On Implementing Swedish Tense and Aspect

Björn Gambäck
Stockholm

Abstract

The paper addresses the problems encountered when implementing a system for the treatment of Swedish tense, mood and aspect. The underlying theory suffered from the same shortcomings as do most implementable linguistic theories: it was designed for English. To extend it to Swedish some aspects of the theory, but also the implementation had to be generalized to allow for a system which treats Swedish verb-phrase syntax and semantics in a uniform way. This paper is concentrated on how this treatment actually has been implemented in a large-scale natural-language processing system.

1. Introduction

The theories for the treatment of tense and aspect phenomena in various languages are so many that it almost seems like any linguist (or at least any slavist and aspectologist) worthy of the name ought to have her own; however, not many of these theories have had any major impact on computational linguistics, possibly partially because most natural language systems are written for English where the "problems" caused by tense and aspect (at least at the surface) are not so complicated as to warrant the spending of too much development time and partially because most NL-systems simply do not have a life-span long enough for the issue to reach the implementation agenda.

The paper concentrates on the problems encountered when implementing a system for the treatment of Swedish tense, mood and aspect. The underlying theory was designed for English, so some aspects of it had to be generalized in order to extend it to Swedish. However, most of the treatment was still relevant, given that Swedish is not a language where aspect is too complicated, either. This objection is not as serious as it may sound, since the generalized version of the theory also should be able to treat such aspectual languages as Polish and Russian: a claim which however is not defended in the paper, neither a main point of it. A full-detailed discussion of Swedish verb-phrases in general will also be left aside; they are treated in length by other authors, for example Andersson (1977) and Tjekalina (1991).

It should be noted that in the title of the paper, as well as in the text following, the term "tense-aspect information" (or just T/A) and the like will be taken to refer to a range of phenomena that in principle just have a few properties in common, namely that they are (to certain extent) visual in the surface syntax, but in general have to be interpreted at a deeper semantic level. This is mostly, but not necessarily, because they are discourse rather than sentence related.

Apart from the obvious "tense" and "aspect" from the title, another category of the same kind that immediately springs to mind is "mood", but this paper will also include "voice" as belonging to the same broad type, while it (admittedly rather arbitrarily) will exclude for example "negation". It will also exclude phenomena which might have some aspects in common with the ones mentioned, but which is outside the current state of the art, for example "metaphor". The term "tense-aspect" used here is for the lack of a better one, but should thus not be taken as defining just those two categories, or defining one category of those two concepts. Or indeed as defining any categories at all, reflecting what (Chatterjee, 1982, p 337) refers to as the *categorial paradox*:

> "... a semantic or grammatical category is one only in relation to other 'neighboring' categories, yet we have not succeeded in isolating or defining a tense/aspect category (giving it *gesamtbedeutungen*) in the most studied languages. (...) Further, even if we did, our category would be language-specific, and so would its interaction with other categories of the language. (...) Aspect being to some extent notational (i.e., an investigative concept) in all languages, a universalist pinning down of the category is impossible."

The phenomena lumped together here as "T/A" will be more or less manifest in different languages, so for example for Swedish "tense" is a rather obvious candidate for discussion, and so is "mood", while "aspect" seems to be far more controversial. There has even been claims that there is no such thing as aspect in Swedish (Jordan Zlatev, personal communication, 1993). The following text will hopefully show that such claims are not to be taken too seriously. A more relevant question is raised by (Gawrónska, 1992), who argues that aspect in English and Swedish is in practice not as relevant as the introduction of (or lack of introduction of) the definite article, thus giving the definite article in these languages a role rather complementary to that of aspect in for example Russian and Polish.

The rest of the paper is outlined as follows: the next section will introduce the natural-language processing system used, the Swedish Core Language Engine. Section 3 contains a discussion of the treatment of verb-phrase syntax and semantics in the grammar, while Section 4 gets into some specific details of the implementation of the tense and aspect system.

