Topological Dependency Analysis of the Dutch Verb Cluster

Denys Duchier
Programming Systems Lab
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
duchier@ps.uni-sb.de

Ralph Debusmann
Computational Linguistics
Universität des Saarlandes
Postfach 15 11 50
D-66041 Saarbrücken, Germany
rade@coli.uni-sb.de

July 31, 2007

Abstract

Topological Dependency Grammar (TDG) is a new lexicalized formalism where word order phenomena arise from the interaction of a non-ordered syntax tree with a projective topological tree, both related by an emancipation mechanism. TDG proved capable of an elegant account of German word-order phenomena. We apply it now to the modelization of word-order variation in the Dutch verb cluster and further extend it with an ordering principle that correctly accounts for the ordering of elements in the Mittelfeld.

1 Introduction

Topological Dependency Grammar (henceforth TDG) is a lexicalized grammar formalism proposed by (Duchier and Debusmann, 2001) where licensed analyses emerge from the interactions of a non-ordered tree of syntactic dependencies with a corresponding projective tree of topological dependencies, both related by an emancipation mechanism. (Gerdes and Kahane, 2001) independently suggested a similar approach but with a representation of topology based on phrase structure rather than immediate dependency. TDG proved capable of an elegant account of many word order phenomena in German (Debusmann, 2001).

In the present article, we examine word order in Dutch subordinate clauses and develop a TDG account which addresses the phenomena of cross-serial dependencies, verb-raising, infinitivus pro participio, inversion, full and partial extraposition. We also outline how to accommodate some dialectal variations. While TDG had so far not paid special attention to word order within the Mittelfeld, this issue must be addressed in Dutch: we propose a simple ordering principle which achieves an effect similar to the argument composition mechanism of (van Noord and Bouma, 1996).

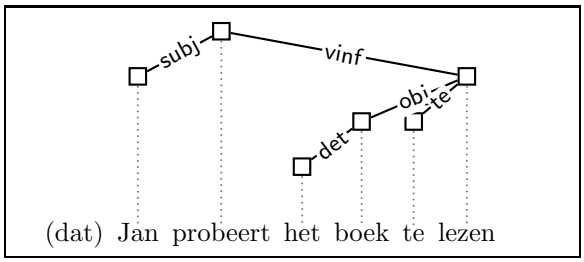
2 TDG Framework

In this section, we introduce the TDG framework informally using our simple Dutch grammar for illustration. The formal foundations can be found in (Duchier, 2001). The full lexicon assumed for this article is summarized in Table 1.

A TDG analysis consists of two trees, the ID tree and the LP tree, which are formed from the same set of nodes (one for each word

of the input) but different sets of edges.

The ID tree is a non-ordered tree of syntactic dependencies where edges are labeled with grammatical rolections such as subj for subject or obj for object.

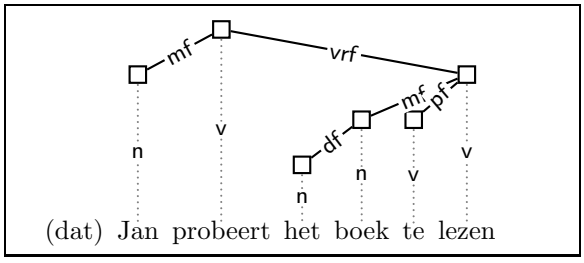


For the purpose of this article, the set $\mathcal{R}$ of grammatical roles is simply:

$$\mathcal{R} = \{\mathsf{subj}, \mathsf{iobj}, \mathsf{obj}, \mathsf{vbse}, \mathsf{vprt}, \underline{\mathsf{vinf}}, \mathsf{vinf}, \overline{\mathsf{vinf}}, \mathsf{te}, \mathsf{det}\}$$

corresponding respectively to subject, indirect object, direct object, bare infinitive, past participle, stripped infinitive, infinitive, saturated infinitive,[1] *te* particle, and determiner. In the ID tree, we say that a node is the syntactic *head* of its immediate daughters.

The LP tree is a tree of topological dependencies. It is ordered and projective, and both nodes and edges are labeled by fields such as mf for Mittelfeld.



In the LP tree, we say that a node is the topological *host* of its immediate daughters, and that the daughters are *guests* of their host.

[1] $\underline{\mathsf{vinf}}$, vinf, and $\overline{\mathsf{vinf}}$ respectively model *te*-infinitives whose arguments *must* all be raised, *may* optionally be raised, or *must not* be raised. Bare infinitives (vbse) behave like $\underline{\mathsf{vinf}}$-infinitives: their arguments must also be raised.

