A Dependency Account of Korean Word Order

Kim Gerdes
ERSS, Université Bordeaux 3

Domaine Universitaire
F-33607 Pessac cedex
kim.gerdes@u-bordeaux3.fr

Hi-Yon Yoo
ERSS, Université Bordeaux 3
and ARP, Université Paris 7
Domaine Universitaire
F-33607 Pessac cedex
hyoo@u-bordeaux3.fr

Abstract

The aim of this paper is to present an analysis of Korean word order based on the idea of topology within the Meaning-Text Theory (MTT). This linear and modular framework distinguishes several representation levels for a sentence; the linearization process taking place between the syntactic (dependency) structure and the topological (constituent) structure. The linearization rules take into account the communicative information, and the resulting topological structures are easily interfaced with phonological representations, including prosody. This approach allows accounting for Korean case marker variations, word order variations and support verb constructions in the MTT framework.

Introduction

Korean is known as a verb final language with relatively free word order for verbal arguments.[1] Word order can only be free in the sense that it is unrestricted under a purely syntactic point of view. In languages that are less restricted syntactically, it is the communicative structure that determines word order. We propose to take into account the possible prosodic structures for a sentence given its communicative structure (~pragmatic/information structure, Lambrecht 1994, Mel'čuk 2001). Any syntactic theory that does not directly include communicative information will suffer under two shortcomings:

1. **The syntactic rules will not be able to determine the appropriate word order among the different possibilities.** This approach inevitably brings along terms like “scrambling” and “non-configurational.”
2. **The syntactic structures are incomplete for spoken language generation.** One of the main goals of the generative grammars is to generate all possible sentence strings. This attributes to the sentence string a place in the language model, whereas the cognitive language generation process has to deal with more complex linguistic representations than word strings. While text parsing remains an interesting and essential part of language engineering, it should not intrude into our theoretical language model. Language is primarily spoken and our language model has to relate sounds and not strings with meanings. In particular, we need constituent trees that are easily interfaced with prosodic groups and words and we cannot compute these structures without communicative information.

The fact that in Korean syntactic information (e.g. verb final) and communicative information (argument placement) jointly determine the word order resembles what has been observed for German, and it is appealing to apply to Korean the classical descriptive method of German: Topology. The term “topology” refers essentially to the idea that the sentence comprises a template (domain) of fixed places, called fields, in which the elements of the sentence are placed during the linearization

---

[1] We will see than under certain conditions, arguments can be placed after the verb.

process. The idea goes back to 19th century description of German sentence structure and is present ever since in teaching of German just as in different linguistic analyses up to Kathol 1995's HPSG analysis (Herling 1821, Erdmann, 1886, Drach 1937, Tesnière 1959, Höhle 1986, Kathol 1995). Calling German a "V2 language" is after all nothing else than saying that the finite verb occupies a specific position behind a constituent that is placed in the beginning of the sentence not for syntactic but for communicative reasons. Korean exhibits similar phenomena: Basically, a fixed position of the verb and relatively free placement of its dependants. Moreover, German and Korean both (optionally) construct sentence final verbal complexes whose arguments "scramble" (Rambow & Lee 1994, Choi 1999), i.e. their placement is determined by communicative factors.

Our main goal is to present a simple linearization grammar for Korean, i.e. we are interested in linking syntactic functional structures (dependency trees) with linear constituent structures (topological trees). We thus separate syntactic functions and linear order into two different structures, contrarily to X-bar based approaches that collapse these two types of information into one structure.[2] We place our analysis in the Meaning-Text Theory where language is primarily seen as a language generation process (although all rules remain bidirectional) that parses through different independent stages: semantic representation (the meaning), syntactic representation (dependencies), topological representation (constituent trees), and phonological representation (phoneme string with prosodic groupings).[3]

We will focus on the dependency-topology interface but we will equally discuss the adjacent structures where they play a role for the input and the output of the linearization process.

