabling analysers to be coupled. Thus, for example, we have been able to integrate our Quintus^{TM} Prolog parser with the C morphological analyser PCKIMMO [Antworth, 1990] via TCP/IP.

For automated translation, we also need to ensure that our grammatical specifications are as modular as possible. For this reason, the semantic actions (which implement the translation) of our grammar of English contain only procedure stubs: the actual definitions are included with the specification of the particular language into which we intend to translate.

The situation is illustrated in Figure 6. The English specification is fed into the parser generator to give an English parser. Then the procedure definitions and look-up tables are used to augment the parser to provide a machine translation system for a specific language pair. Note that to date we have only a specification for English and translation procedures/tables for French and German, although the modularity of the system does allow for extension with other specifications and corresponding tables.

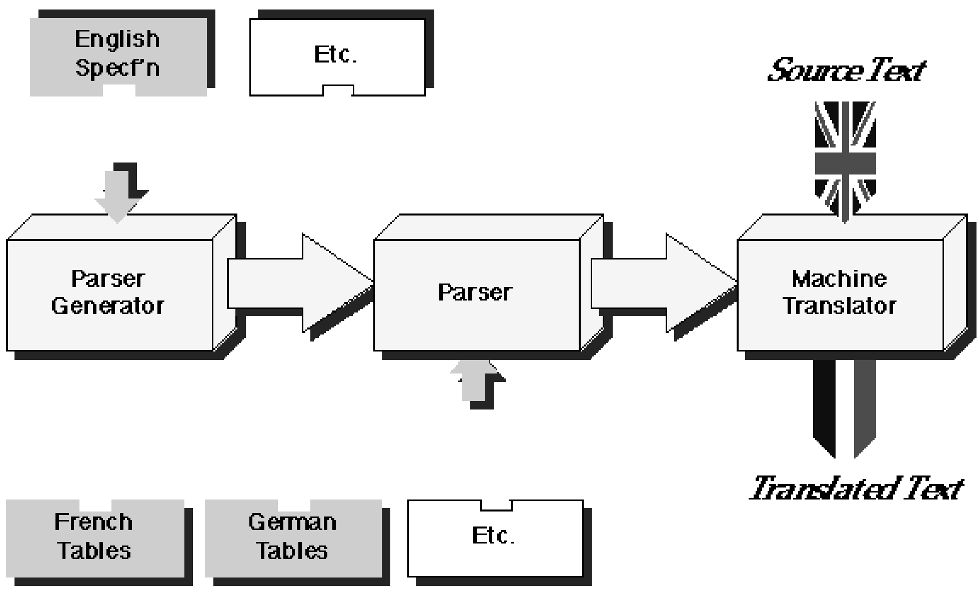

Figure 6: An Automatic Translator Generation System

4.4 Translation examples

Tucker [1991] considers tensed sentences involving progressive and perfective auxiliaries and temporal adverbials such as the English for. In this paper we consider only the most interesting examples, which involve combinations of the progressive, which is in English but not in French or German, the English for adverbial, which has two possible translations in both French and German, and perfective aspect which leads to interesting cases when combined with the progressive and for adverbials. We have chosen only to consider sentences in present or past tense, as, in English at least, ways of expressing the future (e.g. will auxiliary) have modal overtones, so an analysis of these would lead us beyond the scope of this paper.

The sentences with which we are concerned are given below.

Progressive	
1	max is writing his book
Present perfective	
2	max has written his book
3	max has written his book for ten years
4	max has known marie for ten years
Present perfective progressive	
5	max has been writing his book for ten years
Past perfective progressive	
6	max had been writing his book for ten years
Past perfective	
7	max had known marie for ten years

The English features extracted from the syntactic and semantic analysis of these sentences are shown in the following table. Note that as the *ext* feature can have the value + or -, in combination with other features for sentences 6 and 7 there are two possible translations.

No.	Sem	Type	Ref	PROG	PERF	Ext	Recur	Cont
1	write_	dyn	pres	+	⊥	⊥	⊥	event
2	write_	dyn	pres	⊥	+	⊥	⊥	event
3	write_	dyn	pres	⊥	+	+	⊥	event
4	know_	stat	pres	⊥	+	+	⊥	state
5	write_	dyn	pres	+	+	+	⊥	event
6(a)	write_	dyn	past	+	+	+	⊥	event
6(b)	write_	dyn	past	+	+	-	⊥	event
7(a)	know_	stat	past	⊥	+	+	⊥	state
7(b)	know_	stat	past	⊥	+	-	⊥	state

The corresponding features derived from the table look ups are shown in the following table, together with the verb phrase translations that are generated in English and French.