For this reason we call the set $\mathcal{F}_\mathsf{G}$ of edge labels *guest fields* and the set $\mathcal{F}_\mathsf{H}$ of node labels *host fields*.

$$\mathcal{F}_\mathsf{G} = \{\mathsf{df}, \mathsf{mf}, \mathsf{vrf}, \mathsf{vlf}, \mathsf{pf}\} \qquad \mathcal{F}_\mathsf{H} = \{\mathsf{n}, \mathsf{v}\}$$

df is the determiner field, mf the Mittelfeld, vrf the verbal right field (canonical position of verbal arguments), vlf the verbal left field (inverted verbal arguments), and pf the particle field (for *te*). The set $\mathcal{F} = \mathcal{F}_\mathsf{G} \uplus \mathcal{F}_\mathsf{H}$ of fields is totally ordered:

$$\mathsf{df} \prec \mathsf{n} \prec \mathsf{mf} \prec \mathsf{vlf} \prec \mathsf{pf} \prec \mathsf{v} \prec \mathsf{vrf}$$

To be admissible, the LP tree must be well-ordered, i.e. locally for each node, the order of its outgoing edges must respect the order on their labels. Thus:

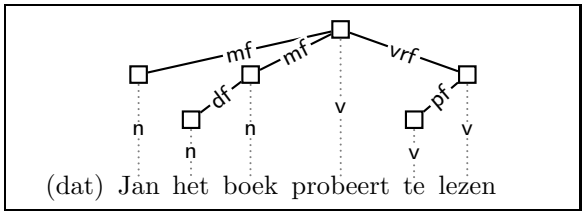
(1) dat Jan probeert het boek te lezen

(2) ∗dat Jan probeert te het boek lezen

(1) is well-ordered, but (2) is not because it violates the order $\mathsf{mf} \prec \mathsf{pf}$ locally on *lezen*. Further, the assignment of a host field to a node determines how it must be placed with respect to its guests and their recursive topological dependents. Thus, in our example, the order $\mathsf{mf} \prec \mathsf{v} \prec \mathsf{vrf}$ mandates the linearization:

$$[\text{Jan}] \prec \text{probeert} \prec [\text{het boek te lezen}]$$

For an analysis to be admissible, the shape of the LP tree must be derived from the ID tree's by a flattening process of emancipation called *climbing*: a node (and its subtree) may migrate upwards in search of a topological host. Thus, in our example, *het boek* may also climb up to the finite verb *probeert*:



A TDG analysis is further constrained by an assignment of lexical entries to nodes. Given a set $\mathcal{L}$ of labels, we write $\Pi_{\mathcal{L}}$ for the set of label patterns π that can be formed according to the following abstract syntax:

$$\pi \quad ::= \quad \ell \mid \ell* \mid \ell? \qquad\qquad \forall \ell \in \mathcal{L}$$

Patterns are used for subcategorization constraints: ℓ means precisely one edge labeled ℓ, $\ell*$ means 0 or more ℓ-edges, and $\ell?$ at most 1 ℓ-edge. A lexical entry has the signature:

$$\begin{bmatrix} \mathsf{cat}_{\text{ID}} & : & 2^{\mathcal{R}} \\ \mathsf{subcat}_{\text{ID}} & : & 2^{\Pi_{\mathcal{R}}} \\ \mathsf{cat}_{\text{LP}} & : & 2^{\mathcal{F}_{\mathsf{G}}} \\ \mathsf{subcat}_{\text{LP}} & : & 2^{\Pi_{\mathcal{F}_{\mathsf{G}}}} \\ \mathsf{hcat}_{\text{LP}} & : & 2^{\mathcal{F}_{\mathsf{H}}} \\ \mathsf{blocks} & : & 2^{\mathcal{R}} \end{bmatrix}$$

For example, one lexical entry for *kussen* is:

$$\begin{bmatrix} \mathsf{cat}_{\text{ID}} & : & \{\overline{\mathsf{vinf}}\} \\ \mathsf{subcat}_{\text{ID}} & : & \{\mathsf{obj}, \mathsf{te}\} \\ \mathsf{cat}_{\text{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\text{LP}} & : & \{\mathsf{mf}*, \mathsf{pf}\} \\ \mathsf{hcat}_{\text{LP}} & : & \{\mathsf{v}\} \\ \mathsf{blocks} & : & \mathcal{R} \end{bmatrix}$$

