Journal of Language Modelling

VOLUME 4 ISSUE 2
DECEMBER 2016

Institute of Computer Science
Polish Academy of Sciences
Warsaw

Journal of Language Modelling

VOLUME 4 ISSUE 2
DECEMBER 2016

Tools and resources

Articles

JOURNAL OF
LANGUAGE MODELLING

ISSN 2299-8470 (electronic version)
ISSN 2299-856X (printed version)
http://jlm.ipipan.waw.pl/

Published by IPI PAN
Institute of Computer Science, Polish Academy of Sciences
ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

Circulation: 100 + print on demand

Layout designed by Adam Twardoch.
Typeset in XeLaTeX using the typefaces: *Playfair Display*
by Claus Eggers Sørensen, *Charis SIL* by SIL International,
JLM monogram by Łukasz Dziedzic.

EDITORIAL BOARD

Design and analysis of a lean interface for Sanskrit corpus annotation

Pawan Goyal[1] *and Gérard Huet*[2]
[1] IIT Kharagpur, India
[2] INRIA Paris Laboratory, France

ABSTRACT

We describe an innovative computer interface designed to assist annotators in the efficient selection of segmentation solutions for proper tagging of Sanskrit corpora. The proposed solution uses a compact representation of the shared forest of all segmentations. The main idea is to represent the union of all segmentations, abstracting from the sandhi rules used, and aligning with the input sentence. We show that this representation provides an exponential saving, in both space and time.

The segmentation methodology is lexicon-directed. When the lexicon does not have full coverage of the corpus vocabulary, some chunks of the input may fail to be recognized. We designed a lexicon-acquisition facility, which remedies this incompleteness and makes the interface more robust.

This interface has been implemented, and is currently being applied to the annotation of the Sanskrit Library corpus. Evaluation over 1,500 sentences from the *Pañcatantra* text shows the effectiveness of the proposed interface on real corpus data.

Keywords: Sanskrit, text segmentation, annotation, interface

1 GENERALITIES ON SANSKRIT LINGUISTICS

Sanskrit is the primary language used as a vehicle of culture in India. Literature in Sanskrit for all fields of human endeavour has been produced continuously over the past four millennia, giving rise to an immense corpus, which is, to date, only partially digitised. It benefits

from a very sophisticated linguistic tradition stemming from the fairly complete grammar composed by Pāṇini by the fourth century B.C.E.

During the last 15 years, significant efforts have been made to develop computational linguistics for Sanskrit, and considerable progress has been achieved in providing computer assistance for Sanskrit corpus processing (Goyal *et al.* 2009; Hellwig 2009; Huet *et al.* 2009; Kulkarni and Huet 2009; Kulkarni and Shukl 2009; Scharf and Hyman 2009; Jha 2010; Kulkarni *et al.* 2010; Kumar *et al.* 2010; Goyal *et al.* 2012). Nevertheless, there does not exist at this time a complete analyser for Classical Sanskrit texts able to compute morphological tagging reliably in a completely automatic way. The main difficulty concerns segmentation, since Sanskrit is represented in writing by continuous phonetic enunciation, which demands complex processing for its analysis in separate word forms. Although complete algorithms for this segmentation preprocessing have been proposed (Huet 2005), human assistance is still needed to focus on the appropriate solution within all possible analyses.

We propose in this paper a new human-machine interface to help a professional annotator to decide quickly between all possible segmentations, in order to select a unique morphological analysis among the many possible ones. Indeed, there exist thousands of such segmentations for simple sentences, and literally billions for complex ones. Once a sufficient amount of tagged corpus data has been made available using such semi-automated annotation tools, it is hoped that it will be possible to use it to train a fully automated parser using statistical methods.

A preliminary version of this paper was presented at the ICON conference in Hyderabad in December 2013 (Huet and Goyal 2013). The novel aspects and contributions of this paper with respect to the previous version are: a) it is an extended version of the conference paper, with a detailed explanation of the segmentation methodology, related work and illustrative examples, b) we conduct a thorough evaluation of the proposed system with respect to robustness as well as convergence time taken, in practice, on real corpus data with 1,500 sentences, and c) we propose a module for error recovery and lexical acquisition, which makes the system much more usable when dealing with a corpus that is error-prone, or contains words that are not present in the lexicon used by the system.

The paper is organized as follows. Section 2 discusses related work on the problem of word segmentation. Section 3 gives the necessary formalisms required for segmentation analysis of Sanskrit text. The concept of aligned segmentations and the graphical display of the interface, which are the central themes of this paper, are detailed in Sections 4, 5 and 6. The use of the proposed interface as a tagging tool is discussed in Section 7, using a complete walk-through example. We present the evaluation of the proposed interface using 1,500 sentences from real corpus data in Section 8. An experimental segmenter for lexical acquisition is described in Section 9. Section 10 concludes the paper.

2 RELATED WORK

The task of word segmentation is a necessary initial step for processing those natural languages where word boundaries are not maintained in the written text. A lot of prior work concerns word segmentation for Chinese text. The two dominant models for Chinese word segmentation are 'word-based' and 'character-based' (Sun 2010). Word-based methods read the input sentences from left to right, predicting whether the current piece of continuous characters is a word token. Once a word is found, they move on and search for the next word. The methods vary in terms of the strategies used for word prediction and disambiguation, if there are multiple possibilities. For instance, the maximum matching approach (Chen and Liu 1992) chooses the longest word for disambiguation, while prediction is based on a dictionary. Recently, machine-learning methods have been employed to solve these problems. Zhang and Clark (2007) used a linear model with an average perceptron algorithm where, given an input sequence of characters c, the model finds a segmentation $\hat{w}$ such that

$$\hat{w} = \max_{w \in GEN(c)} (\alpha \cdot \phi(c, w)),$$

where ϕ is a feature map, α is the parameter vector, which is learnt via training, and $GEN(c)$ enumerates the set of segmentation candidates for the character sequence c.

Character-based approaches, on the other hand, attempt to assign labels to the characters in the sequence, indicating whether a character c_i is a single character word (S), or the beginning (B), middle (I)

or end (*E*) of a multi-character word, thus treating this as a sequence labelling problem. Word tokens are inferred, based on the character classes. Several models, such as Conditional Random Fields (CRFs), have been used for this task (Tseng 2005).

Various approaches have since been proposed to combine word-based and character-based methods (Sun *et al.* 2009). For instance, Wang *et al.* (2014) recently proposed a method based on dual decomposition (Rush *et al.* 2010) to combine these approaches in an efficient framework.

The problem of word segmentation in Sanskrit, however, is more difficult than in languages such as Chinese, where the words are combined without any euphonic assimilation at the boundary. The next section describes in further detail the problem of word segmentation for Sanskrit text.

3 SEGMENTATION ANALYSIS FOR SANSKRIT TEXT

We shall now formalize the word segmentation problem in Sanskrit written text at various levels of abstraction. Sanskrit may be written in all Indian scripts, most usually in the *devanāgarī* script used by languages of North India such as Hindi, but such syllabic representation is awkward for morpho-phonetic computations. It is preferable to translate it into a list of phonemes, with one-to-one translation. We assume the standard set of 50 phonemes, already known from the time of Pāṇini. Such low-level representation issues are discussed at length in Scharf and Hyman (2009) and Huet (2009).

In Sanskrit, where oral tradition dominated the sphere of learning and an advanced discipline of phonetics explicitly described euphonic assimilation, the phonetic transformations at the juncture of successive words, well known by the term *sandhi*, are represented in writing. Assimilation obscures word boundaries in speech, and these word boundaries are correspondingly eliminated in writing as well. For example, *vasati* ‘dwells’ *atra* ‘here’ becomes *vasatyatra* in continuous speech. Some euphonic changes, like this one, can be separated in alphabetic Roman transcription despite the sound alteration, viz. *vasaty atra*. Other assimilation changes, however, preclude word separation, even in alphabetic transcription, because the final sound of the preceding word and the initial sound of the following word merge

into a single sound. Thus *vidyā* 'knowledge' *āpyate* 'is attained' becomes *vidyāpyate*; the single sound *ā* belongs to both words. This phenomenon has been well recognized and formally analysed since antiquity (Pāṇini gave a complete axiomatisation of sandhi in terms of string rewriting in his 4th century B.C. treatise Aṣṭādhyāyī). The most difficult task in parsing a Sanskrit sentence is determining the word boundaries. Solutions to this problem have valuable ramifications for speech analysis, where a similar problem is encountered in virtually all languages.

We assume that the reader is familiar with the use of finite-state methods for morpho-phonemic computations, as explained in standard references such as Roche and Schabes (1997), Kaplan and Kay (1994) and Beesley and Karttunen (2003). We also assume some familiarity with the lexicon-driven Sanskrit segmenter of Huet (2005), from which we extract the following definitions.

Definitions. A *lexical juncture system* on a finite alphabet Σ is composed of a finite set of words $L \subseteq \Sigma^*$ and a finite set R of rewrite rules of the form $[x]u|v \rightarrow w$, with $x, v, w \in \Sigma^*$ and $u \in \Sigma^+$. Note that in the Kaplan and Kay notation, the rule we write $[x]u|v \rightarrow w$ would be written as $u|v \rightarrow w/x\_$.[1]

The word $s \in \Sigma^*$ is said to be a *solution* to the system (L,R) iff there exists a sequence $\langle z_1, \sigma_1\rangle; \ldots; \langle z_p, \sigma_p\rangle$ with $z_j \in L$ and $\sigma_j = [x_j]u_j|v_j \rightarrow w_j \in R$ for $(1 \leq j \leq p)$, $v_p = \epsilon$ and $v_j = \epsilon$ for $j < p$ only if $\sigma_j = o$, subject to the matching conditions: $z_j = v_{j-1}y_jx_ju_j$ for some $y_j \in \Sigma^*$ for all $(1 \leq j \leq p)$, where by convention $v_0 = \epsilon$, and finally $s = s_1 \ldots s_p$ with $s_j = y_jx_jw_j$ for $(1 \leq j \leq p)$. ϵ denotes the empty word. We also say that such a sequence is an *analysis* of the solution word s.

In this formalization, Σ is the set of phonemes, R is the set of sandhi rules, and L is the vocabulary as a set of lexical items. As a first approximation, one may think of L as the lexicon of inflected words. In Section 7, we shall partition L according to lexical sorts, some of which are morphemes such as stems and affixes, in order to segment compound words by the same method as explained here for sentence

[1] This algorithm assumes that the segmenter induces the segment boundaries from a generative lexicon of permitted inflected forms. Another method would guess arbitrary segment boundaries with rules $uv \rightarrow w/x\_$, and attempt morphological analysis of the segments, but this is less efficient. Some rules could also use the v part as right context, when it is unchanged. Many variations exist.

segmentation into words. This extension is necessary to keep L finite, in view of the fact that nominal compounding in Sanskrit is productive to an arbitrary depth. But all the notions defined here will apply easily to this refinement, which allows us to keep notations simple. We shall also assume the system (L, R) to be *non-overlapping*, as defined in Huet (2005). This assumption is met in classical Sanskrit, except for a small number of uni-phonemic morphemes, which are amenable to the general treatment, modulo the introduction of so-called *phantom phonemes*, as explained in Huet (2006); Goyal and Huet (2013).

Note that the sandhi problem is expressed in a symmetric way. Going from $z_1|z_2|\ldots|z_n| \in (L \cdot |)^*$ to $s \in \Sigma^*$ generates a correct phonemic sentence s with word forms $z_1, z_2, \ldots, z_n$, using the sandhi transformations. Whereas going the other way means analysing the sentence s as a possible phonemic stream, using words from the lexicon transformed by sandhi. It is this second problem that the Sanskrit segmenter has to solve, since sandhi, while mostly deterministic in generation, is strongly ambiguous in analysis. Below, we provide a brief summary of the solution proposed by Huet (2005). The basic data structures used by the system are *Tries*, *Decos*, and variations on applicative data structures to represent finite automata and transducers. This methodology has recently been extended to a general paradigm of relational programming, using the notion of effective Eilenberg machines (Huet and Razet 2015).

Definitions. *Tries* are tree structures that store finite sets of strings sharing initial prefixes. We assume that the alphabet of string representations is some initial segment of positive integers. Thus a string is encoded as a list of integers that will from now on be called a *word*.

Definitions. A word may be associated with a non-empty list of information of polymorphic type α, absence of information being encoded by the empty list. We shall call such associations a *decorated trie*, or *deco* for short.

To solve the sandhi problem for analysis, the inflected form tries are decorated with the rewrite rules. The algorithm proceeds in one bottom-up sweep over each inflected form trie. For every accepting node (i.e. lexicon word), at occurrence z, we collect all sandhi rules[2]

[2] The treatment of a contextual rule $[x]u|v \to w$ is similar: we check that $z = \lambda xu$, but the decorated state is now at occurrence λx. In both kinds of rules,

$\sigma : u|v \rightarrow w$ such that u is a terminal substring of z: $z = \lambda u$ for some λ. When we move up the trie, recursively building the automaton graph, we decorate the node at occurrence λ with a choice point labelled with the sandhi rule. This builds in the automaton the prediction structure for rule σ, at distance u above a matching lexicon word. At interpretation time, when we enter the state corresponding to λ, we shall consider this rule as a possible non-deterministic choice, provided the input tape contains w as an initial substring. If this is the case, we shall then move to the state of the automaton at occurrence v (before this, the program checks that all sandhi rules are plausible in the sense that occurrence v exists in the inflected trie, i.e. there are some words that start with string v). When we take this action, the automaton acts as a transducer, by writing on its output tape the pair (z, σ).

Coming back to the solution word, we may think of s as a phonetically correct utterance over vocabulary L, and its analysis $S = \langle z_1, \sigma_1 \rangle; \ldots; \langle z_p, \sigma_p \rangle$ as one of its possible segmentations. Analysis S is completely explicit, in the sense that s may be computed from S, applying sandhi rules σ_i in sequence, going from left to right. Conversely, there may be many possible segmentations S of a given utterance s, typically thousands for a moderately long sentence, although it is proven in Huet (2005) that they are always finite in number. We write $Segs(s)$ for the set of segmentations of s. The algorithm described in Huet (2005) shows how to enumerate the complete set $Segs(s)$ from a given input string s. In view of its possibly enormous size, attempts have been made, e.g. Huet (2007), to filter out non-sensible segmentations by a semantic analysis in the manner of dependency grammars. This method works well for simple sentences, but is not sufficient for more complex sentences, particularly in the presence of ellipses and other anaphoric or discourse operators where dependencies are context-sensitive. Furthermore, the set $Segs(s)$ is not easily amenable to sharing, and as a consequence the segmentation-cum-tagging Web service of the Sanskrit Heritage site[3] has not been of practical use so

the choice point is put at the ancestor of z at distance u. This suggests as implementation to compute at the accepting node z a stack of choice points arranged by the lengths of their left component u. Furthermore, once the matching is done, the context x may be dropped when stacking a contextual rule, since it is no longer needed.

[3] `http://sanskrit.inria.fr/`

far on real corpus data, since it tended to generate very long Web pages, even to the point of choking the server. Wading through such long lists of segmentations was very tedious and error-prone. The new interface described in the present paper completely solves this problem. We shall now explain its main concepts.

4 ALIGNED SEGMENTATIONS

The key idea behind the new interface is to represent an abstraction of the union of all segmentation decompositions, realigned on the input utterance. This new representation is now amenable to sharing, and may thus be represented very compactly on one computer screen.

Definition. We consider a sandhi analysis S as above, generalized to allow empty sequences. It may be defined inductively, as being either empty or of the form $S = \langle z_1, \sigma_1 \rangle ; S'$, with S' a similar sequence. Let n be a natural number. We define the *alignment* of S *with offset* n, noted as $S \hookrightarrow n$, as a set of pairs of *aligned segments* of the form (k, z), with $k \in \mathbb{N}$ and $z \in L$, as follows. If S is the empty sequence, then $S \hookrightarrow n = \emptyset$. Otherwise, let $S = \langle z, \sigma \rangle ; S'$ with $\sigma = [x]u|v \rightarrow w$. We define $S \hookrightarrow n = \{(n, z)\} \cup S' \hookrightarrow n'$, where $n' = n + |z| + |w| - (|u| + |v|)$.

If S is a segmentation analysis of utterance s, we define its corresponding *aligned segment collection* as the set of aligned segments $\overline{S} = S \hookrightarrow 0$. Note that in this new notion we leave aside the precise sandhi rules used in the analysis S, keeping only the tabulation information that allows us to present its set of segments aligned with the original input s.

Let $\mathscr{S}$ be a set of segmentation analyses of utterance s. We define the *tabulated display* of $\mathscr{S}$, noted $D(\mathscr{S})$, as the set of aligned segments obtained as the union of all its corresponding aligned segment collections:

$$D(\mathscr{S}) = \bigcup_{S \in \mathscr{S}} \overline{S}$$

We say that an aligned segment (k, z) is *relevant* to a segmentation analysis S iff $(k, z) \in \overline{S}$. Let $\mathscr{S}$ be a non-empty set of segmentation analyses of some utterance s, and $(k, z) \in D(\mathscr{S})$. We define the *restriction* of $\mathscr{S}$ to (k, z), noted $\mathscr{S} \downarrow (k, z)$, as the set of all segmentation analyses in $\mathscr{S}$ to which (k, z) is relevant:

$$\mathscr{S} \downarrow (k, z) = \{S \in \mathscr{S} \mid (k, z) \in \overline{S}\}$$

We obtain of course $\mathscr{S} \downarrow (k, z) \subseteq \mathscr{S}$.

Fact 1. $(k,z) \in D(\mathscr{S}) \Rightarrow \mathscr{S} \downarrow (k,z) \neq \emptyset$.
Proof. Trivial compactness property of union.

Let $\mathscr{S}$ be a non-empty set of segmentation analyses of some utterance s, and $(k,z) \in D(\mathscr{S})$. We say that (k,z) is *critical* in $D(\mathscr{S})$ iff it is not relevant to some $S' \in \mathscr{S}$. This implies that

$$|D(\mathscr{S} \downarrow (k,z))| < |D(\mathscr{S})|$$

Thus, selecting a critical segment in the interface will effectively reduce the search space. In practice, it will reduce it by half or more, and convergence will be ensured in $\log(N)$ steps, where N is the total number of segmentation solutions. Let us now give a sufficient condition for criticality.

Let (k,z) and (k',z') be two distinct aligned segments in some tabulated display $D(\mathscr{S})$. We say that (k,z) and (k',z') *conflict* if $k \leq k' < k + |z| - 1$ or $k' \leq k < k' + |z'| - 1$.
Fact 2. Let (k,z) and (k',z') conflict in $D(\mathscr{S})$. They are both critical, as they are mutually exclusive – no segmentation may contain both.
Proof. By inspection of sandhi rules, we may check that every rule $[x]u|v \rightarrow w$ is such that $|u|+|v| \leq |w|+1$. Thus any overlap of a segment with its successor in any segmentation is at most of length one. Since every segment is of length at least one, overlap of a segment with some other segment in the same segmentation solution may be at most of length one. Let (k',z') be an aligned segment of $D(\mathscr{S})$ conflicting with (k,z). No segmentation analysis to which (k',z') is relevant may belong to $\mathscr{S} \downarrow (k,z)$, and thus $(k',z') \notin D(\mathscr{S} \downarrow (k,z))$.

Note that the conflicting condition is sufficient to show that two segments may not appear in a common segmentation solution, but that this is not a necessary condition, even for contiguous segments. The interest of this notion is that it is easy to check visually, whereas the necessary and sufficient criterion is not, since sandhi rules are not shown.

We now state a fact which may not be true of all lexical juncture systems, but is verified for Sanskrit sandhi, as we shall argue in Section 7.3.
Fact 3. If $D(\mathscr{S})$ has no critical aligned segment, $\mathscr{S}$ is a singleton.

5 A GRAPHICAL INTERFACE

Let s be the utterance under consideration. Initially, we compute the set $\mathscr{S} = Segs(s)$ of all its possible segmentations, and we display $D(\mathscr{S})$,

where every aligned segment (k,z) is represented as the segment z displayed with an offset of k spaces from the left margin. When two aligned segments overlap, we represent them in separate rows of the display. We sort all segments, so that longer segments are listed above shorter ones. Each segment is displayed either with a blue check sign, if it does not conflict with any other segment, or else with two signs, a green check sign to select the segment, and a red cross sign to discard it. These green check and red cross signs are mouse-sensitive; they trigger as call-back the segmentation routine that will compute all segmentation analyses consistent with this particular choice, that is, for which all aligned segments currently selected with green check signs are present, and those segments discarded with red cross signs are absent. If s is segmentable at all, $\mathscr{S}$ is non-empty, and so is $D(\mathscr{S})$. At any point in the computation, the current display $D(\mathscr{S})$ represents the union of a non-empty set $\mathscr{S}$ of segmentations of s, by repeated application of Fact 1. Consequently, selecting or discarding a segment can never fail.

Furthermore, if the user selects or discards a critical segment, there is visible progress, since all conflicting segments vanish when a segment is selected, while any segment vanishes when discarded. This corresponds to the case where it conflicts with some other segment, which is easy to see in the visual display, since it covers a column that is strictly inside the conflicting segment.

When a segment is selected using the green check sign, both the check and cross signs are replaced by a single blue check sign, which is mouse-insensitive, thus making the segment inert for the rest of the interaction. On the other hand, if a segment is discarded using the red cross sign, it vanishes and in the particular case where it conflicts only with one other segment, the other segment will become inert. Note that the user cannot select a non-critical segment, since these are presented with blue check signs, which are not mouse-sensitive. When there are no more critical segments, we have reached a unique segmentation solution, consistent with Fact 3.

Several other actions, besides the selection of a segment, are possible at any moment. Firstly, users may undo the previous selections, up to an arbitrary depth. Secondly, they may revert to the old interface, which gives a linear listing of all segmentations consistent with the segments currently selected. A counter indicates how many

distinct segmentations remain. Users may also opt to use the semantic pruning mechanism to provide machine assistance, for potentially faster convergence. Finally, it is possible to send the remaining set of segmentations to the more complete dependency parser under development at the University of Hyderabad (Kulkarni *et al.* 2010; Kulkarni and Ramakrishnamacharyulu 2013).

A complexity analysis of the interface is presented in the Appendix. From experimental evidence, it has been observed that the number of solutions grows exponentially with the length of the utterance and the bound $O(C^n)$ has actually been reached for the real corpus (see Figure 10). For instance, the following sentence, excerpted from the *Vikramorvaśī* play by Kālidāsa, has 6,967,296,000 ($\approx 2^{32}$) segmentations. The sentence has 240 phonemes, and the desired solution has 40 segments. This sentence can be managed by our interface in 17 clicks, so the convergence is quite fast.
yā tapasviśeṣapariśaṅkitasya sukumāram praharaṇam mahendrasya pratyādeśaḥ rūpagarvitāyāḥ śriyaḥ alaṃkāraḥ svargasya sānaḥ priyasakhī urvaśī kuberabhavanāt pratinivartamānā samāpattidṛṣṭenakeśinā dānavenacitralekhādvitīyā bandigrāhaṃ gṛhītā. 'Our dear friend *Ūrvaśī*, who is the youthful weapon of *Mahendra*, the one fearful of the power of extra-ordinary penance, who is an overshadower of *Śrī*, who is proud of her beauty, and who is an ornament of heaven, was taken captive, together with *Citralekhā*, by the demon *Keśī*, who had appeared by chance, while she was returning from the house of *Kubera*.' (English translation by Brendan Gillon)

More example cases will be presented in Section 8.

6 GRAPHICAL RENDERING OF THE DISPLAY

Figure 1 presents the graphical rendering presented by our system for the following sentence:
satyaṃbrūyātpriyaṃbrūyānnabrūyātsatyamapriyaṃpriyaṃcanānṛtambrūyādeṣadharmaḥsanātanaḥ.
This is the well-known saying (*subhāṣitam*): 'One should tell the truth, one should say kind words; one should neither tell harsh truths, nor flattering lies; this is a rule for all times.'

As indicated in the display, this diagram summarizes 120 distinct segmentations. The colour code used for the segments indicates var-

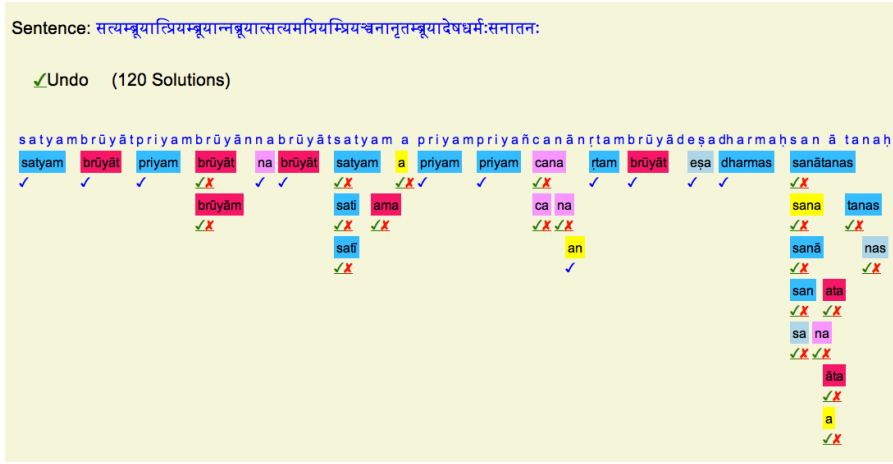


Figure 1: Initial display of the aligned segments for the sentence *satyaṃbrūyātpriyaṃbrūyānnabrūyātsatyamapriyaṃpriyaṃcanānṛtambrūyādeṣa-dharmaḥsanātanaḥ*

ious lexical categories, e.g., blue for substantives, red for finite verb forms, purple for adverbs, pale blue for pronouns and yellow for compounds.

The main notion behind the interface is that of the display $D(\mathscr{S})$ for a consistent set of segmentations $\mathscr{S}$. Initially, we take $\mathscr{S} = Segs(s)$, and we progressively select aligned segments $(k_1, z_1), \ldots, (k_n, z_n)$. The only data kept in memory are the initial sentence s, and the stack of choices $C_n = (k_1, z_1), \ldots, (k_n, z_n)$. The interface interaction is implemented as a CGI coroutine, which receives arguments s and C_n in its invoking URL. The server recomputes the sequence of all segmentations $Segs(s)$ at every step, keeping only those consistent with the stack of choices C_n, sorted by alignments into a sorted list of checkpoints. The display of all consistent segmentations is stored in an array 'display' of size $|s|$. The display value at index i is the list of all segments z such that (i, z) is an aligned segment of some segmentation solution consistent (i.e. not conflicting) with all the checkpoints C_n. This test is easy, since C_n is sorted. One may think of the display as a shared representation of $D(\mathscr{S})$, for $\mathscr{S}$, the set of segmentation solutions consistent with the current stack of choices. Actually the array 'display' may be thought of as a hashcoding array for the set $D(\mathscr{S})$, with the hashcode of an aligned segment (k, z) being its alignment k in the input string.

What is crucial for the efficient sharing of the tree of all segmentations as a directed acyclic graph (DAG) is the abstraction of sandhi rules. Indeed, our methodology is reminiscent of parsers based on tabulation methods, which use such dynamic programming methods (Earley 1983; Tomita 1985; Billot and Lang 1989; Stolcke 1995).

Implementation of 'Undo' is trivial, since it consists in calling the interface with the same stack of choices minus the last choice.

Note the simplicity of this implementation: at every step, all the information is recomputed with the standard segmenter but, since the technology is very fast, this is not noticeable to the user as the reaction seems instantaneous (at least on a localhost server).

Presenting the tabulated display of the aligned segmentations as an HTML page was not entirely trivial. The segmentation analysis gives us all possible segments, appearing at various offsets. First, for an arbitrary offset k_i, the number of segments may be quite large. Also, the length $|z_i|$ of the largest segment (k_i, z_i) at offset k_i might be such that it conflicts with the aligned segments at the next offset k_{i+1}. Since the objective was to have a compact display, keeping the alignment intact, the problem of where to fit the aligned segments at offset k_{i+1} remains, in such a case, once the HTML display has been populated with the segments at offset k_i. The second issue is related to the fact that, while the maximum size of the display array is fixed as the length of the utterance ($|s|$), the size of an aligned segment (k_i, z_i) is $|z_i|$, a variable depending on the segment z_i. Thus, the problem is to show the aligned segment as a single entity.

Now, a simplistic implementation to keep the alignment intact would have been to list all the segments corresponding to the offset k_{i+1}, starting from the next row after all the segments at offset k_i have been enumerated. This would obviously not lead to a compact display. Similarly, a very simple implementation to handle variable-sized segments would be to define an array of $|s|$ columns and display each solution (k_i, z_i) in $|z_i|$ columns, starting from the $k_i{}^{\text{th}}$ column. The problem with this approach is that the display of a word does not appear continuous here. Also, depending upon the transliteration scheme used, some phonemes would require more space than others, so the row length will be variable. And the segment z_i cannot be treated as a single HTML entity in this case, which is a requirement for user-friendly display

of morphological tags, as well as for the callbacks, initiating user interaction.

To alleviate these problems, we sorted the segments at each offset according to length.[4] Thus, the longer segments appear at the top. Now, while filling the segment (k_{i+1}, z_{i+1}) at offset k_{i+1}, we search for the first row from the top where the last filled segment does not conflict with (k_{i+1}, z_{i+1}), and fill this segment in that row. This gives a much more compact display.

Similarly, to handle the second issue, instead of using $|z_i|$ columns for an aligned segment (k_i, z_i), we used the HTML 'colspan' attribute to use variable width columns in a row. Thus, an aligned (k_i, z_i) is displayed, using a $|z_i|$ width column at offset k_i. This allows continuous display of a segment, as well as treating it as a single HTML entity.

7 LEXICAL CATEGORIES AND TAGGING

7.1 *Dealing with lemmatized segments*

Since our method is lexicon-directed, our candidate forms are morphologically generated, and may be kept along with their lemmas. Furthermore, we may restrict our segmenter to recognize only morphologically correct sequences, according to a regular grammar expressing morphological constraints. This refinement is also necessary because the sandhi relation after preverbs (*upasarga*) is different from the external sandhi between words or compound components. This grammar is compiled into the state-transition graph of a finite automaton/transducer, which expresses the control of our lexical scanner in the usual manner. The states of this automaton, called *phases*, correspond to the lexical categories associated with colours in the interface. We may refine the above formalization to this new situation easily, replacing the notion of aligned segment (k, z) by the finer notion of *aligned lemmatized segment* $(k, (l, z))$, where l is the lemmatization of segment z.

We can go back to the example sentence in Section 6, for which the initial display summarizing 120 distinct segmentations is presented in Figure 1. At the right side of the diagram, one sees the long

[4] Note that this sorting is prioritized from left to right, as this is the most natural order for reading Sanskrit text.

segment *sanātanas* (‘eternal’) and below it several choices of smaller words that are obviously overgenerated items. Clicking on the green sign under the blue segment *sanātanas* removes all this noise, and the number of potential solutions drops to 12, generating the display given in Figure 2 – note the blue unlinked check sign indicating the previously selected segment.