	French			German		
No.	Tense	Perf	Translation	Tense	Perf	Translation
1	present	-	ecrit	present	-	schreibt
2	present	+	a ecrit	present	+	hat geschrieben
3	present	-	ecrit	present	-	schreibt
4	present	-	connait	present	-	kennt
5	present	-	ecrit	present	-	schreibt
6(a)	imperf	-	ecrivait	preter	-	schrieb
6(b)	imperf	+	avait ecrit	preter	+	hatte geschrieben
7(a)	imperf	-	connaisait	preter	-	kannte
7(b)	imperf	+	avait connu	preter	+	hatte gekannt

The output of the English-French and English-German translation systems for the example sentences are given below:

	English–French Translations	
1	max is writing his book	*max ecrit son livre*
2	max has written his book	*max a ecrit son livre*
3	max has written his book for ten years	*max ecrit son livre depuis dix ans*
4	max has known marie for ten years	*max connait marie depuis dix ans*
5	max has been writing his book for ten years	*max ecrit son livre depuis dix ans*
6	max had been writing his book for ten years	*max ecrivait son livre depuis dix ans* *max avait ecrit son livre pendant dix ans*
7	max had known marie for ten years	*max connaissait marie depuis dix ans* *max avait connu marie pendant dix ans*
	English–German Translations	
1	max is writing his book	*max schreibt sein buch*
2	max has written his book	*max hat sein buch geschrieben*
3	max has written his book for ten years	*max schreibt seit zehn jahren sein buch*
4	max has known marie for ten years	*max kennt seit zehn jahren marie*
5	max has been writing his book for ten years	*max schreibt seit zehn jahren sein buch*
6	max had been writing his book for ten years	*max schrieb seit zehn jahren sein buch* *max hatte seit zehn jahren sein buch geschrieben*
7	max had known marie for ten years	*max kannte seit zehn jahren marie* *max hatte seit zehn jahren marie gekannt*

5 Model-Based Translation

The model-based approach differs from the feature-based approach in that there is a strong formal, compositional link between the semantic representation and the logical representation. This section describes how that link is established, and shows how appeals to the semantic representation may be made when performing transfer in translation. It was argued in the introduction that such a link could be exploited to provide a logic of events, with a well-defined deduction calculus which could be used to support translation (amongst other things) by performing transfer through deduction rather than lookup. Whilst this claim is not pursued in this paper, the experiment described in this section does serve to illustrate how a contrastive semantic analysis may provide insight into the components of such a logic, above that which may be gleaned from the analysis of a single language. Finally, the experiment also provides some insight into issues concerning the process of a model-based contrastive semantic study, that is the construction of a semantic theory which accounts for linguistic behaviour in two or more languages.

5.1 The logical language

Under the model-based approach the semantics of a natural language expression is represented as a (set of) expressions in some logical language, which has a formal, compositional semantic interpretation, instead of a feature structure with an intuitive, informal interpretation. Expressions in the logical language are formed from a number of basic operators, basic event symbols, time terms and parameters. For example, the sentence

Max has known Marie for ten years

receives the reading

`PRES PERF(e) CUL FOR(ten-years) Max-KNOW-Marie`

in which `PRES`, `PERF`, `CUL` and `FOR` are all basic operators, `e` is a parameter, `ten-years` a time term and `Max-KNOW-Marie` a basic event symbol. A basic event symbol is derived from the remains of the sentence, once the temporal adverbials, aspectual and tense markers have been removed. Semantically, it is treated as primitive. The operators provide representations of the temporal adverbials (e.g. `FOR`), tense (e.g. `PRES`) and aspectual (e.g. `PERF`) markers. Some operators do not correspond to any explicit syntactic marker (e.g. CUL), and are referred to as *implicitly marked*. These are 'forced' into a reading by the classification of events encoded in the semantics; how this works in detail is described in the next section. Parameters (e.g. `e`) are used to represent arbitrary events which are instantiated by context; again, this is explained later. Time terms (e.g. `ten-years`) are derived from temporal referring expressions associated with adverbials; these are also treated as primitive in the semantics. A formal presentation of this syntax may be found in [Kent, 1993]. This also describes a formal mapping from NL (according to QGLS) to these readings, which we will not go into here. Suffice it to say, that in practical, implementation terms, the logical expressions could be "calculated" from the feature-structures derived in the way described in the first part of the paper, although it should be noted that some of the features would become redundant. An example of the latter is the *ext* feature derived from a for adverbial, as illustrated in Figure 5. Under the model-based approach the semantics of the adverbial is determined via the logical operator `FOR`, making the *ext* feature redundant.