It simultaneously constrains the ID tree, the LP tree, and the emancipation relationship between them. $\mathsf{subcat}_{\text{ID}}$ constrains the outgoing edges of the node in the ID tree, while cat_{ID} restricts its incoming edge. We say that *kussen* ID-subcategorizes for obj and te: the node in the ID tree must have precisely one obj and one te outgoing edge and no other. Symmetrically, we say that it ID-categorizes for $\overline{\mathsf{vinf}}$: if an incoming edge exists, it must be labeled with $\overline{\mathsf{vinf}}$. Similarly, we say that it LP-subcategorizes for mf∗ and pf: the node in the LP tree must have 0 or more mf edges, 1 pf edge, and no other. It LP-categorizes for vrf: if an incoming edge exists in the LP tree, it must be labeled vrf; in other words, the lexical entry mandates that *kussen* land in the right verbal field. $\mathsf{hcat}_{\text{LP}}$ indicates that the node must be assigned "host" label v. Now consider an entry for *gekust*:

$$\begin{bmatrix} \mathsf{cat}_{\text{ID}} & : & \{\mathsf{vprt}\} \\ \mathsf{subcat}_{\text{ID}} & : & \{\mathsf{obj}\} \\ \mathsf{cat}_{\text{LP}} & : & \{\mathsf{vlf}, \mathsf{vrf}\} \\ \mathsf{subcat}_{\text{LP}} & : & \emptyset \\ \mathsf{hcat}_{\text{LP}} & : & \{\mathsf{v}\} \\ \mathsf{blocks} & : & \emptyset \end{bmatrix}$$

It LP-categorizes for either vlf or vrf,[2] meaning that it may either land in the right verbal field, or be subject to inversion and land in the left verbal field. Notice that it ID-subcategorizes for obj, but does not LP-subcategorize for mf∗ thus forcing its object argument to "climb" in search of a host offering a Mittelfeld.

Finally, the emancipation mechanism is also subject to lexical constraints. The blocks feature of a lexical entry is a set of grammatical roles for which this node acts as a barrier. For example, the entry for *kussen* above blocks all grammatical roles (because this is an entry for a saturated *kussen* as attested by the fact that it ID-categorizes for $\overline{\mathsf{vinf}}$). The entry for *gekust* blocks no role, while the entries for *moeten* block all verbal roles.

3 The Dutch Verb Cluster

We restrict our attention to non-finite complementation in verb-final sentences and distinguish three classes of verbs taking verbal complements: verb-raising verbs, partial extraposition verbs, and full extraposition verbs. Verb-raising verbs ID-subcategorize for vbse or $\underline{\mathsf{vinf}}$, i.e. verbal complements whose arguments *must* climb. Partial extraposition verbs ID-subcategorize for vinf, i.e. verbal complements whose arguments *may* climb. Full extraposition verbs ID-subcategorize for $\overline{\mathsf{vinf}}$, i.e. saturated verbal complements whose arguments consequently *must not* climb.

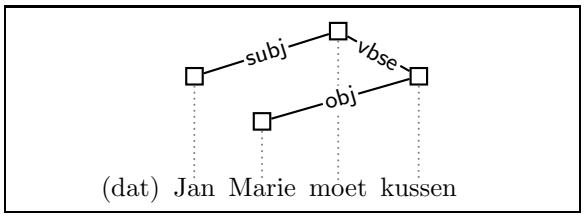
[2] The interpretation of cat_{ID} and cat_{LP} is disjunctive.

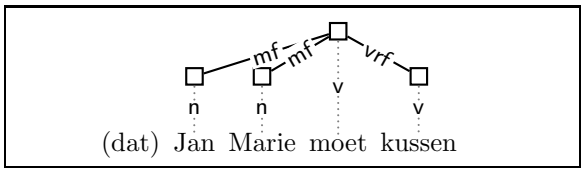
3.1 Verb-raising Verbs

In the ANS (the standard Dutch grammar), this class of verbs is called *obligatory group forming*. In fact, *group forming* is the characteristic property of verb-raising verbs: they force their governed verbs to raise their non-verbal complements into the Mittelfeld, leading to the formation of a contiguous group of verbs called *verb cluster*. The class of verb-raising verbs includes auxiliaries, modals, perception verbs, causatives, certain raising and control-verbs like *proberen* (*to try*) and the verb *helpen* (*to help*). Here is an example featuring the modal *moet*, leading to the formation of the verb cluster *moet kussen*:

(3) (dat) Jan Marie moet kussen
(that) Jan Marie must kiss
"(that) Jan must kiss Marie"

The TDG analysis of (3) is:





In our grammar, verb-raisers ID-subcategorize for either vbse (bare infinitive) or vinf (*te*-infinitive). *moet* for instance ID-subcategorizes for vbse:[3]

(4)
$$\textit{moet} \mapsto \begin{bmatrix} \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{subj}, \mathsf{vbse}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{vlf}?, \mathsf{vrf}?\} \end{bmatrix}$$

[3]In the following, we present only partial lexical entries containing only the relevant information for the context. For the full lexical entries cf. Table 1.

kussen in turn ID-*categorizes* for vbse. It ID-subcategorizes for obj while LP-subcategorizing for no field, thus forcing the object to climb up:

(5) *kussen* $\mapsto$
$$\begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\mathsf{vbse}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{obj}\} \\ \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vlf}, \mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \emptyset \end{bmatrix}$$

With this lexical entry for verbs governed by verb-raisers, we correctly exclude sentences as in (6) below where the object *Marie* has not climbed into the Mittelfeld but stayed within the verb cluster *moet kussen*:

(6) ∗dat Jan moet Marie kussen

3.1.1 Infinitivus pro participio-effect

If governed by the *hebben*-auxiliary, verb-raisers must show infinitival rather than past participle inflection. This is called the *Infinitivus pro participio*-effect (IPP-effect).