2. The Swedish Core Language Engine

The Swedish Core Language Engine (S-CLE, Gambäck & Rayner, 1992) is a general-purpose natural-language processing system for Swedish which was developed from its English counter-part, the SRI Core Language Engine (CLE, Alshawi, 1992). The system is written purely in Prolog and based on unification as the main mechanism. The S-CLE is equipped with a sizable grammar for Swedish covering most common constructions in the language, including: questions (yes/no- and wh-), topicalized clauses, imperatives, passives, relative clauses, negation, cleft constructions, ellipsis, conjunction, noun-phrase and verb-phrase modification by preposition-phrases, adjectives and adverbs, various kinds of complex determiners, proper names, codes, dates and times, possessive constructions and about fifty different kinds of complements to verbs and adjectives. The grammar formalism is a feature-category type with declarative bidirectional rules, that is, the grammar can be used both for language analysis and for generation (it is just compiled in different ways depending on in what direction it is to be used)

A natural-language sentence that is input to the S-CLE is analysed to a logical-form like representation called QLF, Quasi-Logical Form, a conservative representation of the meaning of an input sentence based on purely linguistic evidence. The English and Swedish versions of the CLE have been used together to form a bidirectional translation system, transfer taking place at the QLF level (Alshawi *et al*, 1991), but the QLF can also be used as the basis for further (deeper-level) processing. Deriving a QLF from an NL-sentence involves the processing steps shown in Figure 1.

$$\begin{tikzcd} \text{NL} \arrow[r] & \boxed{\text{Morphology}} \arrow[r] & \boxed{\text{Syntax}} \arrow[r] & \boxed{\text{Semantics}} \arrow[r] & \text{QLF} \end{tikzcd}$$

Figure 1: The analysis steps of the S-CLE

First morphological analysis locate the correct word-senses and inflected forms of the input string, then syntactic parsing and (compositional) semantic analysis derive the parse tree(s) and its corresponding QLF-representation. Later processing steps (e.g., reference resolution and quantifier scoping) will try to further instantiate the QLF, aiming at deriving a "true" logical form (context and application dependent).

The S-CLE lexicon

The lexicon of the S-CLE is rather elaborate, with the lexical entries containing information both about morphological inflection patterns, syntactical subcategorization patterns and some semantical restrictions on the type of arguments. The lexicon form chosen for verbs is the imperative (rather than the "normal" dictionary form, the infinitive) since this form constitutes the stem of most other inflections, so a verb like *gilla* ("like") can be defined as being of the first declension, subcategorizing for an NP (i.e., being a transitive verb) and having the restrictions that it is a physical, nonpropositional, located event obtaining between a human subject and an object which basically can be anything, thus:

```
lr(gilla,v_subj_obj(v1,n),gilla_3p).
sor(gilla_3p,[[physev,nonprop,located],[human],
[]]=>[prop]).
```

where `gilla_3p` is the semantic constant used to identify the verb *gilla*, `v_subj_obj` is the pattern of a regular transitive verb which passivizes, `v1` the first declension of a non-deponent verb (`n`), and the Prolog-type list at the end of the second line introduces the restrictions on the event itself, the subject, the object and finally on the overall statement produced (a proposition, `prop`).

This *implicit* verb entry is in turn expanded out automatically using explicit paradigm (prototype) syntactic and semantic entries for verbs of the `v_subj_obj` (transitive) type. Schematically,[1] the syntactic one is

```
paradigm('verb_subj_obj'(Conjugation,Deponent),
    v:[@conjugation(Conjugation), @deponent(Deponent),
       vform=impera, gaps=Gaps,
       subcat=[np:[gaps=Gaps]]]).
```

Where the notation is to be interpreted so that an element belonging to the `v_subj_obj` paradigm is a verb (i.e., has the *category name* `v`) which has a list of *feature-value pairs* associated with it. So, for example `subcat` (for subcategorization) is a feature of the verb having a value which in turn is a list consisting of just one element, an `np` (the object). That NP also has a list of feature and values; some of the values are unified with the corresponding values on the verb, the `gaps` feature (which holds a list of empty constituents found within the phrase) for one is thus shared between the verb and its object.

[1]All lexicon entries and grammar rules exemplified in this paper are simplified. Features and other information not relevant for the discussion at hand have been removed to improve readability.