5.2 Semantic interpretation

A semantic interpretation for a logical sentence is a set of intervals from an interval structure, which is one component of a *model.* Formally, the latter may be expressed as a pair $\langle \mathcal{I}, f \rangle$ where $\mathcal{I}$ is an interval structure derived from a dense, unbounded, left and right branching point structure, and f is a function mapping basic event symbols, time terms and parameters to sets of intervals. For basic event symbols and parameters, the latter may be thought of as the set of occurrences of the events they represent. For time terms, the intervals are assumed to have the duration specified by the time term.

The set of intervals denoted by the logical sentence is constructed compositionally from its components, by defining an interpretation function which defines the semantic effect of applying the logical operators to their arguments.

To illustrate the process consider

(1) Max was writing his book

with reading

(2) `PAST PROG Max-WRITE-his-book`

`Max-WRITE-his-book` denotes a (culminated process) event e, an occurrence of which is represented by the closed interval in any of the line diagrams in Figure 7. `PROG Max-WRITE-his-book` is the (state) event $prog(e)$ whose occurrences are all those intervals within the open interior of the occurrences of e, which are represented by the open interval within the closed interval on the diagrams. (2) denotes the set of all intervals before which there is an occurrence of $prog(e)$.

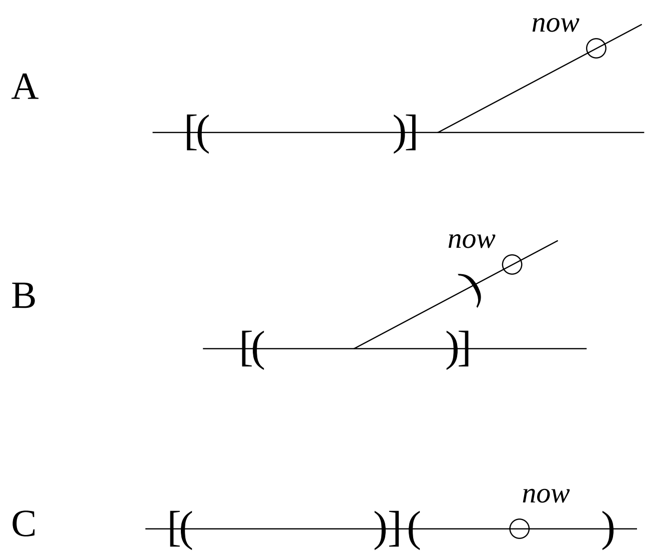

Figure 7: Models of Events

As has already been indicated, sets of intervals are used to characterise events: the intervals correspond to occurrences of the event. Crucial to the semantic theory, is a classification of events. This formalises work dating back to [Vendler, 1967], in particular the classification described in [Moens, 1987] and its partial formalisation in [Lascarides, 1988]. The classification contains five classes: states, processes, culminations, culminated processes and repeated culminated processes. These are outlined below; for the full definitions and detailed reasons underpinning these definitions see [Kent, 1993].

An example of a process event is that denoted by `Max-WRITE`. A process event is characterised as a set of closed intervals, whose maximal intervals are disjoint when restricted to any single branch,[2] and are homogeneous in the sense that all closed intervals contained within an occurrence of a process are also occurrences of that process. This captures the intuition that whilst a process event is in progress, it can be said to have happened. For example, this ensures that if Max is writing then it is always true to say that Max has

[2]A stronger condition than 'disjoint' is required in discrete time.

written. In addition, a further restriction is imposed on process events to ensure that for any branch br intersecting an interval i during which the event is in progress, there will always be an occurrence of that event containing i on br. This captures the intuition that if a process is in progress then it must complete on every future branch, [3] and contrasts with the situation for culminated processes (see below).

A state is an event whose maximal occurrences are open intervals, and which is strictly homogeneous in the sense that the state holds on all subintervals of an interval during which the state holds. Making the maximal occurrences open intervals, enables state change to be modelled (see the description of culminations below). Homogeneity seems a natural property of states: if Max was writing between 1pm and 2pm, then he was writing at all points and during all periods between those times. [4] As may have already been guessed, the progressive of an event e, written $prog(e)$ is defined to be a state.