Similarly, one immediately notices the segment *satyam* (‘truth’), together with conflicting noisy alternatives. Similarly, for *cana* (‘and not’), these two selections will leave us with only one choice between segments *brūyāt* and *brūyām* (two forms of root *brū* ‘to say’ in the optative mood of the present active voice in the singular number, respectively in the 3rd and 1st person). By obvious symmetry with its other occurrences in the sentence, *brūyāt* must be chosen, obtaining the correct segmentation in a total of 4 easy clicks, shown in Figure 3. At this point, one may click on the explicit button “Unique Solution”, where fine tuning of the final morphological parameters, such as ambiguities of gender of substantival forms, may be effected through a final user interface, shown in Figure 4. This last stage is necessary, because our lemmas label a given form with a multi-tag, factoring out all values of morphological parameters usable to generate this form. The user can select the appropriate options to produce the final unambiguous tagging of the sentence as a list of lemmas, where segments are hyperlinks to the digital lexicon, as shown in Figure 5.

This page may be stored, and the next sentence may then be read from the corpus input stream, in order to progressively annotate the digital library.

Sometimes it is useful for annotators to see the lemmatization of a segment in order to make a decision with more information than merely its lexical category (indicated by the colour code). This facility

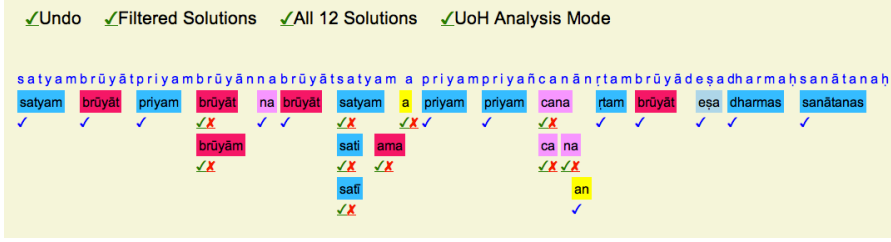


Figure 2: Aligned segments after selection of segment *sanātanas*

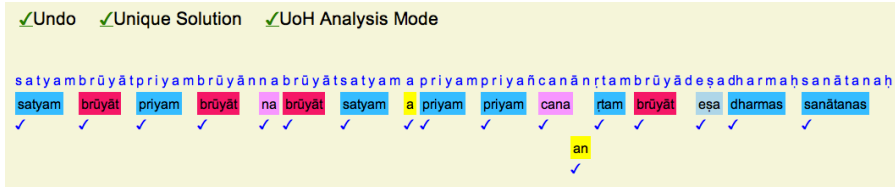


Figure 3: Aligned segments after 4 clicks

Figure 4: The interface for selecting unique tags from multi-tags

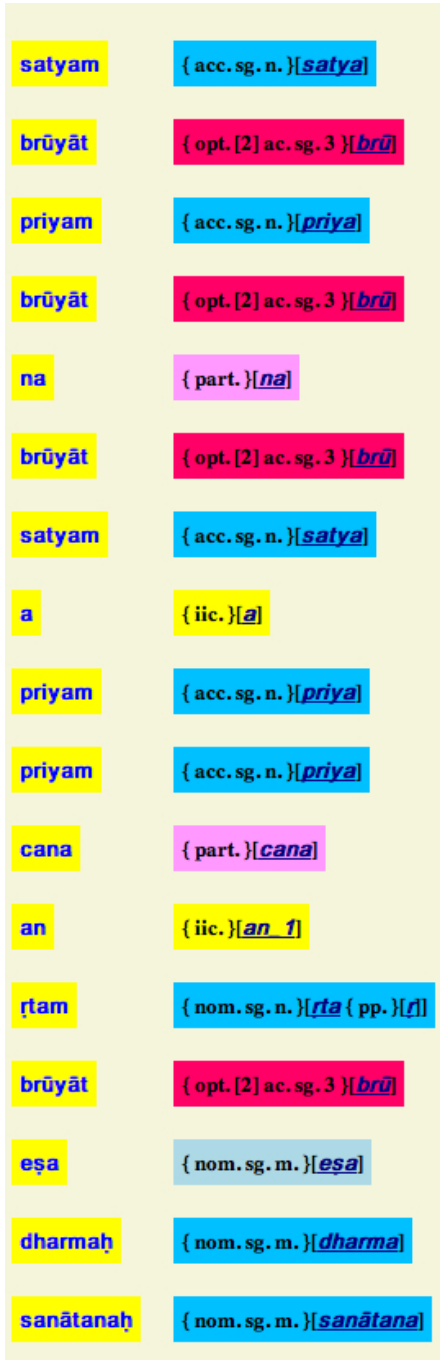


Figure 5:
Final tagging

is available in the interface: every segment is mouse-sensitive, and clicking on it yields its lemma, as shown in Figure 6 for the segment *brūyāt*.

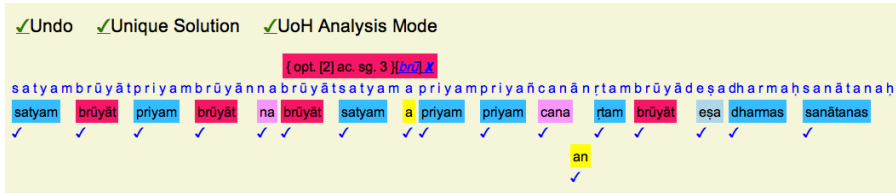


Figure 6: Asking for the lemma of the segment *brūyāt*

Note that, in this lemma, the root *brū* is itself mouse-sensitive; it is a hyperlink to its lexical entry, allowing access to its meaning. We provide two aligned digital lexicons, our original Sanskrit-to-French Heritage dictionary, and also the more complete classical Sanskrit-to-English Monier-Williams (MW) dictionary (Monier-Williams *et al.* 1899).[5] Thus annotators have all available information at their disposal at any point, but with minimal cluttering of the workspace.

It should be noted that this interface is not only easy to use, it is actually fun to play with. It may be thought of as some kind of electronic game.

7.2 *Rationale for using the cross signs*

The cross signs presented for conflicting segments are used to discard a particular segment. However, it may be argued that this result may more efficiently be achieved by selecting the correct segment. While this would be the appropriate in majority of cases, there are a few instances where one would need to use a cross sign to select the appropriate solution. Figure 7 describes the possible segmentations for the utterance, *ihehi*, which can be analysed either as *iha* + *ā* + *ihi* or as *iha* + *ihi*. The interface presents the segments *iha* and *ihi* with blue check signs, indicating that these do not conflict with any other segment, but the segment *ā* is presented with a green check and a red cross sign. If we had used only a green check sign, it would not have been possible for the annotator to select the solution *iha* + *ihi*,

[5] The protocol for the non-trivial task of mutually linking these lexical resources has been discussed in Goyal *et al.* (2012)

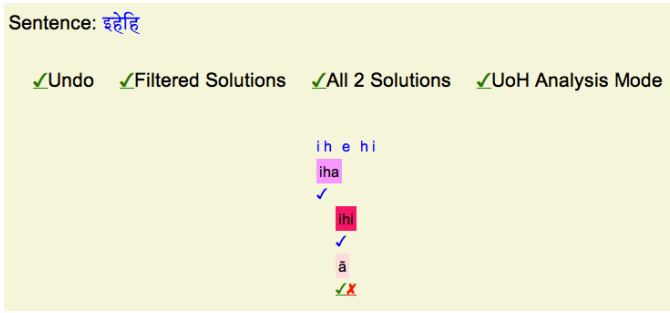


Figure 7:
The aligned segmentations for *ihehi*.

since there would have been no opportunity to discard the *ā* segment. With this facility, the annotator is free to choose either of these two analyses.[6]

7.3 *Justifying Fact 3*

In the example just shown, we assumed implicitly that when no more choices were available to the user, there was only one segmentation solution left, and we could then proceed to the final disambiguation of the remaining multi-tags of this unique solution. This assumption is precisely what we called Fact 3 above, and that we now restate:

Fact 3. If $D(\mathscr{S})$ has no critical aligned segment, $\mathscr{S}$ is a singleton.

Proof. Assume that $D(\mathscr{S})$ has no critical aligned segment. In other words, all the segments are marked with a blue mark, indicating that they belong to all remaining solutions. Thus, all remaining solutions have the same segments. We shall need to prove that all the aligned segments are strictly ordered within one unique solution. Consider any two remaining segments (k, z) and (k', z') where, without loss of generality, we may assume $k \leq k'$. If $k < k'$, the z segment must precede the z' one. Now let $k = k'$. It is not the case that both $|z| > 1$ and $|z'| > 1$, since the two segments would conflict with each other. Assume without loss of generality $|z| = 1$. If $|z'| > 1$, the z segment must precede the z' one. We are left to consider the case where $|z| = |z'| = 1$. The only relevant mono-phonemic segments in classical Sanskrit are the privative prefix *a*, forming so-called *nañ-tatpuruṣa* compounds, and the

[6] Another possible way to achieve this would have been to use null segments and allow the annotators to choose between the null segment or the other possibility. We, however, prefer to use the cross sign, since it also helps a reader to reduce the number of possibilities by discarding some nonsensical combinations.

preposition *ā*, used as prefix (*upasarga*) to final (*tiṅanta*) and propositional (*kṛdanta*) verbal forms.[7] We thus only have to consider the proper ordering of co-aligned *a* and *ā* segments. The privative particle *a* can prefix only consonant-initial nouns, since it alternates with the form *an* for vowel-initial ones. The preposition *ā* is assumed not to be iterated, which would be redundant. Thus, the only possible ordering is that an *ā* segment could precede an *a* segment (but we do not know of even one concrete example). This explanation justifies Fact 3 in the case of classical Sanskrit.

7.4 *Robustness*

The interface is remarkably robust for realistic sentences, as shown in the example in Section 5. Figure 8 shows the initial display of our interface, where the sentence from the *Vikramorvaśī* play by Kālidāsa, as mentioned in Section 5, is processed by the Sanskrit reader. The interface shows all the 6,967,296,000 possible solutions in a compact display. The display presents various choice points to the user, and is manageable in 17 clicks. A full evaluation of the interface, for robustness as well as convergence analysis, is presented in the next section.

8 EVALUATION

To evaluate the robustness of the proposed system, we used a dataset consisting of 1,500 sentences from *Pañcatantra*. These sentences were annotated, using a software-assisted human interface for morphological tagging, built on top of the Sanskrit Heritage Reader (Goyal *et al.* 2012). The annotators were allowed to give their own annotations, when the correct segmentation did not appear in the system. The length distribution of the sentences used in this study is shown in Figure 9. Clearly, many of the sentences contained more than 90 characters.

To study its robustness, we checked whether the sets of segmentation analyses for these sentences contained the segmentations identified by the annotators. When these sentences were given as input to the system, the system gave a summary page in *each of the cases*.

[7] Vedic Sanskrit offers additional difficulties, with autonomous prepositions and the mono-phonemic interjection *u*.

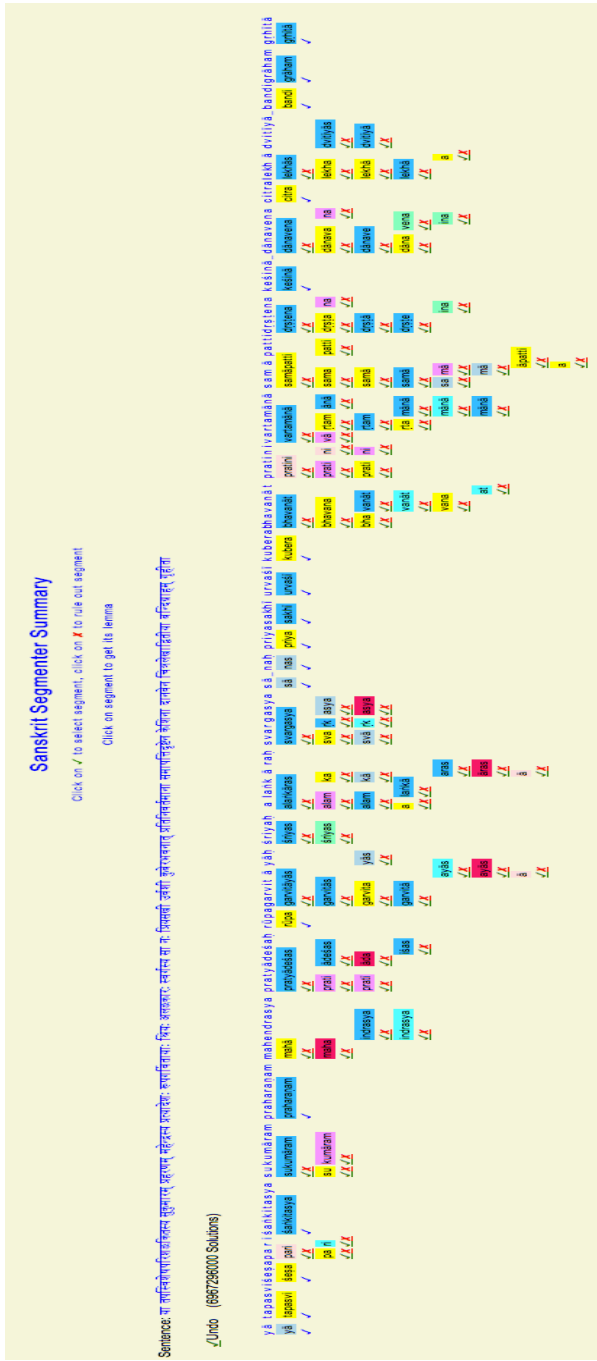


Figure 8: Checking the interface for a long sentence

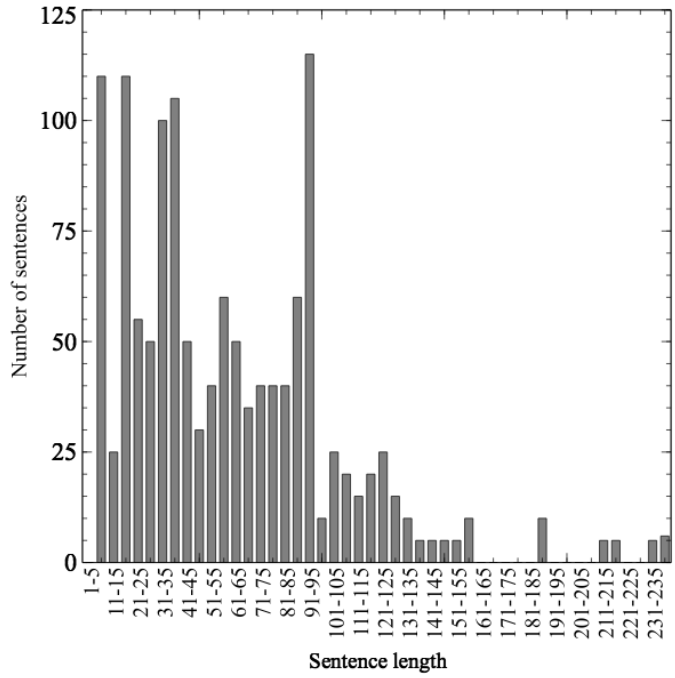


Figure 9: Distribution of length for the sentences used in the evaluation

Figure 10 plots the log of the number of solutions identified by the system, with respect to the length of the sentence. The plot is truncated at 125 characters both for the sake of visibility and also because few sentences exceed that limit. We can clearly see that the number of solutions increased exponentially with the length of the sentence.

On further analysis, we found that in 1,092 out of 1,500 sentences, all the segmentations from the gold standard were present in the set of segmentation analyses, returned by the summary interface of the system. On analysing the rest of the cases, we found that, in 59 cases, the annotated sentence did not match the input sentence, and a few changes had been introduced by the annotators. We therefore studied the performance of the system on the remaining 1,441 sentences. First, we measured the recall of the system by identifying how many of the words in the segmentation were also present in the summary interface. We measured both micro- and macro-averaged recall. As per the standard definition, for macro-averaged recall, we computed recall for each of the sentences and then took an average for all 1,441 values, one for each sentence. For micro-averaged recall, we computed the

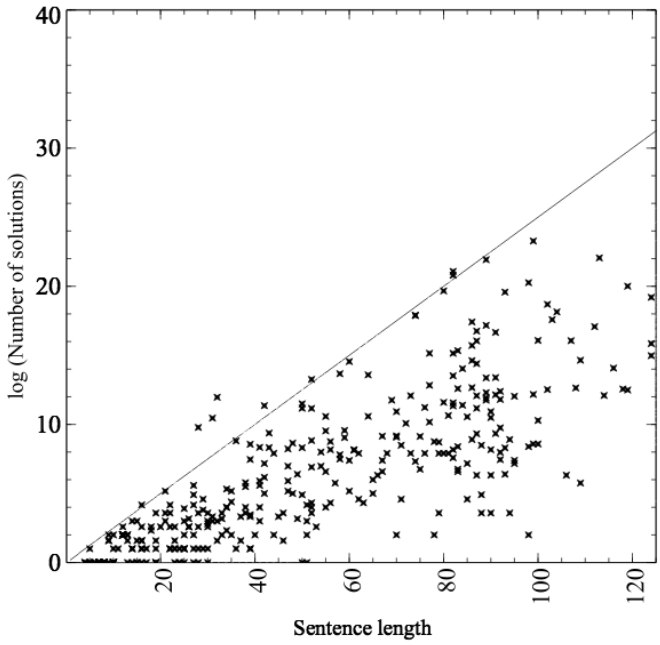


Figure 10:
Scatter plot showing the distribution of the number of solutions (log) with respect to the length of the sentence

fraction of words present overall. These values were found to be high: 0.96 and 0.97 respectively.

For 349 cases, one or more words in the gold-standard segmentations could not be mapped to the segmentations returned by the summary interface. A further analysis revealed that, in 204 of these cases, the system could not recognize a word from the sentence, mostly because of the incompleteness of the lexicon used by the Sanskrit Heritage system. This problem can be solved by supplementing the lexicon with new words specific to the particular corpus under consideration. In the next section, we describe how the current system helps annotators to do this in an interactive manner. For most of the other cases, the main issue was that the original sentence contained a quote, or that a punctuation mark occurred in the middle of the sentence, e.g., the sentence,[8] *tatas tayā, manorathānām apy agamyam, iti matvā, tathā, iti pratipannam* was pre-processed and the following was the input to the system: *tatas tayā manorathānām apy agamyam iti matvā tathā iti prati-*

[8] And she assented, for she thought: "It is a thing beyond my fondest aspirations." (English translation)

pannam, resulting in the words *tathā iti* being separated by a space in 'sandhied' mode.[9] This was not recognized by the system, as it denotes the *pada-pāṭha* ('unsandhied') form and not the sandhied form. In 'sandhied' mode, the correct input would have been *tatheti*, as *tathā iti* leads to other interpretations, such as *tathās* + *iti*, *tathau* + *iti*, etc. In future, we might be able to auto-detect this, along the lines of spell-correction.

In many such cases, the system could not make use of sentence breaks and punctuation information, which were removed during pre-processing. The system has to be adapted to allow such information in the input, to be able to make adjustments during segmentation.

Further, to empirically evaluate the convergence time to get the unique solution out of all the possible segmentations returned by the system, we took a sample of 10 sentences of length ≥ 100, and the annotators were asked to use the summary interface to come up with the unique solution. We noted down the actual time taken, as well as the number of clicks used by the evaluators. The details are provided in Table 1 below. The length of the sentences varied between 113 and 224, and the total solutions were as high as 3,736,212,480. However, in all cases, at most 19 clicks were required to achieve the unique

Table 1: Empirical evaluation of the convergence time for 10 different sentences

S. No.	Sentence length	Total solutions	Number of clicks	Time (in sec.)
1	150	22,394,880	14	59.2
2	115	4,368	6	28.2
3	156	19,051,200	17	56.3
4	224	248,832,000	17	73.6
5	149	18,662,400	14	42.9
6	149	3,736,212,480	19	78.7
7	113	2,880	8	32.3
8	122	9,216	8	17.0
9	122	17,600	10	36.4
10	169	167,215,104	17	65.8

[9] This is one of the parameters of the system. The user can choose the mode 'sandhied' to read a sentence that is not segmented and the mode 'unsandhied' to read text that has already been sandhi analysed (*pada-pāṭha* form).

solution, and the maximum time taken was 78.7 seconds, which is quite fast, as well as practical.

The segmentation method is lexicon-directed. Thus, for any aligned segment (k_i, z_i) to appear in the segmentation solution, the segment z_i must belong to the vocabulary L. It will thus be incomplete, if the generative lexicon does not completely cover the vocabulary of the targeted corpus. In the next section, we will describe how our interface is robust enough to handle the cases when a chunk (part of the utterance s, which is segmentable, independent of the rest of the utterance) is not recognized by the system.

9 PARTIAL SEGMENTATION, ERROR RECOVERY, LEXICAL ACQUISITION

In general, given an utterance s, a chunk may remain 'unanalysed' or 'ill-analysed'. The case of 'unanalysed' chunks might occur due to one of the following reasons:

- The utterance s contains an invalid chunk z_i, not allowed by the grammar, or
- s contains a segment (chunk) z_i, which is a valid segment, but does not appear in the vocabulary L, due to the incompleteness of the lexicon.[10]

A chunk may remain ill-analysed if the desired solution does not appear in the segmentation returned by the system. This mostly occurs because of the incompleteness of the lexicon.

In order to deal with this incompleteness, and make our interface robust, we extended it in such a way that it will report the unanalysed chunks of input, and allow for their correction. This facility has been provided by adding a supplementary phase to the lexer, allowing any phonemic string. Thus, when the system is unable to recognize a segment z_i in utterance s at offset k_i, this unanalysed segment is displayed in grey along with a spade sign. This spade sign triggers as callback another CGI routine called 'user-aid', initiating an interaction loop with the user.

[10] Note however that our lexicon is 'generative' to a certain extent: most participles (*kṛdantas*) are systematically generated from root entries, and compounds are analysed, and thus do not need to be explicitly listed in the lexicon.

For the 'ill-analysed' chunks, since there is at least a partial solution, no explicit link is provided to the 'user-aid' CGI. Instead, if the user decides that the analysis presented by the system is not correct for some particular chunk, clicking on the chunk will provide access to the 'user-aid' CGI for the given chunk. This routine provides the following options:

Edit and resubmit the sentence. If the user has entered a wrong sentence *s* (for instance due to misspelling), this option allows the user to edit the sentence *s* and submit it to the system for re-analysis.

Edit and resubmit the chunk. This option allows the user to edit only the wrong chunk in *s* and does not disturb the rest of the sentence. The user can edit the chunk and the system will show the analysis corresponding to the modified chunk, keeping the segmentation solution of the other chunks intact.

Show partial solution without this chunk. This option appears only when there are at least two chunks in the sentence. This option allows the user to see the partial solution without using that chunk.

Select among possible lemmatizations. This module tries to guess the possible lemmatizations (analyses) of a segment from the 'Unknown' phase. This module is developed using finite state methods and will be discussed in Section 9.1 below.

Enter your own lemmatization. If users feel that none of the suggested lemmatizations are correct, this option allows them to enter the lemmatizations of their choice. This module will be discussed in Section 9.2 below.

9.1 *Experimental Stemmer*

We have implemented an experimental stemmer, in order to attempt semi-automatic lexicon acquisition, at least for substantive stems. This is a very difficult problem, in the presence of retroflexion rules by internal morphology. This progressive assimilation of the retroflex articulatory feature operates on a non-bounded left context of the rule application, and thus cannot be directly modelled as an invertible regular transduction. Fortunately, retroflexion rules do not cross word boundaries, and thus do not pollute external sandhi.

The experimental module for guessing the possible lemmatizations for an 'unanalysed' chunk is built using the suffix segmentation

rules, learnt from the database of inflected forms, available with the Heritage lexicon. To give an example of the rules learnt, consider the following entry in the database of inflected noun forms:

rāmas nom.sg.m. [rāma] 'Rama, name of a person'

The entry has three different parts, the inflected form *rāmas*, the stem corresponding to this form *rāma* and the morphological information of this inflected entry 'nom.sg.m.'. This entry is used to learn the following rule:

$$x.a \xrightarrow{nom.sg.m.} x.as \tag{1}$$

where $x.a$ denotes any phonetic string ending in the phoneme 'a'. Similarly, for the entry,

takṣan loc.sg.m. [takṣṇi] 'carpenter'

The rule learnt would be

$$x.an \xrightarrow{loc.sg.m.} x.ṇi \tag{2}$$

Note that, in both these cases, the context (right context, or ending) is chosen based on the following criteria:

- The context should not be empty. This condition was used so that the rule would not cause an over-generation. Thus we will not learn the rule $x.\phi \xrightarrow{nom.sg.m.} x.s$ in the first case where ϕ denotes a null context.
- The minimum possible context should be used to describe the rule. This condition was used to avoid the segmenter failing because of having too long a context. Thus we will not learn the rule $x.ma \xrightarrow{nom.sg.m.} x.mas$ in the first case, because then it would not allow us to recognize that the word *mohanas* is a declined form of the stem *mohana* 'Mohana, name of a person' because the context *nas* would not match the one used in the rule (*mas*).

Now, since the database also contains very special rules, which might be applicable to only a few stems, a simple probabilistic model is used to filter these rules. The first filter is based on the frequency count of a certain rule, that is, how many times this rule is encountered while declining the nominal forms in the lexicon. Rule 1, above, is used 5079 times, while rule 2 is used only 13 times.

The next filter is based on the conditional probability of a stem and morphological analysis being associated with a given suffix. Thus, a rule is selected only if the probability of the stem and morphologi-

cal analysis given the suffix is greater than the threshold. For rules 1 and 2 discussed above, these probability values were found to be 0.96 and 0.05 respectively. For rule 2, this probability was low because, given the ending *ṇi*, the stem ending in *n* with the same morphology is more likely (probability of 0.63). An example of one such entry in the database is: *dīrghasūtrin loc.sg.m. [dīrghasūtriṇi]* 'spinning a long yarn, procastinating'.

Thus, two different thresholds are used, frequency count and probability. The criteria for selecting these thresholds involved a trade-off. A low value for these thresholds would allow too many unnecessary solutions for a given segment. A very high threshold, on the other hand, might not be able to provide the desired solution. Thus, these values were tuned on a corpus,[11] resulting in optimal values of 3 for frequency and 0.02 for probability.

Once these rules were learnt from the database, they were fed into a finite-state transducer, which could then be used to guess all the possible stems along with the morphological analysis for a previously unanalysed form. All the possible lemmatizations produced by the transducer are displayed to the user. The interaction loop for lexicon acquisition is discussed in the next section.

9.2 *Lexicon acquisition*

For a segment in the 'Unknown' phase, various lemmatizations are proposed by the transducer. They are presented to the user accompanied by radio buttons. These radio buttons allow the user to select among the various suggested lemmatizations. It should be noted that one of the objectives of this module is to acquire the stems that appear in the corpus but are not available in the Heritage lexicon. To assist the user, we search for each suggested stem in the Monier-Williams (MW) dictionary, which is one of the most complete lexicons for Sanskrit. If a stem appears in the MW, the stem is displayed with a hyperlink to the online MW dictionary, and the radio button corresponding to this entry is preset. This is based on the intuition that, among all the possible choices, any choice that is already present in a more complete lexicon is more likely to be correct, and will, in any case, be verified

[11] We collected examples of unanalysed segments from the Bhagavad Gītā text. These examples were used to tune the thresholds.

by the user. If the user selects any of these suggestions, the base entry and gender information are saved in a 'cache' database.

If the users cannot find the desired solution among the suggested lemmatizations, they are allowed to enter their own lemmatization. A text area is provided for the user to enter the stem, with various select boxes, to be completed with morphological information, such as gender, case and number. Once the user submits this information, the base entry and gender information is saved in the 'cache' database.

To illustrate this procedure, we input the following sentence from *Pañcatantra* into the Sanskrit reader: *ye punar ātmīyāḥ śṛgālā āsan te sarve 'py ardhacandraṃ dattvā niḥsāritāḥ*. 'But to all the jackals, his own kindred, he administered a cuffing, and drove them away.'

Figure 11 shows the aligned segments as returned by the system. The system does not present any analysis for the segment *ātmīyāḥ* (his own kindred), which is displayed in grey, along with a spade sign.

Once the annotator clicks on the spade sign, it opens the 'user-aid' CGI routine. Various options presented to the annotator by this routine are shown in Figure 12. In this particular case, the segment *ātmīyāḥ* is a valid segment, which remains unanalysed because the stem *ātmīya* is not present in the lexicon. Thus, we will focus on the option 'Select among possible lemmatizations'. The annotator is presented with various possible analyses but the specific analysis with the stem *ātmīya* present in MW has been shown with a hyperlink. The annotator can select the radio button corresponding to the first analysis in the

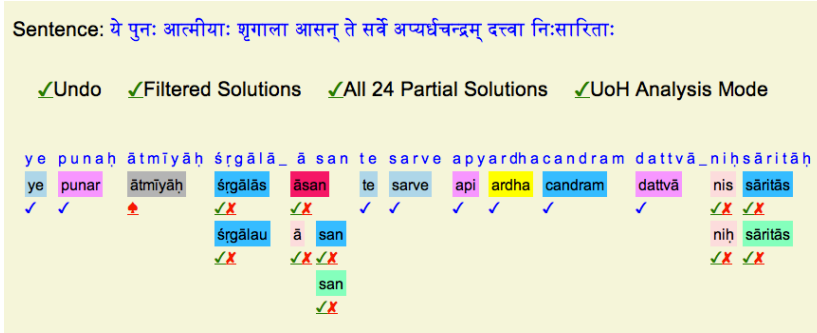


Figure 11: The partial segmentations for the sentence *ye punar ātmīyāḥ śṛgālā āsan te sarve 'py ardhacandraṃ dattvā niḥsāritāḥ* with the segment *ātmīyāḥ* remaining unanalysed

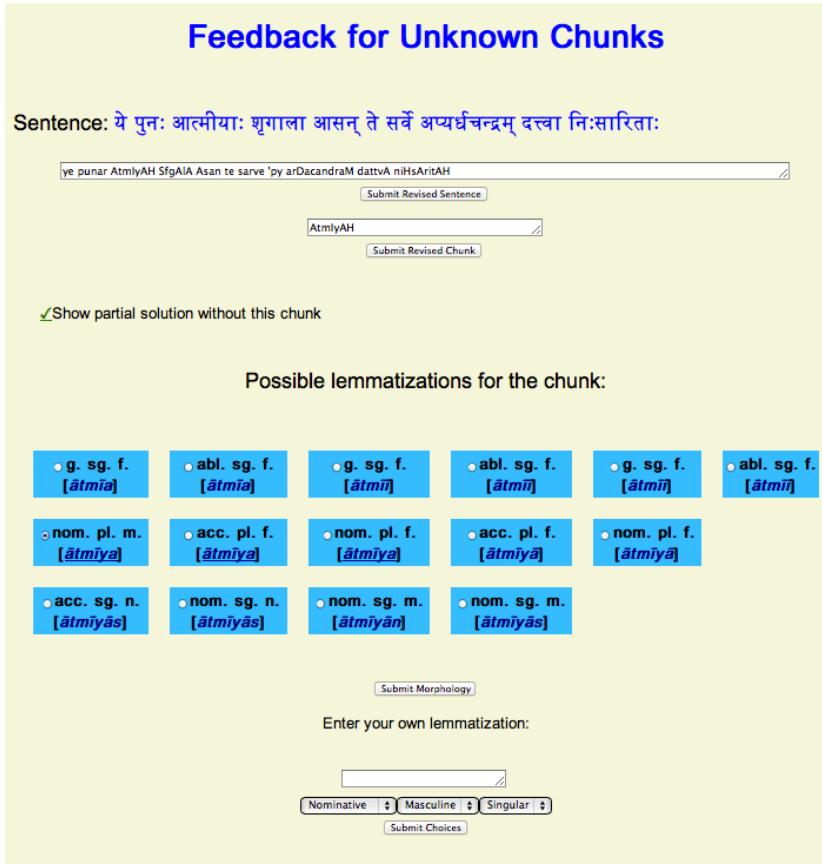


Figure 12: Options provided to the annotator for the unanalysed chunk *ātmīyāḥ*

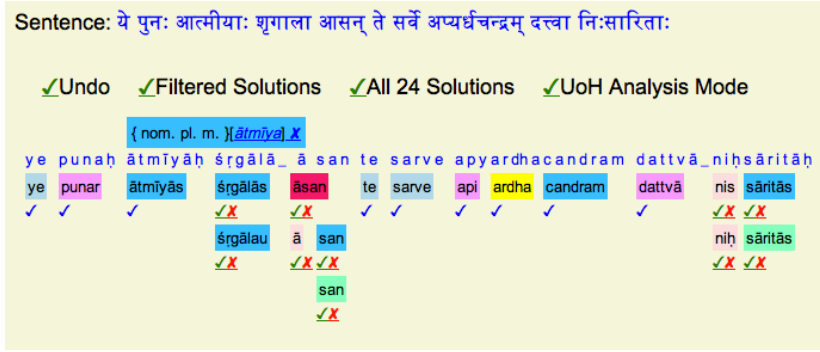


Figure 13: Revised interface for the annotator with *ātmīyāḥ* analysed as chosen, using the options shown in Figure 12

second row (nom. pl. m. [*ātmīya*]) and click on the button 'Submit Morphology'.