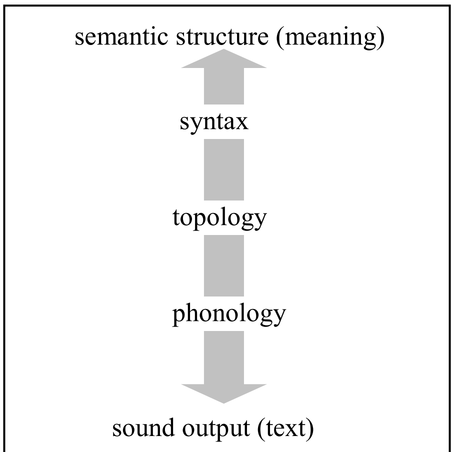

Fig. 1: The simplified MTT model

Sentences corresponding to a simple propositional structure

The following 15 sentences have the same propositional structure in common: The predicate "mokta" '*eat*' has two arguments, an agent "Yongi" and a patient "sakwa" '*apple*'. In the MTT framework, a semantic dependency graph represents this fact (Fig. 2)

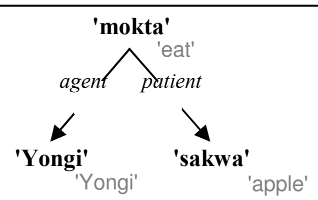

Fig. 2: A semantic (propositional) graph

(1) a. Yongi-nun sakwa-rul mok-ot-so.
Yongi-top apple-obj eat-past-decl
'*As for Yongi, she ate the apple*'.

b. Yongi-ka sakwa-rul mok-ot-so.
Yongi-subj apple-obj eat-past-decl
'*Yongi ate the apple*'.

---

[2] Labelling "the boat" as an NP constituent hides two types of information: 1. These words form a linear constituent where "the" precedes "boat" and 2. the name "NP" implies that the noun "boat" is the head and "the" its dependent.

[3] The term "text" is misleading for this last representation (although it is usual and name-giving in MTT).

c. Yongi-ka sakwa-nun mok-ot-so.
Yongi-subj apple-foc eat-past-decl
'*(I don't know about other fruits but) Yongi ate APPLES.*'

d. Yongi-nun sakwa-nun mok-ot-so.
Yongi-top apple-foc eat-past-decl
'*(I don't know about other fruits but) As for Yongi, she ate APPLES.*'

e. sakwa-rul Yongi-ka mok-ot-so.
apple-obj Yongi-subj eat-past-decl
'*Yongi ate the apple*'.

f. apple-nun Yongi-ka mok-ot-so.
apple-top Yongi-subj eat-past-decl
'*As for the apples, Yongi ate them.*'

g. sakwa-rul mok-ot-so Yongi-ka .
apple-obj eat-past-decl Yongi-subj
'S*he ate the apples, Yongi.*'

h. mok-ot-so Yongi-ka sakwa-rul .
eat-past-decl Yongi-subj apple-obj
'S*he ate them, Yongi, the apples.*'

i. *Yongi-nun mok-ot-so sakwa-nun .
Yongi-top eat-past-decl apple-top

j. sakwa-rul mok-ki-nun Yongi-ka he-so .
apple-obj eat-nominal-top Yongi-subj does
'Eating apples, *Yongi does it.*'

k. *sakwa-rul mok-ki-nun he-so Yongi-ka .
apple-obj eat-nominal-top does Yongi-subj

l. ?mok-ki-nun Yongi-ka sakwa-rul mok-ot-so .
eat-nominal-top Yongi-subj apple-obj eat-past-decl
'As for eating, *Yongi ate apples.*'

m. ???sakwa-rul mok-ki-nun Yongi-ka mok-ot-so
apple-obj eat-nominal-top Yongi-subj eat-past-decl
'???Eating apples, *Yongi ate them.*'

n. sakwa-ka Yongi-robuto mok-hyot-so .
apple-subj Yongi-instr(by) eat-pass/past-decl
'*Apples were eaten by Yongi.*'

o. Yongi sakwa mok-ot-so.
Yongi apple eat-past-decl
'*Yongi ate the apple*'.

Note that subjet and object can appear with different case markings and we encounter light verb constructions with *hada* as well as doubled verb constructions. In the following, we will attempt to link these variations with semantic and syntactic considerations. Note that the same propositional graph can be expressed by passive constructions like (1n) or without explicit case marking (1o); however we do not present their analysis here.