All occurrences of a culminated process event are like the maximal occurrences of a process, namely closed intervals which are disjoint when restricted to any single branch. This captures the intuiton that a culminated process comprises those occurrences of some process event which terminates. In addition, a culminated process need not complete on every branch through an interval during which that event is in progress. For example, diagram B of Figure 7 represents a situation where a culminated process occurs on one branch (the closed interval), so is in progress on all branches which intersect with the interior of that interval, but need not occur on some of those branches - e.g. the top branch depicted. This captures the intuition that if a culminated process is in progress it need not complete in the future.

The occurrences of a culmination are those points at which a state stops or starts, which are those points bounded on one side by an occurrence of the state and on the other by an occurrence of the negation of a state, where negation (for states) is different to classical negation. For example, in diagram C, the point coinciding with the right bound of the closed interval (say an occurrence of the culminated process e denoted by `Max-WRITE-his-book`) is bounded on the left side by an open interval (a maximal occurrence of $prog(e)$) and on the right by a (left-maximal) occurrence of $neg(prog(e))$, which is the state corresponding to the negation of $prog(e)$. It is, therefore, an occurrence of the culmination corresponding to the 'stopping' of $prog(e)$. In this case, it also happens to be an occurrence of the culmination corresponding to the 'finishing' of the culminated process, which is the start of some consequent state. This is used in the interpretation of

(3) Max has written his book

through reading

(4) `PRES PERF(e) CUL Max-WRITE-his-book`

[3]Our analysis does not at present cater for non-terminating processes, but it could be extended to do so.

[4]Of course there may be narrow and broad senses of writing, where the latter may allow for some gaps during which Max may not actually be writing (narrow sense). But that is to do with the nature of the event chosen to give a semantics to the process of writing, not with the progressive.

Here, `e` denotes the consequent state which the operator `PERF` forces to be a state whose start is the culmination denoted by `CUL Max-WRITE-his-book`, which is the culmination associated with Max finishing his book.

We will not need to consider repeated culminated processes in this paper. Suffice it to say that they are used to account for the behaviour of sentences such as Max was eating sandwiches, which describe events formed from repeatedly performing some other event (in this case Max eating a sandwich). For a full account see [Kent, 1993].

5.3 Accounting for linguistic behaviour

The definition of semantic interpretation, indeed the whole semantic theory, has been driven by the linguistic behaviour which it has been constructed to explain. Two aspects of behaviour have been considered: entailment and felicitousness. A sentence A entails another sentence B if and only if there is a reading α of A which entails a reading β of B. α entails β if and only if β is true in every situation in which α is true. A sentence is felicitous if and only if it is syntactically, semantically and pragmatically grammatical.

These definitions are given a formal characterisation by considering all possible interpretations of α and β, noticing that one obtains a different interpretation for each model that can be constructed (and there are infinitely many of these). In addition, a designated time-point *now* is declared, against which the *satisfiability* of a logical sentence is evaluated. Thus, in the definition above, a *situation* may be viewed as a tuple comprising a model, an interpretation function and a designated time point *now*.

For example, (2) is satisfiable in all interpretations of Figure 7, as it is true at all points 'now', assuming that the closed interval depicted is an occurrence of `Max-WRITE-his-book`. On the other hand, (6) is only satisfiable in interpretations A and C (assuming that the occurrence of `Max-WRITE-his-book` depicted is the only one on the same branch as 'now'). B illustrates a situation where `PROG Max-WRITE-his-book` is true before 'now', but not `Max-WRITE-his-book`. This situation is allowed because `Max-WRITE-his-book` denotes a culminated process. Thus there is an interpretation (B) in which (2) is satisfied but not (6). Hence (2) does not entail (6). Assuming that these are the only felicitous readings of (1) and (5), respectively, this means that (1) does not entail (5). It is left to the reader to satisfy herself that (5) entails (1).

(5) Max wrote his book

(6) `PAST Max-WRITE-his-book`

With regard to felicitousness, a sentence is syntactically grammatical if a reading in the logical language can be constructed for it, e.g. in the way described earlier in the paper. A sentence is semantically grammatical if there is some reading and some situation in which that reading is true. A sentence is pragmatically grammatical if the situation in which the reading is true is a situation which is regarded as "reasonable" according to context. For example, an alternative reading of (3), which is syntactically grammatical, might be (7).