This information provided by the annotator is stored in the 'cache' database. The morphological generator is used to generate all the forms corresponding to the stems stored in this database. This cache database augments the lexicon L, and thus enables the system to recognize a segment that was previously unanalysed or ill-analysed. Figure 13 shows the revised interface that is presented to the annotator, with the segment *ātmīya* analysed as chosen. Now the annotator can complete the tagging by going through the normal process, as already described in detail.

It is to be noted that, once this stem is processed to augment the lexicon L, the system can recognize all other inflected forms corresponding to this base stem as well.[12] This feature is particularly useful for annotators working on a specific corpus, since an unanalysed stem is likely to appear in that corpus again, possibly as a different utterance in another morphological context. The information about the selected stem and gender is stored in a local file on the annotator's workstation, which may be passed on to the lexicon manager for lexicon acquisition.

9.3 *Evaluating the experimental stemmer*

To evaluate the experimental stemmer, we used the 53 nominal forms that were not recognized by the system. These 53 words were passed to the 'user-aid' CGI routine. Among the suggestions provided by the system, we selected the particular lemma that corresponds to the stem in the Monier-Williams dictionary, which would have given that nominal form. On manual verification, we found that, in 52 out of 53 cases, the lemma matched the one provided by the annotators. This confirms that this experimental stemmer can be used very effectively by the annotators to deal with words that are unknown to the system.

[12] All the paradigms for generating the nominal and verbal forms are already available in the system. Thus, given a new nominal stem as input, the system can generate all its inflected forms, which are added to the database. At the time of analysis, all these forms are therefore recognizable.

10 CONCLUSION

We have presented a new interface for interactive segmentation-cum-tagging of Sanskrit sentences. This technology is not limited to Sanskrit. It can be adapted for interactive feedback, with a segmenter, tagger or parser, where sentences are presented as a finite collection of sequences of annotated word forms (lemmas). It may also operate at the generative morphology level, where words are presented as a combination of morphemes.

This interface enables a human annotator to visualise a sentence as a sequence of words, readable in one compact hypertext page. Word forms are vertically aligned with the original input. This allows the sharing of lemmas, and avoids cluttering the visual display with redundant information. Segments at a given offset are sorted by length, in decreasing order, which permits easy selection, with a heuristic of maximum overlap of segments with the input sentence. This heuristic, which tends to minimize the number of segments, is very often correct. Small word forms or morphemes, which agglutinate by chance into larger chunks of the input, get relegated as noise to the bottom of the display screen.

Fast recomputation of solutions respecting selection or rejection of a given segment achieves an exponential convergence rate. Even for long sentences admitting billions of solutions, the effect of these selections is instantaneous. Selection mistakes may be fixed rapidly using the undo facility. Morphological information is hidden in order not to clutter the screen, since appropriate use of colours for lexical categories usually facilitates the right decision. In case of doubt, the annotator may click on any puzzling segment and instantly obtain its full lemmatization, including lexicon access, if required, to check the meaning.

The main concept behind the data structure containing the display information is dynamic programming, i.e. sharing a tree structure as a directed acyclic graph, a standard technique in tabulated parsers. The originality of our approach is that the tree structure is not the forest of parse trees, but the union of all possible segmentation solutions, from which sandhi justification has been erased. This representation allows exponential savings, both in space (the displayed graph) and in time (the number of disambiguation operations).

The main ideas of this interface have been reused to summarize all possible dependencies between word forms in the dependency parser developed at the Sanskrit Studies Department of the University of Hyderabad.[13] This parser may be accessed as a second pass of our segmenter, leading to a smooth combination of the two processes – the user switches seamlessly between tagging and parsing (Huet and Kulkarni 2014). When to call the parser is actually an interesting tradeoff. If we call it too early, it will just choke under the enormous number of possible taggings. On the other hand, if we use our manual interface until we have produced a single set of tags, we lose many of the benefits of automation, since the dependency analysis would discard many inconsistent word combinations.

We have presented a novel technique for lexicon acquisition during corpus tagging by annotators, which makes our interface robust to lexicon incompleteness, but also to corpus mistakes and to non-standard enunciations (non-Paninian forms, Prakrit,[14] onomatopoeia, foreign words, etc.). The current module is developed only for nominal forms and needs to be extended to handle verbal forms as well. Another limitation of this module is that the system would only be able to guess a stem if the unanalysed chunk contains only one word. Handling cases where the unanalysed chunk contains more than one word is the next logical goal for our project.

Our interface has been tested successfully by the Sanskrit Library team[15] for the annotation of a variety of classical Sanskrit texts (Scharf *et al.* 2015).

APPENDIX: COMPLEXITY ANALYSIS

The convergence of the selection via the interface is very fast. Since the method is dichotomic, it converges on average in $\log(N)$ steps, where N is the total number of segmentation solutions. Indeed, when the input may be split as $s = s_1 \cdot s_2$, with s_1 and s_2 independently segmentable, with respectively n_1 and n_2 segmentations, presented with displays of sizes respectively d_1 and d_2, the global display has a size of $d_1 + d_2$ for

[13] `http://sanskrit.uohyd.ac.in/scl/`

[14] By the term 'Prakrit', we mean Middle Indo-Aryan languages such as Pāli.

[15] `http://sanskritlibrary.org/`

a total of $n_1 \times n_2$ segmentations. This interface thus gives an exponential improvement over the recursive dove-tailing of the segmentation process. In any case, the number of selections will be smaller than the number of words of the intended segmentation, i.e. of the order of the length of the sentence divided by the average length of a word. In practice, convergence is very fast.

Theorem. Let $\mathscr{S}$ be the set of segmentation analyses of some utterance s of length n. $|\mathscr{S}|$ is of asymptotic order $O(C^n)$, whereas $|D(\mathscr{S})|$ is of asymptotic order $O(n)$.

Proof. This theorem depends on the lexicon being used and can have, at best, an average complexity analysis. Let m be the length of an average segment of an utterance s. For our analysis, we will also assume that each segment in a valid solution has length ≥ 2.

Consider s of length n. We will try to find an upper bound on the number of segmentation solutions for this utterance. Let us consider the i^{th} phoneme of this utterance. A valid solution can have this phoneme participating in a segment of length 2, 3, ... up to m. Analysing further, a segment of length 2 can start at 2 possible offsets, $i-1$ or i. Similarly, a segment of length 3 can start at 3 possible offsets, and so on. In general, let of_j denote the number of offsets at which a segment of length j may start for the i^{th} phoneme. Then, $of_j \leq j$ for $j \in \{2, 3, \ldots, m\}$. Every such offset k for a segment of length j defines a set with aligned segments (k, z_l) such that $|z_l| = j$. Thus, for the i^{th} phoneme, an upper bound on the number N_{ss_i} of possible sets is:

$$\begin{aligned} N_{ss_i} &\leq of_2 + of_3 + \cdots + of_m \\ &\leq 2 + 3 + \cdots + m \\ &< \frac{m(m+1)}{2} \end{aligned} \tag{3}$$

For each of these N_{ss_i} sets, the possible number of segments depends on the sandhi rules R. For any segment in such a set, permutations are possible only at the first and last phonemes because of the sandhi rules applied at the junction. Let $left_w$ denote the number of possible v's, such that $u|v \rightarrow w \in R$ for an arbitrary u. Similarly, let $right_w$ denote the number of possible u's, such that $u|v \rightarrow w \in R$ for an arbitrary v. Now let *maxleft* be the maximum of all such $left_w$ and *maxright* be the maximum of all such $right_w$. Thus, such a set can contain at most $|ss_i| = (maxleft \times maxright)$ segments. Then, the maximum number of

segments N_i that the i^{th} phoneme can participate in is:

$$N_i \leq |ss_i| \cdot N_{ss_i} \tag{4}$$

Now that we have the maximum number of possible segments for the phoneme at position i, we can use this to obtain an upper bound on the number of segments $|\mathscr{S}|$ for the utterance s. We will use the fact that the set $|\mathscr{S}|$ will be a subset of all the possible segments in which phonemes at various positions can participate. Thus

$$\begin{aligned} |\mathscr{S}| &\leq N_1 \times N_2 \times \cdots \times N_n \\ &= (|ss_i|)^n \cdot {N_{ss_i}}^n \\ &= \left(C \cdot \frac{m(m+1)}{2}\right)^n \end{aligned} \tag{5}$$

Similarly, an upper bound on the number of segments in the tabulated display is the sum of all possible segments at various positions. Thus

$$\begin{aligned} |D(\mathscr{S})| &\leq N_1 + N_2 + \cdots + N_n \\ &= \left(C \cdot \frac{m(m+1)}{2}\right) \cdot n \end{aligned} \tag{6}$$

Hence, it follows from Equations 5 and 6 that $|\mathscr{S}|$ is of asymptotic order $O(C^n)$ at worst, whereas $|D(\mathscr{S})|$ is of asymptotic order $O(n)$.

REFERENCES

Kenneth R. Beesley and Lauri Karttunen (2003), *Finite-state morphology: Xerox tools and techniques*, CSLI Publications, The University of Chicago Press.

Sylvie Billot and Bernard Lang (1989), The structure of shared forests in ambiguous parsing, in *Proceedings of the 27th annual meeting on Association for Computational Linguistics*, ACL '89, pp. 143–151, Association for Computational Linguistics, Stroudsburg, PA, USA, doi:10.3115/981623.981641.

Keh-Jiann Chen and Shing-Huan Liu (1992), Word identification for Mandarin Chinese sentences, in *Proceedings of the 14th conference on Computational linguistics-Volume 1*, pp. 101–107, Association for Computational Linguistics.

Jay Earley (1983), An efficient context-free parsing algorithm (reprint), *Communications of the ACM - Special 25th Anniversary Issue*, 26(1):57–61.

Pawan GOYAL, Vipul ARORA, and Laxmidhar BEHERA (2009), Analysis of Sanskrit text: Parsing and semantic relations, in Gérard HUET, Amba KULKARNI, and Peter SCHARF, editors, *Sanskrit Computational Linguistics 1 & 2*, pp. 200–218, Springer-Verlag LNAI 5402.

Pawan GOYAL and Gérard HUET (2013), Completeness analysis of a Sanskrit reader, in Malhar KULKARNI, editor, *Recent Researches in Sanskrit Computational Linguistics (Proceedings, 5th International Symposium on Sanskrit Computational Linguistics)*, pp. 130–171, D.K. Printworld.

Pawan GOYAL, Gérard HUET, Amba KULKARNI, Peter SCHARF, and Ralph BUNKER (2012), A distributed platform for Sanskrit processing, in *COLING*, pp. 1011–1028.

Oliver HELLWIG (2009), SanskritTagger, a stochastic lexical and POS tagger for Sanskrit, in Gérard HUET, Amba KULKARNI, and Peter SCHARF, editors, *Sanskrit Computational Linguistics 1 & 2*, pp. 266–277, Springer-Verlag LNAI 5402.

Gérard HUET (2005), A functional toolkit for morphological and phonological processing, application to a Sanskrit tagger, *Journal of Functional Programming*, 15,4:573–614, `http://yquem.inria.fr/~huet/PUBLIC/tagger.pdf`.

Gérard HUET (2006), Lexicon-directed segmentation and tagging of Sanskrit, in Bertil TIKKANEN and Heinrich HETTRICH, editors, *Themes and Tasks in Old and Middle Indo-Aryan Linguistics*, pp. 307–325, Motilal Banarsidass.

Gérard HUET (2007), Shallow syntax analysis in Sanskrit guided by semantic nets constraints, in *Proceedings of the 2006 International Workshop on Research Issues in Digital Libraries*, pp. 6:1–6:10, ACM, New York, NY, USA, doi:http://doi.acm.org/10.1145/1364742.1364750, `http://yquem.inria.fr/~huet/PUBLIC/IWRIDL.pdf`.

Gérard HUET (2009), Formal structure of Sanskrit text: Requirements analysis for a mechanical Sanskrit processor, in Gérard HUET, Amba KULKARNI, and Peter SCHARF, editors, *Sanskrit Computational Linguistics. First and Second International Symposia Rocquencourt, France, October 29-31, 2007 Providence, RI, USA, May 15-17, 2008*, pp. 162–199, Springer.

Gérard HUET and Pawan GOYAL (2013), Design of a lean interface for Sanskrit corpus annotation, in *Proceedings of ICON 2013, the 10th International Conference on NLP*, pp. 177–186.

Gérard HUET and Amba KULKARNI (2014), Sanskrit linguistics web services, in *COLING (Demo)*, pp. 48–51.

Gérard HUET, Amba KULKARNI, and Peter SCHARF, editors (2009), *Sanskrit computational linguistics 1 & 2*, Springer-Verlag LNAI 5402.

Gérard HUET and Benoît RAZET (2015), Computing with relational machines, *Mathematical Structures in Computer Science*, FirstView:1–20, ISSN 1469-8072, doi:10.1017/S0960129515000390, `http://journals.cambridge.org/article_S0960129515000390`.

Girish Nath JHA, editor (2010), *Sanskrit computational linguistics 4*, Springer-Verlag LNAI 6465.

Ronald M. KAPLAN and Martin KAY (1994), Regular models of phonological rule systems, *Computational Linguistics*, 20,3:331–378.

Amba KULKARNI and Gérard HUET, editors (2009), *Sanskrit computational linguistics 3*, Springer-Verlag LNAI 5406.

Amba KULKARNI, Sheetal POKAR, and Devanand SHUKL (2010), Designing a constraint based parser for Sanskrit, in Girish N. JHA, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*, pp. 70–90, Springer-Verlag LNAI 6465.

Amba KULKARNI and K. V. RAMAKRISHNAMACHARYULU (2013), Parsing Sanskrit texts: Some relation specific issues, in Malhar KULKARNI, editor, *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium*, pp. 191–212, D. K. Printworld(P) Ltd.

Amba KULKARNI and Devanand SHUKL (2009), Sanskrit morphological analyser: Some issues, *Indian Linguistics*, 70(1-4):169–177.

Anil KUMAR, Vipul MITTAL, and Amba KULKARNI (2010), Sanskrit compound processor, in Girish N. JHA, editor, *Proceedings of the 4th International Sanskrit Computational Linguistics Symposium*, pp. 57–69, Springer-Verlag LNAI 6465.

Monier MONIER-WILLIAMS, Ernst LEUMANN, and Carl CAPPELLER (1899), *A Sanskrit-English Dictionary: Etymological And philologically arranged with special reference to cognate Indo-European languages*, Oxford, The Clarendon Press, `http://www.sanskrit-lexicon.uni-koeln.de/scans/csldoc/dictionaries/mw.html`.

Emmanuel ROCHE and Yves SCHABES (1997), *Finite-State Language Processing*, MIT Press.

Alexander M. RUSH, David SONTAG, Michael COLLINS, and Tommi JAAKKOLA (2010), On dual decomposition and linear programming relaxations for natural language processing, in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 1–11, Association for Computational Linguistics.

Peter SCHARF, Anuja AJOTIKAR, Sampada SAVARDEKAR, and Pawan GOYAL (2015), Distinctive features of poetic syntax preliminary results, *Sanskrit syntax*, pp. 305–324.

Peter SCHARF and Malcolm HYMAN (2009), *Linguistic issues in encoding Sanskrit*, Motilal Banarsidass.

Andreas STOLCKE (1995), An efficient probabilistic context-free parsing algorithm that computes prefix probabilities, *Computational Linguistics*, 21(2):165–201.

Weiwei SUN (2010), Word-based and character-based word segmentation models: Comparison and combination, in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 1211–1219, Association for Computational Linguistics.

Xu SUN, Yaozhong ZHANG, Takuya MATSUZAKI, Yoshimasa TSURUOKA, and Jun'ichi TSUJII (2009), A discriminative latent variable Chinese segmenter with hybrid word/character information, in *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 56–64, Association for Computational Linguistics.

Masaru TOMITA (1985), *Efficient parsing for natural language: A fast algorithm for practical systems*, The Springer International Series in Engineering and Computer Science - Volume 8, Springer.

Huihsin TSENG (2005), A conditional random field word segmenter, in *Fourth SIGHAN Workshop on Chinese Language Processing*.

Mengqiu WANG, Rob VOIGT, and Christopher D. MANNING (2014), Two knives cut better than one: Chinese word segmentation with dual decomposition, in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), Baltimore, MD*.

Yue ZHANG and Stephen CLARK (2007), Chinese segmentation with a word-based perceptron algorithm, in *Annual Meeting of the Association for Computational Linguistics*, volume 45, p. 840.

Representing syntax by means of properties: a formal framework for descriptive approaches

Philippe Blache
CNRS & Aix-Marseille Université
Laboratoire Parole et Langage

ABSTRACT

Linguistic description and language modelling need to be formally sound and complete while still being supported by data. We present a linguistic framework that bridges such formal and descriptive requirements, based on the representation of syntactic information by means of *local properties*. This approach, called *Property Grammars*, provides a formal basis for the description of specific characteristics as well as entire constructions. In contrast with other formalisms, all information is represented at the same level (no property playing a more important role than another) and independently (any property being evaluable separately). As a consequence, a syntactic description, instead of a complete hierarchical structure (typically a tree), is a set of multiple relations between words. This characteristic is crucial when describing unrestricted data, including spoken language. We show in this paper how local properties can implement any kind of syntactic information and constitute a formal framework for the representation of *constructions* (seen as a set of interacting properties). The *Property Grammars* approach thus offers the possibility to integrate the description of local phenomena into a general formal framework.

Keywords: syntax, constraints, linguistic theory, usage-based theories, constructions, Property Grammars

1 INTRODUCTION

The description and modelling of local language phenomena contributes to a better understanding of language processing. However,

this data-driven perspective needs to provide a method of unifying models into a unique and homogeneous framework that would form an effective theory of language. Reciprocally, from the formal perspective, linguistic theories provide general architectures for language processing, but still have difficulty in integrating the variability of language productions. The challenge at hand is to test formal frameworks using a large range of unrestricted and heterogeneous data (including spoken language). The feasibility of this task mainly depends on the ability to describe all possible forms, regardless of whether they are well-formed (i.e. grammatical) or not. Such is the goal of the linguistic trend known as *usage-based* (Langacker 1987; Bybee 2010), which aims to describe how language works based on its concrete use. Our goal is to propose a new formal framework built upon this approach.

Moving away from the generative framework. Addressing the question of the syntactic description independently of grammaticality represents an epistemological departure from the generative approach in many respects. In particular, it consists in moving away from the representation of competence towards that of performance. Several recent approaches in line with this project consider grammar not as a device for generating language, but rather as a set of statements, making it possible to describe any kind of input, addressing at the same time the question of gradience in grammars (Aarts 2004; Blache and Prost 2005; Fanselow *et al.* 2005). To use a computational metaphor, this comes to replace a *procedural approach* where grammar is a set of operations (rules), with a *declarative approach* where grammar is a set of descriptions. This evolution is fundamental: it relies on a clear distinction between linguistic knowledge (the grammar) and parsing mechanisms that are used for building a syntactic structure. In most current formalisms, this is not the case. For example, the representation of syntactic information with trees relies on the use of phrase-structure rules which encode both a syntactic relation (government) and operational information (the local tree to be used in the final structure). Such merging of operational information within the grammar can also be found in other formalisms such as *Tree-Adjoining Grammars* (Joshi *et al.* 1975) in which the grammar is made of sub-parts of the final syntactic tree. It is also the case in *Dependency Grammars* (Tesnière 1959) with the projectivity principle (intended to control tree well-

formedness) as well as in HPSG (Pollard and Sag 1994; Sag and Wasow 1999) and its feature percolation principles.

We propose disentangling these different aspects by excluding information solely motivated by the kind of structure to be built. In other words, linguistic information should be encoded independently of the form of the final representation. Grammar is limited then to a set of descriptions that are linguistic facts. As explained by Pullum and Scholz (2001), doing this enables a move away from *Generative-Enumerative Syntax* (GES) towards a *Model-Theoretic Syntax* (MTS) (Cornell and Rogers 2000; Blackburn and Meyer-Viol 1997; Blache 2007).

Several works are considered by Pullum and Scholz (2001) to exhibit the seeds of MTS, in particular *HPSG* and *Construction Grammars* (Fillmore 1988; Kay and Fillmore 1999). These two approaches have recently converged, leading to a new framework called *Sign-Based Construction Grammars* (Sag 2012; Sag *et al.* 2012). SGBG is motivated by providing a formal basis for *Construction Grammars*, paving the way towards modelling language usage. It starts to fulfill the MTS requirements in that it proposes a monotonic system of declarative constraints, representing different sources of linguistic information and their interaction. However, there still remains a limitation that is inherent to HPSG: the central role played by *heads*. Much information is controlled by this element, as the theory is *head-driven*. All principles are stipulated on the basis of the existence of a context-free skeleton, implemented by dominance schemas. As a consequence, the organization of the information is *syntacto-centric*: the interaction of the linguistic domains is organized around a head/dependent hierarchical structure, corresponding to a tree.

In these approaches, representing the information of a domain, and more to the point the interaction among the domains, requires one to first build the schema of mother/daughters. Constraints are then applied as filters, so as to identify well-formed structures. As a side effect, no description can be given when no such structures can be built. This is a severe restriction both for theoretical and cognitive reasons: one of the requirements of MTS is to represent all linguistic domains independently of each other (in what Pullum and Scholz 2001 call a *non-holistic* manner). Their interaction is to be implemented directly, without giving any priority to any of them with respect to the others. Ignoring this requirement necessarily entails a modular and se-

rial conception of language processing, which is challenged now both in linguistics and in psycholinguistics (Jackendoff 2007; Ferreira and Patson 2007; Swets *et al.* 2008). Evidence supporting this challenge includes: language processing is very often underspecified; linguistic information comes from different and heterogeneous sources that may vary depending on usage; the understanding mechanisms are often non-compositional; etc.

One goal of this paper is to propose an approach that accommodates such different uses of language so as to be able to process canonical or non-canonical, mono- or multimodal inputs.

Describing any kind of input. Linguistic information needs to be represented separately when trying to account for unrestricted material, including non-canonical productions (e.g. in spoken language). The main motivation is that, whatever the sentence or the utterance to be parsed, it becomes then possible to identify its syntactic characteristics independently of the structure to be built. If we adopt this approach, we still can provide syntactic information partly describing the input even when no structure can be built (e.g. ill-formed realizations). In other words, it becomes possible to provide a description (in some cases a partial description) of an input regardless of its form.

This type of approach allows one to describe any type of sentence or utterance: it is no longer a question of establishing whether the sentence under question is grammatical or not, but rather of describing the sentence itself. This task amounts to deciding which descriptions present in the grammar are relevant to the object to be described and then to assessing them.

Grammar as set of constructions. One important advance for linguistic theories has been the introduction of the notion of *construction* (Fillmore 1988; Kay and Fillmore 1999). A construction is the description of a specific linguistic phenomenon, leading to a specific form-function pairing that is conventionalized or even not strictly predictable from its component parts (Goldberg 2003, 2009). These pairings result from the convergence of several properties or characteristics, as illustrated in the following examples:

1. Covariational conditional construction
   `The Xer the Yer`: "*The more you watch the less you know*"

2. Ditransitive construction
   `Subj V Obj1 Obj2`: "*She gave him a kiss*"
3. Idiomatic construction: "*kick the bucket*"

Several studies and new methodologies have been applied to syntactic description in the perspective of modelling such phenomena (Bresnan 2007). The new challenge is to integrate these constructions, which are the basic elements of usage-based descriptions, into a homogeneous framework of a grammar. The problem is twofold: first, how to represent the different properties characterizing a construction; and second, how to represent the interaction between these properties in order to form a construction.

Our proposal. We seek an approach where grammars comprise usage-based descriptions. A direct consequence is to move the question away from building a syntactic structure to describing the characteristics of an input. Specifically, grammatical information should be designed in terms of statements that are not conceived of with the aim of building a structure.

We propose a presentation of a theoretical framework that integrates the main requirements of a *usage-based* perspective. Namely, it first integrates constructions into a grammar and secondly describes non-grammatical exemplars. This approach relies on a clear distinction of operational and declarative aspects of syntactic information. A first step in this direction has been achieved with *Property Grammars* (Blache 2000; Blache and Prost 2014), in which a grammar is only made of properties, all represented independently of each other. *Property Grammars* offer an adequate framework for the description of linguistic phenomena in terms of interacting properties instead of structures. We propose going one step further by integrating the notion of construction into this framework. One of the contributions of this paper, in comparison to previous works, is a formal specification of the notion of construction based on constraints only, instead of structures as in SBCG. It proposes moreover a computational method for recognizing them.

In the first section, we present a formal definition of the syntactic properties; these are used for describing any type of input. We then discuss more theoretical issues that constitute obstacles when trying to represent basic syntactic information independently of the rest of the

grammar.[1] We explore in particular the consequences of representing relations between words directly, without the mediating influence of any higher-level structures or elements (i.e. without involving the notion of phrases or heads). Last, we describe how this framework can incorporate the notion of construction and detail its role in the parsing process.

2 NEW PROPERTIES FOR GRAMMARS

We seek to abstract the different types of properties that encode syntactic information. As explained above, we clearly separate the representation of such information from any pre-defined syntactic structure. In other words, we encode this information by itself, and not in respect to any structure: a basic syntactic property should not be involved in the building of a syntactic structure. It is thus necessary to provide a framework that excludes any notion of hierarchical information, such as heads or phrases: a property is a relation between two words, nothing more. Disconnecting structures and relations is the key towards the description of any kind of input as well as any type of construction.

Unlike most syntactic formalisms, we limit grammar to those aspects that are purely descriptive, excluding *operational* information. Here, the grammatical information as well as the structures proposed for representing syntactic knowledge are not determined by how they may be used during analysis. We want to avoid defining (e.g. as in constituency-based grammars) a phrase-structure rule as a step in the derivational process (corresponding to a sub-tree). In this case, the notions of projection and sisterhood eclipse all other information (linear order, co-occurrence, etc.), which becomes implicit. Likewise, in *dependency grammars*, a dependency relation corresponds to a branch on the dependency tree. In this context, subcategorization or modification information becomes dominant and supersedes other information which, in this case too, generally becomes implicit. This issue also affects modern formalisms, such as HPSG (Pollard and Sag 1994; Sag and Wasow 1999; Sag 2012) which, strictly speaking does not use

[1] Pullum and Scholz (2001) emphasize this characteristic as a requirement for moving away from the holistic nature of generative grammars.

phrase-structure rules but organizes syntactic information by means of principles in such a way that it has to percolate through the heads, building as a side-effect a tree-like structure.

Our approach, in the context of *Property Grammars* (hereafter PG) consists in identifying the different types of syntactic information in order to represent them separately. At this stage, we will organize grammatical statements around the following types of syntactic information:

- the *linear order* that exists among several categories in a construction
- the *mandatory co-occurrence* between two categories
- the *exclusion of co-occurrence* between two categories
- the impossibility of *repeating* a given category
- syntactic-semantic *dependency* between two categories (generally a category and the one that governs it)

This list of information is neither fixed nor exhaustive and could be completed according to the needs of the description of specific languages, for example with adjacency properties, completing linearity, or morphological dependencies.

Following previous formal presentations of *Property Grammars* (Duchier *et al.* 2010; Blache and Prost 2014) we propose the following notations: x, y (lower case) represent individual variables; X, Y (upper case) are set variables. We note $C(x)$ the set of individual variables in the domain assigned to the category C (cf. Backofen *et al.* (1995) for more precise definitions). We use the binary predicates $\prec$ and $\approx$ respectively for linear precedence and equality.

2.1 *Linearity*

In PG, word order is governed by a set of linearity constraints, which are based on the clause established in the ID/LP formalism (Gazdar *et al.* 1985). Unlike phrase-structure or dependency grammars, this information is, therefore, explicit. The linearity relationship between two categories is expressed as follows ($pos(x)$ being the function returning the position of x in the sentence):

$$\textit{Prec}(A, B) : (\forall x, y)[(A(x) \wedge B(y) \rightarrow pos(x) < pos(y))] \qquad (1)$$

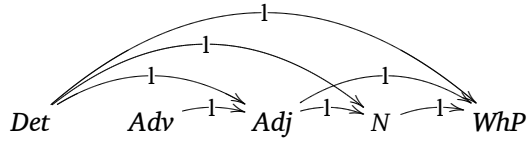
This is the same kind of linear precedence relation as proposed in GPSG (Gazdar *et al.* 1985). If the nodes x and y, respectively of category A and B, are realized,[2] then y cannot precede x.

For example, in a nominal construction in English, we can specify the following linearity properties:

$$Det \prec Adj; \quad Det \prec N; \quad Adj \prec N; \quad N \prec WhP; \quad N \prec Prep \qquad (2)$$

Note that, in this set of properties, relations are expressed directly between the lexical categories (the notion of phrase-structure category is no longer used). As such, the $N \prec Prep$ property indicates precedence between these two categories regardless of their dependencies. This aspect is very important and constitutes one of the major characteristics of PG: all properties can be applied to any two items, including when no dependency or subcategorization link them.

The following example illustrates all the linearity relationships in the nominal construction "*The very old reporter who the senator attacked*" (the relative clause is not described here):



(3)

In this example, the linearity properties between two categories are independent of the *rection* (government) relations that these categories are likely to have. The linearity between *Det* and *Adj* holds even if these two categories have other dependencies (for example between the *Adj* and a modifier such as *Adv*). In theory, it could even be possible that a word dependent from the second category of the relation is realized before the first one: as such, there is no projectivity in these relations.[3] The same situation can be found for non-arguments: a linearity can be directly stipulated for example between a negative adverb and a verb. This is an argument in favour of stipulating properties directly between lexical categories rather than using phrase-structures.

[2] A word or a category is said to be *realized* when it occurs in the sentence to be parsed.

[3] Such a phenomenon does not exist in languages with fixed word order such as English or French.