Let us first informally consider simple contexts given by the following set of questions. We indicate which of the above sentences can function as an answer to these questions:

Context	**Possible answers**
A. What's going on?	b,e
B. Who ate the apple?	b, e, f, j

C. What did Yongi eat?	a, b, c, d, e, g, $^{?}$l
D. What did Yongi do with the apple?	a, b, e, f, g, h
E. What did Yongi do?	a, b, c, d
F. How about the apple?	e
G. Who ate what?	$^{?}$l

The fact that sentences (1c) and (1d) do not differ in respect to their contexts although there is a perceptible difference (in the degree of the topicalization of "Yongi") shows that finer grained contexts should be considered to distinguish these sentences. As a first approximation, we follow Choi 1999's idea that we should (at least) consider a partition of communicative structure in four parts that we call *topic* (the prominent theme), *rheme* (the non prominent rhematic part of the sentence), *focus* (the contrastive or prominent rheme) and finally *tail* (non-prominent theme). We call *topic* and *tail* also "*thematic,*" and *focus* and *rheme* "*new.*"[4]

For example, the context C corresponds to a *new* (rhematic or focus) object, the rest of the sentence being thematic (topic or tail). Context D requires only the verb to be new while in context E, only the subject is thematic. We will have to explain why sentence (1a) can appear in these three contexts and not in the others; said in an MTT way, we want to explain how the speaker produces sentence (1a) when she/he wishes to express the semantic representation of Figure 3, i.e. the propositional content of Figure 2 in a communicative context that we represent by superposed areas carrying communicative values.

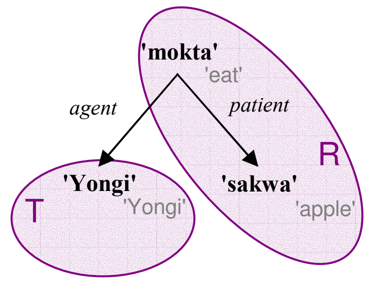

Fig. 3:
A semantic graph with communicative areas (context E)

Semantic – syntax

As a first step in the synthesis, a semantic representation (a propositional graph and communicative areas) is translated to a syntactic structure (dependency structure and communicative areas). We will only sketch this step for Korean. For a discussion of the formal and computational properties of these constrained graph grammars, see Bohnet & Wanner 2001. We suppose a complete partition of the semantic graph into communicative areas. The only restriction that we observed on the communicative structure is that there is only one *new* area, which covers a part of the dependency tree

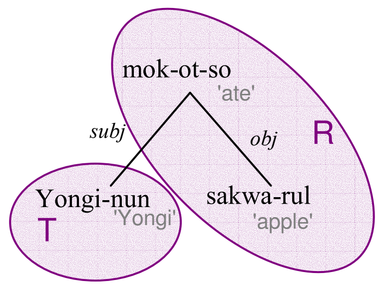

Fig. 4:
One of the syntactic dependency trees with communicative markup corresponging to Fig. 3 (In the other corresponding tree , *Yongi* carries the case marker –*ka*)

4 Mel'cuk 2001 has eight opositions but rigorous testing of oppositions other than *theme/rheme* and *prominent/non prominent* is much harder and goes beyond the scope of this present work. However, our formals system allows to take into account other communicative oppositions. Moreover, each of these four communicative values corresponds to a distinguished prosodic pattern (see section *Topology – Phonology*).

that is not interrupted by another communicative area.[5] In the general case, each node of the semantic graph (each semanteme) is translated to a node of the dependency tree that carries the same communicative value. The communicative information determines some of the pragmatic case markings (Park 1995): In the case of active diathesis, an agent can be translated with the default case marker “ka” in all communicative settings, and with “nun” if the agent is either topic or focus.[6] The patient has the default active case marking “rul”, and “nun” only in a focus situation.

Sentences (1g) and (1l) have a different dependency structure. If the semantic predicate is topic, we cannot simply pass on the communicative value to the verb in the dependency tree, but we have to nominalize it with the affixe “ki” and the new semantically empty verb “hada” appears:

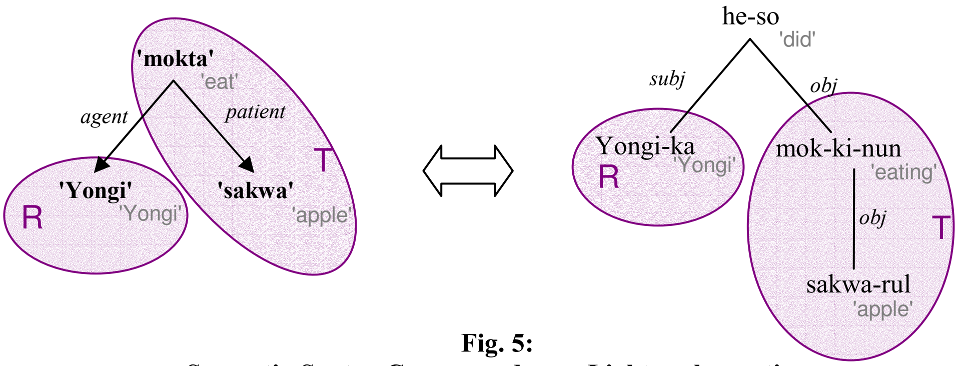

Fig. 5:
Semantic-Syntax Correspondence: Light verb creation
Semantic graph **Syntactic dependency tree**