Of the other features, the `vform` specifies that the verb-form found in the lexicon is the imperative, while `@conjugation` and `@deponent` are macros (introduced by the `@` operator). a phenomena to be further discussed below. For now it is enough to note that they instantiate some morphological features with values following the `v1` and `n` declarations from an implicit entry as the one for *gilla* shown above.

3. Verb-phrase syntax and semantics

As indicated by the title, this paper is mainly devoted to how a theory of Swedish verb-phrases actually was implemented. The next section will go into the some of the more specific implementation details, but we will start out with a discussion of the overall verb-phrase grammar rules, illustrated by the how the rules actually have been implemented.[1]

Syntax

From a theoretical view-point, the aim of this work is to establish a *uniform* treatment of Swedish verb-phrases of any kind, be it with or without modification or with different types of verbal complements. In order to reach that goal, a syntactic grammatical rule as the following is central:

```
vp:[gaps=G, vform=Vf]
 -->
v:[gaps=G, vform=Vf, subcat=Complements]
 +
Complements
```

This rule should be read so that the feature `subcat` on a verb in effect specifies the number of constituents to be found in a verb-phrase, since the rest of the right-hand side of the rule only is specified as being something which is unified with the value `Complements.` As we saw already in the previous section, the value of `subcat` for a particular verb is specified in its lexical entry, so if the verb found is “gilla”, its subcategorization will be instantiated to be an NP.

Thus the above rule actually is a rule *schema*, replacing a multitude of verb-phrase formation rules which could have been written explicitly. The rule is both elegant in its simplicity and useful in that it helps in avoiding redundancy in the grammar and saves the grammar writer time:

[1]For a more consistent and implementation independent description of the theory, see Gambäck (1993).

the S-CLE contains some 50 different types of verbs all of which are treated with the same schema; writing a separate rule for each verb-type would of course have been possible, but hardly feasible.

Semantics

Looking at the semantic side the situation gets a bit more complicated; while main verbs still can be treated easily by a verb-phrase formation rule parallel to the single syntactic one, care has to be taken while treating auxiliaries.

The main verb case simply adds semantic information to the syntactic rule:

```
(V,
 vp:[@shared_tense_aspect(T,U)])
 -->
(V,
 v:[@shared_tense_aspect(T,U), arglist=Complements])
 +
Complements
```

Here, each constituent of the rule is a pair `(QLF,Category)`, where the `Category` holds the same information as in the syntactic case (i.e., consists of the category name followed by a list of feature-value pairs), while the QLF is the semantic information, a logical form fragment. Thus `arglist` is a feature performing exactly the same function as `subcat` above, but with the semantic information added. The value `V` for both the verb's and the verb-phrase's logical form indicates that the verb's semantic interpretation is passed up (by unification) to become the interpretation of the entire verb-phrase.

`@shared_tense_aspect` is a macro which (also by unification) passes the tense-aspect information up from the main verb to the verb-phrase. The rule does not explicitly take care of the semantic interpretations of the complements: this information is, however, simply unified into the verb's semantics in the lexicon.

Tense auxiliaries

Auxiliaries that change the tense of the verb-phrase (e.g., to past as *hade*, "had", or future as *ska*, "shall") must be treated separately from the main-verb case. As was shown above, both the semantic interpretation and the tense information for main verbs and for their "mother" verb-phrases are the same; however, in the auxiliary case, the semantic interpretation of the mother verb-phrase should still be the one of the daughter verb-phrase, but the tense should be taken from the auxiliary, so this case of the rule becomes:

```
(V,
 vp:[@shared_tense_aspect(T,U)])
 -->
(empty,
 v:[@shared_tense_aspect(T,U), arglist=(V,vp:[])])
+
(V,
 vp:[])
```

Where the auxiliary is shown to be a verb which subcategorizes for a verb-phrase with the same semantics as the mother VP, but itself carrying no semantic information proper (i.e., the QLF of the V is empty). The tense-aspect information of the daughter verb-phrase is left out from the rule, indicating that it should not influence the T/A of the mother; however, it may, but this should be treated in the lexical entry for the auxiliary.[1]

Modal auxiliaries

Modal auxiliaries complicate the picture somewhat: we need to treat two cases, one for finite and one for non-finite (i.e., infinite plus supine) verb forms, the difference being that the former (in Swedish, but not for example in English) can modify other modals as in a sentence like