In addition to the representation of syntactic relations, properties may be used to instantiate attribute values. For example, we can distinguish the linearity properties between the noun and the verb, depending on whether N is *subject* or *object* by specifying this value in the property itself:

$$N_{[subj]} \prec V; \qquad V \prec N_{[obj]} \tag{4}$$

As we shall see, all properties can be used to instantiate certain attribute values. As is the case in *unification grammars*, attributes can be used to reduce the scope of a property by limiting the categories to which it can be applied. Generally speaking, a property (playing the role of a constraint) has a dual function: control (limiting a definition domain) and instantiation (assigning values to variables, by unification).

2.2 *Co-occurrence*

In many cases, some words or categories must co-occur in a domain, which is typically represented by subcategorization properties. For example, the transitive schema for verbs implies that a nominal object (complement) must be included in the structure. Such co-occurrence constraint between two categories x and y specifies that if x is realized in a certain domain, then y must also be included. This is formally represented as follows:

$$Req(A, B) : (\forall x)[A(x) \rightarrow \exists y B(y)] \tag{5}$$

If a node x of category A is realized, so too is a node y of category B. The co-occurrence relation is not symmetric.

As for verbal constructions, a classical example of co-occurrence concerns nominal and prepositional complements of ditransitive verbs, which are represented through the following properties:

$$V \Rightarrow N; \qquad V_{[dit]} \Rightarrow Prep \tag{6}$$

As described in the previous section, a property is stipulated over lexical categories, independently of their dependents and their order.

Co-occurrence represents not only complement-type relations; it can also include co-occurrence properties directly between two categories independently from the head (thus regardless of rection rela-

tions). For example, the indefinite determiner is not generally used with a comparative superlative:[4]

(1) a. *The most interesting book of the library*
b. **A most interesting book of the library*

In this case, there is a co-occurrence relation between the determiner and the superlative, which is represented by the property:

$$Sup \Rightarrow Det_{[def]} \tag{7}$$

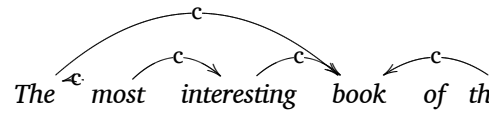
Furthermore, this example shows that we can also specify variable granularity properties by applying general or more specific categories by means of attribute values.

A key point must be emphasized when using co-occurrence properties: the notion of head does not play a preponderant role in our approach. Moreover, we do not use sets of constituents within which, in constituency-based grammar, the head is distinct and indicates the type of projection. Classically in syntax, the head is considered to be the governing category, which is also the minimum mandatory component required to create a phrase. This means that the governed components must be realized together with the head. As such, this information is represented by properties establishing co-occurrence between the head and its complements. Defining a specific property that identifies the head is, therefore, not necessary.

In the case of nominal construction, the fact that N is a mandatory category is stipulated by a set of co-occurrence properties between the complements and the adjuncts to the nominal head:

$$Det \Rightarrow N_{[common]}; \quad Adj \Rightarrow N; \quad WhP \Rightarrow N; \quad Prep \Rightarrow N \tag{8}$$

The set of co-occurrence properties for the nominal construction described so far can be represented by the following graph:



(9)

[4] This constraint is limited to comparative superlatives. In some cases the use of an indefinite determiner entails a loss of this characteristic. In the sentence "*In the crowd, you had a former fastest man in the world.*" the superlative becomes absolute, identifying a set of elements instead of a unique one.

We shall see later how the conjunction between co-occurrence and dependency properties is used to describe the syntactic characteristics of a head, without the need for other types of information. As such (unlike previous versions of PG), using specific properties for describing the head is not required.

At this stage, we can note that different solutions exist for representing non-headed constructions, for example when no noun is realized in a nominal construction. As we will see later, all constraints are violable. This means that a nominal construction without a noun such as in "*The very rich are different from you and me*" can be described with a violation of the co-occurrence properties stipulated above. This comes to identify a kind of implicit relation, not to say an empty category. Another solution consists in considering the adjective as a possible head of the nominal construction. In such a case, the grammar should contain another set of co-occurrence and dependency properties that are directly stipulated towards the adjective instead of the noun.

2.3 *Exclusion (co-occurrence restriction)*

In some cases, restrictions on the possibilities of co-occurrence between categories must be expressed. These include, for example, cases of lexical selection, concordance, etc. An exclusion property is defined as follows:

$$Excl(A, B) : (\forall x)(\nexists y)[A(x) \wedge B(y)] \tag{10}$$

When a node x of category A exists, a sibling y of category B cannot exist. This is the *exclusion* relation between two constituents, that corresponds to the co-occurrence restriction in GPSG. The following properties show a few co-occurrence restrictions between categories that are likely to be included in nominal constructions:

$$Pro \otimes N; \qquad N_{[prop]} \otimes N_{[com]}; \qquad N_{[prop]} \otimes Prep_{[\text{inf}]} \tag{11}$$

These properties stipulate that, in a nominal construction, the following co-occurrences cannot exist: a pronoun and a noun; a proper noun and a common noun; a proper noun and an infinitive construction introduced by a preposition.

Likewise, relative constructions can be managed based on the syntactic role of the pronoun. A relative construction introduced by a subject relative pronoun, as indicated in the following property, cannot

contain a noun with this same function. This restriction is compulsory in French, where relative pronouns are case marked:

$$WhP_{[subj]} \otimes N_{[subj]} \tag{12}$$

It is worth noting that a particularity of this type of property is that it can only be verified when the entire government domain (i.e. a head and its complements/adjuncts) is known. We will discuss later the different cases of constraint satisfiability, which depend on their scope.

2.4 *Uniqueness*

Certain categories cannot be repeated inside a rection domain. More specifically, categories of this kind cannot be instantiated more than once in a given domain. This property is defined as follows:

$$Uniq(A) : (\forall x, y)[A(x) \wedge A(y) \rightarrow x \approx y] \tag{13}$$

If one node x of category A is realized, other nodes y of the same category A cannot exist. Uniqueness stipulates that constituents cannot be replicated in a given construction. Uniqueness properties are common in domain descriptions, although their importance depends upon the constructions to which they belong. The following example describes the uniqueness properties for nominal constructions:

$$Uniq = \{Det,\ Rel,\ Prep_{[inf]},\ Adv\} \tag{14}$$

These properties are well established for the determiner and the relative pronoun. They also specify here that it is impossible to replicate a prepositional construction that introduces an infinitive ("*the will to stop*") or a determinative adverbial phrase ("*always more evaluation*").

Uniqueness properties are represented by a loop:

(15)

2.5 *Dependency*

The dependency relation in *PG* is in line with the notion of syntactic-semantic dependency defined in *Dependency Grammars*. It describes

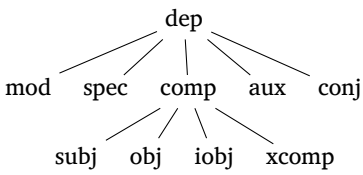


Figure 1:
The hierarchy of the *dependency* relation

dep	generic relation, indicating dependency between a constructed component and its governing component
mod	modification relation (typically an adjunct)
spec	specification relation (typically *Det-N*)
comp	the most general relation between a head and an object (including the subject)
subj	dependency relation describing the subject
obj	dependency relation describing the direct object
iobj	dependency relation describing the indirect object
xcomp	other types of complementation (for example between *N* and *Prep*)
aux	relation between the auxiliary and the verb
conj	conjunction relation

Table 1:
The sub-types of the *dependency* relation

different types of relations between two categories (complement, modifier, specifier, etc.). In terms of representation, this relation is arbitrarily oriented from the dependent to the head. It indicates the fact that a given object complements the syntactic organization of the target (usually the governor) and contributes to its semantic structure. In this section, we we leave aside semantics and focus on the syntactic aspect of the dependency relation.

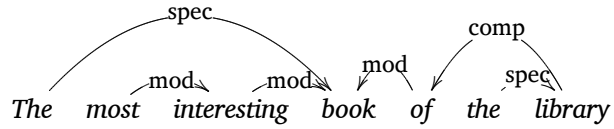
Dependency relations are type-based and follow a type hierarchy (Figure 1); note that this hierarchy can be completed according to requirements of specific constructions or languages.

Since the dependency relation is a hierarchy, it is possible to use in a description one of these types, from the most general to the most specific, depending on the required level of precision. Each of these types and/or sub-types corresponds to a classic syntactic relation (Table 1).

Dependency relations (noted $\rightsquigarrow$) possibly bear the dependency sub-type as an index. The following properties indicate the dependency properties applied to nominal constructions:

$$Det \rightsquigarrow_{spec} N_{[com]}; \quad Adj \rightsquigarrow_{mod} N; \quad WhP \rightsquigarrow_{mod} N \tag{16}$$

The following example illustrates some dependencies into a nominal construction:



(17)

In this schema, we can see the specification relations between the determiners and the corresponding nouns, and the modification relations between the adjectival and prepositional constructions as well as between the adverb and the adjective inside the adjectival construction.

Feature control: The types used in the dependency relations, while specifying the relation itself, also provide information for the dependent element. In PG, the dependency relation also assigns a value to the FUNCTION attribute of the dependent. For example, a *subject* dependency between a noun and a verb is expressed by the following property:

$$N_{[subj]} \rightsquigarrow_{subj} V \tag{18}$$

This property instantiates the function value in the lexical structure [FUNCTION *subject*]. Similarly, dependency relations (as it is also the case for properties) make it possible to control attribute values thanks to unification. This is useful, for example, for agreement attributes that are often linked to a dependency. For instance, in French, a gender and number agreement relation exists between the determiner, the adjective and the noun. This is expressed in the following dependencies:

$$Det_{[agr_i]} \rightsquigarrow_{spec} N_{[agr_i]}; \quad Adj_{[agr_i]} \rightsquigarrow_{mod} N_{[agr_i]} \tag{19}$$

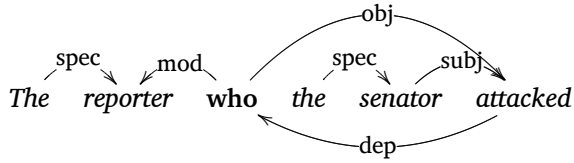
Formal aspects: Unlike dependency grammars, this dependency relation is not strict. First of all, as the dependencies are only a part of the syntactic information, a complete dependency graph connecting all the categories/words in the sentence is not required. Moreover, dependency graphs may contain cycles: certain categories may have dependency relations with more than one component. This is the case,

for example, in relative constructions: the relative pronoun depends on the main verb of the construction (a complementation relation with the verb of the relative, regardless whether it is the subject, direct object, or indirect object). But it is also a dependent of the noun that it modifies.

In PG, a cycle may also exist between two categories. Again, this is the case in the relative construction, between the verb and the relative pronoun. The relative pronoun is a complement of the main verb of the relative. It is also the target of the dependency relation originating from the verb. This relation indicates that the verb (and its dependencies) will play a role in establishing the sense of the relative construction. In this case, the dependency relation remains generic (at the higher level of the type hierarchy). The dependency properties of the relative construction stipulate:

$$\textit{WhP}_{[comp]} \rightsquigarrow_{comp} V; \qquad \textit{WhP} \rightsquigarrow_{mod} N; \qquad V \rightsquigarrow_{dep} \textit{WhP} \tag{20}$$

It should be noted that the dependency relation between *WhP* and *V* bears the *comp* type. This generic type will be specified in the grammar by one of its sub-types *subj*, *obj* or *iobj*, each generating different properties (in particular exclusion) for the relative. The following schema illustrates an example of a relative construction, with two particularities (the double dependency for the *WhP*, and the cycle between *WhP* and *V*):



(21)

As we can see, the dependency graph in PG (as with the other properties) is not necessarily connected or cycle-free. Table 2 summarizes the main characteristics of the dependency relation.

It should be noted that these relations are stipulated taking into account the precise type of the dependency relations: they are true

Antisymmetric:	if $A \rightsquigarrow_x B$, then $B \not\rightsquigarrow_x A$
Antireflexive:	if $A \rightsquigarrow B$, then $A \neq B$
Antitransitive:	if $A \rightsquigarrow_x B$ and if $B \rightsquigarrow_x C$ then $A \not\rightsquigarrow_x C$

Table 2: Characteristics of the *dependency* relation

only for a given type, but not as a general rule. For example, a symmetric complementation relation cannot exist (if A is a complement of B, then B cannot be a complement of A). However, a cycle can appear when the dependency types are different (as seen above for $V - WhP$ dependencies).

Apart from the type-based restrictions, properties are identical to those found in dependency grammars. The main difference in PG is that the dependency graph is not necessarily connected and does not necessarily have a unique root.

Furthermore, we can see that when two realized categories (i.e. each corresponding to a word in the sentence) are linked by a property, they are usually in a dependency relation, directly or otherwise. Formally speaking, this characteristic can be expressed as follows:

Let $\mathscr{P}$ be a relation expressing a PG property, let x, y and z be categories:

$$\text{If } x \mathscr{P} y, \text{ then } x \rightsquigarrow y \vee y \rightsquigarrow x \vee [\exists z \text{ such that } x \rightsquigarrow z \wedge y \rightsquigarrow z] \quad (22)$$

Finally, dependency relations comprise two key constraints, ruling out some types of dual dependencies:

- A given category cannot have the same type of dependency with several categories[5]:

$$\text{If } x \rightsquigarrow_{dep_i} y, \text{ then } \nexists z \text{ such that } y \not\approx z \wedge x \rightsquigarrow_{dep_i} z \quad (23)$$

  Example : $Pro_i \rightsquigarrow_{subj} V_j$; $Pro_i \rightsquigarrow_{subj} V_k$
  The same pronoun cannot be subject of two different verbs.

- A given category cannot have two different types of dependencies with the same category:

$$\text{If } x \rightsquigarrow_{dep_i} y, \text{ then } \nexists\, dep_j \neq dep_i \text{ such that } x \rightsquigarrow_{type\_dep_j} y \quad (24)$$

  Example : $Pro_i \rightsquigarrow_{obj} V_j$; $Pro_i \rightsquigarrow_{subj} V_j$
  A given pronoun cannot simultaneously be the subject and object of a given verb.

Note that such restrictions apply for dependencies at the same level in the dependency type hierarchy. In the above example, this is

[5] This constraint is to be relaxed for some phenomena such as coordination, depending on the conjuncts are considered at the same level or not.

$Det \prec \{Det, Adj, WhP, Prep, N\}$	$Det \rightsquigarrow_{spec} N$
$N \prec \{Prep, WhP\}$	$Adj \rightsquigarrow_{mod} N$
$Det \Rightarrow N_{[com]}$	$WhP \rightsquigarrow_{mod} N$
$\{Adj, WhP, Prep\} \Rightarrow N$	$Prep \rightsquigarrow_{mod} N$
$Uniq = \{Pro, Det, N, WhP, Prep\}$	$Pro \otimes \{Det, Adj, WhP, Prep, N\}$
	$N_{[prop]} \otimes Det$

Table 3: Properties of the nominal construction

the case for *subj* and *obj*: such dual dependency cannot exist. Also note that these constraints do not rule out licit double dependencies such as that encountered in control phenomena (the same subject is shared by two verbs) or in the case of the relative pronoun which is both the modifier of a noun and the complement of the verb of the relative:

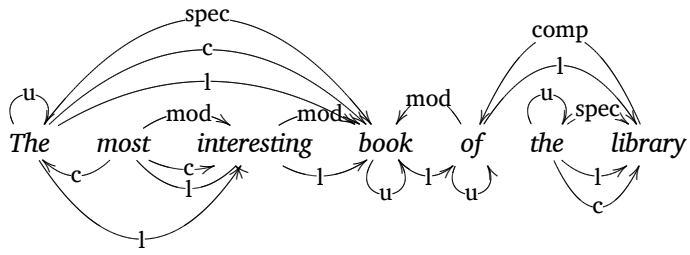
$$WhP \rightsquigarrow_{comp} V; \qquad WhP \rightsquigarrow_{mod} N \tag{25}$$

In this case, the relation types represent dependencies from both inside and outside the relative clause.

2.6 *A comprehensive example*

Each property as defined above corresponds to a certain type of syntactic information. In PG, describing the syntactic units or linguistic phenomena (chunks, constructions) in the grammar consists in gathering all the relevant properties into a set. Table 3 summarizes the properties describing the nominal construction.

In this approach, a syntactic description, instead of being organized around a specific structure (for example a tree), consists in a set of independent (but interacting) properties together with their status (satisfied or violated). The graph in the figure below illustrates the PG description of the nominal construction: "*The most interesting book of the library*".



(26)

In *PG*, a syntactic description is therefore the graph containing all the properties of the grammar that can be evaluated for the sentence to be parsed. As illustrated in the example, this property graph represents explicitly all the syntactic characteristics associated to the input; each is represented independently of the others.

3 BRINGING CONSTRUCTIONS INTO PROPERTY GRAMMARS

A *construction* is defined as the convergence of several properties. For example, the ditransitive construction is, among other features, characterized by the fact that the argument roles are filled by two nominal objects in a specific order. The first step towards the recognition of a construction consists in identifying such basic properties. At this stage, no other process but the spotting of the properties needs to be used. This means that all properties should be identified directly and independently of the rest of the grammar. For example, in the case of the ditransitive construction, this consists in identifying the linear order between the nominal objects.

The issue, then, is to describe such local and basic properties, without relating them to any higher level information. As a consequence, we propose a representation in which all properties are self-contained (as presented in the previous section) in the sense that their evaluation should not depend on the recognition of other elements or structure. However, the two classical means of representing syntactic information (*constituency* or *dependency*) consist either in structuring higher-level groups (phrases in the case of constituency-based grammars) or assigning a specific role to the head in the definition of a branching structure (in the case of dependency grammars). In this section, we explore in greater detail these aspects and their consequences when trying to represent basic properties directly. Our analysis is built around three issues: the notion of syntactic group, the status of the head, and the kind of information to be encoded in the lexicon for the representation of basic properties.

3.1 *Constructions as sets of properties*

Constituency-based approaches rely on the definition of syntactic properties in terms of membership: a syntactic object is characterized

by its set of constituents. This approach offers several advantages in describing the distributional properties of syntactic groups, for example. Moreover, it constitutes a direct framework for controlling the scope of local properties (such as linearity or co-occurrence restriction): they are valid within a *domain* (a phrase).

Using this notion of domain proves interesting for constraint-based frameworks in which a phrase is described by a set of categories to which several constraints apply (offering a direct control of the scope of constraints). However, such an approach requires the organization of syntactic information into two separate types, forming two different levels: on the one hand, the definition of the domain (the set of categories, the phrase) and, on the other hand, their linguistic properties. In terms of representation (in the grammar), this means giving priority to the definition of the domain (the identification of the set of constituents, for example by means of rules or schemas). The constraints come on top of this first level, adding more information. In terms of parsing, the strategy also follows this dual level organization: first recognizing the set of categories (for example *Det, N, Rel,* ... for the *NP*), then evaluating constraint satisfaction.

The problem with this organization is that it gives priority to a certain type of information, namely constituency, that is motivated by operational matters (representation and construction of the syntactic structure) more than by linguistic considerations: sisterhood in itself does not provide much syntactic knowledge or, more precisely, is too vague in comparison with the syntactic properties binding two categories (e.g. co-occurrence, restriction, dependency). Moreover, this organization has a severe drawback: a linguistic description is only possible when the first level (identification of the set of categories) is completed. In other words, it is necessary to build a phrase before being able to evaluate its properties. This approach does not fit with the notion of construction for several reasons. First, a construction is not necessarily composed of adjacent constituents. A constituency-based grammar cannot handle such objects directly. Moreover, constructions can be formed with a variable structure (elements of varying types, non-mandatory elements, etc.), due to the fact that they encode a convergence of different sources of information (phonology, morphology, semantics, syntax, etc.). An organization in terms of constituents relies on a representation driven by syntax, which renders impossible a

description in terms of interaction of properties and domains as is the case with construction-based approaches.

Our goal is to integrate a multi-domain perspective, based on a description in terms of constructions, that is capable of dealing with any kind of input (including ill-formed or non-canonical realizations). We propose a representation of the linguistic information in terms of properties that are all at the same level. In other words, all information needs to be represented in the same manner, without any priority given to one type of information over another. No domain, set of categories or phrase should be built before being able to describe the linguistic characteristics of an input: a linguistic property should be identified directly, independently of any other structure.

As a consequence, properties need to be represented as such in the grammar (i.e. independently of any notion of constituency) and used directly during parsing (i.e. without needing to build a set of categories first). This goal becomes possible provided that the scope of the property is controlled. One way to do this consists in specifying precisely the categories in relation. Two types of information can be used with this perspective: the specification of certain features (limiting the kinds of objects to which the property can be applied), and the use of an HPSG-like category indexing (making it possible to specify when two categories from two properties refer to the same object).

As such, integrating the notion of construction should not make use of the notion of constituency but rather favour a description based on direct relations between words (or lexical categories). Thus, we fall in line with a perspective that is akin to dependency grammars, except for the fact that we intend to use a larger variety of properties to describe the syntax and not focus exclusively on dependency. In the remainder of this section we will present a means of representing constructions only using such basic properties.

3.2 *The question of heads: to have or not to have?*

The notion of head plays a decisive role in most linguistic theories: syntax is usually described in terms of government or dependency between a head and its dependents. In *constituency-based grammars*, the head bears a special relation to its projection (the root of the local tree

it belongs to). In *dependency grammars*, a head is the target of the relations from the depending categories. The role of the head can be even more important in lexicalized theories such as LFG (Bresnan 1982) or HPSG. In this case, the head is also an operational element in the construction of the syntactic structure: it represents the site through which all information (encoded by features) percolates. All exocentric syntactic relations (between a phrase constituent and another component outside this phrase) are expressed as feature values which, as a result of a number of principles, move from the source constituent to the target, passing through the head.

A direct consequence is that when heads play a central role, syntactic information needs to be represented in a strictly hierarchical manner: as the head serves as a gateway, it is also a *reduction* point from which all information relating to the head's dependents may be accessed. Such a strict hierarchical conception of syntax has a formal consequence: the syntactic structure must be represented as a hierarchical (or a tree-like) structure in which every component (word, category, phrase, etc.) is dependent on a higher-level element. Such a syntactic organization is not suited for the description of many phenomena that we come across in *natural* language. For example, many constructions have no overt head:

(2) a. *John sets the red cube down and takes the black*.
b. *First trip, New York*.
c. *Monday, washing, Tuesday, ironing, Wednesday, rest*.

Example (2a) presents a classical elision as part of a conjunction: the second NP has no head. This is also the case in the nominal sentences in examples (2b) and (2c), which correspond to binary structures where each nominal component holds an argumentative position (from the semantic point of view) without a head being realized. We already gave some arguments towards the non-headed construction analysis in the second section. In the case of the last two examples, little information can be given at the syntactic level; it mainly comes from the interaction of morphology, prosody and discourse. The solution in PG (not developed in this paper) consists in implementing interaction constraints for controlling the alignment of properties coming from the different domains (Blache and Prévot 2010).

This raises the issue of structures that can be adapted to the representation of linguistic relations outside the head/dependent relation. The example of collective nouns in French illustrates such a situation:

(3) a. *un ensemble de catégories* (a set of categories)
b. **un ensemble des catégories* (a set of-plu categories)
c. *l'ensemble de catégories* (the set of categories)
d. *l'ensemble des catégories* (the set of-plu categories)

If a collective noun is specified by an indefinite determiner, then the complex category preposition-determiner *de* ("of") – which, in this case, is a partitive – can only be used in its singular form. This construction is controlled by the exclusion property:

$$Det_{[ind]} \otimes \{Prep + Det_{[plu]}\} \tag{27}$$

Inside a nominal construction with a collective noun, we have a direct constraint between the type of determiner (definite or indefinite) and the preposition agreement feature without any mediation of the head. In order to be complete, this property has to be restricted to those determiners which specify a collective noun. This is implemented by a co-indexation mechanism between categories, that is described in section 3.4 below.

Generally speaking, the head plays a fundamental role in specifying the subcategorization or the argument structure. It is not, however, necessary to give it an operational role when constructing the syntactic structure. We shall see that the head, even with no specific role, can be identified only as being the category to which all dependency relations converge.

3.3 *The structure of lexical entries*

As in unification grammars, the lexical information is highly important. Nonetheless, the lexicalization of syntactic information (emphasized in theories such as LFG or HPSG) is more limited in PG. In particular, the lexicon does not play a direct role in the construction of the syntactic structure; rather, all information is borne by the properties. Lexical information, although rich, is only used on the one hand to control the scope of the properties (as described above) and on the other hand to instantiate the subcategorization or the specific dependencies that one category can have with others.

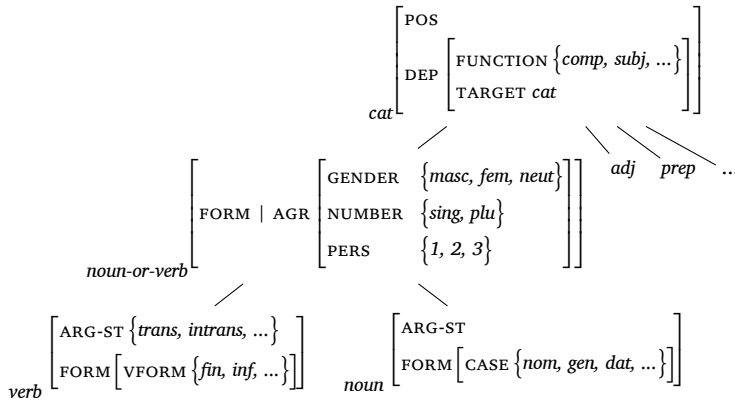


Figure 2: Inheritance in nominal and verbal categories

In general, a lexical entry is associated with an attribute-value matrix which basically contains the category, agreement, morpho-syntactic features, subcategorization list and grammatical function (when relevant). This structure can be enriched with other features, for example those describing semantics, phonology, etc. It can also be completed depending on the category, with more specific information such as mood, tense, person, or the valence feature that gives the list of arguments required.

Figure 2 summarizes the main features of nominal and verbal categories. It represents a type hierarchy, while the subtypes inherit "appropriate" features from the higher-level types.

The most general type, *cat*, comprises features appropriate to the description of all categories: the category label as well as the description of its dependency with other categories. This relation is described by the type of the dependency and the target value of the relation. In the above example, the lower level subtypes describe the features appropriate to *N* and *V*: both categories take agreement. Moreover, the verb has an argument structure which specifies its valence as well as its form attributes. As for the noun, it is associated with case features.

3.4 *The role of features*

Properties are relations between two lexical categories (that may potentially have other dependencies). For example, a linear property such as $V \prec N_{[obj]}$ indicates that the verb precedes the direct object.

This relation holds regardless of the other dependency relations of V and N. However, in this example, specifying the function value is mandatory: without it, the property would not be valid ($V \prec N$ is not licit as such in English).

The instantiation of feature values of a category involved in a property reduces its definition domain and, as a side effect, the scope of the property. Moreover, with all properties being independent of each other, it is necessary to provide as much information as possible to identify precisely the categories to be linked. Representing a property in this way renders them absolute, in the manner of *Optimality Theory* (Prince and Smolensky 1993), in which all constraints are universal. In this approach, a property can be evaluated directly, without needing any knowledge of the context or the rest of the syntactic structure. This condition is imperative when trying to consider a grammar as a set of properties.

We present two series of examples illustrating how feature instantiation helps in controlling the application of a property.

Control by feature values. The specification of feature values in properties can be used in order to describe certain phenomena directly. For example, the argument structure can be described by means of linearity and dependency properties, assigning subcategorization and case feature values:

$$\begin{array}{ll} V \Rightarrow N_{[subj]} & V_{[trans]} \Rightarrow N_{[obj]} \\ V_{[intrans]} \otimes N_{[obj]} & V_{[ditrans]} \Rightarrow N_{[iobj]} \end{array} \tag{28}$$

Likewise, the different possible constructions of the relative in French can be described by specifying the case of the relative pronoun:

$$\begin{array}{ll} WhP_{[nom]} \otimes N_{[subj]} & WhP_{[nom]} \rightsquigarrow_{subj} V \\ WhP_{[acc]} \otimes N_{[obj]} & WhP_{[nom]} \rightsquigarrow_{obj} V \end{array} \tag{29}$$

These properties stipulate that the nominative relative pronoun *qui* (“who”) excludes the possibility to realize a subject within the relative construction and specifies a subject-type dependency relation between the relative pronoun and the verb. The same type of restriction is specified for the accusative pronoun *que* (“which”) and could also be extended to the dative pronoun *dont* (“of which/of whom”). These properties implement the long-distance dependency between *WhP* and the “gap” in the argument structure of the main verb.

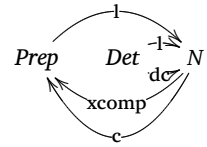
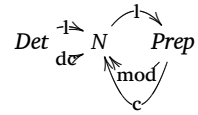
Construction	Properties	Example	Property graph
Prepositional	$Prep \prec N$ $N \rightsquigarrow_{xcomp} Prep$	"*on the table*"	
Nominal	$N \prec Prep$ $Prep \rightsquigarrow_{mod} N$	"*the book on ...*"	

Table 4: Inverse dependencies between *Prep* and *N*

Control by co-indexation. We illustrate here the possibility of controlling the application of properties thanks to the co-indexation of the categories involved in different properties. The following example describes the relative order between *Prep* and N, which is governed by the type of construction in which they are involved: the preposition precedes the noun in a prepositional construction whereas it follows it in a nominal one. Table 4 presents a first description of these different cases, illustrated with an example.

As such, it is necessary to specify the *linearity* and *dependency* properties between *Prep* and N according to the construction they belong to. In order to distinguish between these two cases, we specify the syntactic functions. The following feature structures specify the dependency features of N, illustrating here the cases of the subject of a V and a complement of a *Prep*:

$$\text{(a) N} \left[\text{DEP} \left[\begin{array}{l}\text{FUNCTION } mod\\ \text{TARGET } V\end{array}\right]\right] \qquad \text{(b) N} \left[\text{DEP} \left[\begin{array}{l}\text{FUNCTION } xcomp\\ \text{TARGET } Prep\end{array}\right]\right] \tag{30}$$

Using this representation, the distinction between the two cases of dependency between N and *Prep* relies on the specification of the function and target features of the categories (Table 5). Moreover, a co-indexation makes it possible to link the properties.

These properties stipulate an order and a dependency relation; these are determined by the syntactic roles. In a nominal construction, the noun precedes the prepositional construction that modifies it, whereas the preposition precedes the noun in the other construction. Two classical mechanisms, based on unification, are used in these

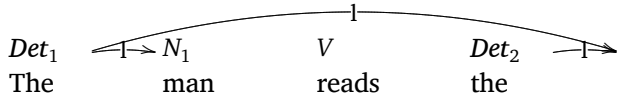
Table 5: Co-indexation between constraints

Construction type	Constraints
Nominal	$N_i \prec Prep \quad \begin{bmatrix} \text{FCT } mod \\ \text{TGT } N_i \end{bmatrix}$ $Prep \quad \begin{bmatrix} \text{FCT } mod \\ \text{TGT } N_i \end{bmatrix} \rightsquigarrow_{mod} N_i$
Prepositional	$Prep_i \prec N \quad \begin{bmatrix} \text{FCT } xcomp \\ \text{TGT } Prep_i \end{bmatrix}$ $N \quad \begin{bmatrix} \text{FCT } xcomp \\ \text{TGT } Prep_i \end{bmatrix} \rightsquigarrow_{xcomp} Prep_i$

properties: first, the specification of the dependency attribute controls the application of the properties (the *N* following *Prep* is its complement, the *Prep* that follows *N* modifies it). Moreover, index unification (marked by the use of the same index *i* in the previous examples) ensures that the category is identical across all relations: the co-indexation of the categories in the different properties imposes a reference to the same object.