As we will see later, this dependency tree does not have a corresponding linearization if not all dependants of the nominalized verb are in the same communicative area.

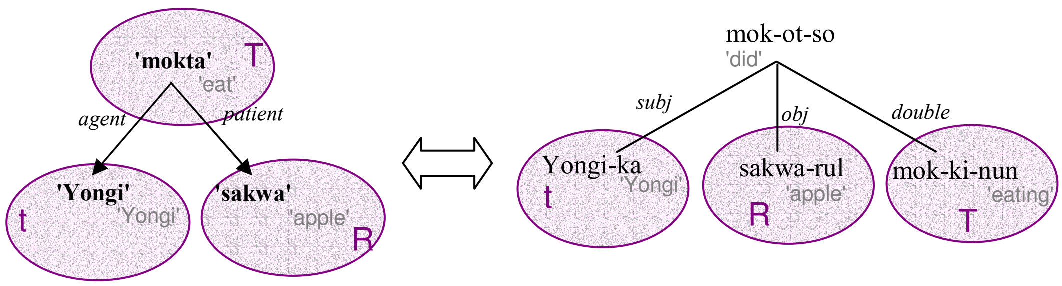

Fig. 6:
Semantic-Syntax Correspondence: Verb duplication
Semantic graph **Syntactic dependency tree**

The second possibility when the verb is topicalized, is to double it instead of using the light verb “hada”. Our linearization grammar does not produce an output when not only the verb is doubled, but also one of its arguments. This allows sentences like (1l) and avoids sentences like (2)

(2) *	sakwa-rul	mok-ki-nun	Yongi-ka	sakwa-rul	mok-ot-so.
	apple-obj	eat-nominal-top	Yongi-subj	apple-obj	eat-past-decl

[5] Our linearization grammar does not produce a result in other cases. Contexts like G (‘Who ate what’) trigger syntactic constructions that allow the *new* area to be connected through a communicatively non marked node.

[6] The fact that “nun” can express topic and focus, the two prominent communicative values, may be due to “nun” expressing only the prominence (the contrastive exclusion of other possibilities, see Han 1997).

Dependency – topology

A topological linearization grammar has the following parameters:
A *domain* is a linear template of fields, each *field* carrying information on how many elements it can or has to contain. Domains are assigned a numeral value indicating the permeability of its borders (controlling what is sometimes called "long distance dependencies").
Domains are created by words in accordance to *box creation rules* that are of the following type:
`(origField, cat, CV, domain, finalField)`
An element in a field of name *origField* of category *cat* and communicative value *CV* can open a domain of type *domain* and it will occupy the field *finalField*. The created domain keeps the communicative value of the element that created it.
Elements are positioned into fields following *placement rules*:
`(catGov, CVGov, fieldGov, rel, cat, CV, field, maxperm)`
The element of category *cat* and communicative value CV can go into a field of name *field* if its governor is in field *fieldGov* and it has the category *catGov* and the communicative value *CVGov*, the syntactic function of the element is *rel*, and the *field* is not separated by a domain border greater than *maxperm*.

The linearization starts with an empty initial field *i* into which the root of the dependency tree is placed. Then the first box creation rule can apply. When all applicable box creation rules applied, the dependents of the root find their positions corresponding to the placements rules and so on.