Jag skulle vilja kunna flyga.	*"I would like to be able to fly"*
	(lit. "I should want could fly")

In examples like this one (where at least the *skulle vilja* construction is very common), finite modals behave quite a bit like tense auxiliaries; they do not affect the semantic content as such, but rather the modal information, which (as we shall see in the next section) can be taken to be part of the tense-aspect information, so that finite modals actually can be treated with exactly the same case of the verb-phrase formation rule as tense auxiliaries. Non-finite modals on the other hand behave just like ordinary verbs in the effect they have on the semantic interpretation

[1]It should be noted that in the actual implementation the auxiliary QLF may be non-empty (it can contain imformation about verb-modification by so called "mobile adverbs" – e.g., the negation marker *inte)* and is thus taken care of properly, anyhow. A full description of the implementation of negation would, however, hardly add to the present discussion and is thus (and for space considerations) left out here.

proper. They are thus treated with the same rule instance as the main verbs.

4. Implementational aspects

For an inflectional language like Swedish, where most of the tense and aspect information can be found in the suffix of the main verb, it is natural to view the tense-aspect information as forming a function of the affix. For the actual implementation, we represent it in the compositional semantics as a functor

```
verb(Tense,Aspect,Action,Mood,Voice)
```

where the information is filtered up from the verb-affix to the verb phrase. The arguments of `verb` will be explained further on; first, however, we should note that the choice of this functor is rather (but not completely) arbitrary. For a language such as Finnish where the aspect information is carried on the object rather than the predicate some other functor name of course should be chosen. Also, the number of arguments and their interpretation could certainly vary between languages (or between linguists and linguistic theories treating the same language), but in general we need a strategy as the one suggested by (Alshawi & Crouch, 1992): first a way to packet the tense-aspect information declaratively in the compositional semantics and then a way to unpack this information later on to determine the implicit points in time, etc., not shown in the surface form of the sentence. This "packaging" is the main function of the functor `verb`, whose arguments are in order:

`Tense`	the relation of the event to the present time of the speaker: past, present, or future.
`Aspect`	the relation of the event to the action time of the verb: perfective or imperfective.
`Action`	the way in which an event happens: progressive or non-progressive.
`Mood`	the speaker's view on the event: a modal, imperative, etc.
`Voice`	the relation of the meaning of the verb to the subject: active or passive.

Morphology

As noted above, the lexicon form chosen for the Swedish verbs is the imperative, since this form constitutes the stem of most other inflections. For tense and aspect purposes, however, the imperative is a bit peculiar:

it stands almost on the side of the entire tense-aspect system. Thus the lexicon contains stems for which the T/A information is only partially instantiated (viz., `verb(no,Pf,Pg,imp,A)`). The (normally) full instantiation is obtained by the inflection in morphology rules as the following

```
(@verb_semantics(Sense,TA,Event,Args),
 v:[@shared_tense_aspect(TA,U)])
  -->
(@verb_semantics(Sense,_,Event,Args),
 v:[])
  +
(suffix,
 suffix:[@shared_tense_aspect(TA,U)])
```

which shows that the mother verb is formed by adding a suffix to the daughter verb (i.e., the stem form). Just as in the sematic grammar rule above, each of the three components of the rule consists of two parts: the semantic information (here, a QLF fragment) and the category name followed by a list of feature-value pairs. The variables `TA` and `U` together carry the tense-aspect information: `TA` holding the T/A information proper, while `U` keeps track of the as-of-yet uninstantiated information. The T/A information from the suffix is passed up to the inflected verb by unification. This is also the only (semantic) information added by the suffix; the other parts of the mother-verb semantics come from the daughter, i.e., the sense name `Sense` (as `gilla_3p` above), the variable `Event` representing the event itself and the list of the verb's arguments' (e.g., objects') QLF-fragments, `Args`.