4 REPRESENTING AND PROCESSING CONSTRUCTIONS

Syntactic information is usually defined with respect to a specific domain (a set of categories). For example, the precedence property between *Det* and *N* only makes sense within a nominal construction. The following example illustrates this situation, showing the possible relations corresponding to the linearity property $Det \prec N$. These relations are represented regardless of any specific domain (i.e. between all the determiners and nouns of the sentence). Same-category words are distinguished by different indices:



(31)

In this example, the relation $Det_1 \prec N_2$ connects two categories that clearly do not belong to the same domain. More generally, the

subsets of categories $\{Det_1, N_1\}$ and $\{Det_2, N_2\}$ form possible units, unlike $\{Det_1, N_2\}$. The problem is that, as explained in the previous section, properties need to be assessed and evaluated independently of any a priori knowledge of a specific domain: a property in the grammar is not specifically attached to a set of categories (a phrase or a dependent). However, linguistic description relies mainly on the identification of local phenomena that corresponds to the notion of *construction* such as that specified in *Construction Grammars* (Fillmore 1988). It is, therefore, necessary to propose an approach fulfilling both requirements: the representation of properties independently and the description of local phenomena as sets of properties.

We propose in the next two sections to examine constructions through two different perspectives: one concerning their representation and the other describing their processing. In the first perspective, constructions are described as sets of interacting properties. In the latter, constructions are recognized on the basis of topological characteristics of the property graph (representing sets of evaluated properties).

4.1 *In grammar: construction = set of properties*

Grammars organize syntactic information on the basis of structures to which different relations can be applied. In phrase-structure grammars, the notion of *phrase* implicitly comprises the definition of a domain (the set of constituents) in which the relations are valid. This notion of domain also exists in theories like HPSG, using generic tree schemata that are completed with the subcategorization information borne by lexical entries (both pieces of information together effectively correspond to the notion of constituency). Dependency grammars, in contrast, integrate syntactic information in the dependency relation between a head and its dependents. In both cases, the question of the scope of syntactic relations relies on the topology of the structures: a relation is valid inside a local tree. Therefore, a domain typically corresponds to a set of categories that share common properties.

Our approach relies on a *decentralized* representation of syntactic information by means of relations that can be evaluated independently of the entire structure. In other words, any property can be assessed alone, without needing to evaluate any other. For example, the assessment of linearity between two categories is done without taking

into account any other information such as subcategorization. In this case, we can evaluate the properties of a construction without having to create a syntactic tree: *PG* is based on a *dynamic* definition of the notion of construction. This means that all properties are assessed separately, a construction being the set of independently evaluated properties.[6]

In *Construction Grammars*, a construction is defined by the interaction of relations originating from different sources (lexical, syntactic, semantic, prosodic, etc.). This approach makes it possible to describe a wide variety of facts, from lexical selection to syntactico-semantic interactions (Goldberg 2003; Kay and Fillmore 1999; Lambrecht 1995). A construction is then intended as a linguistic *phenomenon* that is composed of syntactic units as well as other types of structures such as multi-word expressions, specific turns, etc. The notion of construction is, therefore, more general than that of syntactic unit and not necessarily based on a structured representation of information (e.g. a tree).

PG provides an adequate framework for the representation of constructions. First, a syntactic description is the interaction of several sources of information and properties. Moreover, *PG* is a constraint-based theory in which each piece of information corresponds to a constraint (or property). The description of a construction in a *PG* grammar is a set of properties connecting several categories. This definition gives priority to the relations instead of their arguments, which means that a prior definition of the set of constituents involved in the construction is not necessary.[7] As a consequence, the notion of constraint scope is not directly encoded: each property is specified independently and the grammar is a set of constructions, each described by a set of properties.

The following example illustrates the encoding of the ditransitive construction, focusing on the relation between the type of categories (*N* or *Prep*), their linear order and their function:

[6] A direct implementation of this mechanism consists in assessing all the possible properties, for all the combinations of words/categories, which is exponential. Different possibilities of controlling this complexity exists, such as delayed evaluation or probabilistic selection.

[7] In previous versions of *PG*, all categories belonging to a construction were indicated in a list of constituents.

$V_{[ditrans]} \Rightarrow N_{[obj]}$ $\quad$ $N_{[obj]} \rightsquigarrow_{obj} V_{[ditrans]}$
$V_{[ditrans]} \Rightarrow X_{[iobj]}$ $\quad$ $N_{[iobj]} \rightsquigarrow_{iobj} V_{[ditrans]}$
$N_{[iobj]} \prec N_{[obj]}$ $\quad$ $Prep_{[iobj]} \rightsquigarrow_{iobj} V_{[ditrans]}$
$N_{[obj]} \prec Prep_{[iobj]}$

The two first co-occurrence properties stipulate that the ditransitive verb governs a nominal object plus an indirect object of unspecified category encoded by X (that could be, according to the rest of the properties, either a nominal or a prepositional construction). Linearity properties stipulate that in the case of a double nominal construction, the nominal indirect object should precede the direct object. Otherwise, the direct object precedes the indirect prepositional construction. Finally, the dependency relations instantiate, according to their function, the type of the dependency with the verb.

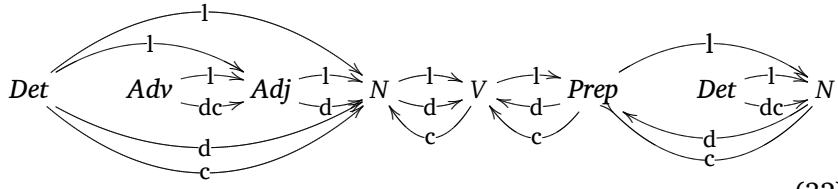
4.2 *In analysis: construction = government domain*

The theoretical and naïve parsing principle in PG consists in evaluating all properties that may exist between all categories corresponding to the words in a sentence. This set of properties contains considerable noise: most of the properties evaluated in this way link categories which do not belong to the same domain. The issue is to elicit the constructions existing in this set. Concretely, the set of properties forms a graph from which the connected categories may correspond to a construction. In the following, we put forward a formal characterisation of the notion of construction in terms of graph topology.

Generally speaking, two types of properties can be distinguished, based on the number of categories they involve:

- Binary properties, where two categories are connected: linearity, dependency, co-occurrence
- Unary properties: uniqueness, exclusion

Unary relations, because of their specificity, do not have any features that may be used to identify the construction. On the contrary, the three types of binary properties are the basis of the domain identification mechanism. The following graph illustrates the characterisation of the sentence "*A very old book is on the table.*":



(32)

It is noteworthy that in this graph, it is possible to identify several subgraphs in which all the categories are interconnected. Formally, they are referred to as being *complete*: a complete graph is a graph where all nodes are connected[8]. In this example, the nodes labelled by *Adv* and *Adj* form a complete subgraph: both categories are connected. On the other hand, the set of categories {*Det*,*Adv*,*Adj*} does not form a complete subgraph, the *Det* and *Adv* categories being disconnected.

Furthermore, when eliciting a construction, it is necessary to take into account all the categories of the same constraint network. For example, the *Adj* and *N* nodes could form a complete subgraph, but it would be a subset of another more complete subgraph {*Det*,*Adj*,*N*} subset. As a consequence, we only take into consideration *maximal complete subgraphs*.

The maximal complete subgraphs in the previous example correspond to the subsets of the following nodes (Table 6) to which we have associated a construction type.

Table 6: Constructions as complete subgraphs

Adv − *Adj*	Adjectival construction
Det − *Adj* − *N*	Nominal construction
N − *V*	Subject/verb construction
V − *Prep*	Verb/indirect object construction
Prep − *N*	Prepositional construction
Det − *N*	Nominal construction

As such, based on a graph topology, we can identify constructions for which the following definition can be given:

[8] For clarity's sake, only such subgraphs have been represented here. A complete graph would bear all possible relations, including not relevant ones, such as linearity between the first *Det* and the lat *N*. This would not change the identification and the properties of the complete subgraphs such as described here.

Definition: *A construction is a maximal complete subgraph of the property graph*.

Concretely, these subsets correspond to syntactic units. Yet, where classical approaches rely on the definition of constructions a priori in the grammar, this definition proposes a dynamic and a posteriori description. This is fundamental: it makes it possible to describe any type of sentence, regardless of its grammaticality. Analyzing a sentence consists in interpreting the property graph. This structure may contain constructions that lead directly to a semantic interpretation. But it can also be the case that the property graph contains subparts that are not necessarily connected with the rest of the sentence. This situation occurs with ungrammatical sentences.

At this stage, exhibiting the set of relevant constructions for the description of a sentence consists in identifying, among the set of maximal complete subgraphs, those that cover the set of words: in the optimal case, the set of nodes of the exhibited constructions corresponds to the set of words in the sentence. Note that in theory, constructions can overlap, which means that the same node could belong to different constructions. This characteristic is useful when combining different domains of linguistic description, including prosody, discourse, etc. However, when studying a single domain, for example syntax, it is useful to reduce overlapping: a category belonging to a construction can contribute to another construction provided it is its head. The task is therefore to exhibit the optimal set of constructions, covering the entire input.

5 PARSING BY SATISFYING CONSTRAINTS

Parsing a sentence S consists in firstly determining and evaluating the set of properties relevant for the input and secondly in exhibiting the constructions. In the second stage, it is necessary to establish all the partitions of the suite of categories that correspond to S. The issue is to know which parts correspond to a construction and whether an *optimal* partition exists.

In the first stage, an operational semantics describing conditions of satisfiability must be assigned to the properties. In this perspective, we introduce some preliminary notions:

- **Set of property categories**: Let p be a property. We define a function **Cat**(**p**) building the set of categories contained in p. For example, $Cat(Det \prec N) = \{Det, N\}$.
- **Applicable properties**: Given a grammar G and a set of categories C, the set of *C-applicable properties* is the set of all the properties of G in which the categories of C appear. More specifically, a property p is *applicable* when its evaluation becomes possible. Two types of properties can be distinguished: those requiring the realization of all the categories they involve (uniqueness, linearity and dependency) and the properties needing at least one of their categories to be evaluated (co-occurrence and exclusion). As such, we have:
  Definition: Let $p \in G$:
  - p is a *uniqueness*, *linearity* or *dependency* property: p is an *applicable property* for C iff $[Cat(p) \subset C]$
  - p is a *co-occurrence* or *exclusion* property: p is an *applicable property* for C iff $[Cat(p) \cap C \neq \emptyset]$
- **Position in the string** : We define a function **Pos**(**c**, **C**), returning the rank of c in the category suite C

An operational semantic definition may be assigned to each property as in Table 7 (C being a set of categories).

Table 7: Properties' operational semantics

- `Uniqueness`: $Uniq_x$ holds in C iff $\forall y \in C - \{x\}$, then $x \not\approx y$
- `Exclusion`: $x \otimes y$ holds in C iff $\forall z \in C - \{x\}$, then $z \not\approx y$
- `Co-occurrence`: $x \Rightarrow y$ holds in C iff $\{x, y\} \subset C$
- `Linearity`: $x \prec y$ holds in C iff $pos(x, C) < pos(y, C)$

These definitions provide the conditions of satisfiability of the different properties. It now becomes possible to illustrate how the description of the syntactic structure can be built.

The construction of the syntactic description (called the *characterisation*) of a construction consists in evaluating the set of its applicable properties. In more general terms, parsing a sentence consists in evaluating all the relevant properties and then determining the corresponding constructions. Formally:

let S be the set of categories of a sentence to be parsed,

let $Part_S$ be a partition of S,

let p be one subpart of $Part_S$,

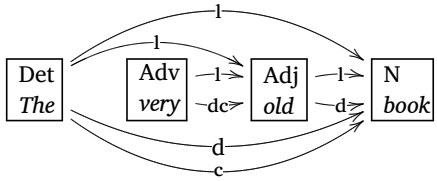
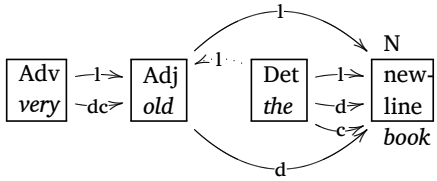
let $Prop_p$ be the set of applicable properties of p.

The categories belonging to p part are instantiated: their feature values, as determined by the corresponding lexical entries, are known insofar as they correspond to the words of the sentence to be parsed. The properties in $Prop_p$ stipulate constraints in which the categories are fully instantiated (by the unification of the categories of the properties in the grammar and those realized in the sentence). We define $Sat(Prop_p)$ as the constraint system formed by both applicable properties and the state of their satisfaction after evaluation (true or false).

Table 8 presents two examples of nominal constructions along with their characterisations; the second example contains a linear constraint violation between *Det* and *Adj*.

This example illustrates a key aspect of *Property Grammars*: their ability to describe an ill-formed sentence. Furthermore, we also note that in this description, in spite of the property violation, the nominal construction is characterized by a large number of satisfied constraints. This characteristic allows one to introduce a crucial element for usage-based grammars: *compensation* phenomena between positive and negative information. We know that constraint violation can be an element of difficulty for human or automatic processing. The idea is that the violation of constraints can be compensated by the satis-

Table 8: Characterisations of nominal constructions

Property graph	Characterisation
	$P^+ = \{Det \prec Adj,\ Det \prec N,$ $Adv \prec Adj,\ Adj \prec N,$ $Det \rightsquigarrow N,\ Adj \rightsquigarrow N,\ Adv \rightsquigarrow Adj,$ $Det \Rightarrow N,\ Adv \Rightarrow Adj,\ Adj \Rightarrow N\}$ $P^- = \emptyset$
	$P^+ = \{Det \prec N,\ Adv \prec Adj,$ $Adj \prec N,\ Det \rightsquigarrow N,\ Adj \rightsquigarrow N,$ $Adv \rightsquigarrow Adj,\ Det \Rightarrow N,$ $Adv \Rightarrow Adj,\ Adj \Rightarrow N\}$ $P^- = \{Det \prec Adj\}$

faction of some others. For example, the violation of a precedence constraint can be compensated by the satisfaction of co-occurrence and dependency ones. PG offers the possibility to quantify these compensation effects, on the basis of complexity evaluation (Blache *et al.* 2006; Blache 2011).

One important issue when addressing the question of parsing is that of ambiguity. The problem is twofold: how to represent ambiguity and how to deal with it. With syntactic information being represented in terms of graphs, it is theoretically possible to represent different types of attachment at the same time. It is possible to have in the property graph two dependency relations of the same type, which are then mutually exclusive. The control of ambiguity resolution can be done classically, thanks to preference options implemented by property weights.

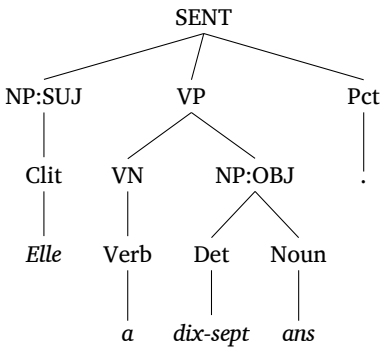
6 AN APPLICATION TO TREEBANKING

The use of treebanks offers a direct framework for the experimentation and the comparison of syntactic formalisms. Most of them have been developed using classical constituency or dependency-based representations. They have then to be adapted when studying more specific proposals. We present in this section an approach making it possible to extract properties from existing treebanks.

Most of the properties presented in this paper can be extracted automatically under some conditions, following a method presented in Blache *et al.* (2016). This is in particular the case with linearity, uniqueness, co-occurrence and exclusion, on which we focus in this section. The three first properties can be inferred fully automatically, the last one has to be filtered manually after its automatic extraction. The mechanism consists of two steps:

1. Extraction of the implicit context-free grammar
2. Generation of the properties from the CFG

In order to validate the approach, we have tested the method on several treebanks that offer different representations. We used first a set of four large constituency-based treebanks: the *Penn Treebank* (Marcus *et al.* 1994) itself, the *Chinese Treebank* (Xue *et al.* 2010), the *Arabic Treebank* (Maamouri *et al.* 2003), and the *French Treebank*



SENT → **NP:SUJ VP** *Pct*
NP:SUJ → *Clit*
VP → **VN NP:OBJ**
VN → *Verb*
NP:OBJ → *Det Noun*

Figure 3: Constituent tree and inferred CFG rules

(Abeillé *et al.* 2003). In a second stage, we have applied property extraction to the *Universal Dependencies Treebank* (Nivre *et al.* 2015). We offer a brief overview of this ongoing work presently.

The extraction of a context-free grammar (CFG) from a constituency treebank is based on a simple method described in Charniak (1996). Each internal node of a tree is converted into a rule in which the left-hand side (LHS) is the root and the right-hand side (RHS) is the sequence of constituents. The implicit grammar is composed of the complete set of rules. Figure 3 shows the syntactic tree associated with the French sentence *Elle a dix-sept ans* (“She is seventeen”), together with the corresponding CFG rules.

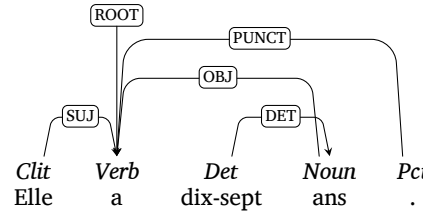
We applied a similar approach to dependency treebanks. In this case, a root node (LHS of a rule) is a head, while the constituents (RHS) form its list of dependents, following the projection order by which the head is added (encoded with the symbol *).

Figure 4 illustrates the dependency tree of the same sentence as in Figure 3 with the extracted CFG rules.

Using these grammars, it is straightforward to extract the properties that we consider in this experiment, which we describe in Figure 5.

The treebanks and the generated resources are serialized as XML; this facilitates editing and visualization. We have developed software to view the different types of information: treebanks, tagset, extracted grammar, rules, and properties. Each type of information is associated

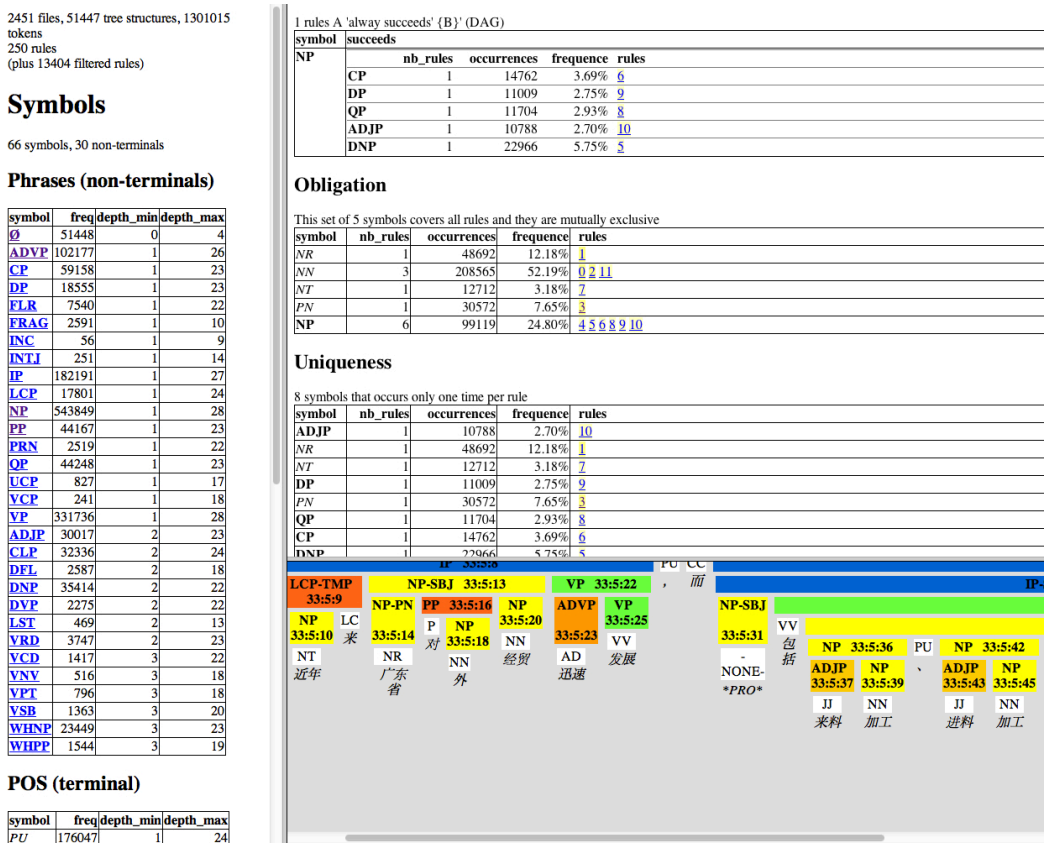
Figure 4: Dependency tree and inferred CFG rules



Verb:ROOT → *Clit:SUJ * Noun:OBJ Pct:PUNCT*
Noun:OBJ → *Det:DET **

Figure 5: Property extraction procedures

Linearity: the precedence table is built while verifying – for each category preceding another category into a construction (or a right-hand side) – whether this relation is valid throughout the set of constructions	$\forall\, rhs_m \in RHS(XP)$ **if** $((\exists\, (c_i, c_j) \in rhs_m \mid c_i \prec c_j)$ **and** $(\nexists\, rhs_n \in RHS(XP) \mid (c_i, c_j) \in rhs_n \wedge c_i \prec c_j))$ **then** add $prec(c_i, c_j)$
Uniqueness: the set of categories that cannot be repeated in a right-hand side	$\forall\, rhs_m \in RHS(XP)$ $\forall\, (c_i, c_j) \in rhs_m$ **if** $c_i \neq c_j$ **then** add $uniq(c_i)$
Requirement: identification of two categories that co-occur systematically in all constructions of an *XP*	$\forall\, rhs_m \in RHS(XP)$ $bool \leftarrow ((c_i \in rhs_m) \wedge (c_j \in rhs_m))$ **if** *bool* **then** add $req(c_i, c_j)$
Exclusion: when two categories never co-occur in the entire set of constructions, they are supposed to be mutually exclusive; this is a strong interpretation, which causes an overgeneration of such constraints, but there is no other way to identify this phenomenon automatically	$\forall\, rhs_m \in RHS(XP)$ $bool \leftarrow \neg((c_i \in rhs_m) \wedge (c_j \in rhs_m))$ **if** *bool* **then** add $excl(c_i, c_j)$



2451 files, 51447 tree structures, 1301015 tokens
250 rules
(plus 13404 filtered rules)

Symbols

66 symbols, 30 non-terminals

Phrases (non-terminals)

symbol	freq	depth_min	depth_max
Ø	51448	0	4
ADVP	102177	1	26
CP	59158	1	23
DP	18555	1	23
FLR	7540	1	22
FRAG	2591	1	10
INC	56	1	9
INTJ	251	1	14
IP	182191	1	27
LCP	17801	1	24
NP	543849	1	28
PP	44167	1	23
PRN	2519	1	22
QP	44248	1	23
UCP	827	1	17
VCP	241	1	18
VP	331736	1	28
ADJP	30017	2	23
CLP	32336	2	24
DFL	2587	2	18
DNP	35414	2	22
DVP	2275	2	22
LST	469	2	13
VRD	3747	2	23
VCD	1417	3	22
VNV	516	3	18
VPT	796	3	18
VSB	1363	3	20
WHNP	23449	3	23
WHPP	1544	3	19

POS (terminal)

symbol	freq	depth_min	depth_max
PU	176047	1	24

1 rules A 'alway succeeds' {B}' (DAG)

symbol	succeeds				
NP		nb_rules	occurrences	frequence	rules
	CP	1	14762	3.69%	6
	DP	1	11009	2.75%	9
	QP	1	11704	2.93%	8
	ADJP	1	10788	2.70%	10
	DNP	1	22966	5.75%	5

Obligation

This set of 5 symbols covers all rules and they are mutually exclusive

symbol	nb_rules	occurrences	frequence	rules
NR	1	48692	12.18%	1
NN	3	208565	52.19%	0 2 11
NT	1	12712	3.18%	7
PN	1	30572	7.65%	3
NP	6	99119	24.80%	4 5 6 8 9 10

Uniqueness

8 symbols that occurs only one time per rule

symbol	nb_rules	occurrences	frequence	rules
ADJP	1	10788	2.70%	10
NR	1	48692	12.18%	1
NT	1	12712	3.18%	7
DP	1	11009	2.75%	9
PN	1	30572	7.65%	3
QP	1	11704	2.93%	8
CP	1	14762	3.69%	6
DNP	1	22966	5.75%	5

Figure 6: Properties from the *Chinese Treebank*

with a link to a corresponding example in the treebank. Figure 6 illustrates some properties of a *NP* extracted from the *Chinese Treebank*.

In our interface, the left part of the window lists the set of categories of the grammar, together with frequency information. Non-terminals are hyperlinked to their corresponding syntactic description (corresponding PS-rules and properties). This information is displayed in the top right of the window. Each property (in this example *Obligation* and *Uniqueness*) comes with the set of rules starting from which it has been generated. Links to the different occurrences of the corresponding trees in the treebank are also listed. The lower right side of the window contains a graphical representation of the tree structure.

7 CONCLUSION

Describing linguistic phenomena by means of atomic, low-level, and independent properties makes possible the joining of formal and descriptive linguistics. We are now in position to propose a general account of language processing, capable of integrating the description of local phenomena into a global architecture and making it possible to benefit from the best of the descriptive and formal approaches.

Usage-based theories describe language starting from the data, identifying different linguistic phenomena and gathering them into a set of descriptions. In the same perspective, *Construction Grammars* represent phenomena in terms of constructions. We have proposed in this paper an extended version of *Property Grammars* (PG), that represents all syntactic information by means of properties that can interact. *PG* has the advantage of being very flexible: properties are local and independent of each other, able to represent any local relation between words or categories. This characteristic solves the issue raised by Pullum and Scholz (2001), showing the limits of a *holistic* approach in grammars, in which all statements are dependent on each other (for example, a phrase-structure rule is not considered in and of itself, but rather as a step in the derivation process corresponding to a piece of the final syntactic tree). In *PG* all information is described by means of properties; these can remain local or can interact with other properties.

PG thus offers a formal framework for representing *constructions*, which are considered as a set of interacting properties. It also constitutes a homogeneous approach integrating both views of syntactic description: a usage-based one, aimed at describing specific phenomena; and a formal one that proposes a general organization in terms of grammars. Moreover, a syntactic description given in terms of properties makes it possible to describe ill-formed inputs: a property graph is not necessarily connected, and can even contain violated properties.

As a perspective, on top of being an adequate framework for a precise description of unrestricted linguistic material, *Property Grammars* also offer a framework for an evaluation of the quality of syntactic information associated to an input, based on an analysis of the syntactic description (the quantity and the importance of satisfied properties,

their coverage, etc.). This also paves the way towards a cognitive account of language processing, capable of evaluating the relative importance of local phenomena within a general description.

REFERENCES

Bas AARTS (2004), Modelling Linguistic Gradience, *Studies in Language*, 28(1):1–49.

Anne ABEILLÉ, Lionel CLÉMENT, and François TOUSSENEL (2003), Building a Treebank for French, in A. ABEILLÉ, editor, *Treebanks*, Kluwer, Dordrecht.

Rolf BACKOFEN, James ROGERS, and K. VIJAY-SHANKER (1995), A First-Order Axiomatization of the Theory of Finite Trees, *Journal of Logic, Language, and Information*, 4(1).

Philippe BLACHE (2000), Constraints, Linguistic Theories and Natural Language Processing, in D. CHRISTODOULAKIS, editor, *Natural Language Processing*, volume 1835 of *Lecture Notes in Artificial Intelligence (LNAI)*, Springer-Verlag.

Philippe BLACHE (2007), Model Theoretic Syntax is not Generative Enumerative Syntax with Constraints: at what Condition?, in *Proceedings of CSLP07*.

Philippe BLACHE (2011), Evaluating Language Complexity in Context: New Parameters for a Constraint-Based Model, in *CSLP-11, Workshop on Constraint Solving and Language Processing*.

Philippe BLACHE, Barbara HEMFORTH, and Stéphane RAUZY (2006), Acceptability Prediction by Means of Grammaticality Quantification, in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 57–64, Association for Computational Linguistics, Sydney, Australia, `http://www.aclweb.org/anthology/P/P06/P06-1008`.

Philippe BLACHE and Laurent PRÉVOT (2010), A Formal Scheme for Multimodal Grammars, in *Proceedings of COLING-2010*.

Philippe BLACHE and Jean-Philippe PROST (2005), Gradience, Constructions and Constraint Systems, in Henning CHRISTIANSEN, Peter Rossen SKADHAUGE, and Jorgen VILLADSEN, editors, *Constraint Solving and Language Processing - CSLP 2004*, volume 3438 of *Lecture Notes in Artificial Intelligence (LNAI)*, pp. 74–89, Springer, Roskilde, Denmark.

Philippe BLACHE and Jean-Philippe PROST (2014), Model-Theoretic Syntax: Property Grammar, Status and Directions, in P. BLACHE, H. CHRISTIANSEN, V. DAHL, D. DUCHIER, and J. VILLADSEN, editors, *Constraints and Language*, pp. 37–60, Cambridge Scholar Publishers.

Philippe BLACHE, S. RAUZY, and G. MONTCHEUIL (2016), MarsaGram: an Excursion in the Forests of Parsing Trees, in *Proceedings of LREC16*.

Patrick BLACKBURN and Wilfried MEYER-VIOL (1997), Modal Logic and Model-Theoretic Syntax, in M. DE RIJKE, editor, *Advances in Intensional Logic*, pp. 29–60, Kluwer.

Joan BRESNAN (1982), *The Mental Representation of Grammatical Relations*, MIT Press Series on Cognitive Theory and Mental Representation, MIT Press.

Joan BRESNAN (2007), Is Syntactic Knowledge Probabilistic? Experiments with the English Dative Alternation, in Sam FEATHERSTON and Wolfgang STERNEFELD, editors, *Roots: Linguistics in Search of Its Evidential Base*, pp. 75–96, Mouton de Gruyter.

Joan BYBEE (2010), *Language, Usage and Cognition*, Cambridge University Press.

Eugene CHARNIAK (1996), Tree-bank Grammars, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1031–1036.

Thomas CORNELL and James ROGERS (2000), Model Theoretic Syntax, in Cheng L. LAI-SHEN and R. SYBESMA, editors, *The Glot International State of the Article Book I*, Holland Academic Graphics.

Denys DUCHIER, Thi-Bich-Hanh DAO, Yannick PARMENTIER, and Willy LESAINT (2010), Property Grammar Parsing Seen as a Constraint Optimization Problem, in *Proceedings of Formal Grammar 2010*, pp. 82–96.

Gisbert FANSELOW, Caroline FÉRY, Ralph VOGEL, and Matthias SCHLESEWSKY, editors (2005), *Gradience in Grammar: Generative Perspectives*, Oxford University Press, Oxford.

Fernanda FERREIRA and Nikole D. PATSON (2007), The 'Good Enough' Approach to Language Comprehension, *Language and Linguistics Compass*, 1(1).

Charles J. FILLMORE (1988), The Mechanisms of "Construction Grammar", in *Proceedings of the Fourteenth Annual Meeting of the Berkeley Linguistics Society*, pp. 35–55.

Gerald GAZDAR, Ewan KLEIN, Geoffrey PULLUM, and Ivan SAG (1985), *Generalized Phrase Structure Grammars*, Blackwell.