(7) (dat) Piet Jan Marie heeft zien kussen
(that) Piet Jan Marie has $see_{(inf)}$ *kiss*
"that Piet has seen Jan kiss Marie"

(8) ∗(dat) Piet Jan Marie heeft gezien kussen
(that) Piet Jan Marie has $seen_{(part)}$ *kiss*
"that Piet has seen Jan kiss Marie"

In our grammar, verb-raisers ID-categorizing for vprt must always show infinitival inflection. Therefore we exclude (8) by not including a lexical entry for *gezien as a verb-raiser* at all. However, as *zien* is ambiguous between a verb-raiser and a transitive main verb, we do add a lexical entry for the past participle *gezien*:

(9) *gezien* $\mapsto$
$$\begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\mathsf{vprt}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{obj}\} \end{bmatrix}$$

Notice that (9) only ID-subcategorizes for obj, i.e. it is a transitive verb having no verbal complement. Having this lexical entry, we cover e.g. the following sentence:

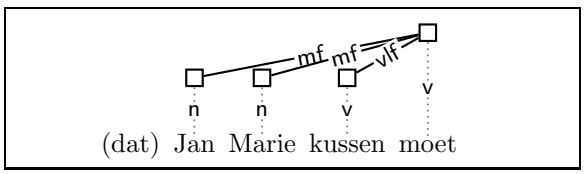
(10) (dat) Jan Marie heeft gezien
(that) Jan Marie has $seen_{(part)}$
"that Jan has seen Marie"

3.1.2 Inversion of Finite Verb-raisers

The standard order among verbs in the Dutch verb cluster prescribes that verbal governors precede their verbal complements.[4] However, some verbs permit a phenomenon called *inversion* where the governor follows its verbal complement rather than preceding it. (11) shows an example of inversion, where, in contrast to (3), *moet* follows its verbal complement *kussen*:

(11) (dat) Jan Marie kussen moet
(that) Jan Marie kiss must
"(that) Jan must kiss Marie"

Lexical entries (4) and (5) for *moet* and *kussen* already take inversion into account: *moet* LP-subcategorizes for both vlf and vrf, thus allowing verbal complements to land either to its left or to its right. Since *kussen* LP-categorizes for both vlf and vrf, it can land either to the left or to the right of its host. In the LP tree for (11) below, it lands in the vlf to the left of *moet*:



Of all the finite verb-raising verbs, only auxiliaries and modals permit inversion. Here is an example of inversion featuring the auxiliary *heeft*:

(12) (dat) Jan Marie gekust heeft
(that) Jan Marie kissed has
"(that) Jan has kissed Marie"

No other finite verb-raisers permit inversion, as shown in the following examples featuring the verb-raiser *ziet*:

(13) (dat) Piet Jan Marie ziet kussen
(that) Piet Jan Marie sees kiss
"(that) Piet sees that Jan kisses Marie"

[4]This is precisely the mirror image of the order in the German verb cluster where typically verbal governors *follow* their verbal complements.

(14)∗dat Piet Jan Marie kussen ziet

We exclude sentences like (14) as follows: all verb-raisers which do not permit inversion only LP-subcategorize for vrf but not for vlf. Hence, verbal complements can only land to the right of those verbs but not to the left. This is reflected in the following lexical entry for *ziet*:

(15)
$$\textit{ziet} \;\mapsto\; \begin{bmatrix} \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{subj}, \mathsf{obj}, \mathsf{vbse}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{vrf}?\} \end{bmatrix}$$

Inversion is disallowed if the governed verb is itself a verb-raiser. This requirement effectively rules out recursive inversion. In the following examples, the verbal complement of *moet* is the verb-raiser *hebben*:[5]

(16) (dat) Jan Marie **moet**$_1$ **hebben**$_2$ gekust$_3$
(that) Jan Marie must have kissed
"(that) Jan must have kissed Marie."