The Korean topological grammar for verbs and their arguments:[7]
<u>The domains</u> (The symbols *,?,! follow the conventions of regular expressions):

The main domain:	`md = T* R? V? t*`	`(permeability = 2)`
The verbal domain:	`vd = A* V!`	`(permeability = 2)`
The verbal complex:	`vc = Vdep? Vgov!`	`(permeability = 1)`

<u>The box creation rules:</u>
The verb at the root of the dependency tree is either rhematic or tail, or it can be a communicatively neutral *hada* (for the topic case see precedent section).
For any communicative value (indicated by the question mark), the verb can open a main domain and go into the *V*-field:
`(i, verb, ?, md, V)`
In case that the verb is *tail*, it has a second option : opening the main domain and going into the field *t*:
`(i, verb, ?, md, t)`
A verb in the V-field opens a verbal complex:
`(V, verb, ?, vc, Vgov)`
A verb in any other field (T, R or t) opens a verbal domain:
`((T,R,t), verb, ?, vd, V)`
<u>The placement rules:</u>
Nominal arguments:
If the verb did not open a verbal domain: If a nominal argument has CV T it goes into the field T, if it has CV *F* or *R* it goes into the field *R*, and if it has the CV *t*, it goes into the field *t*.
```
(verb, ?, Vgov, ?, noun, T, T, 1)
(verb, ?, Vgov, ?, noun, (F,R), R, 1)
(verb, ?, Vgov, ?, noun, t, t, 1)
```

[7] In this paper, we treat the noun phrase as one element. Its topological analysis follows the same type of rules, see for example Yoo 2003b.

If the verb is inside of a verbal domain: Any nominal argument has to have the same CV as its governor to go into the field *A*. Otherwise, the linearization will not succeed.

```
(verb, T, Vgov, ?, noun, T, A, 1)
(verb, F, Vgov, ?, noun, F, A, 1)
(verb, R, Vgov, ?, noun, R, A, 1)
(verb, t, Vgov, ?, noun, t, A, 1)
```

Verbal arguments:

If the dependant and the governing verb share the same communicative value, they create a verbal complex:

```
(verb, T, Vgov, ?, verb, T, Vdep, 0)
(verb, F, Vgov, ?, verb, F, Vdep, 0)
(verb, R, Vgov, ?, verb, R, Vdep, 0)
(verb, t, Vgov, ?, verb, t, Vdep, 0)
```

If the CV of the dependant and the governing verb differ, the dependant verb has to emancipate itself to the field corresponding to its CV:

```
(verb, R, Vgov, ?, verb, t, t, 0)
(verb, F, Vgov, ?, verb, t, t, 0)
(verb, t, Vgov, ?, verb, F, V, 0)
(verb, t, Vgov, ?, verb, F, R, 0)
(verb, t, Vgov, ?, verb, T, T, 0)
```

A (nominalized) verbal topic governor does not allow its verbal dependents to emancipate, thus, if they have a different CV, the linearization will fail. For the case of a communicatively neutral verb (CV *N*), their dependent is topic and goes into the *T* field:

```
(verb, N, Vgov, ?, verb, T, T, 0)
```

This simple grammar relates the syntactic dependency structure with all possible linearizations, i.e. all possible topological constituent trees. As an example, it relates the syntactic dependency tree of Fig. 4 with the following (unique) topological constituent tree.

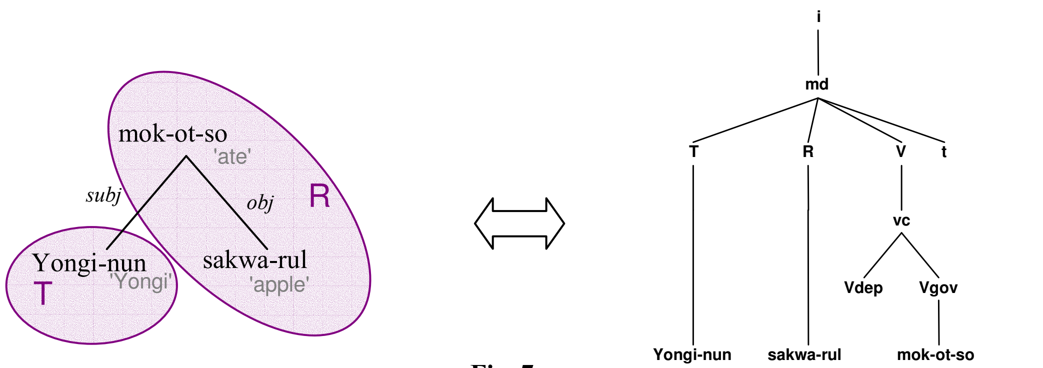

Fig. 7:
Syntax-Topology Correspondence: a simple construction
Syntactic dependency tree **Topological constituent tree**
(simplified tree without the internal analysis of the noun phrase)

Verbal dependents and verb clusters

The above topological grammar allows the generation (and analysis) of complex verbal predicates. As an example, consider the following semantics-syntax-topology correspondence:

A control verb construction with “yaksok-het-so” ‘promise’ can form a verb cluster with its verbal dependent “moketa-go” ‘to eat’ if and only if they share the same communicative value. Figure

8 illustrates this case. The rhematic part covers the *R* and the *V* field, the other elements are placed according to their communicative value.