(7) `PRES PERF(e) Max-WRITE-his-book`

However, (7) is unsatisfiable, as, whatever the interpretation, `PERF(e) A` denotes the null event (i.e. has no occurrences) if `A` does not denote a culmination. If this were the only reading, then (3) would be said to be semantically ungrammatical.

Context is also used to refine our definition of entailment, by insisting that only situations which are reasonable according to context need be considered.

We will not consider in any detail the felicitousness and entailment behavior of examples in this paper, as we are mainly interested in translation— the above summary has been included mostly to give an overall picture of the semantic theory being used. However, it is worth pointing out that the definition of logical operators for the languages under consideration should be such that an account of this behaviour may be constructed; and that, clearly, the source and target sentences of the translation should be felicitous. If one is interested in conducting a full contrastive study, then one might also wish to consider entailment relations between source and target. For example, one imagines that the source must at least entail the target. One would also expect that the entailment should also be in the reverse direction, to obtain semantic equivalence between translations. However, in what follows there seems to be an example where this is not the case.

5.4 Translation examples

Listed below are the English-French translation examples, as considered for the feature-based analysis, with their readings in logical form. The choice of readings and their semantic interpretations are discussed in the sequel. Only French translations are considered as the German case is very similar (for this collection of examples).

	English–French Translations
1	max is writing his book PRES PROG max-WRITE-his-book *max ecrit son livre* PRES PROG max-ECRIT-son-livre
2	max has written his book PRES PERF(e) CUL max-WRITE-his-book *max a ecrit son livre* PRES PERF(e) CUL max-ECRIRE-son-livre
3	max has written his book for ten years PRES PERF(e) CUL FOR(ten-years) PROG max-WRITE-his-book *max ecrit son livre depuis dix ans* PRES DEPUIS(dix-ans,e) PROG max-ECRIRE-son-livre
4	max has known marie for ten years PRES PERF(e) CUL FOR(ten-years) max-KNOW-marie *max connait marie depuis dix ans* PRES DEPUIS(dix-ans,e) max-CONNAITRE-marie
5	max has been writing his book for ten years PRES PERF(e) CUL FOR(ten-years) PROG max-WRITE-his-book *max ecrit son livre depuis dix ans* PRES DEPUIS(dix-ans,e) PROG max-ECRIRE-son-livre
6	max had been writing his book for ten years PAST PERF(e) CUL FOR(ten-years) PROG max-WRITE-his-book *max ecrivait son livre depuis dix ans* PAST DEPUIS(dix-ans,e) PROG max-ECRIRE-son-livre *max avait ecrit son livre pendant dix ans* PAST PERF(e) CUL PENDANT(dix-ans) max-ECRIRE-son-livre
7	max had known marie for ten years PAST PERF(e) CUL FOR(ten-years) max-KNOW-marie *max connaissait marie depuis dix ans* PAST DEPUIS(dix-ans,e) max-CONNAITRE-marie *max avait connu marie pendant dix ans* PAST PERF(e) CUL PENDANT(dix-ans) max-CONNAITRE-marie

Progressive

This case is represented by example 1. The translation is accounted for semantically, if one is prepared to believe that `max-ECRIT-son-livre` is semantically the same as `max-WRITE-his-book`. The interesting aspect of this example is that we have chosen to implicitly mark `PROG` in the French reading. An alternative reading would omit the `PROG` operator, and this would be the translation of the English simple present. It could not be the translation of the present progressive, as it would not be semantically equivalent. This implicit marking is supported by the observation that sometimes it is desirable to distinguish the progressive reading of the present by the use of e.g. être en train de. The past progressive follows similarly, assuming the imperfective tense in French is interpreted via the operator `PAST`.

Present perfective [progressive]

The present perfective is represented by examples 2 – 4, and the perfective progressive by 5. For 2, the translation is direct. In 3 and 4, the insertion of the for adverbial in the English, causes the perfective construction to be omitted from the French translation. To account for this, a new operator DEPUIS has been introduced.

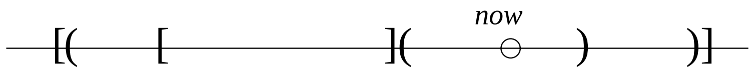

Figure 8: Present Perfective

Figure 8 represents an interpretation in which both English and French readings of 3 are true. The larger closed interval is an occurrence of the event denoted by max-WRITE-his-book / max-ECRIRE-son-livre; the larger open interval represents the span over which the progressive of this event holds true. The smaller closed interval is a period lasting ten years/dix ans (i.e. an occurrence of ten-years / dix-ans).