(17)∗(dat) Jan Marie **hebben**$_2$ gekust$_3$ **moet**$_1$

(18)∗(dat) Jan Marie **hebben**$_2$ **moet**$_1$ gekust$_3$

We exclude (17) and (18) by stipulating that non-finite verb-raisers can only land to the right of their governors, i.e. they LP-categorize only for vrf but not for vlf. For instance, here is the lexical entry for *hebben*:

(19) $$\textit{hebben} \;\mapsto\; \begin{bmatrix} \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{vrf}?\} \\ \mathsf{blocks} & : & \emptyset \end{bmatrix}$$

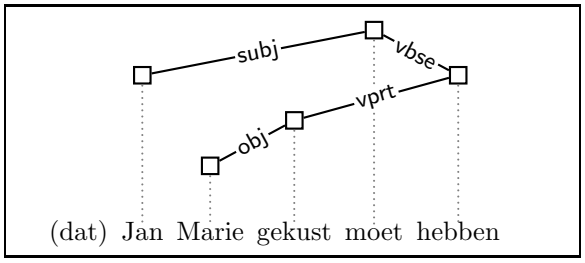
3.1.3 Inversion of Non-finite Verb-raisers

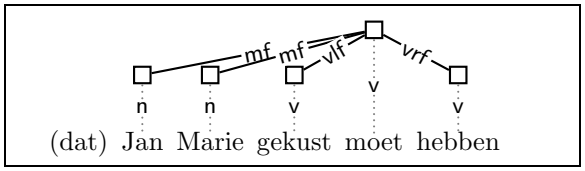
Non-finite auxiliaries also permit inversion. We give an example below where the verbal complement *gekust* of *hebben* is inverted and placed at the left periphery of the verb cluster:

[5]In examples involving more than two verbs we use indices to indicate the depth of embedding and use boldface to indicate the two verbs involved in inversion.

(20) (dat) Jan Marie **gekust**$_3$ moet$_1$ **hebben**$_2$
(that) Jan Marie kissed must have
"(that) Jan must have kissed Marie."

Here is the TDG analysis of the sentence:





Except for auxiliaries, all other verb-raising verbs do not permit inversion if in non-finite form. An example is the non-finite modal *moeten*:

(21) (dat) Jan Marie heeft$_1$ **moeten**$_2$ **kussen**$_3$
(that) Jan Marie has moeten kiss
"(that) Jan has had to kiss Marie."

(22)$*$(dat) Jan Marie **kussen**$_3$ heeft$_1$ **moeten**$_2$

We exclude (22) using barriers: we stipulate that non-finite verb-raising verbs which do not permit inversion (such as the modal *moeten*) *block* all verbal roles $VRoles = \{\mathsf{vbse}, \mathsf{vprt}, \underline{\mathsf{vinf}}, \mathsf{vinf}, \overline{\mathsf{vinf}}\}$:

(23) *moeten* $\mapsto$ $\begin{bmatrix} \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{vrf?}\} \\ \mathsf{blocks} & : & VRoles \end{bmatrix}$

Thus, while in (20) *gekust* climbed through *hebben* up to the finite verb *moet*, in (22) *kussen* would have to climb through *moeten* up to *heeft*. But this is ruled out by the fact that *moeten* blocks all verbal roles.

3.1.4 Dialectal Variation

In example (20), the inverted verb *gekust* moves to the left periphery of the verb-cluster. However, in some Dutch dialects (including Flemish) it is also possible to place the inverted verb directly to the left of its governor:
(24)?(dat) Jan Marie moet gekust hebben

In our TDG analysis, we can account for this form of dialectal variation as follows. For standard Dutch, we assume the lexical entry (19) for *hebben* which only offers vrf. For dialects such as Flemish, we employ a modified lexical entry which more leniently offers both vlf and vrf:

(25) *hebben* $\mapsto$ $\begin{bmatrix} \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{vlf?}, \mathsf{vrf?}\} \\ \mathsf{blocks} & : & \emptyset \end{bmatrix}$

3.2 Partial extraposition verbs

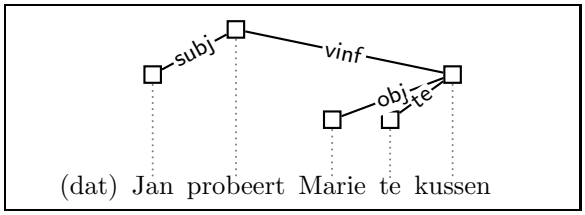
Partial extraposition verbs give rise to the so-called *third construction*. They exhibit properties similar to verb-raising verbs (den Besten and Rutten, 1989), but with a crucial difference: they do not force their governed verbs to raise their non-verbal arguments. As a result, both of the examples below are grammatical:

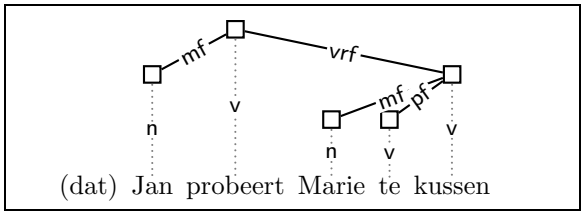
(26) (dat) Jan Marie probeert te kussen
(that) Jan Marie tries to kiss
"that Jan tries to kiss Marie"

(27) (dat) Jan probeert Marie te kussen

In (26), the object *Marie* of *kussen* climbs up into the Mittelfeld of the finite verb, just like in the earlier verb-raising example (3). In (27) however, *Marie* does not climb up into the Mittelfeld but stays directly to the left of *kussen*.