When the verbal dependent “moketa-go” ‘to eat’ has a different communicative value as in the example of Figure 9, it is placed into the field according to its CV and opens there a verbal domain (and its own verb cluster for eventual verbal dependents). The linearization of this case results in two linear orders, because the precedence between the two nominal rhematic arguments is not specified.

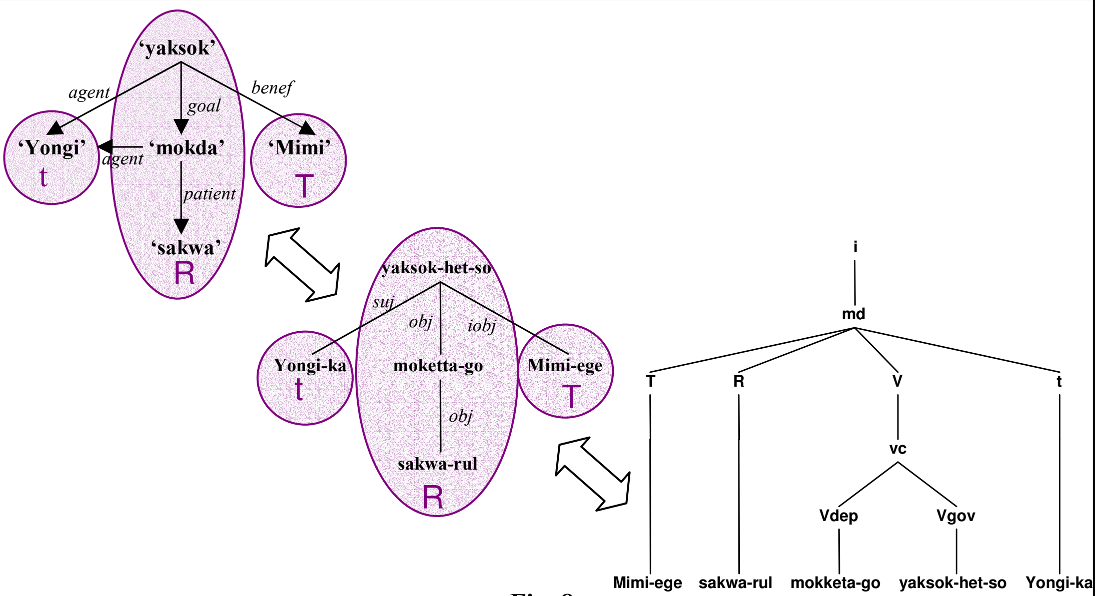

Fig. 8:
Semantics-Syntax-Topology Correspondence:
Governing and dependent verb form a verb cluster