Adele E GOLDBERG (2003), Constructions: a New Theoretical Approach to Language, *Trends in Cognitive Sciences*, 7(5):219–224.

Adele E GOLDBERG (2009), The Nature of Generalization in Language, *Cognitive Linguistics*, 20(1):1–35.

Ray JACKENDOFF (2007), A Parallel Architecture Perspective on Language Processing, *Brain Research*, 1146(2-22).

Aravind JOSHI, Leon LEVY, and M. TAKAHASHI (1975), Tree Adjunct Grammars, *Journal Computer Systems Science*, 10(1).

Paul KAY and Charles FILLMORE (1999), Grammatical Constructions and Linguistic Generalizations: the *What's X doing Y?* Construction, *Language*, 75(1):1–33.

Knud LAMBRECHT (1995), Compositional vs. Constructional Meaning: The Case of French "comme-N", in M. SIMONS and T. GALLOWAY, editors, *SALT V*.

Ronald LANGACKER (1987), *Foundations of Cognitive Grammar, vol. 1 : Theoretical Prerequisites*, Stanford University Press.

Mohamed MAAMOURI, Ann BIES, Hubert JIN, and Tim BUCKWALTER (2003), Arabic Treebank, Technical report, Distributed by the Linguistic Data Consortium. LDC Catalog No.: LDC2003T06.

Mitchell P. MARCUS, Beatrice SANTORINI, and Mary Ann MAREINKIEWICZ (1994), Building a Large Annotated Corpus of English: the Penn Treebank, *Computational Linguistics*, 19(2):313–330.

Joakim NIVRE, C. BOSCO, J. CHOI, M.-C. DE MARNEFFE, T. DOZAT, R. FARKAS, J. FOSTER, F. GINTER, Y. GOLDBERG, J. HAJIC, J. KANERVA, V. LAIPPALA, A. LENCI, T. LYNN, C. MANNING, R. MCDONALD, A. MISSILÄ, S. MONTEMAGNI, S. PETROV, S. PYYSALO, N. SILVEIRA, M. SIMI, A. SMITH, R. TSARFATY, V. VINCZE, and D. ZEMAN (2015), Universal Dependencies 1.0., Technical report, http://hdl.handle.net/11234/1-1464.

Carl POLLARD and Ivan SAG (1994), *Head-driven Phrase Structure Grammars*, Center for the Study of Language and Information Publication (CSLI), Chicago University Press.

Alan PRINCE and Paul SMOLENSKY (1993), *Optimality Theory: Constraint Interaction in Generative Grammars*, Technical Report RUCCS TR-2, Rutgers Optimality Archive 537.

Geoffrey PULLUM and Barbara SCHOLZ (2001), On the Distinction Between Model-Theoretic and Generative-Enumerative Syntactic Frameworks, in Philippe DE GROOTE, Glyn MORRILL, and Christian RÉTORÉ, editors, *Logical Aspects of Computational Linguistics: 4th International Conference*, number 2099 in Lecture Notes in Artificial Intelligence, pp. 17–43, Springer Verlag, Berlin.

Ivan SAG (2012), Sign-Based Construction Grammar: An Informal Synopsis, in H. BOAS and I. SAG, editors, *Sign-Based Construction Grammar*, pp. 69–200, CSLI.

Ivan SAG, Hans BOAS, and Paul KAY (2012), Introducing Sign-Based Construction Grammar, in H. BOAS and I. SAG, editors, *Sign-Based Construction Grammar*, pp. 1–30, CSLI.

Ivan SAG and T. WASOW (1999), *Syntactic Theory. A Formal Introduction*, CSLI.

Benjamin SWETS, Timothy DESMET, Charles CLIFTON, and Fernanda FERREIRA (2008), Underspecification of Syntactic Ambiguities: Evidence from Self-Paced Reading, *Memory and Cognition*, 36(1):201–216.

Lucien TESNIÈRE (1959), *Éléments de syntaxe structurale*, Klincksieck.

Nianwen XUE, Zixin JIANG, Xiuhong ZHONG, Martha PALMER, Fei XIA, Fu-Dong CHIOU, and Meiyu CHANG (2010), Chinese Treebank 7.0, Technical report, Distributed by the Linguistic Data Consortium. LDC Catalog No.: LDC2010T07.

The sequencing of adverbial clauses of time in academic English: Random forest modelling of conditional inference trees

Abbas Ali Rezaee and Seyyed Ehsan Golparvar
University of Tehran, Tehran, Islamic Republic of Iran

ABSTRACT

Adverbial clauses of time are positioned either before or after their associated main clauses. This study aims to assess the importance of discourse-pragmatics and processing-related constraints on the positioning of adverbial clauses of time in research articles of applied linguistics written by authors for whom English is considered a native language. Previous research has revealed that the ordering is co-determined by various factors from the domains of semantics and discourse-pragmatics (bridging, iconicity, and subordinator) and language processing (deranking, length, and complexity). This research conducts a multifactorial analysis on the motivators of the positioning of adverbial clauses of time in 100 research articles of applied linguistics. The study will use a random forest of conditional inference trees as the statistical technique to measure the weights of the aforementioned variables. It was found that iconicity and bridging, which are factors associated with discourse and semantics, are the two most salient predictors of clause ordering.

Keywords: positioning, discourse, semantics, processing, iconicity

1 INTRODUCTION

Previous research on subordinate adverbial clauses has revealed that the majority of these clauses are mainly put in initial and final posi-

tions (Aarts 1988; Kirk 1997; Diessel 1996, 2001; Givón 2011). These two clause positions serve different discourse functions.

Adverbial clauses that are sentence-final usually play a local function. They illustrate the conditions of their matrix clause by specifying reason, temporal circumstances, result, etc. Further, such adverbial clauses are usually unidirectional; they are linked to their main clauses as already stated. Post-posed adverbial clauses offer information which is more integrated with the matrix clause at the local level (Thompson *et al.* 2007). Moreover, such adverbial clauses are mostly placed in the middle position of a paragraph; that is, adverbial clauses in final position are usually in the middle of a firmly coherent thematic chain (Givón 2001). In terms of semantics, the information encoded in sentence-final clauses tends to be in line with the information expressed in clauses that are in coordination (Ford 1993; Givón 2001).

On the other hand, sentence-initial adverbial clauses play a stringently local function, but have broader discourse-organizing functions by dint of enumerating a new frame for the coming discourse or connecting it to the preceding discourse. Furthermore, the cohesive function of pre-posed clauses may occur at different levels, from the whole discourse to inter-paragraph and inter-sentential levels. The inter-sentential function may be deemed as a local back-referencing function yielding a close connection between two sentences, "while the higher-level function marks the episode boundary or thematic discontinuity" (Thompson *et al.* 2007, p. 289). It should be observed that whether local or global, initial adverbial clauses play a bidirectional function, connecting what has been stated before to what is to be expressed. In addition, semantic information offered by pre-posed clauses is less significant due to the fact that they often repeat or give predictable information from what has already been stated (Thompson *et al.* 2007).

Thus, the two ordering patterns of adverbial clauses are not necessarily interchangeable in the academic discourse and writers of research articles should be cognizant of when to employ each of these positions in their texts.

The present study intends to examine the constraints on the positioning of temporal adverbial clauses in research articles of applied linguistics. Further, this research seeks to measure the weight of processing-based and discourse-pragmatics constraints on the po-

sitioning of finite, temporal adverbial clauses by means of a random forest of conditional inference trees, which has proved to be more efficient than ordinary regression models (Tagliamonte and Baayen 2012; Wiechmann and Kerz 2013).

2 BACKGROUND

Two approaches have attempted to account for the positioning tendency of adverbial clauses in English. The first approach is grounded upon the fact that the order of linguistic items, including finite adverbial clauses, is primarily influenced by the information structure of the string. Proponents of this discourse-based account (e.g., Chafe 1984; Birner and Ward 1998) have put forward the argument that users of a language tend to produce new, inaccessible information, which is reflected in the main clause, after given, accessible information that is expressed by the subordinate clause.

Two factors encourage speakers and writers to place adverbial clauses in the initial position, namely the 'bridging' function and the 'setting the stage' function. Sentence-final adverbial clauses serve local functions, whereas sentence-initial adverbial clauses play discourse-organizing functions. Two instances of discourse-organizing functions are connecting the sentence to the preceding discourse and introducing new frames for upcoming discourse (Ford 1993; Verstraete 2004; Thompson *et al.* 2007; Givón 2011).

The current study, like Wiechmann and Kerz (2013), only focuses on one discourse-pragmatic factor: bridging. It refers to a context in which an initial adverbial clause acts like a bridge between the previous and the upcoming discourse. The presence of an anaphoric item in an adverbial clause marks the bridging function of that clause. In example (1), the underlined part is a sentence-initial temporal clause and the anaphoric item THEIR plays a bridging function, connecting the previous sentence to the upcoming discourse.

(1) This article explores the citing behaviours of 16 undergraduates in a North American university. After completing a research paper for *their* disciplinary courses, each participating student was interviewed to identify in his/her writing words and ideas borrowed from source texts and to explain why and how the relevant texts were appropriated with or without citations. (Shi, 2010)

The semantic nature of the subordinate clauses is the other factor examined in the discourse-based approach. To put it differently, the semantic difference observed among different types of adverbial clauses (such as adverbial clauses of time, condition, concession, etc.) cause them to occupy different positions within a complex sentence (Quirk *et al.* 1985; Biber *et al.* 1999; Diessel 2005, 2008; Wiechmann and Kerz 2013). For example, Diessel (2001) showed that conditional clauses usually precede their associated matrix clauses, causal clauses are usually sentence-final, and there is a roughly even distribution between initial and final temporal adverbial clauses. Diessel (2001) also revealed that adverbial clauses of reason and purpose are predominantly placed in the final slot. Moreover, concessive clauses show a slight preference for the final position (Biber *et al.* 1999; Diessel 2001; Wiechmann and Kerz 2013). Clauses headed by different subordinators display slight differences in meaning. Thus, any subordinator selected for adverbial clauses is deemed as a predictor of the positioning of these clauses (Wiechmann and Kerz 2013). For example, IF and UNLESS are the most common subordinators for adverbial clauses; however, IF is the most versatile conditional subordinator, According to Quirk *et al.* (1985), WHEN, AFTER, and BEFORE are the most frequent temporal subordinators in academic English, which will be the focus of this study.

Iconicity is another factor that affects the order of temporal adverbial clauses. According to Croft (2003), the main idea underlying iconicity is that the structure of language is a reflection of the structure of experience. Haiman (2015) has asserted that some of the most basic principles and rules of language tend to be ironically motivated. “The meaning of a complex expression is in some way the sum of the meanings of its parts”, “Conceptual closeness of ideas is reflected in physical closeness of their expression”, “The same form is used for same meaning”, and “More form reflects more meaning” (Haiman 2015, p. 512) are some of the iconic principles of language.

It has been suggested that the order of clauses in complex sentences often corresponds to the order of events they describe (Diessel 2008; Haiman 2015). Previous studies have demonstrated that this tendency is able to account for the positioning of some types of subordinate clauses. For instance, Haiman (1983) showed that conditional clauses are usually placed in the sentence-initial position since the

event they describe is conceptually prior to the one denoted by the matrix clause. Similarly, Greenberg (1963) has argued that purpose clauses follow their associated matrix clause because they express the upshot of the action denoted in the main clause. In a similar vein, it has been suggested that AFTER clauses are put in initial position more frequently than BEFORE clauses, because the former denote an event that takes place prior to the one in the matrix clause, while BEFORE clauses describe a posterior event (Clark 1971).

The other approach attempting to account for the ordering of dependent clauses considers processing-related factors. These accounts expound the positioning of an adverbial clause on the grounds of constraints like the relative length of the clause, complexity, and deranking. The most prominent supporter of this account is John Hawkins (Hawkins 1994, 2004), claiming that the constituent order is basically determined by processing difficulty. Hawkins has explained that information structure comes to the scene only when two alternative orders are equally demanding with regard to processing.

The first processing-related factor co-determining the order of temporal adverbial clauses is the length of the constituents. Past research has clearly demonstrated that in languages like English longer constituents usually come after shorter ones (Quirk *et al.* 1985). This tendency can be explained based on the notion that the processing of the whole construction (complex sentence) appears to be more smooth with this order (Hawkins 1994, 2004; Gibson 1998, 2000). In line with Hawkins' performance-based theory of constituent ordering (Hawkins 2004), constituents deemed to be heavy tend to appear in the final slot, because this ordering is cognitively more efficient in languages which are head-initial, rendering both production and parsing easier.

In a similar vein, the dependency locality theory propounded by Gibson (2000) assumes that the processing complexity of a linguistic string rests on the length of its syntactic dependencies. The complexity effects on ordering follow from the integration cost component determining that longer distance attachments are more demanding to produce in comparison with shorter distance ones (Hawkins 1994). Temporal adverbial clauses that are placed in the initial slot yield longer dependencies and are hence more burdensome to process.

We may also resort to a pragmatics, information-structural account to shed light on the tendency of 'lighter' constituents to precede

'heavier' ones in accordance with the 'given-new' principle (Arnold *et al.* 2000), paying attention to the fact that new information requires more linguistic materials to be encoded compared to given information. The discourse-pragmatics account has also revealed that the informativeness increases towards the end of each grammatical unit, for both clauses and multi-clause expressions. Thus, length is a salient predictor of the positioning of adverbial clauses.

The second predictor of clause positioning that is related to processing difficulty is complexity. There are a number of definitions and accounts of complexity such as relative complexity (Vulanovic 2007), absolute complexity (Miestamo 2004), language complexity (Hawkins 1994, 2004), and complexity in terms of informativeness (Li and Vitányi 1997). Adverbial clauses of time may show different degrees of complexity. It may be expected that pre-posed adverbial clauses are structurally less complex. Following Diessel (2008) and Wiechmann and Kerz (2013), in this study we consider only those dependent clauses as complex that contain at least another dependent clause of any kind. We should bear in mind that linguistic complexity and the length of adverbial clause are closely tied to each other. Adverbial clauses containing another subordinate clause – complex adverbial clauses – tend to be longer and hence are more demanding to process. Consequently, it can be assumed that complex adverbial clauses of time are usually post-posed.

Wiechmann and Kerz (2013) have noted that deranking is another processing-related factor affecting the ordering of temporal adverbial clauses. Based on Stassen (1985), languages may apply two basic strategies in coding two linked clauses coming in a fixed temporal order. In the first strategy, called *balancing*, the two clauses have verb forms that are structurally equivalent, each of them occurring in one independent clause. Example (2) is an illustration of this strategy.

(2) His father died before he was born.

In the second strategy, *deranking*, a verb form that cannot come in an independent clause is used in the dependent clause. A deranked verb form is different from its balanced counterpart in two ways: (1) the categorical distinctions usually associated with verbs in language, like tense, aspect, mood, or person distinctions, are totally or partially absent, (2) particular markings that are not allowed to be used in in-

dependent clauses are used in dependent clauses (Cristofaro 2003). Consider example (3):

(3) Coming home, he directly went to bed.

In other words, an adverbial clause in English is 'balanced' if it is tensed, whereas it is perceived as 'deranked' provided that it is not tensed but reduced in some way. Deranked adverbial clauses consist of a non-finite verb form or are used as a verbless construction (Wiechmann and Kerz 2013). Consider Example (4):

(4) The findings indicate that a significant percentage of the subjects experience difficulties *when studying content subjects through the medium of English.*
(Evans and Green 2007)

In Example (4), the italic part is a deranked temporal adverbial clause in which 'studying' is a verb without tense. It might be assumed that balanced adverbial clauses tend to be longer than deranked ones and consequently their processing can be more difficult. However, this is not always true; as Cristofaro (2003) and Wiechmann and Kerz (2013) have noted, non-finite or verbless adverbial clauses present information in a more condensed format. Therefore, reduced or deranked adverbial clauses involve greater syntactic integration and more informational compactness and can be much more demanding in processing, which can move them to the final slot.

Recent inquiries on clause positioning have demonstrated that a variety of constraints, the effects of which may be in conflict, condition the ordering of finite adverbial clauses. They have revealed that the ordering of main and adverbial clauses is determined by the interaction between processing, discourse, pragmatics, and semantics (Wasow 2002; Diessel 2005, 2008; Wiechmann and Kerz 2013).

Diessel (1996) examined the processing factors of initial and final adverbial clauses. Particularly, Diessel examined the ordering of finite adverbial clauses (such as adverbial clauses of condition, concession, time, reason, and manner) in light of Hawkins' processing principles (Hawkins 1994). Diessel (2008) also explored the impact of several factors (including: length, complexity, pragmatic import, and the principle of iconicity) on the ordering of adverbial clauses of time, and demonstrated that iconicity of sequence is the most powerful predictor of the positioning of temporal adverbial clauses. Finally, Wiech-

mann and Kerz (2013) made an assessment of the weight of discourse-pragmatics and processing-based constraints on the ordering of concessive adverbial clauses. They revealed that discourse-pragmatics factors, namely bridging and subordinator choice, are the stronger factors predicting the positioning of concessive adverbial clauses.

3 METHOD

3.1 *Corpus*

A corpus of 100 research articles written by native speakers of English was compiled for this experiment.

The articles were randomly sampled[1] from a set of articles published in each of ten applied linguistics/language learning/language teaching journals.[2] Ten articles were selected from each journal. All these articles were filtered so that only those with the standard IMRD (Introduction, Methods, Results, and Discussion) format were included.

3.2 *Data annotation*

In this study, the position of temporal adverbial clauses (POS) is the dependent variable which is measured as a binary factor having two levels that are final (POS 1) and initial (POS 0). In addition, the predictors of clause ordering are bridging, subordinator, iconicity, length, complexity, and deranking. Bridging (BRG) is measured on a binary basis with two levels that are containing an anaphoric item indicating a bridging context (BRG 1) and absence of such an item (BRG 0). Subordinator is a categorical variable with three levels, namely WHEN (SUB 0), AFTER (SUB 1), and BEFORE (SUB 2). According to Quirk

[1] We enumerated all articles published in all ten journals between 2001 and 2014, then performed stratified random sampling using random number tables from Stat Trek (`http://stattrek.com/Tables/Random.aspx`).

[2] The ten journals from which we sampled articles are: Annual Review of Applied Linguistics; Applied Linguistics; ESP Journal; EAP Journal; Language Learning; Language Teaching Research; System; Second Language Research; Second Language Writing; and TESOL Quarterly. These journals may contain different types of articles (research articles, reviews, editorials); only research articles were included in the corpus compiled.

et al. (1985), these three subordinators are the most frequent temporal subordinators in the academic register. Iconicity is also measured on a categorical basis with three levels that are clauses referring to a prior event (ICN 0), clauses denoting a simultaneous event (ICN 1), and clauses expressing a posterior event (ICN 2).

As in Wiechmann and Kerz (2013), the relative length of dependent clauses (LNG) is measured as a continuous variable, which is defined as the proportion of the number of words in the adverbial clause to that of the whole complex sentence containing that clause. Complexity (COM) is binary variable with two categories: simple (COM 0) and complex (COM 1). Finally, deranking (DRK) is similarly a binary variable encoding balanced (DRK 0) and deranked (DRK 1).

3.3 *Data analysis*

The present research uses a *random forest of conditional inference trees*. Each forest is a large number of decision trees used for variable selection. Each decision tree is able to cope with missing values; nonetheless, use of one single tree may be unreliable due to the fact that minor changes in the input variables may bring about significant changes in the output. Therefore, selecting variables by means of a random forest of such trees is a far more efficient tool (Breiman 2001).

The preference for random forest modelling with conditional inference trees is rooted in the fact that it provides an unbiased tool for variable selection in the individual classification trees, enabling us to reliably assess the relative importance of variables coded on different scales or different with regard to the number of their factor levels. This is a salient deficiency of traditional tree-based models. In addition, the coefficients of logistic regression models are far more complex to interpret (Hothorn *et al.* 2006; Strobl *et al.* 2007).

Classification trees generally try to predict a binary outcome on the basis of a group of predictors. The algorithms by which classification trees operate work through the data and determine a number of if-then logical (split) conditions yielding definite classification of cases. To put it another way, in the initial step, the algorithm will split the data based on the most significant predictor. The algorithm will go on separating each resulting subset of the data until it can no longer find significant associations between the dependent variable and any of the predicting variables (Wiechmann and Kerz 2013).

By contrast, mixed effects models are grounded upon various assumptions concerning the distribution of the data and require the data to satisfy given requirements in order that such models' parameters be estimable: whereas random forests are non-parametric, rendering them a more flexible tool allowing the researcher to incorporate all potential predictors in the analysis concurrently, even if there exist severe interactions among these factors, there are highly unequal cell counts or even empty cells, or are collinear. If, instead, a linear modelling framework were to be used, it could result in potentially unsolvable computational problems (Strobl *et al.* 2007).

4 RESULTS

The findings of this study demonstrate that the majority of adverbial clauses of time (64.8%) are in final position. In addition, a considerable proportion of them are simple (88.4%), balanced (72.8%), have no anaphoric item suggesting a bridging context (91.6%), and begin with WHEN as the subordinator (80.8%). Moreover, their average length relative to the size of the whole complex sentence is 0.43. In addition, more than half of the adverbial clauses of time in this corpus (52.9%) denote a simultaneous event. Furthermore, temporal clauses expressing a prior event (37.8%) are far more frequent that those referring to a posterior event (9.3%). Table 1 reports some descriptive statistics with regard to the sample.

Figure 1 illustrates the distribution of these predictors across the two clause positions in this corpus of adverbial clauses of time. Figure 1 depicts that there is a significant distribution difference between initial and final adverbial clauses with regard to iconicity and bridging, and to a lesser extent, length. In addition, Figure 1 suggests that temporal adverbial clauses without a bridging function are mostly in final position, whereas those involving a bridging context are mainly sentence-initial. With regard to complexity, it is observed that in both simple and complex clauses, post-posed adverbial clauses outnumber pre-posed ones. Likewise, in both balanced and deranked clauses, temporal clauses in final position are more frequent than those in initial position. Further, in the three subordinators, sentence-final clauses are more frequent than sentence-initial ones. Finally, Figure 1 shows that temporal adverbial clauses expressing a prior event are

Dependent variable	POS	**Initial**		**Final**
		35.2%		64.8%
Predictors	BRG	**Bridging**		**Non-bridging**
		91.6%		8.4%
	COM	**Simple**		**Complex**
		88.4%		11.6%
	DRK	**Balanced**		**Deranked**
		72.8%		27.2%
	SUB	**When**	**After**	**Before**
		80.8%	11.9%	7.3%
	ICN	**Prior**	**Simultaneous**	**Posterior**
		37.8%	52.9%	9.3%
	LNG	**Mean**		**Standard Deviation**
		0.43		0.16

Table 1: Descriptive statistics

more often in sentence-initial position, while temporal clauses referring to a simultaneous or posterior event usually follow their associated main clauses.

Figure 2 depicts the conditional inference tree. The analysis of this tree reveals that three of the predictors of the positioning of temporal clauses (subordinator, bridging, and iconicity) turn out to be significant predictors. Each oval denotes a split variable and the corresponding *p* value estimating the significance level. Moreover, the numbers on the lines linking the nodes of the tree show the particular categories of the nominal predictors or the value range of the numerical predictors.[3]

In order to interpret the tree, we should examine it from top to bottom. At the top of the tree representing all data in the first subset, the first split is made based on iconicity. Temporal adverbial clauses that are prior (ICN ≤ 0) are split based on bridging (Node 2), whereas adverbial clauses of time that are simultaneous or posterior (ICN > 0) are further split based on their subordinator (SUB, Node 5).

The conditional inference tree clearly demonstrates that among the prior adverbials that lack a bridging context (BRG ≤ 0), sentence-

[3] It should be noted that the only numerical variable in this research is length.

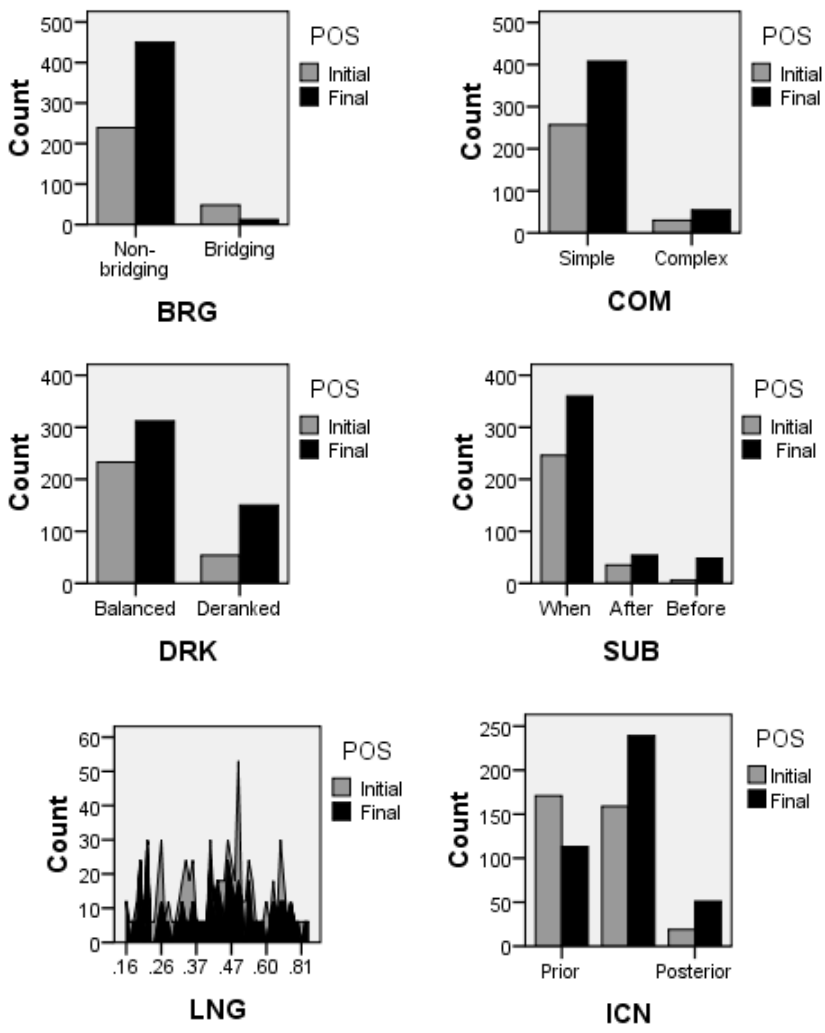


Figure 1: The distribution of the six predictors across the two positions

initial clauses slightly outnumber sentence-final ones (Node 3, 468 cases), whereas those with a bridging context are mostly in the initial position (Node 4, 40 cases). On the other hand, among temporal clauses that are simultaneous or posterior (ICN > 0), those that are headed by WHEN or AFTER (SUB ≤ 1) are predominantly in final position. This is observed in Node 6 with 128 cases. Moreover, adverbial clauses of time that are introduced by BEFORE are all post-posed (Node 7, 20 cases).

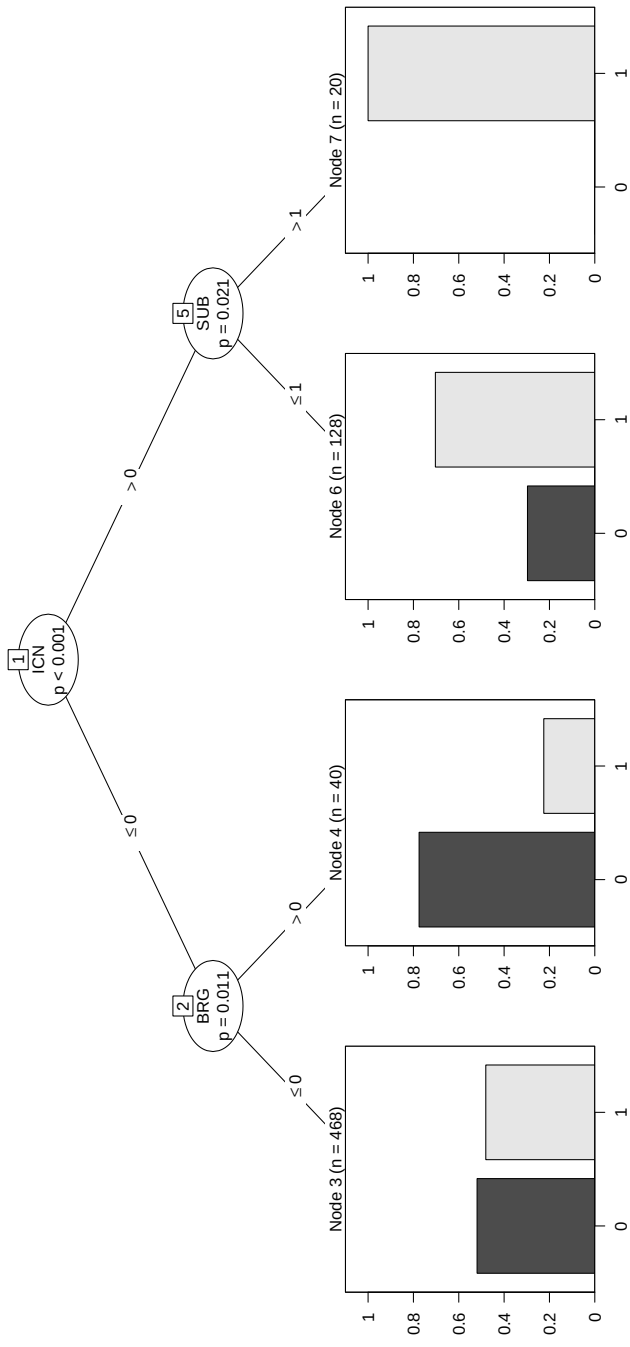


Figure 2: Conditional inference tree for clause positioning

A salient point with regard to a single-tree model is that such a model can yield problematic results. In order to solve this problem, a forest of such trees – rather than a single one – is built. This will produce more robust and generalizable findings (Breiman 2001). In this study a total set of 500 trees is built by means of a bootstrapping technique, in which 500 different random subsamples are taken from the original data.

In order to measure how salient each variable is for predicting the ordering of concessive clauses, a permutation variable importance measure is calculated. We used the conditional variable importance measure implemented in the cforest function of the Party package in R. In this estimation, the original values of the predictor are permutated to decouple the original association of the predictor and the response. This will demonstrate how much the overall classification accuracy of the model drops. The greater the decrease in classification accuracy is, the more useful that predictor is for modelling clause positioning. The superiority of the conditional variable importance measure over alternatives (e.g., Gini importance) lies in the fact that it is not biased in cases where explanatory variables are different in terms of their number of categories or scale of measurement (Breiman 2001). Figure 3 illustrates the variable importance plot for the six predictors measured by the random forest model.