Here is a TDG analysis of (27):





(dat) Jan probeert Marie te kussen

We model partial extraposition as follows: (a) partial extraposition verbs ID-subcategorize for vinf and (b) verbs ID-categorizing for vinf do LP-subcategorize for mf. The latter is in contrast to words ID-categorizing for vbse and vinf as *kussen* in (5). Here are lexical entries for *probeert* and *kussen* (ID-categorizing for vinf):

(28)
$$\textit{probeert} \;\mapsto\; \begin{bmatrix} \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{subj}, \mathsf{vinf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{vrf}?\} \end{bmatrix}$$

(29) $$\textit{kussen} \;\mapsto\; \begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\mathsf{vinf}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{obj}, \mathsf{te}\} \\ \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{pf}\} \\ \mathsf{blocks} & : & \{\mathsf{te}\} \end{bmatrix}$$

Notice that verbs ID-categorizing for vinf can only land to the right of their governors: they LP-categorize only for vrf. Also, they block no role, which allows their arguments to climb up (26).

3.3 Full extraposition verbs

The third class of verbs taking verbal complements are full extraposition verbs. They ID-subcategorize for a fully saturated VP, i.e. no arguments of the governed verb may climb into the Mittelfeld of the finite verb, which explains the ungrammaticality of (31):

(30) (dat) Piet Jan dwingt Marie te kussen
(that) Piet Jan forces Marie to kiss
"that Piet forces Jan to kiss Marie"

(31)∗(dat) Piet Jan Marie dwingt te kussen

We model this phenomenon similar to how we modeled partial extraposition: (a) full extraposition verbs ID-subcategorize for $\overline{\mathsf{vinf}}$ and (b) verbs ID-categorizing for $\overline{\mathsf{vinf}}$ LP-subcategorize for mf. In addition, we require that verbs ID-categorizing for $\overline{\mathsf{vinf}}$ block all roles. Thus, no argument of the governed verb can climb up. This idea is reflected in the lexical entries for *dwingt* and *kussen* (ID-categorizing for $\overline{\mathsf{vinf}}$) below:

(32)
$$\textit{dwingt} \;\mapsto\; \begin{bmatrix} \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{subj}, \mathsf{obj}, \overline{\mathsf{vinf}}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{vrf}?\} \end{bmatrix}$$

(33) $$\textit{kussen} \;\mapsto\; \begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\overline{\mathsf{vinf}}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{obj}, \mathsf{te}\} \\ \mathsf{cat}_{\mathrm{LP}} & : & \{\mathsf{vrf}\} \\ \mathsf{subcat}_{\mathrm{LP}} & : & \{\mathsf{mf}*, \mathsf{pf}\} \\ \mathsf{blocks} & : & \mathcal{R} \end{bmatrix}$$

Notice in particular that (33) blocks all roles, contrary to (29).

3.4 The Two Sides of *proberen*

The word *proberen* is a special case: if governed by the perfect auxiliary *hebben*, it is ambiguous between being a partial extraposition verb or a verb-raiser. In the following two examples, *proberen* (in its past participle form *geprobeert*) acts as a partial extraposition verb:

(34) (dat) Jan Marie heeft geprobeert te kussen
(that) Jan Marie has tried$_{(part)}$ to kiss
"that Piet has tried to kiss Marie"

(35) (dat) Jan heeft geprobeert Marie te kussen

However in the two examples below, *proberen* acts as a verb-raising verb:

(36) (dat) Jan Marie heeft proberen te kussen

(37)∗(dat) Jan heeft proberen Marie te kussen

There are two indications that *proberen* is a verb-raising rather than a partial extraposition verb here: (a) it shows infinitival inflection although being the past participle complement of *heeft* (IPP-effect) and (b) sentence

(37), where *Marie* does not climb up into the Mittelfeld, is ungrammatical.