Semantic graph **Syntactic dependency tree** **Topological constituent trees**

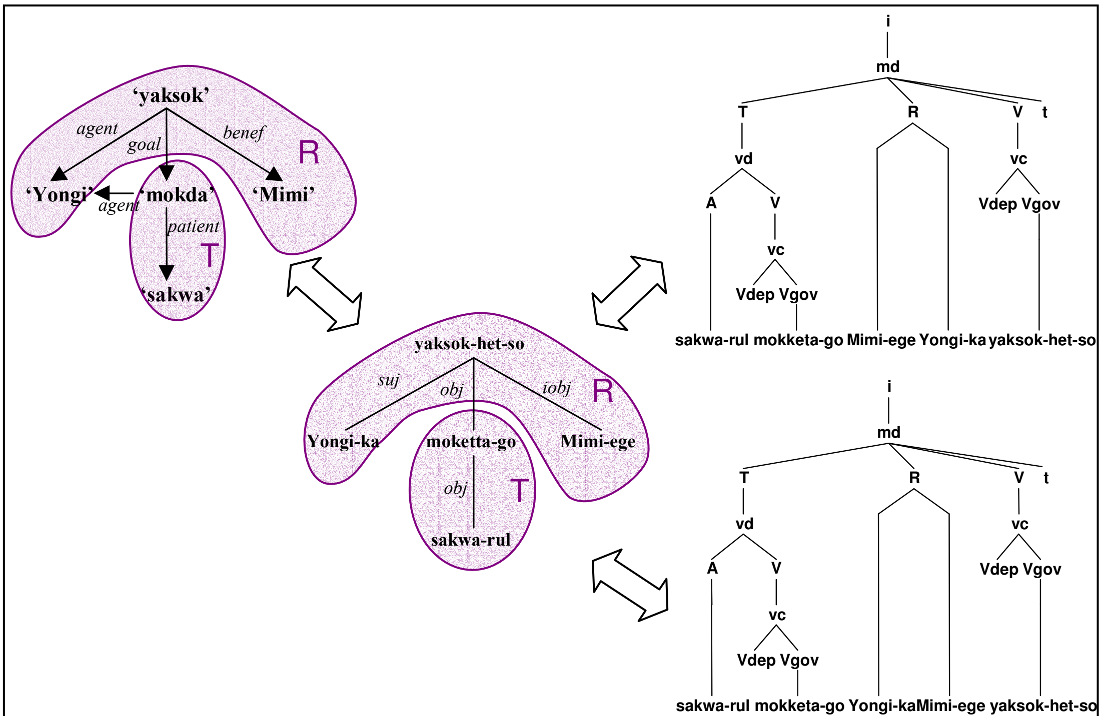

Fig. 9:
Semantics-Syntax-Topology Correspondence:
a dependent verb creates its own verbal domain, the order between Mimi and Yongi is not specified

Semantic graph **Syntactic dependency tree** **Topological constituent trees**

Topology – Phonology

The synthesis is not finished with the linearization, because we are interested in spoken language. From the topological constituent tree, we need to compute the correct phoneme string and the prosodic groupings. The goal of the topology is precisely to simplify this next step of language generation, and the rules we have to impose to compute the right prosodic groupings are in fact quite simple. We do not treat here the computation of the phoneme string. The phonological level is represented by a phonological tree, a hierarchical representation with three levels: Prosodic words form prosodic groups and prosodic groups form the prosodic utterance. The computation of this structure from the topological tree consists in flattening the topological tree. We thus have to give rules indicating in which case two words join to form a prosodic word, and which domain are flattened into a unique prosodic group. To the prosodic groups we associate the prosodic representation, which is computed from the communicative value carried by each domain and from the position of the prosodic word within the prosodic group.

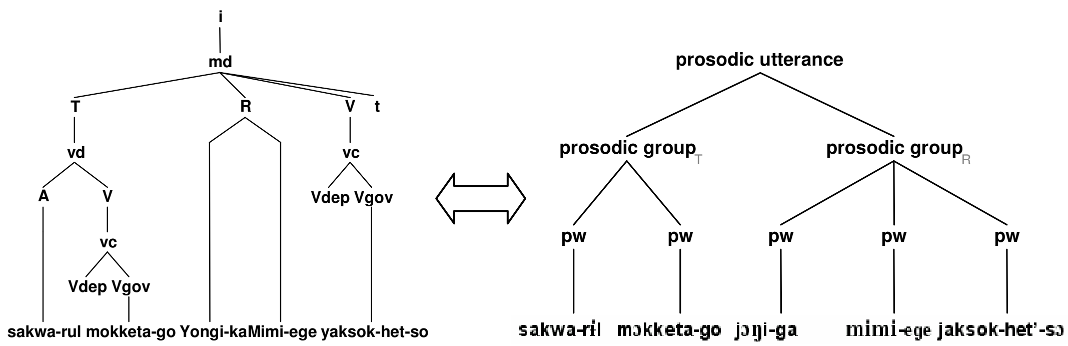

Fig. 10:
Topology-phonology Correspondence:
Topological constituent trees **Phonological tree (pw=prosodic word)**
Again, the internal structure of the nominal domains is not shown

The correspondence topology-phonology takes place in two steps:

- Prosodic word rules: The leaves of the topological tree can combine to form a prosodic word depending on 1. the domain and field it is in, 2. lexical information on the accentuability of the leaf. In case the leaf cannot form a prosodic word on its own, the rule indicates if it combines with an element to its left or to its right. (We do not treat these rules here, as we do not want to discuss the syntactic role of case markers and support verb constructions with "hada" 'do'. In our example, we present these structures as one word.)
- Prosodic group rules are of type `(D1, C1, F, D2, C2, PG)`: The domains *D1* and *D2* are contracted to form a prosodic group *PG* if *D1* dominates *D2* via the field *F*, and *D1* has the communicative value *C1* and *D2* has the communicative value *C2*. All (lowest) non-contracted domains form an independent prosodic group.