Figure 3 demonstrates that iconicity is by far the most important variable for predicting the ordering of adverbial clauses of time in

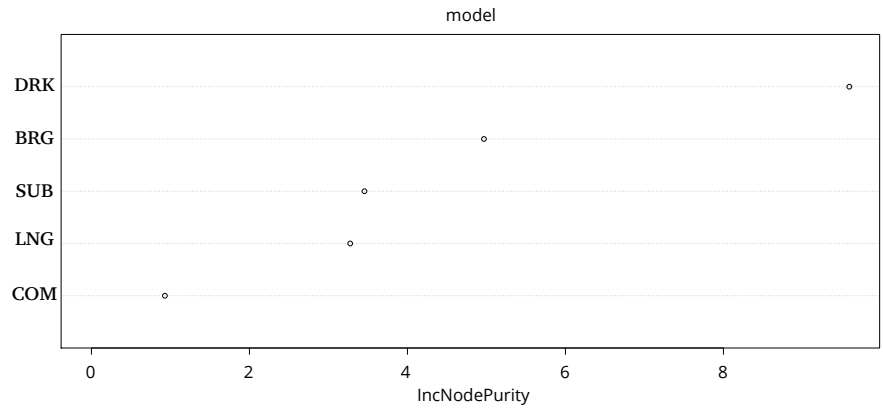


Figure 3: The variable importance plot

academic English. To put it differently, whether a temporal adverbial clause refers to a prior, simultaneous, or posterior event is the most significant factor determining the ordering of these constructions. Initial clauses mostly suggest a prior event and final clauses usually denote a simultaneous or posterior event. As depicted in Figure 3, bridging (whether temporal adverbial clauses contain an anaphoric item indicating a bridging context or not) is the second most important predictor of the positioning of adverbial clauses of time. Initial clauses mostly suggest a bridging context and final clauses usually do not have such a function. The conditional variable importance plot (Figure 3) also revealed that discourse-pragmatics motivators play a more important role in sequencing adverbial clauses of time.

5 DISCUSSION

The results of this research revealed that sentence-final adverbial clauses of time are more frequent than sentence-initial ones. In addition, the random forest of conditional inference trees demonstrated that iconicity of sequence is the most powerful predictor of the positioning of temporal clauses. Further, bridging turns out to be the second most important variable for predicting the position of these clauses. Consequently, factors associated with discourse and pragmatics can offer a better explanation for the ordering of adverbial clauses of time. Finally, among the motivators of clause order that are related to processing, length is a more powerful predictor of clause positioning in research articles of applied linguistics.

On the descriptive side, the findings of this research indicated that in adverbial clauses of time produced by native writers, sentence-final clauses (64.8%) outnumber sentence-initial ones (38.2%). This is supported by previous research (Chafe 1984; Quirk *et al.* 1985; Diessel 1996; Biber *et al.* 1999; Diessel 2001, 2005, 2008; Wiechmann and Kerz 2013). The descriptive analysis also revealed that in both balanced and simple clauses post-posed temporal clauses are more frequent than pre-posed ones. In addition, in all of the three temporal subordinators, final constructions are more frequently observed than initial ones. Therefore, adverbial clauses of time mostly follow their matrix clauses in the academic corpus of this study.

The findings of this study revealed that iconicity of sequence has a strong impact on the linear ordering of adverbial clauses of time. Temporal clauses expressing a prior event are more often in sentence-initial position, while temporal clauses referring to a simultaneous or posterior event usually follow their associated main clauses. This is in line with Diessel (2008) who has claimed that iconicity of sequence, in spite of being semantic by nature, can be regarded as a processing principle affecting the overall processing load of a complex sentence since a clause order that is not iconic is more demanding to process. This is also supported by Ohtsuka and Brewer (1992) who demonstrated that temporal clauses headed by NEXT are easier to store and retrieve than non-iconic clauses headed by BEFORE.

Random forest modelling of the competing motivators of the ordering of adverbial clauses of time also indicated that the presence of an anaphoric item with a bridging context is the second most powerful predictor of clause ordering in this corpus of temporal clauses written by researchers of applied linguistics for whom English is deemed as a native language. This is supported by Wiechmann and Kerz (2013) in which bridging emerged as the first predictor of positioning in adverbial clauses of concession. This finding corroborates the idea that adverbial clauses of time are mostly put in the initial position when their function is to organize the flow of information in the ongoing discourse, and their use is affected by factors related to information structuring and cohesion (Givón 2001; Verstraete 2004; Diessel 2005, 2008; Wiechmann and Kerz 2013). Consider Example (5):

(5) This second rater reviewed 15% of the data and then results were compared with those obtained by the researcher. A minimum of an 80% coincidence was needed. When *this 80%* was not achieved, which only happened in one case, the case was discussed until both raters agreed on the mark.

(Llanes and Muñoz 2009)

In Example (5), the underlined part is a temporal adverbial clause in which 'this 80%' is an anaphoric item indicating a bridging context. The anaphoric item and the adverbial clause of time in which it is embedded establish a link between the main clause and the previous discourse. The results of this research demonstrated that the majority of these bridging-functioning clauses precede their main clauses.

The relative length of the adverbial clauses of time investigated in this study was the most closely associated variable among those motivated by processing-based theories. It only emerged as the third predictor of the ordering of these constructions. This is in line with Diessel (2008) and Wiechmann and Kerz (2013) who also found that length plays a marginal role in predicting temporal ordering. According to Hawkins' parsing theory, it can be assumed that post-posed adverbial clauses are easier to process than pre-posed ones since complex sentences containing final adverbial clauses enjoy a shorter recognition domain than complex sentences involving initial adverbial clauses (Hawkins 2004). This offers a cogent explanation for the predominance of sentence-final adverbial clauses of time in English (Diessel 2008). This also provides further support for the fact that the ordering of adverbial clauses of time is first and foremost determined by discourse-pragmatic and semantic constraints rather than processing-based explanations.

6 CONCLUSION

The findings of this research demonstrated that in a corpus of 100 research articles in the field of applied linguistics written by those for whom English is considered as a native language, post-posed temporal clauses outnumber pre-posed ones. In addition, this study provided further support for previous research on clause positioning (Diessel 2005; Wasow 2002; Diessel 2008; Wiechmann and Kerz 2013) suggesting that the ordering of adverbial clauses is co-determined by constraints of cognitive processing and discourse-pragmatics. Moreover, discourse-pragmatics motivators (iconicity and bridging) are significantly better predictors of the position of temporal adverbial clauses than processing-related constraints. Further, the length of these clauses, as a processing-related factor, emerged as the third significant predictor of the positioning of temporal clauses in this corpus. Finally, the random forest of conditional inference trees technique was found to be a robust statistical means for assessing the relative weight of these constraints.

REFERENCES

Bas AARTS (1988), Clauses of concession in written present-day British English, *Journal of English Linguistics*, 21(1):39–58.

Jennifer E. ARNOLD, Anthony LOSONGCO, Thomas WASOW, and Ryan GINSTROM (2000), Heaviness vs. newness: The effects of structural complexity and discourse status on constituent ordering, *Language*, 76(1):28–55.

Douglas. BIBER, Stig JOHANSSON, Geoffrey LEECH, Susan CONRAD, and Edward FINEGAN (1999), *Longman Grammar of Spoken and Written English*, Longman, London, `http://eprints.lancs.ac.uk/1275/`.

Betty J. BIRNER and Gregory WARD (1998), *Information Status and Noncanonical Word Order in English*, number 40 in Studies in Language Companion Series, John Benjamins Publishing Company.

Leo BREIMAN (2001), Random forests, *Machine Learning*, 45(1):5–32.

Wallace CHAFE (1984), How people use adverbial clauses, in *Proceedings of the Tenth Annual Meeting of the Berkeley Linguistics*, pp. 437–449, Linguistic Society of America.

Eve V. CLARK (1971), On the acquisition of the meaning of before and after, *Journal of Verbal Learning and Verbal Behavior*, 10(3):266–275, `http://www.sciencedirect.com/science/article/pii/S0022537171800543`.

Sonia CRISTOFARO (2003), *Subordination*, Oxford Studies in Typology and Linguistic Theory, Oxford University Press, Oxford.

William CROFT (2003), *Radical Construction Theory: Syntactic Theory in Typological Perspective*, Cambridge University Press, 2nd edition.

Holger DIESSEL (1996), Processing factors of pre-and postposed adverbial clauses, in *Proceedings of the 22th Annual Meeting of the Berkeley Linguistics Society*, pp. 71–82, Linguistic Society of America.

Holger DIESSEL (2001), The ordering distribution of main and adverbial clauses: A typological study, *Language*, 77(3):433–455.

Holger DIESSEL (2005), Competing motivations for the ordering of main and adverbial clauses, *Linguistics*, 43(3):449–470.

Holger DIESSEL (2008), Iconicity of sequence: A corpus-based analysis of the positioning of temporal adverbial clauses in English, *Cognitive Linguistics*, 19(3):465–490.

Stephen EVANS and Christopher GREEN (2007), Why EAP is necessary: A survey of Hong Kong tertiary students, *Journal of English for Academic Purposes*, 6(1):3–17.

Cecilia E. FORD (1993), *Grammar in Interaction: Adverbial Clauses in American English Conversations*, Cambridge University Press, Cambridge.

Edward GIBSON (1998), Linguistic complexity: Locality of syntactic dependencies, *Cognition*, 68(1):1–76, http://dx.doi.org/10.1016/S0010-0277(98)00034-1.

Edward GIBSON (2000), The Dependency Locality Theory : A distance-based theory of linguistic complexity, in Alec P. MARANTZ, Yasushi MIYASHITA, and Wayne O'NEIL, editors, *Image, Language, Brain*, pp. 95–126, MIT Press.

Talmy GIVÓN (2001), *Syntax: Introduction*, volume 2, John Benjamins Publishing Company, Amsterdam and Philadelphia.

Talmy GIVÓN (2011), *Ute Reference Grammar*, volume 3 of *Culture and Language Use*, John Benjamins Publishing.

Joseph H. GREENBERG (1963), Some universals of grammar with particular reference to the order of meaningful elements, in Joseph H. GREENBERG, editor, *Universals of language*, volume 2, pp. 73–113, MIT Press.

John HAIMAN (1983), Iconic and economic motivation, *Language*, 59(4):781–819, http://www.jstor.org/stable/413373.

John HAIMAN (2015), Iconicity in linguistics, in James D. WRIGHT, editor, *International Encyclopedia of the Social & Behavioral Sciences*, Elsevier, Oxford, second edition.

John A. HAWKINS (1994), *A Performance Theory of Order and Constituency*, volume 73 of *Cambridge Studies in Linguistics*, Cambridge University Press.

John A. HAWKINS (2004), *Efficiency and Complexity in Grammars*, Oxford University Press, http://tocs.ub.uni-mainz.de/pdfs/125604661.pdf.

Torsten HOTHORN, Kurt HORNIK, and Achim ZEILEIS (2006), Unbiased recursive partitioning: A conditional inference framework, *Journal of Computational and Graphical Statistics*, 15(3):651–674.

John M. KIRK (1997), Subordinate clauses in English, *Journal of English Linguistics*, 25(4):349–364.

Ming LI and Paul VITÁNYI (1997), *An Introduction to Kolmogorov Complexity and its Applications*, Graduate Texts in Computer Science, Springer Verlag, 2nd edition.

Àngels LLANES and Carmen MUÑOZ (2009), A short stay abroad: Does it make a difference?, *System*, 37(3):353–365.

Matti MIESTAMO (2004), On the feasibility of complexity metrics, in *FinEst Linguistics: Proceedings of the Annual Finnish and Estonian Conference of Linguistics*, pp. 11–26, Tallinn University Press.

Keisuke OHTSUKA and William F. BREWER (1992), Discourse organization in the comprehension of temporal order in narrative texts, *Discourse Processes*, 15(3):317–336.

Randolph QUIRK, Sidney GREENBAUM, Geoffrey LEECH, and Jan SVARTVIK (1985), *A Comprehensive Grammar of the English Language*, Longman, London.

Leon STASSEN (1985), *Comparison and Universal Grammar*, Basil Blackwell, Oxford.

Caroline STROBL, Anne-Laure BOULESTEIX, Achim ZEILEIS, and Torsten HOTHORN (2007), Bias in random forest variable importance measures: Illustrations, sources and a solution, *BMC Bioinformatics*, 8(1), http://www.biomedcentral.com/1471-2105/8/25.

Sali A. TAGLIAMONTE and R. Harald BAAYEN (2012), Models, forests, and trees of York English: Was/were variation as a case study for statistical practice, *Language Variation and Change*, 24(02):135–178.

Sandra A. THOMPSON, Robert A. LONGACRE, and Shin Ja J. HWANG (2007), Adverbial clauses, in Timothy SHOPEN, editor, *Language Typology and Syntactic Description*, pp. 237–300, Cambridge University Press, Cambridge.

Jean-Christophe VERSTRAETE (2004), Initial and final position for adverbial clauses in English: The constructional basis of the discursive and syntactic differences, *Linguistics*, 42(4):819–853.

Relja VULANOVIC (2007), On measuring language complexity as relative to the conveyed linguistic information, *SKY Journal of Linguistics*, 20:399–427.

Thomas WASOW (2002), *Postverbal Behavior*, CSLI Publications, Stanford, CA.

Daniel WIECHMANN and Elma KERZ (2013), The positioning of concessive adverbial clauses in English: Assessing the importance of discourse-pragmatic and processing-based constraints, *English Language and Linguistics*, 17(01):1–23.

Query responses

Paweł Łupkowski[1] *and Jonathan Ginzburg*[2]
[1] Department of Logic and Cognitive Science
Institute of Psychology, Adam Mickiewicz University
Poznań, Poland
[2] Laboratoire de Linguistique Formelle (LLF) (UMR 7110) & LabEx-EFL
Université Paris-Diderot (Paris 7)
Sorbonne Paris-Cité

ABSTRACT

In this article we consider the phenomenon of answering a query with a query. Although such answers are common, no large-scale, corpus-based characterization exists, with the exception of clarification requests. After briefly reviewing different theoretical approaches on this subject, we present a corpus study of query responses in the British National Corpus and develop a taxonomy for query responses. We identify a variety of response categories that have not been formalized in previous dialogue work, particularly those relevant to adversarial interaction. We show that different response categories have significantly different rates of subsequent answer provision. We provide a formal analysis of the response categories within the framework of KoS.

Keywords: questions, query responses, corpus study, KoS

1 INTRODUCTION

Responding to a query with a query is a common occurrence, representing on a rough estimate more than 20% of all responses to queries found in the British National Corpus (BNC).[1] Research on dialogue has long recognized the existence of such responses. However, with

[1] In the spoken part of the BNC, using SCoRE (Purver 2001), we found 9,279 ?/? cross-turn sequences, whereas 41,041 ?/. cross-turn sequences, so the ?/? pairs constitute 22.61%.

the exception of one of its subclasses – albeit a highly substantial one – the class of query responses has not been characterized empirically in previous work.

The class that has been studied in some detail is that of Clarification Requests (hereafter referred to as CRs) (see e.g., Purver *et al.* 2001; Rodriguez and Schlangen 2004; Rieser and Moore 2005). However, CRs can be triggered by any utterance, interrogative or otherwise. Researchers working on the semantics and pragmatics of questions (see e.g., Carlson 1983; Wiśniewski 1995) have been aware for many years of the existence of one class of query responses – responses that express questions dependent in some sense on the question they respond to, as in (1a,b). This led Carlson to propose (1d) as a sufficient condition for a query response (cf. (1a,c)).

(1) a. **A:** Who murdered Smith? **B:** Who was in town?
b. **A:** Who is going to win the race? **B:** Who is going to participate?
c. Who killed Smith depends on who was in town at the time.
d. $q2$ can be used to respond to $q1$ if $q1$ depends on $q2$.

How to define question dependence is an important issue if the criterion in (1d) is to have much substance. A number of proposals concerning dependence have been made in the literature, for instance Ginzburg (2012) offers the definition in (2):

(2) $q1$ depends on $q2$ iff any proposition p such that p resolves $q2$, also satisfies p entails r such that r is about $q1$. (Ginzburg 2012, (61b), p. 57)

For Ginzburg, this notion of dependence is an agent-relative notion, given the agent-relativity of the relation resolves.[2] An arguably more open-ended view is taken by Roberts (1996), who suggests that a query move m is relevant in a context where q is the question under discussion if *m is part of a strategy to answer q* (Roberts 1996, p. 17). In similar fashion, Larsson (2002) and Asher and Lascarides (2003) argue

[2] The agent-relativity of the relation resolves is argued for in great detail in Ginzburg 1995. resolves is the answerhood notion implicated in examples such as '… knows where she is' and '… knows who came to the talk', which is, arguably, relativized by agent goals and background knowledge.

that the proper characterization of query responses is pragmatically based. Asher and Lascarides (2003) propose to characterize non-CR query responses by means of the rhetorical relation of *question elaboration* (Q-Elab), with stress on the plan-oriented relation between the initial question and the question expressed by the response. Q-Elab might be informally summarized as follows:

(3) If Q-Elab(α, β) holds between an utterance α uttered by A, where g is a goal associated by convention with utterances of the type α, and the question β uttered by B, then any answer to β must elaborate a plan to achieve g.

Q-Elab, motivated by interaction in cooperative settings, is vulnerable to examples such as those in (4). There is a reading of (4a) that can be characterized by using dependence (*What I like depends on what* YOU *like*), but it can also be used simply as a coherent retort. (4b) could possibly be used in political debate without necessarily involving any attempt to discover an answer to the first question

(4) a. **A:** What do you like? **B:** What do you like?
b. **A:** What is Sarkozy going to do about it? **B:** Well, what is Papandreou going to do?

In the field of the logic of questions we can mention approaches proposed within Inferential Erotetic Logic (IEL) (Wiśniewski 1995, 2013) and inquisitive semantics (INQ) (Groenendijk 2009; Groenendijk and Roelofsen 2011). Although INQ and IEL represent different approaches to questions, both frameworks share a similar treatment of question dependency. In IEL, the central notion used to express dependency between questions is *erotetic implication*. Erotetic implication is a semantic relation between a question, Q, a (possibly empty) set of declarative well-formed formulae, X, and a question, Q_1. Intuitively, erotetic implication ensures the following: (i) if Q has a true direct answer and X consists of truths, then Q_1 has a true direct answer as well ('transmission of soundness[3] and truth into soundness' – cf. Wiśniewski 2003, p. 401), and (ii) each direct answer to Q_1, if true, and if all elements of X are true, narrows down the class for

[3] A question Q is *sound* iff it has a true direct answer (with respect to the underlying semantics).

which a true direct answer to Q can be found ('open-minded cognitive usefulness' – cf. Wiśniewski 2003, p. 402).

In the framework of inquisitive semantics, the dependency relation has been analysed in terms of *compliance*. Roughly speaking, INQ treats questions as sets of possibilities or, in other words, as an issue to be resolved. The intuition behind the notion of compliance is to provide a criterion to "judge whether a certain conversational move makes a significant contribution to resolving a given issue" (Groenendijk and Roelofsen 2011, p. 167).

Other question generation mechanisms in a broadly dialogical context have been proposed in the literature. One such notion is *askability*. The intuition behind askability relates to the issue – when is it reasonable to (publicly) ask a question? Peliš and Majer (2010), applying a dynamic epistemic logic of questions combined with a public announcements' logic for modelling communicative interaction and knowledge revision during this process, propose three conditions that have to be met by an agent in order to ask a question within a group of agents: (i) the answer is not known to the agent posing the question (non-triviality); (ii) each direct answer is considered as possible by the agent (admissibility); and (iii) at least one of the direct answers must be the right one in a given context (context condition).

Van Kuppevelt (1995) proposes *topicality* as the general organizing principle in discourse. The topic (for a discourse unit) is provided by an explicit or implicit question. Van Kuppevelt does not consider simple question – query response pairs, but rather speaks about discourse units. However, the relation between such units is determined by the relation between the previously mentioned topic-providing questions. From the current perspective, the most interesting is the notion of *subtopic-constituting sub-question*:

(5) An explicit or implicit question Q_p is a subtopic-constituting subquestion if it is asked as the result of an unsatisfactory answer A_{p-n} to a preceding question Q_{p-n} and is intended to complete A_{p-n} to a satisfactory answer to Q_{p-n}. (Kuppevelt 1995, p. 125)

Graesser *et al.* (1992) propose four question generation mechanisms for natural settings (especially in educational contexts). The first group consists of *knowledge deficit* questions. The other three groups

are: *common ground* questions, *social coordination* questions and *conversation control* questions. Common ground questions, like 'Are we working on the third problem?' or 'Did you mean the independent variable?', are asked to check whether knowledge is shared between dialogue participants. Social coordination questions relate to different roles of dialogue participants, such as in student – teacher conversations. Social coordination questions are requests for permission to perform a certain action or might be treated as indirect request for the addressee to perform such an action (e.g., 'Could you graph these numbers?', 'Can we take a break now?'). Conversation control questions, as it is indicated by their name, aim at manipulating the flow of a dialogue or the attention of its participants (e.g., 'Can I ask you a question?').

How many kinds of query responses are there and what aspects of context or agents' information states are needed to characterise them? In order to better understand the nature of query response, we ran a corpus study on one large, balanced corpus, the British National Corpus (BNC), and several smaller, more domain-specific corpora, a selection from CHILDES (parent/child interaction; MacWhinney 2000), AMEX (interactions in the travel domain; Kowtko and Price 1989), and BEE (tutor/student interaction; Rosé *et al.* 1999). The results we obtained, discussed in Section 3 of this paper, show that, apart from CRs, dependent questions are indeed by far the largest class of query responses. However, our results reveal also the existence of a number of response categories characterisable neither as dependent questions nor as plan-supporting responses. These include:

- a class akin to what Conversation Analysts refer to as *counters* (Schegloff 2007) – responses that attempt to foist on the conversation an issue that differs from the current discourse topic and
- *situation-relevant responses* – responses that ignore the current topic but address the situation it concerns.

Just as wide coverage is an important goal for any computational theory of sentential grammar (tempered by some notion of 'strong generative capacity', i.e., attaining this in a principled way), the same goal *mutatis mutandis* applies to theories of dialogue; their corresponding aim is to characterise in a principled way the relevance or coherence of a wide range of utterance sequences. Attaining wide coverage for

the particular case of the response space of a query naturally has significant practical importance for dialogue management and the design of user interfaces. Beyond that general goal, a better understanding of e.g., *counters* and *situation-relevant responses* is important for adversarial interaction (e.g., courtroom, interrogation, argumentation, certain games).

The rest of the paper is structured as follows: in Section 2 we present the taxonomy underlying our corpus study; Sections 3 and 4 describe the results, whereas issues concerning annotation reliability are discussed in Section 5; in Section 6 we show how to analyse the relevance of each of the response categories emergent in the corpus study. We do this in terms of information state transitions of two interlocutors participating in a dialogue, using the dialogue framework, KoS.[4] We conclude with a brief cross-theoretical evaluation of potential analyses of the various response classes, and with possibilities for future work.

2 A CORPUS-BASED TAXONOMY OF QUERY RESPONSES

In this section, we present a corpus-based taxonomy of query responses. It was designed on the basis of 1,051 examples of query – query-response pairs obtained from the BNC. Initially, examples were obtained using the search engine SCoRE (Purver 2001). Subsequently, cross talk and tag questions were eliminated manually. The annotation was performed by the first author; we discuss the reliability of this annotation in Section 5. In what follows, we describe and exemplify each class of the resulting taxonomy. To make the description clearer we use $q1$ for the initial question and $q2$ for a question given as a response to $q1$. The taxonomy is focused on the *function of* $q2$ in a dialogue.

2.1 *Clarification requests (CR)*

Clarification requests are all query responses that concern any aspect pertaining to the content or form of $q1$ that was not completely un-

[4] KoS is a toponym – the name of an island in the Dodecanese archipelago – bearing a loose connection to *conversation-oriented semantics* (Ginzburg 2012, p. 2).

derstood. This class contains intended content queries (see example (6a)), repetition requests (example (6b)) and relevance clarifications (example (6c)). In this article, we will not consider this class in detail, mainly because of existing, detailed work on this subject (see e.g., Purver 2006; Ginzburg 2012).

(6) a. **A:** What's Hamlet about?
B: Hamlet? [KPW, 945–946][5]

b. **A:** Why are you in?
B: What?
A: Why are you in? [KPT, 469–471]

c. **A:** Is he knocked out?
B: What do you mean? [KDN, 3170–3171]

2.2 *Dependent questions (DP)*

By a *dependent question*, we understand $q2$ where a dependency statement as in (1d) (see page 246) could be assumed to be true. The following examples illustrate this:

(7) a. **A:** Do you want me to <*pause*> push it round?
B: Is it really disturbing you? [FM1, 679–680]
(cf. *Whether I want you to push it around depends on whether it really disturbs you.)*

b. **A:** Any other questions?
B: Are you accepting questions on the statement of faith at this point? [F85, 70–71]
(cf. *Whether more questions exist depends on whether you are accepting questions on the statement of faith at this point.*)

c. **A:** Does anybody want to buy an Amstrad? <*pause*>
B: Are you giving it away? [KB0, 3343–3344]
(cf. *Whether anybody wants to buy an Amstrad depends on whether you are giving it away.*)

2.3 *'How should I answer this?' questions (FORM)*

This class consists of query responses addressing the issue of the *way* the answer to $q1$ should be given. In other words, whether the answer

[5] This notation indicates the sentence numbers (945–946) of a BNC file (KPW).

to $q1$ will be satisfactory to A depends on $q2$. This relation between $q1$ and $q2$ is illustrated in the following examples. Consider (8a). The way B answers A's question in this case will be dictated by A's answer to $q2$ – whether or not A wants to know details point by point.

(8) a. **A:** Okay then, Hannah, what, what happened in your group?
B: Right, do you want me to go through every point? [K75, 220–221]

b. **A:** Where's that one then?
B: Erm, you know Albert Square? [KBC, 506–507]

c. **A:** Another thing I found out today was do we know where our main supplier of our coffee is.
Any guesses?
B: Which country? [G3U, 251–253]

2.4 *Requests for underlying motivation (MOTIV)*

In the case of *requests for underlying motivation*, $q2$ addresses the issue of the motivation underlying asking $q1$. Whether an answer to $q1$ will be provided depends very much on receiving a convincing answer to $q2$ (i.e., one that provides good reasons for asking $q1$). In this respect this class differs from the previous classes, where we may assume that an agent wishes to provide an answer to $q1$. Most query responses of this kind are of the form 'Why?' (32 out of 41 examples, see e.g., (9a)) but also other formulations were observed (9 out of 41, see e.g., (9b) and (9c)). Most direct questions of this kind are: *What's it got to do with you?*, *What's it to you?*, *Is that any of your business?*, *What's that gotta do with anything?*.

(9) a. **A:** What's the matter?
B: Why? [HDM, 470–471]

b. **A:** Out, how much money have you got in the building society?
B: What's it got to do with you? [KBM, 2086–2087]

c. **A:** Just what the fucking do you think you're doing?
B: Is that any of your business? [KDA, 1308–1309]

2.5 *'I don't want to answer your question' (NO ANSW)*

The role of query responses of this class is to signal that an agent does not want to provide an answer to $q1$, at least at the current stage of the

conversation. Instead of answering $q1$, the agent provides $q2$ and attempts to "turn the table" on the original querier, as in examples (10).

(10) a. **A:** Yeah what was your answer?
B: What was yours? [KP3, 636–637]

b. **A**: come on Stacey get on with it <*pause*> can you move up a bit?
B: What? <*unclear*> why didn't you pull the bench out? [KCG, 378–379]

c. **A**: What about my fifty p?
B: Fucking hell, where's my tenner? [KDA, 3527–3528]

d. **A:** Why is it recording me?
B: Well why not? [KSS, 43–44]

2.6 *Indirect responses (IND)*

This class consists of query responses, which provide (indirectly) an answer to $q1$. Interestingly, providing an answer to $q2$ is not necessary in this case. Consider (11a). By asking the question *Do you know how old this sweater is?*, B clearly suggests that the answer to A's question is negative. Moreover, B does not expect to obtain an answer to his/her question. This may also be observed in examples (11b) (*of course I am Gemini*) and (11c) (*no, my job is not safe*).

(11) a. **A:** Is that an early Christmas present, that sweater?
B: Do you know how old this sweater is? [HM4, 7–8]

b. **A:** Are you Gemini?
B: Well if I'm two days away from your, what do you think? [KPA, 3603–3604]

c. **A:** Is your job safe?
B: Well, whose job's safe? [G5L, 130–131]

Another means of providing indirect answers can also be observed in the corpus data. These are cases where by asking $q2$ an agent already presupposes the answer to $q1$. (12a) illustrates this – we note that a positive answer to $q1$ is presupposed in B's question. A similar situation can be observed in examples (12b) (*no, I have not tasted this*) and (12c) (*I will help you*).

(12) a. **A:** I've got to do the washing up?
B: Shall I, shall I come and help you? [KPU, 1861–1862]

b. **A:** have you tasted this?
B: are they nice? [KPY, 653–654]

c. **A:** Will you help with the <*pause*> the paint tonight?
B: What can I do? [KE4, 3263–3264]

2.7 *'I ignore your question' (IGNORE)*

The final class in the taxonomy involves cases where $q2$ does not address $q1$, but is, nonetheless, related to the situation associated with $q1$. This is evident in example (13c). A and B are playing Monopoly. A asks a question, which is ignored by B. It is not that B does not wish to answer A's question and therefore asks $q2$. Rather, B ignores $q1$ and asks a question related to the situation (in this case, the board game).

(13) a. **A:** Well do you wanna go down and have a look at that now? <*pause*> While there's workmen there?
B: Why haven't they finished yet? [KCF, 617–619]

b. **A:** Just one car is it there?
B: Why is there no parking there? <*unclear*> [KP1, 7882–7883]

c. **A:** I've got Mayfair <*pause*> Piccadilly, Fleet Street and Regent Street, but I never got a set did I?
B: Mum, how much, how much do you want for Fleet Street? [KCH, 1503–1504]

2.8 *Summary*

In this section, a corpus-based taxonomy of query responses was presented. Seven classes of query-responses were described. The classification is focused on the function $q2$ (the question given as a response) serves in relation to $q1$ (the initial question). In what follows, we present the corpus study that led to the classification. First, a study using the BNC is discussed, then the class distribution over specific genres is presented. Subsequently, we consider the issue of annotation reliability.

3 RESULTS

As we noted, this study used a sample of 1,051 query – query response pairs from the BNC. The procedure for obtaining the sample was the following. First the search engine SCoRE was used on the whole spoken part of the BNC using as the search string: `? $ | ? $.`[6] Following this, the search results were checked manually. The collected sample covers a wide range of dialogue domains, including interviews, radio and TV broadcasts, tutorials, meetings, training sessions or medical consultations (blocks D, F, G, H, J, and K of the BNC). The summary of dialogue domains for the sample is presented in Table 1.

Domain	Frequency	% of the Total
free conversation	940	89.44
educational context (lesson, tutorial, training)	36	3.43
meeting (public meeting, seminar, conference)	27	2.57
radio broadcast	25	2.38
interview	15	1.43
medical consultation	4	0.38
TV broadcast	4	0.38
Total	1,051	100

Table 1: Dialogue domains in the research sample (BNC)

The sample was classified and annotated by the first author with the tags presented in Table 2.

Tag	Query-response type
`CR`	clarification requests
`DP`	dependent questions
`FORM`	questions considering means of answering $q1$
`MOTIV`	questions about the motivations underlying asking $q1$
`NO ANSW`	questions aimed at avoiding to answer $q1$
`IND`	questions with a presupposed answer
`IGNORE`	questions ignoring $q1$

Table 2: Tags used to annotate the query – query response sample

[6] The expression '? $' matches any sentence/turn with a question mark at the end and the pipe character matches the break between sentences/turns. For more details about the SCoRE syntax see `http://www.dcs.qmul.ac.uk/imc/ds/score/help.html`.

Table 3: Frequency of query – query response categories in the BNC (The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.)