We capture *proberen*'s ambivalent behaviour by letting *geprobeerd* ID-subcategorize for vte and *proberen* for vte:

(38) *geprobeerd* $\mapsto$ $\begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\mathsf{vprt}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\mathsf{vte}\} \end{bmatrix}$

(39) *proberen* $\mapsto$ $\begin{bmatrix} \mathsf{cat}_{\mathrm{ID}} & : & \{\mathsf{vprt}\} \\ \mathsf{subcat}_{\mathrm{ID}} & : & \{\underline{\mathsf{vte}}\} \end{bmatrix}$

4 Order in the Mittelfeld

Little overt inflection remains in Dutch and word order tends to follow obliqueness, i.e. $\mathsf{subj} \prec \mathsf{iobj} \prec \mathsf{obj}$:

(40) dat $\underbrace{\text{Jan}}_{\mathsf{subj}}$ $\underbrace{\text{Marie}}_{\mathsf{iobj}}$ $\underbrace{\text{het boek}}_{\mathsf{obj}}$ probeert te geven
"that Piet tries to give the book to Marie"

It is tempting to postulate that the Mittelfeld can be partitioned into a sequence of subfields, each one dedicated to a specific degree of obliqueness. However, verb-raisers demonstrate that this simplistic explanation does not suffice:

(41) dat $\underbrace{\text{Piet}}_{\mathsf{subj}_1}$ $\underbrace{\text{Jan}}_{\mathsf{obj}_1}$ $\underbrace{\text{Marie}}_{\mathsf{iobj}_2}$ $\underbrace{\text{het boek}}_{\mathsf{obj}_2}$ ziet_1 geven_2
"that Piet sees Jan give the book to Marie"

In (41) the indirect object *Marie* of the embedded verb *geven* is raised to a position that follows the direct object *Jan* of the main verb *ziet*. However a pattern emerges: the arguments of the same verb are ordered among themselves according to obliqueness, but the raised arguments of *geven* follow the arguments of its governor *ziet*. The pattern continues to hold with additional levels of embedding:

(42) dat $\underbrace{\text{ze}}_{\mathsf{subj}_1}$ $\underbrace{\text{Piet}}_{\mathsf{obj}_1}$ $\underbrace{\text{Jan}}_{\mathsf{obj}_2}$ $\underbrace{\text{Marie}}_{\mathsf{iobj}_3}$ $\underbrace{\text{het boek}}_{\mathsf{obj}_3}$ laat_1 zien_2 geven_3
"that she lets Piet see Jan give the book to Marie"

and suggested the "cross-serial dependency principle", whereby the order of the raised arguments of embedded verbs follows the order of their governors in the verb cluster, thus giving rise to patterns of the form $ABCABC$. However, this principle fails in the presence of inversion:

(43) dat Jan_1 het boek_2 wil_1 lezen_2

(44) dat Jan_1 het boek_2 lezen_2 wil_1
"that Jan wants to read the book"

A common way to repair the principle is to state that the primary order of raised arguments is determined not by the linear order of their governors in the verb cluster, but by their nesting order in the syntactic structure. This is the essence of Hinrichs and Nakasawa's (Hinrichs and Nakazawa, 1989) argument composition technique. We are thus led to formulate for TDG a similar ordering principle:

Principle. Elements of the Mittelfeld are ordered first according to the relative depth of their respective governors in the dependency tree, and second according to the obliqueness of their grammatical function.

5 Related work

The HPSG accounts of Dutch word order proposed by Rentier (Rentier, 1994), Kathol (Kathol, 1996), and van Noord and Bouma (van Noord and Bouma, 1996; Bouma and van Noord, 1998) all make use of Hinrichs and Nakazawa's technique of *argument composition*.

Rentier does not address inversion nor order of arguments in the Mittelfeld. Kathol properly handles inversion. Van Noord and Bouma cover the greatest number of phenomena. They correctly account for all cases of inversion, full and partial extraposition, and address order of the non-verbal arguments in the

Mittelfeld. In contrast to Rentier and Kathol, Bouma and van Noord assume a flat syntactic analysis where all verbs and non-verbal arguments are directly dominated by the S-node. They order the arguments in the Mittelfeld based on (a) their degree of nesting in the verb cluster and (b) their degree of obliqueness.

6 Conclusion

We proposed a TDG analysis of word-order variations in the verb cluster of Dutch verb-final sentences. Our account addresses the phenomena of cross-serial dependencies, verb-raising, infinitivus pro participio, inversion, partial and full extraposition. Furthermore, in order to properly model the order of elements in the Mittelfeld, we formulated a simple ordering principle based first on the embedding depth of verbal governors in the dependency tree, and second on obliqueness.

A property of TDG is that valid analyses can be characterized as the solutions of a constraint satisfaction problem amenable to efficient processing through constraint propagation (Duchier, 2001). A grammar development environment including an efficient constraint-based is publicly available,[6] and a Dutch grammar written in this environment, covering all of the phenomena mentioned in this article, was used to prepare this article.