The semantic interpretation of FOR(t) A is such that the event denoted by this expression satisfies the following conditions:

- they must be occurrences of t
- they must not be instants
- all sub-intervals of their interiors must be occurrences of A

The last condition ensures that FOR may only be applied to stative events; this explains the implicit marking of PROG, which in turn may explain why 3 seems a bit strange, in contrast to 5, where the progressive is explicitly marked and which has the same reading as 3.

We also insist that (durative) time terms (e.g. ten-years) themselves denote events that are compatible with culminated processes (i.e. their occurrences must be closed intervals, and there must be enough of them to be able to construct a culminated process from a subset). This means that it is quite possible for FOR(t) A to denote a culminated process. Supposing that such is the case in this interpetation, the smaller closed interval is then an occurrence of the culminated process denoted by FOR(ten-years) PROG max-WRITE-his-book, so is ten years long and during the interior of which Max is in the progress of writing his book. CUL returns the culmination of this culminated process, so the right bound of this interval is an occurrence of CUL FOR(ten-years) PROG max-WRITE-his-book.

PERF(e) A denotes a *consequent* state, selected by context via e, of the culmination denoted by A. If A does not denote a culmination then the null event is returned. An occurrence of the consequent state, in this case, is represented by the smaller open interval which is left-bounded by the occurrence of the culmination mentioned above. PRES just requires this consequent state to be true at *now*. To summarise, the semantic interpretation of the English reading of 3 requires *now* to be in the consequent state of Max having written his book for a period of ten years.

The semantic interpretation of the French reading is similar, except that `DEPUIS` takes on the combined effect of `PERF`, `CUL` and `FOR`. `DEPUIS` is also stricter than `FOR` in the sense that it does not allow the larger closed interval to coincide with the smaller closed interval. In other words, it insists that the culminated process must be in progress for longer than the specified time period. `FOR` does allow these intervals to coincide: below, we will see how this allows for alternative translations in the past perfective case.

The basic event in 4 is a state, so to include the `PROG` operator would make no sense. Otherwise the analysis is the same as for 3 and 5.

Past perfective [progressive]

In contrast to the previous case, 6 and 7 each have two translations, involving *depuis* and *pendant*, respectively. The *depuis* translations are similar to the present perfective case, the only difference being that *now* appears after an occurrence of the consequent state, rather than within it—i.e. their was a time in the past when the consequences of max writing his book for ten years were being felt. The *pendant* translations exploit the flexible interpretation of `FOR`, which, you will recall, allows the occurrence of `max-WRITE-his-book` and `FOR(ten-years) PROG max-WRITE-his-book` to coincide. The interpretation of `PENDANT` is like `FOR` except that it forces these occurrences to coincide. This captures the intuition that the *pendant* translations require Max to have stopped writing / to no longer know Marie when in the consequent state.

One may observe that there is no reason in our theory why similar *pendant* translations could not be given in the present perfective case. One possible way of accounting for this is to say that the default interpretations of the English readings of 3, 4 and 5 would be ones in which the relevant intervals do *not* coincide; to counteract this default one would insert an extra qualification, as in Max has been writing his book for ten years, but has now decided to give up.

6 Conclusions

In summary, this paper has described how both feature and model based approaches to the semantics of verb phrases may assist with translation and contrastive semantics. The feature-based approach is relatively simple, and readily mechanisable. The model-based approach provides a much deeper semantic analysis and promises to support a more sophisticated approach to translation via logical proof; although, for this to be realised, more research is required into determining an underlying logic of events, including an appropriate (mechanisable!) deduction calculus. The translation examples considered in this paper are handled correctly by both approaches, but do not provide sufficient data to favour one approach over the other. We suspect that the feature-based approach would come unstuck sooner than the model-based, as language coverage was increased.