Note that these rules do not permit to create new prosodic groups where we did not have a domain in the topological tree. This means that we suppose that two prosodic words that are sisters in a domain, will always be in the same prosodic group.
See Yoo 2003a for a more detailed discussion of this correspondence.

Implementation

The syntax-topology and the topology-phonology formalism are implemented in the direction of generation in a program called *Deplin* (programmed in Java, with an interface that allows to directly edit trees and grammars; all files are in XML to permit easy interfacing with other applications.) This program allows verifying the interaction of our rules with more complicated examples. For Korean, a simple grammar has been created which gives satisfactory results. Work is in progress to widen the coverage of the grammar. Equally, a new version of Deplin will be able to use the same rules in sentence analysis.

Conclusion

An analysis of free word order phenomena in X-bar based structures requires recourse to particularly unwieldy transformations, which makes it preferable to separate argument structure from linear order, as it is done in dependency grammars. The linearization of the (unordered) dependency tree demands access to the underlying communicative structures, which has lead research in the dependency framework early on to a formalization of communicative notions. However, just as work in the transformational approaches focused on English, most work in Dependency Grammars pondered on Slavic languages and did not encounter the difficulties put up by languages like German or Korean, where communicative and syntactic information interact on equal bases in the determination of word order (which can lead to what is called "non-projective" structures).

In this paper, we have shown that it is possible to formalize Korean word order in a dependency approach. The classical notion of topology in combination with a simple communicative structure proves to be useful for the treatment of "long distance dependencies." However, subsequent work will have to show whether the formalism is sufficient also for a wide coverage grammar. In particular, we will have to handle efficiently large dictionaries, which has not yet been a goal for our testing implementation.

The same formalism has been used for grammars of German, French and Modern Greek (Gerdes & Kahane 2001, Yoo 2003b). The interaction of syntactic and communicative constraints in the same formalism allows us to establish a typology of these languages, because it makes it possible to distinguish and quantify the part played by syntax and by the communicative structure in the description of word order.

Some of the references

Bohnet B. and Wanner L. 2001. "On Using a Parallel Graph Rewriting Grammar Formalism in Generation." *Proceedings of the 8th European Natural Language Generation Workshop at the Annual Meeting of the Association for Computational Linguistics*, Toulouse.

Choi, H-W. 1999. *Optimizing Structure in Context – Scrambling and Information Structure*, CSLI, Stanford.

Drach, E. 1937. *Grundgedanken der deutschen Satzlehre*, Diesterweg, Frankfurt.

Gerdes, K. & S. Kahane. 2001. "Word order in German: A formal dependency grammar using a topological hierarchy", *Proceedings of ACL 2001*, Toulouse.

Han, C.-H. 1997. "Asymmetry in the interpretation of -(n)un in Korean" *Japanese/Korean Linguistics 7*. CSLI, Stanford, 1-15.

Kathol, A., 1995. *Linearization-based German Syntax*, Phd thesis, Ohio State University.

Lambrecht, K. 1994. Information Structure and Sentence Form, Cambridge Studies in Linguistics 71, Cambridge University Press.

Mel'čuk, I. 1988. *Dependency Syntax*. New-York, State University of New York Press.

Mel'čuk, I. 2001. *Communicative Organization in Natural Language (The Semantic-Communicative Structure of Sentences)*. Amsterdam, Benjamins.

Park K-S. 1995 *The Semantics and Pragmatics of Case Marking in Korean: A Role and Reference Grammar Account*, Phd thesis, State University of New York at Buffalo.

Rambow, O. and Lee, Y.-S. 1994. "Word order variation and Tree Adjoining Grammar". *Computational Intelligence*, 10(4):386-400.

Tesnière, L. 1959. *Eléments de syntaxe structurale*, Kliencksieck, Paris.

Yoo H-Y. 2003a "De la topologie à la chaîne parlée : les niveaux de représentation pour une Théorie Sens-Parole." *Proceedings of the first international conference on Meaning-Text Theory*, Paris

Yoo H-Y. 2003b. *Prosodie et ordres des mots. Description et formalisation pour le grec moderne*, Phd thesis University of Paris 7.