Suffixes

An example of a suffix entry is the one for the ending "-r", which is added to the stem of some verbs to form the present tense:

```
sense('-r',
suffix:[@pres_mainv(TenseAspect),
        vform=(fin/\present),
        synmorphv=(1\/3\/43),
        lexform='-r'],
suffix).
```

The value of the feature `vform` indicates that the inflected verb produced will be in present and finite form,[1] while the feature `synmorphv` restricts

[1]The symbols "/ \" and "\ /" functions as the normal logical "and" and "or" operators, respectively.

the syntactic morphological categories (i.e., verb declensions) for which the ending "-r" is appropriate. Here, they are verbs belonging to the 1st and 3rd declension as well as those belonging to the 4th declension., 3rd subgroup.

The first `'-r'` is the sense name of the suffix, while the second (the value of the feature `lexform`) is its actual realization in the surface string. In the same fashion, the first `suffix` is the rather arbitrary name of the suffix' category in the grammar, while the second holds the semantic content (the logical-form fragment) obtained from the suffix. The latter is in reality none at all (apart from the T/A information), so the second `suffix` is mainly a place-holder.

For the syntactic analysis of the system implemented, the logical form of the suffix entry could be completely uninstantiated; however, it is worth noting that given that the grammar is bidirectional, we do not want to leave the suffix' semantic content uninstantiated, the generation algorithm used in the S-CLE ("Semantic-Head-Driven Generation", Shieber *et al*, 1990) actually requiring all logical-form fragments to be instantiated.

Macros

For the purposes of this paper, the most important part of the suffix entry above is `@pres_mainv(TenseAspect)`, which actually is a macro call. The full macro definition used for present tense main verbs is

```
macro(pres_mainv([pres,no,no,no,y]),
        [tense=pres, perf=no, prog=no, modal=no, active=y,
         uninstTA=l([])]).
```

which shows that the tense-aspect information in reality is carried by a whole group of feature-value pairs, which together hold the same information as in the previously described `verb` functor. Thus the first feature-value pair indicates the present tense, the second shows the imperfective aspect and the third the non-progressive action. No specific modal information is added by this macro, but the voice of the verb has to be active. Since all the other five T/A features are instantiated, the final `uninstTA` feature just holds an empty (Prolog-type) list. The `@pres_mainv` macro is complemented with a number of macros for all the different inflectional forms of verbs and for both main verbs and auxiliaries; the main ones are listed in the appendix.

In the same fashion, the entries @shared_tense_aspect and verb_semantics in the verb-affixing rule above are also macro calls, their full definitions being

```
    macro(shared_tense_aspect([T,Pf,Pg,M,A],U),
          [tense=T, perf=Pf, prog=Pg, modal=M, active=A,
           uninstTA=U]).

    macro(verb_semantics(Sense, [T,Pf,Pg,M,A],Event,Args),
          @form(verb(T,Pf,Pg,M,A), Event,
                P^[P, [Sense,Event|Args]], _)).
```

The first macro is just a convenient way to address all the T/A features at once, while the second one gives the current version of the semantics chosen for event verbs. It is out of the scope of the present chapter to go into the details of the QLF formalism (the interested reader is referred to Alshawi, 1992), so we only note that the semantics of the verb is a form which includes the verb functor as defined above, the Event variable and the actual (body) semantics of the verb which is a lambda-abstraction[1] with the Sense name as a function whose parameters are the Event variable followed by the logical forms of the complements (Args).

[1] ^ is a type-writer version of the more common λ, so P^[P,Q] is equivalent to $\lambda P.Q(P)$.

References

Alshawi, Hiyan (ed.). 1992. *The Core Language Engine*. The MIT Press, Cambridge, Massachusetts.

Alshawi, Hiyan and Richard Crouch. 1992. *Monotonic Semantic Interpretation*. pp 32–39, *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, Newark, Delaware.

Alshawi, Hiyan, David M. Carter, Björn Gambäck and Manny Rayner. 1991. *Translation by Quasi Logical Form Transfer*. pp 161–168, *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, University of California, Berkeley, California.

Andersson, Erik. 1977. *Verbfrasens struktur i svenskan: en studie i aspekt, tempus, tidsadverbial och semantisk räckvidd*. Doctor of Philosophy Thesis, Åbo Akademi, Åbo, Finland (in Swedish).