The model-based approach also raised some interesting issues concerning the process of contrastive semantic study. The discipline of the model-based approach requires the semantic theory to account for linguistic behaviour, expressed in terms of felicitousness and entailment constraints. It is not clear how to express the relationship between source and target sentences in a translation in these terms: must there be semantic equivalence, or is some form of 'strengthened' entailment sufficient? Where does context fit in? This issue manifests itself in the past perfective case, where there are two possible translations of the for adverbial. Clearly each of these is not semantically equivalent to the English, although perhaps considered together, in some sense, they are. Alternatively one could stick with semantic equivalence (possibly with respect to some context) as the relationship in translation. But then, in this case, one would have to modify the semantic theory for the English fragment, by interpreting for using two operators, say `FOR1` and `FOR2`, corresponding to the two different facits of its meaning. This suggests that a contrastive semantics study may help to construct a better semantic theory for each of the participant languages, than if they were analysed in isolation.

Further work includes

- The extension of the analyses to a wider coverage of English, French and German verb phrases and temporal adverbials, and to the consideration of sentences embedded in discourse, rather than being treated wholly in isolation.
- Consideration of other languages, in particular languages which have less in common.
- The development of a logic of events, to support the translation process under the model-based approach. This can only be influenced positively by examining languages in addition to English.
- Further investigation into the process of contrastive semantic analysis from a formal perspective.

Acknowledgements

We would particularly like to thank Ann Tucker for her collaboration, and also to thank Anja Belz, Jennifer Jetter, and Gabrielle McLellan for their advice on German and French.

References

[Antworth, 1990] E. Antworth. *PCKIMMO: A Two-Level Processor for Morphological Analysis.* Summer Institute of Linguistics, Dallas, TX, 1990.

[Cooper, 1985] R. Cooper. Aspectual Classes in Situation Semantics. CSLI Stanford Rep. No. CSLI-84-14C, 1985.

[Dowty, 1979] D. Dowty. *Word Meaning and Montague Grammar.* D. Reidel, 1979.

[Gazdar *et al.*, 1985] G. Gazdar, E. Klein, G. Pullum, and I. Sag. *Generalized Phrase Structure Grammar*. Blackwell, 1985.

[Grover *et al.*, 1992] C. Grover, J. Carroll, and E. Briscoe. The Alvey Natural Language Tools (4th Release). Cambridge University Computer Laboratory Tech. Rep. No. 284, 1992.

[Hinrichs, 1985] E. Hinrichs. *A Compositional Semantics for Aktionsarten and NP Reference in English*. PhD thesis, Ohio State University, 1985.

[Johnson, 1974] S. Johnson. YACC: Yet Another Compiler-Compiler. Bell Laboratories Computer Science Technical Report No. 39, 1974.

[Kent, 1993] S. Kent. *Modelling Events from Natural Language*. PhD thesis, Department of Computing, Imperial College of Science Technology and Medicine, 1993.

[Lascarides, 1988] A. Lascarides. *A Formal Semantic Analysis of the Progressive*. PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1988.

[Moens, 1987] M. Moens. *Tense, Aspect and Temporal Reference*. PhD thesis, Centre for Cognitive Science, University of Edinburgh, 1987.

[Montague, 1974] R. Montague. *Formal Philosophy*. Yale University Press, 1974.

[Parsons, 1985] T. Parsons. Underlying Events in the Logical Analysis of English. In *Actions and Events: Perspectives on the Philosophy of Donald Davidson*. Blackwell, 1985.

[Pitt and Cunningham, 1990] J. Pitt and J. Cunningham. Attributed Translation and the Semantics of Natural Language. In P. Deransart and M. Jourdan, editors, *Attribute Grammars and Their Applications*, volume 461 of *LNCS*, pages 284–297. Springer-Verlag, 1990.

[Pitt, 1991] J. Pitt. *Compositional Grammars and Natural Language Processing*. PhD thesis, Dept. of Computing, Imperial College of Science Technology and Medicine, 1991.

[Quirk *et al.*, 1985] R. Quirk, S. Greenbaum, G. Leech, and J. Svartvik. *A Comprehensive Grammar of the English Language*. Longmans, 1985.

[Reape, 1991] M. Reape. An Introduction to the Semantics of Unification Based Grammar Formalisms. Technical Report DYANA (Esprit BRA3175) Deliverable R3.2.A, 1991.

[Taylor, 1985] B. Taylor. *Modes of Occurrence: Verbs, Adverbs and Events*. Blackwell, 1985.

[Tucker, 1991] A. Tucker. Machine Translation of the English Verb Phrase. Master's thesis, Department of Computing, Imperial College of Science Technology and Medicine, 1991.

[Vendler, 1967] Z. Vendler. *Linguistics in Philosophy*. Cornell University Press, 1967.