References

Gosse Bouma and Gertjan van Noord. 1998. Word order constraints on verb clusters in german and dutch. In *Complex Predicates in Nonderivational Syntax*. Academic Press.

Ralph Debusmann. 2001. A declarative grammar formalism for dependency grammar. Master's thesis, University of Saarland.

Hans den Besten and Jean Rutten. 1989. On Verb Raising, Extraposition, and Free Word Oorder in Dutch. In *Sentential Complementation and the Lexicon*. Dordrecht.

Denys Duchier and Ralph Debusmann. 2001. Topological dependency trees: A constraint-based account of linear precedence. In *39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse/FRA.

Denys Duchier. 2001. Lexicalized syntax and topology for non-projective dependency grammar. In *Eighth Meeting on Mathematics of Language*, Helsinki/FIN.

Kim Gerdes and Sylvain Kahane. 2001. Word order in german: A formal dependency grammar using a topological hierarchy. In *39th Annual Meeting of the Association for Computational Linguistics (ACL 2001)*, Toulouse/FRA.

Erhard Hinrichs and Tsuneko Nakazawa. 1989. Flipped out: Aux in German. In *Papers from the 25th Meeting of the Chicago Linguistic Society*, pages 193–202, Chicago/IL.

Andreas Kathol. 1996. Order variability in german and dutch verb clusters. In *Computational Linguistics in The Netherlands 1995*.

Gerrit M. Rentier. 1994. A lexicalist approach to dutch cross dependencies. In *Papers from the 30th Regional Meeting of the Chicago Linguistic Society*, pages 376–390.

[6] `http://www.ps.uni-sb.de/~duchier/mogul/info/duchier/coli/dg.html`

	$\mathsf{cat}_{\mathrm{ID}}$	$\mathsf{subcat}_{\mathrm{ID}}$	$\mathsf{cat}_{\mathrm{LP}}$	$\mathsf{hcat}_{\mathrm{LP}}$	$\mathsf{subcat}_{\mathrm{LP}}$	blocks
te	{te}	∅	{pf}	{v}	∅	∅
het	{det}	∅	{df}	{n}	∅	∅
boek	{subj, iobj, obj}	{det}	{mf}	{n}	{df?}	{det}
Piet	{subj, iobj, obj}	∅	{mf}	{n}	∅	∅
heeft	∅	{subj, vprt}	∅	{v}	{mf∗, vlf?, vrf?}	$\mathcal{R}$
moet	∅	{subj, vbse}	∅	{v}	{mf∗, vlf?, vrf?}	$\mathcal{R}$
ziet	∅	{subj, obj, $\underline{\text{vbse}}$}	∅	{v}	{mf∗, vrf?}	$\mathcal{R}$
dwingt	∅	{subj, obj, $\overline{\text{vinf}}$}	∅	{v}	{mf∗, vrf?}	$\mathcal{R}$
probeert	∅	{subj, vinf}	∅	{v}	{mf∗, vrf?}	$\mathcal{R}$
hebben	{vbse}	{vprt}	{vrf}	{v}	{vrf?}	∅
moeten	{vbse}	{vbse}	{vrf}	{v}	{vrf?}	*VRoles*
moeten	{vprt}	{vbse}	{vrf}	{v}	{vrf?}	*VRoles*
zien	{vbse}	{obj, vbse}	{vrf}	{v}	{vrf?}	*VRoles*
zien	{vprt}	{obj, vbse}	{vrf}	{v}	{vrf?}	*VRoles*
zien	{vbse}	{obj}	{vlf, vrf}	{v}	∅	∅
gezien	{vprt}	{obj}	{vlf, vrf}	{v}	∅	∅
proberen	{vprt}	{$\underline{\text{vinf}}$}	{vrf}	{v}	{vrf?}	∅
geprobeerd	{vprt}	{$\overline{\text{vinf}}$}	{vlf, vrf}	{v}	{mf∗, vrf?}	∅
kussen	{vbse}	{obj}	{vlf, vrf}	{v}	∅	∅
kussen	{$\underline{\text{vinf}}$}	{obj, te}	{vrf}	{v}	{pf}	{te}
kussen	{$\overline{\text{vinf}}$}	{obj, te}	{vrf}	{v}	{mf∗, pf}	$\mathcal{R}$
kussen	{vinf}	{obj, te}	{vrf}	{v}	{mf∗, pf}	{te}
gekust	{vprt}	{obj}	{vlf, vrf}	{v}	∅	∅

Table 1: The lexicon

Gertjan van Noord and Gosse Bouma. 1996. Dutch verb clustering without verb clusters. In *Specifying Syntactic Structures*. CSLI Publications.