Chatterjee, Ranjit. 1982. *On Cross-Linguistic Categories and Related Problems*. pp 335–345, *Tense-Aspect: Between Semantics & Pragmatics*. Paul J. Hopper, editor, John Benjamins, Amsterdam, Holland.

Gambäck, Björn. 1993. *Towards a Uniform Treatment of Swedish Verb Syntax and Semantics*. *Proceedings of the 16th Scandinavian Conference of Linguistics and the 8th Conference of Nordic and General Linguistics*, University of Gothenburg, Gothenburg, Sweden.

Gambäck, Björn and Manny Rayner. 1992. *The Swedish Core Language Engine*. pp 71–85, *Papers from the 3rd Nordic Conference on Text Comprehension in Man and Machine*, Linköping University, Linköping, Sweden. Also available as SICS Research Report, R92013, Stockholm, Sweden.

Gawrónska, Barbara. 1992. *An MT Oriented Model of Aspect and Article Semantics*. Doctor of Philosophy Thesis, Lund University, Lund, Sweden.

Shieber, Stuart M., Gertjan van Noord, Fernando C.N. Pereira, and Robert C. Moore. 1990. *Semantic-Head-Driven Generation*. pp 30–43, COMPUTATIONAL LINGUISTICS, Volume 16.

Tjekalina, Elena. 1991. *Om kategorierna modus och tempus i nutida svenska*. pp 139–155, SPRÅK & STIL, Volume 1 (in Swedish).

Appendix: Verb inflection macros

Imperatives (no tense)

```
macro(imperative([no,Pf,Pg,imp,A]),
      [tense=no, perf=Pf, prog=Pg, modal=imp, active=A,
       uninstTA=l([A,Pf,Pg])]).
```

Main verbs and VP complements

```
macro(pres_mainv([pres,no,no,no,y]),
       [tense=pres,perf=no,prog=no,modal=no,active=y,
        uninstTA=l([])]).
macro(past_mainv([past,no,no,no,y]),
       [tense=past,perf=no,prog=no,modal=no,active=y,
        uninstTA=l([])]).

macro(inf_mainv([T,no,no,M,y]),
       [tense=T,perf=no,prog=no,modal=M,active=y,
        uninstTA=l([T,M])]).
macro(perfp_mainv([T,yes,Pg,M,A]),
       [tense=T,perf=yes,prog=Pg,modal=M,active=A,
        uninstTA=l([A,T,Pg,M])]).
macro(perfp_intrans([past,yes,no,M,y]),
       [tense=past,perf=yes,prog=no,modal=M,active=y,
        uninstTA=l([M])]).
macro(perfp_transevent([past,yes,no,M,n]),
       [tense=past,perf=yes,prog=no,modal=M,active=n,
        uninstTA=l([M])]).
macro(perfp_transstate([pres,yes,yes,M,n]),
       [tense=pres,perf=yes,prog=yes,modal=M,active=n,
        uninstTA=l([M])]).
macro(presp_mainv([pres,Pf,yes,M,y]),
        [tense=pres,perf=Pf,prog=yes,modal=M,active=y,
        uninstTA=l([Pf,M])]).

macro(pass_mainv([T,Pf,Pg,M,n]),
        [tense=T,perf=Pf,prog=Pg,modal=M,active=n,
        uninstTA=l([T,Pf,Pg,M])]).

macro(supinev([T,yes,no,M,y]),
       [tense=T,perf=yes,prog=no,modal=M,active=y,
        uninstTA=l([T,M])]).
```

Modals (depends on the verb-form – the second argument)

```
macro(modal_tense_aspect(M,no,[no,Pf,Pg,M,A]),
       [tense=no, perf=Pf, prog=Pg, modal=M, active=A]).
macro(modal_tense_aspect(M,pres,[pres,Pf,Pg,M,A]),
       [tense=pres, perf=Pf, prog=Pg, modal=M, active=A]).
macro(modal_tense_aspect(M,past,[past,Pf,Pg,M,A]),
       [tense=past, perf=Pf, prog=Pg, modal=M, active=A]).
macro(modal_tense_aspect(M,inf,TenseAspect),
       [@inf_mainv(TenseAspect)]).
macro(modal_tense_aspect(M,supine,TenseAspect),
       [@supinev(TenseAspect)]).
```