	Category	Frequency	% of the Total	
1.	CR	832	79.16	
2.	DP	108	10.28	(49.31)
3.	MOTIV	41	3.90	(18.72)
4.	NO ANSW	26	2.47	(11.87)
5.	FORM	16	1.52	(7.31)
6.	IND	22	2.09	(10.05)
7.	IGNORE	6	0.57	(2.74)
	Total	**1,051** (219)	100	

To guide the classification process, we used the following questions:

1. (CR) Is $q2$ a query about something not completely understood in $q1$?
2. (DP) Is it the case that the answer to $q1$ depends on the answer to $q2$?
3. (MOTIV) Does $q2$ address the motivation underlying asking $q1$?
4. (NO ANSW) Is it the case that $q2$ enables the speaker to avoid answering $q1$ while attempting to force the other speaker to answer $q2$ first?
5. (FORM) Is it the case that the way the answer to $q1$ will be given depends on the answer to $q2$?
6. (IND) Is it the case that $q2$ is rhetorical and in this sense does not need to be answered and provides (indirectly) an answer to $q1$?
7. (IGNORE) Does $q2$ relate to the situation described by $q1$?

The results of the classification are presented in Table 3. The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.

The largest class after CRs is DP. What is rather striking is the relatively large frequency of adversarial responses (the classes MOTIV, NO ANSW, and IGNORE).

We also compared which query categories lead to a subsequent answer, either about $q2$ or about $q1$. Bearing in mind that our taxonomy is focused on the function of $q2$ in a dialogue, we would expect the following results.

Category	Ans. to $q2$	Ans. to $q1$
DP	76.85	62.96
MOTIV	78.05	51.22
NO ANSW	80.77	11.54
FORM	68.75	81.25
IND	53.85	100
IGNORE	50	16.67

Table 4:
Answers provided to query responses in % of the total per category

DP Answering $q2$ should lead to answer concerning $q1$. The figures for $q1$ and $q2$ should be similar.

FORM Whether the answer to $q1$ will be useful for A depends on $q2$. $q2$ addresses only the form of the answer to $q1$, so is somewhat less important than with DP. Hence, the number of answers to $q1$ could be higher than for $q2$.

MOTIV Whether an answer to $q1$ will be provided depends on a satisfactory answer to $q2$. The numbers for $q1$ and $q2$ should be comparable, though $q1$ may be somewhat smaller.

NO ANSW Instead of answering $q1$, the agent provides $q2$ and attempts to "turn the table" on the original querier. The original querier is pressured to answer $q2$ and put $q1$ aside. Hence, the numbers for $q1$ should be significantly smaller than for $q2$.

IND $q2$ (indirectly) provides an answer to $q1$, so the latter is answered by definition. Providing an answer to $q2$ is not necessary in this case, so the numbers should be low here.

IGNORE The person posing $q2$ shows a lack of interest in $q1$, but since $q2$ relates to the situation associated with $q1$, there is some expectation that $q2$ be responded to. Thus, the numbers for $q1$ should be significantly smaller than for $q2$. Moreover, the numbers for $q2$ should also be rather low (asking $q2$ is not very cooperative).

The results of the data analysis are presented in Table 4. They are in line with the intuitions underlying the taxonomy.

4 CLASS DISTRIBUTION OVER SPECIFIC GENRES

We conducted our study using the BNC since it is a general corpus with a variety of domains and genres. However, we also wanted to

Table 5:
Frequency of query – query response categories (CHILDES) (The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.)

Category	Frequency	% of the Total	
CR	319	88.12	
DP	11	3.04	(25.58)
MOTIV	2	0.55	(4.65)
NO ANSW	5	1.38	(11.63)
FORM	3	0.83	(6.98)
IND	5	1.38	(11.63)
IGNORE	17	4.70	(39.53)
Total	**362** (43)	100	

Table 6:
Frequency of query – query response categories (BEE) (The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.)

Category	Frequency	% of the Total	
CR	10	22.22	
DP	28	62.22	(80)
NO ANSW	6	13.33	(17.14)
IGNORE	1	2.22	(2.86)
Total	**45** (35)	100	

check how the classes are distributed in more genre-specific corpora. To do this, we decided to study the following corpora:

- the Child Language Data Exchange System (CHILDES; MacWhinney 2000), which contains adult-child conversations,
- the Basic Electricity and Electronics Corpus (BEE; Rosé *et al.* 1999), which contains tutorial dialogues from electronics courses,
- the SRI/CMU American Express dialogues (AMEX; Kowtko and Price 1989), which contains conversations with travel agents.

As with the BNC study, the data was initially obtained by using the search engine SCoRE. Subsequently, cross talk and tag questions were eliminated manually. The annotation was then performed by the first author. 362 examples were obtained from the sample of the CHILDES corpus (files *bates*, *belfast*, and *manchester/anne*); 45 examples were obtained from the whole BEE corpus and 8 from the whole AMEX corpus (the low numbers for BEE and AMEX are caused by the significantly smaller size of these corpora in comparison to BNC and CHILDES). The results of the classification applied to these corpora are presented in Tables 5, 6 and 7. The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.

Category	Frequency	% of the Total
CR	1	12.5
DP	7	87.5 (100)
Total	**8** (7)	100

Table 7:
Frequency of query – query response categories (AMEX) (The parenthesized percentage is the percentage recalculated once the CRs are excluded from the sample.)

As is readily apparent, the DP class is the second largest class in the CHILDES corpus and is the largest class in the task-oriented dialogues obtained from the BEE and AMEX corpora. As for the adversarial classes (MOTIV, NO ANSW, IGNORE), these are very rare in task oriented dialogues. One exception is the NO ANSW class in the case of the BEE corpus. Here the percentage of NO ANSW questions is even higher than in the BNC and in CHILDES. This type of query response is used in a teaching context to encourage a student to provide his/her answer to the teacher's question (e.g., Student: can you remind me which colors mean what on the different resistors?; Tutor: Is that the first thing you need to know? [log-stud31]). When it comes to the CHILDES corpus, a large percentage of IGNORE query responses was observed – in all the examples, it was a child who used this kind of query response. One can also note that for NO ANSW, FORM and IND, the frequency is similar for CHILDES and the BNC. The summary of the distributions for the BNC, CHILDES, AMEX and BEE is presented in Figure 1.

We also compared which query categories lead to a subsequent answer, either about $q2$ or about $q1$. The results are presented in Table 8. In terms of answer analysis, task-oriented dialogues are inter-

Table 8: Answers provided to query responses (CHILDES, BEE and AMEX) in % of the total

	CHILDES		BEE		AMEX	
Category	Ans. to $q2$	Ans. to $q1$	Ans. to $q2$	Ans. to $q1$	Ans. to $q2$	Ans. to $q1$
DP	72.73	45.45	96.43	96.43	100	100
MOTIV	50	0	––	––	––	––
NO ANSW	80	20	100	50	––	––
FORM	100	100	––	––	––	––
IND	0	100	––	––	––	––
IGNORE	70.59	5.88	0	0	––	––

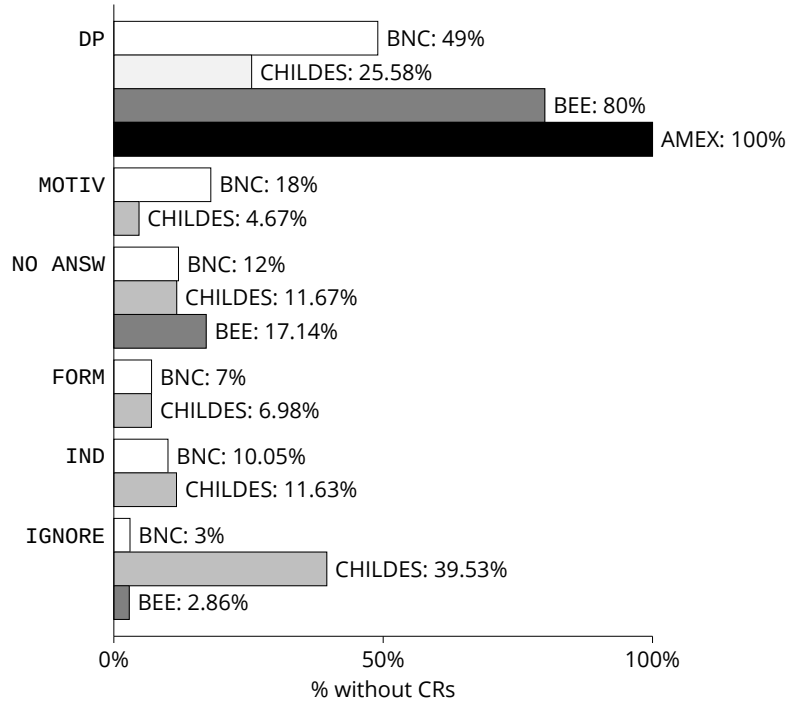


Figure 1: Summary of the distributions in each corpus (in % of the total, without CRs)

esting in the context of the DP query response class. For all observed examples, an answer was provided to $q2$ and to $q1$. The NO ANSW category also behaves in line with the observations for the BNC. We can observe the fulfilment of some of our predictions in the case of the CHILDES corpus. When it comes to interaction with children, neither maintaining attention nor topic continuity are a given, and this can be observed in the data. In the case of DP questions, we still have a high number of answers provided for $q2$, but the number of answers provided to $q1$ is relatively low. As for the IGNORE class, our prediction in general was that the number of answers provided for $q2$ should be low (since the behavior it represents is not very cooperative). However, for CHILDES we observe a high number of provided answers. In our sample it was a child who posed the IGNORE query response, and this offers the basis for an explanation of the results: child-adult conversation generally requires an adult to provide answers to a child's questions, even if the question somehow deviates from the topic of the conversation.

5 ANNOTATION RELIABILITY

As we mentioned above, the annotation process was performed by the first author. In order to check the reliability of the classification process, inter- and intra-annotator studies were performed.

For the inter-annotator study, a sample of 100 randomly chosen examples of query – query responses (retrieved from all four corpora we utilised) was used. The distribution of the classes was in line with the distribution observed by the primary annotator: `CR`: 31 examples; `DP`: 32 examples; `MOTIV`: 11 examples; `NO ANSW`: 8 examples; `FORM`: 5 examples; `IND`: 7 examples; `IGNORE`: 6 examples.

All the examples were supplemented with a context. The guideline for annotators contained explanations of all the classes and examples of question-responses for each category. Also the `OTHER` category was included. The instruction was to annotate the query response to the first question in each example. The control sample was annotated by four annotators (three experienced linguists and a logician with moderate experience in corpus annotation).

The reliability of the annotation was evaluated using κ (Carletta 1996), established by using the R statistical software (R Core Team 2013; version 3.1.2) with the *irr* package (Gamer *et al.* 2012). The interpretation of the kappa values is based on that of Viera and Garrett 2005.

The Fleiss κ for all five annotators was 0.64 (i.e. substantial) with 51% agreement over 100 cases. The agreements between the main annotator and others were all substantial:

1. main and second annotator: $\kappa = 0.67$ with 73% agreement;
2. main and third annotator: $\kappa = 0.65$ with 72% agreement;
3. main and fourth annotator: $\kappa = 0.61$ with 69% agreement;
4. main and fifth annotator: $\kappa = 0.63$ with 70% agreement.

When it comes to detailed analysis of the annotation we start with the `OTHER` category. The annotators were given the option of using this category and, in fact, this option was not used frequently (from a sample of 100 cases): annotator 1: 0, annotator 2: 3, annotator 4: 6, annotator 4: 8, annotator 5: 0.

When we take a closer look at the disagreements between the main and the fourth annotator (the lowest agreement observed) what

becomes clear is that the most problematic cases were DP vs. IGNORE (5 cases). This fact is quite surprising since these categories are rather distant. This suggests that the fourth annotator had misunderstood the category IGNORE.

An analysis of the cases involving the most disagreement suggests that these are not infrequently cases where genuine ambiguities exist given the intentional nature of many of the dialogical relations tying together queries. These naturally enough get exacerbated for annotators required to make decisions in a largely context independent manner.

Thus, (14) was annotated as DP by two annotators, as IGNORE by two annotators, and as NO ANSW by one annotator; in the context of adult/child interaction, IGNORE is possibly more likely – the child observing the same situation as the parent but ignoring politeness in trying to impose the issue momentarily captivating her own interest; at the same time a DP reading is potentially plausible given the plausibility of the assumption *What a fireman does with his axe depends on where his axe is*.

(14) between DP and IGNORE: Parent: what does a fireman do with his axe? Child: where's his axe is?

In a similar fashion, (15) was annotated as DP by three annotators and by two annotators as NO ANSW. Both are potentially plausible classifications: in a cooperative setting (e.g., a dinner enjoyed by a couple in a restaurant) DP is more appropriate (What Norrine will have as a starter depends on what Chris wants), whereas in a more adversarial setting the query response can simply be a means of avoiding the initial question.

(15) between DP and NO ANSW: Chris: What would you like to have start with? Norrine: What do you want?

In light of this issue, we hypothesize that a more satisfying account of annotator reliability for this task would involve developing an annotation model that accommodates ambiguity in annotation, as for instance in work on the basis of crowdsourced labels, as pioneered in Passonneau and Carpenter 2013. This constitutes work we hope to perform in the future.

For the intra-annotator study another control sample of 100 examples was randomly chosen from the data. The distribution of the classes was similar to that in the first control sample. In this case the agreement of the coding between the first annotation and that obtained in the study was substantial ($\kappa = 0.78$).

6 MODELING QUERY RESPONSE CATEGORIES IN KOS

6.1 *Dialogue gameboards, conversational rules, and dialogical relevance*

We offer a formal explication of the coherence that underlies the various different types of query responses within the framework of KoS (Ginzburg 2012). We offer here an analogy to formal syntax. When one discovers a class of constructions C in need of analysis, one means of showing that a given formalism F is adequate involves showing that F's weak (strong) generative capacity properly includes the string set (analysis trees, etc.) corresponding to C. Within dialogue similar desiderata exist, where constructions are replaced by pairs (or longer sequences) of coherent utterances.[7] We seek to show that KoS's notion

[7] An anonymous reviewer for this journal cautions us about the analogy between syntactic grammaticality and dialogue coherence. We agree that the analogy with syntax should not be exaggerated. There are differences. But the analogy between syntactic ungrammaticality and dialogical incoherence is not entirely far-fetched: if one says something incoherent, one could be adjudged to be linguistically incompetent, just as with ungrammaticality. With the latter one can use repair mechanisms to fix ungrammaticality ('I know who did Mary like, I mean who she liked.'), just as with the former (**A:** Who came yesterday? **B:** I'm having a coffee. **A:** Did you hear what I said? **B:** Oh sorry um yes, no I have no idea.). Of course, given the possibility of interpreting incoherence as intended irrelevance, one can often draw that as a possible inference, but grammaticality errors also potentially push us to expect repair, to view the other speaker as momentarily confused etc. The same reviewer points out that weak/strong generative capacity are not necessarily notions to be held as some kind of ideal for scientific explanation. Whatever one thinks of those notions, we think that they were, nonetheless, useful in stimulating syntactic research in the 60s to 80s, in trying to figure out which constructions stretch a given formalism to its limit (e.g., phrase structure grammars and cross-serial dependencies.). Similar considerations apply at present *mutatis mutandis* to formal dialogue theory. We return to this issue and how it relates to cross-theoretical comparison in Section 7.

of coherence properly includes the class of queries and their questions responses, a demonstration that to the best of our knowledge has not hitherto been attempted for any dialogue formalism.[8]

KoS is a framework for dialogue formulated using Type Theory with Records (TTR; Cooper 2005, 2012; Cooper and Ginzburg 2015). It provides formal underpinnings for the information state approach to dialogue management (Larsson and Traum 2003) and underlies dialogue systems such as GoDiS (Larsson 2002) and CLARIE (Purver 2006). On the approach developed in KoS, there is actually no single context. Instead, analysis is formulated at a level of information states, one per conversational participant. The type of such information states is given in (16a). We leave the structure of the private part unanalysed here, as with one exception none of our characterizations makes reference to this; for one approach to *private*, see Larsson 2002. The dialogue gameboard represents information that arises from publicized interactions. Its structure is given in (16b) – the *spkr, addr* fields allow one to track turn ownership, *Facts* represents conversationally shared assumptions, *Pending* and *Moves* represent respectively moves that are in the process of being or have been grounded, *QUD* tracks the questions currently under discussion:[9]

(16) a. TotalInformationState (TIS) $\stackrel{\text{def}}{=}$ $\begin{bmatrix} \text{dialoguegameboard} : \text{DGBType} \\ \text{private} : \text{Private} \end{bmatrix}$

b. DGBType $\stackrel{\text{def}}{=}$ $\begin{bmatrix} \text{spkr: Ind} \\ \text{addr: Ind} \\ \text{utt-time} : \text{Time} \\ \text{c-utt} : \text{addressing(spkr,addr,utt-time)} \\ \text{Facts} : \text{Set(Prop)} \\ \text{Pending} : \text{list(LocProp)} \\ \text{Moves} : \text{list(LocProp)} \\ \text{QUD} : \text{poset(Question)} \end{bmatrix}$

[8] Ginzburg (2010, 2012) sketched such a characterization for the entire class of queries and their responses, though without a detailed corpus study.

[9] The motivation for *Pending* and the type Loc(utionary)Prop(osition) is explained in Section 6.4.

A dialogue gameboard $c1$ will be a record $r1$ such that (17a) holds; by definition this means that:[10] (i) the set of labels of $r1$ needs to be a superset of the set of labels of DGBType and (ii) for each judgement constituent of DGBType $l_k : T_k$, the value r1 gets for that label, denoted by $r1.l_k$, it is the case that $r1.l_k : T_k$. Thus, concretely in this case, $r1$ should have the make-up in (17b), and the constraints in (17c) need to be met:

(17) a. r1 : DGBType

b. $\begin{bmatrix} \text{spkr} = \text{A} \\ \text{addr} = \text{B} \\ \text{utt-time} = \text{t1} \\ \text{c-utt} = \text{p}_{utt(A,B,t1)} \\ \text{Facts} = \text{cg1} \\ \text{Pending} = \langle \text{p1},\ldots,\text{p}k \rangle \\ \text{Moves} = \langle \text{m1},\ldots,\text{m}k \rangle \\ \text{QUD} = \text{Q} \end{bmatrix}$

c. A: Ind, B: Ind, t1 : Time, $\text{p}_{utt(A,B,t1)}$: addressing(A,B,t1),
cg1 : Set(Prop), $\langle \text{p1},\ldots,\text{p}k \rangle$: list(LocProp) $\langle \text{m1},\ldots,\text{m}k \rangle$: list(LocProp),
Q : poset(Question)

The basic units of change are mappings between dialogue gameboards that specify how one gameboard configuration can be modified into another on the basis of dialogue moves. We call a mapping between DGB types a *conversational rule*. The types specifying its domain and its range we dub, respectively, the *preconditions* and the *effects*, both of which are subtypes of DGBType. We explain briefly how this allows one to capture the coherence of responses.

We start by specifying how a question becomes established as in the DGB. The rule in (18) says that given a question q and ASK(A,B,q) being the LatestMove, one can update QUD with q as the maximal element of QUD (henceforth, a *QUD-maximal* element or Max-QUD, the "discourse topic"):[11]

[10] For a more detailed discussion and exemplification, see Cooper and Ginzburg 2015, Section 2.2.

[11] Here, as in the rest of the paper, we make use of *manifest fields* (Coquand *et al.* 2003). A manifest field $\left[\ell = a{:}T\right]$ is a convenient notation for $\left[\ell{:}T_a\right]$ where

(18) Ask QUD-incrementation

$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{spkr} : \text{Ind} \\ \text{addr} : \text{Ind} \\ \text{utt-time} : \text{Time} \\ \text{c-utt} : \text{addressing(spkr,addr,utt-time)} \\ \text{Facts} : \text{Set(Prop)} \\ \text{Pending} : \text{list(LocProp)} \\ \text{q} : \text{Question} \\ \text{Moves} = \langle \text{Ask(spkr,addr,q),m0} \rangle : \text{list(LocProp)} \\ \text{QUD} : \text{poset(Question)} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{spkr} = \text{pre.spkr} : \text{Ind} \\ \text{addr} = \text{pre.addr} : \text{Ind} \\ \text{utt-time} = \text{pre.utt-time} : \text{Time} \\ \text{c-utt} : \text{addressing(spkr,addr,utt-time)} \\ \text{Facts} = \text{pre.Facts} : \text{Set(Prop)} \\ \text{Pending} = \text{pre.Pending} : \text{list(LocProp)} \\ \text{Moves} = \text{pre.Moves} : \text{list(LocProp)} \\ \text{QUD} = \langle \text{pre.q,pre.QUD} \rangle : \text{poset(Question)} \end{bmatrix} \end{bmatrix}$$

In order to avoid the prolixity exemplified in (18), the rules in this paper employ a number of abbreviatory conventions. First, instead of specifying the full value of the list Moves, we usually record merely its first member, which we call 'LatestMove'. Second, the preconditions can be written as a *merge* of two record types $DGBType^{-} \wedge_{merge} PreCondSpec$, one of which $DGBType^{-}$ is a subtype of DGBType and therefore represents predictable information common to all conversational rules; *PreCondSpec* represents information specific to the preconditions of this particular interaction type. Similarly, the effects can be written as a merge of two record types $DGBType^{0} \wedge_{merge} ChangePrecondSpec$, where $DGBType^{0}$ is a supertype of the preconditions and *ChangePrecondSpec* represents those aspects

T_a is a singleton type whose only witness is a. Singleton types are introduced by the clauses in (18).
1. If $a : T$ then T_a is a type.
2. $b : T_a$ iff $b = a$.

of the preconditions that have changed.[12] So we can *abbreviate* (18) as (19b):

(19) a.
$$\begin{bmatrix} \text{pre} & : & \text{PreCondSpec} \\ \text{effects} & : & \text{ChangePrecondSpec} \end{bmatrix}$$

b. Ask QUD-incrementation
$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{q} : & \text{Question} \\ \text{LatestMove} & = & \text{Ask(spkr,addr,q):} \quad \text{LocProp} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{QUD} & = & \langle \text{q,pre.QUD} \rangle : \quad \text{poset(Question)} \end{bmatrix} \end{bmatrix}$$

We can exemplify how this rule works. Assume (20a) to be a record that satisfies the preconditions of the type (19b), in other words it is a record which is of the type assigned to 'pre' in (18) or in abbreviated form in (19b). Hence, it constitutes the appropriate context for Ask QUD-incrementation. The output of that rule is (20b):

(20) a.
$$\begin{bmatrix} \text{spkr} & = & \text{A} \\ \text{c1} & = & \text{p1} \\ \text{addr} & = & \text{B} \\ \text{c2} & = & \text{p2} \\ \text{r} & = & \text{q0} \\ \text{LatestMove} & = & \text{Ask(A,B,q)} \\ \text{QUD} & = & \langle \rangle \\ \text{FACTS} & = & \text{cg1} \end{bmatrix}$$

b.
$$\begin{bmatrix} \text{spkr} & = & \text{A} \\ \text{addr} & = & \text{B} \\ \text{r} & = & \text{q0} \\ \text{LatestMove} & = & \text{Ask(A,B,q0)} \\ \text{QUD} & = & \langle \text{q0} \rangle \\ \text{FACTS} & = & \text{cg1} \end{bmatrix}$$

We also assume an analogue of (19b) for assertion, given in (21). In an interactive setting A asserting p raises the issue p? for B – s/he

[12] This procedure is described in much more general terms using the operation of *asymmetric* merge by Cooper (2016), who shows the use of this operation for a wide range of semantic uses.

can then either decide to discuss this issue (as a consequence of the rule QSPEC introduced below as (24)) or accept it as positively resolved (as a consequence of the rule (22)):

(21) Assertion QUD-incrementation

$$\begin{bmatrix} \text{pre} : & \begin{bmatrix} \text{p} : & \text{Prop} \\ \text{LatestMove} & = \text{Assertion(spkr,addr,p)}: & \text{LocProp} \end{bmatrix} \\ \text{effects} : & \begin{bmatrix} \text{QUD} & = \langle \text{p?,pre.QUD} \rangle : & \text{poset(Question)} \end{bmatrix} \end{bmatrix}$$

An obvious complement to QUD incrementation is a principle controlling QUD downdate. Since QUD consists of questions that are *unresolved* relative to FACTS, QUD downdate is formulated simultaneously with FACTS update: when p is added to FACTS, one needs to verify for all existing elements of QUD that they are not resolved by the new value of FACTS. This joint process of FACTS update / QUD downdate is formulated in (22): given an acceptance or confirmation of p by B, p can be unioned into FACTS, whereas QUD is modified by the function NonResolve. NonResolve is a function that maps a partially ordered set of questions *poset*(q) and a set of propositions P to a partially ordered set of questions *poset*$'(q)$ which is identical to *poset*(q) modulo those questions in *poset*(q) resolved by members of P.

(22) a. Fact Update/ QUD Downdate $\stackrel{\text{def}}{=}$

$$\begin{bmatrix} \text{pre} & : & \begin{bmatrix} \text{p} & : & \text{Prop} \\ \text{LatestMove} = \text{Accept(spkr,addr,p)} & : & \text{LocProp} \\ \text{QUD} = \langle \text{p?,pre.QUD} \rangle & : & \text{poset(Question)} \end{bmatrix} \\ \text{effects} & : & \begin{bmatrix} \text{FACTS} = \text{pre.FACTS} \cup \{\text{p}\} & : & \text{Set(Prop)} \\ \text{QUD} = \text{NonResolve(pre.QUD,FACTS)} & : & \text{poset(Question)} \end{bmatrix} \end{bmatrix}$$

b. NonResolve $\stackrel{\text{def}}{=}$

$$\text{r} : \left(\begin{bmatrix} \text{B} : \text{Ind} \\ \text{P} : \text{set(Prop)} \\ \text{Q} : \text{poset(InfoStruc)} \end{bmatrix} \right) \begin{bmatrix} \text{Q}' : \text{poset(InfoStruc)} \\ \text{c1} : \text{Q}' \subset \text{r.Q} \\ \text{c2} : \forall q_0 \in \text{Q}' \neg \exists f \in P \\ \textit{Resolve}(f, q_0.q) \end{bmatrix}$$

With this in hand, we now turn to explaining how dialogical relevance is handled in KoS. Pre-theoretically, relevance relates an utterance u to an information state I just in case there is a way to suc-

cessfully update I with u. Ginzburg (2012) defines two notions of relevance, a simpler one at the level of moves, i.e. illocutionary contents of utterances, and a somewhat more complex one at the level of utterances. For expository simplicity, we restrict attention here to the former and refer the reader to Ginzburg 2010, 2012 for the more complex notion.

The basic concept introduced here is contextual *m(ove)-coherence* defined in (23a) as applying to m_1 and dgb_0 just in case there is a conversational rule c_1 which maps dgb_0 to dgb_1 and such that dgb_1's LatestMove value is m_1. **Pairwise M(ove)-Coherence**, defined in (23b), applies to a pair of moves m_1, m_2, if m_1 is **M-Coherent** relative to some DGB dgb_0 and there is a sequence of updates leading from LatestMove being m_1 to LatestMove being m_2. Finally, **Sequential M(ove)-Coherence**, defined in (23c), applies to a sequence of moves $m_1, \ldots, m_n$ just in case each successive pair of moves are **Pairwise M-Coherent**:

(23) a. **M(ove)-Coherence**: Given a set of conversational rules $\mathscr{C}$ and a dialogue gameboard dgb_0 : DGBType, a move m_1 : LocProp is $\mathbf{m(ove)}_{\mathscr{C}}{}^{dgb_0}$**-coherent** iff
(i) there exists dgb_1 : *DGBType*, $c_1 \in \mathscr{C}$ such that $c_1(dgb_0) = dgb_1$ and
(ii) $dgb_1.LatestMove = m_1$.

b. **Pairwise M(ove)-Coherence**: Given a set of conversational rules $\mathscr{C}$ two moves m_1, m_2 are $\mathbf{m(ove)}_{\mathscr{C}}$**-pairwise-coherent** iff there exists dgb_0 : *DGBType* and $dgb_i, c_i, (1 \leq i \leq k-1, dgb_i$: *DGBType*, $c_i \in \mathscr{C})$ such that
(i) m1 is $\mathbf{m(ove)}_{\mathscr{C}}{}^{dgb_0}$**-coherent** and
(ii) $c_{i+1}(dgb_i) = dgb_{i+1}$ and $dgb_i.LatestMove = m_1$, whereas $dgb_k.LatestMove = m_2$.

c. **Sequential M(ove)-Coherence**: A sequence of moves m_1 ,..., m_n is $\mathbf{m}_{\mathscr{C}}$**-coherent** iff for any $1 \leq i$ m_i, m_{i+1} are $\mathbf{m}_{\mathscr{C}}$**-pairwise-coherent**.

6.2 *Question accepting responses*

6.2.1 The class DP

We start by characterizing the moves in which the responder B accepts question $q1$ as an issue to be resolved. The potential for DP responses

is explicated on the basis of QSPEC, the conversational rule in (24a). This rule characterizes the contextual background of reactive queries and assertions. It specifies that if q is QUD-maximal, then subsequent to this either conversational participant may make a move constrained to be q-specific, conveying either a proposition p which is a partial answer to q (p is *about* q) or a question $q1$ on which q *depends*, as defined in (24c); one possible definition of *dependence* is given in (24d); intuitively the idea is that if q is dependent on $q1$, then once one knows an answer that resolves $q1$, some information about q (viz. a partial answer) becomes available. This originates in Ginzburg (2012), where formal characterizations of *aboutness* and *resolvedness* can be found.[13]

(24) a. QSPEC

$$\left[\begin{array}{l} \text{pre} : \left[\text{QUD} = \left\langle \text{q}, \text{Q} \right\rangle : \text{poset(Question)}\right] \\ \text{effects} : \text{TurnUnderspec} \wedge_{merge} \\ \left[\begin{array}{l} \text{r} : \text{Question} \vee \text{Prop} \\ \text{R: IllocRel} \\ \text{LatestMove} = \text{R(spkr,addr,r)} : \text{LocProp} \\ \text{c1} : \text{Qspecific(r,q)} \end{array}\right] \end{array}\right]$$

b. q-specific utterance: an utterance whose content is either a proposition p About q or a question $q1$ on which q Depends

c. $q1$ depends on $q2$ iff any proposition p such that p resolves $q2$, also satisfies that for some r p entails r such that r is about $q1$.

Other characterizations of dependency are conceivable and could replace the one given here. For now, we illustrate how dependent responses emerge as relevant responses: in (25) A asks $q1$, responded to by B with a dependent question response $q2$. A answers $q2$, which gets accepted by B, leading to an answer to $q1$:

[13] We notate the underspecification of the turn holder as *TurnUnderspec*, an abbreviation for the following specification which gets unified together with the rest of the rule:

$$\left[\begin{array}{lll} \text{PrevAud} = \left\{\text{pre.spkr,pre.addr}\right\} & : & \text{Set(Ind)} \\ \text{spkr} & : & \text{Ind} \\ \text{c1} & : & \text{member(spkr, PrevAud)} \\ \text{addr} & : & \text{Ind} \\ \text{c2} & : & \text{member(addr, PrevAud)} \\ & & \wedge\ \text{addr} \neq \text{spkr} \end{array}\right]$$

(25) **A**(1)**:** Who should we invite for tomorrow?
B(2)**:** Who will agree to come?
A(3)**:** Helen and Jelle and Fran and maybe Sunil.
B(4)**:** (a) I see. (b) So, Jelle I think.
A(5)**:** OK.

(26)

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = $\langle\rangle$	
	QUD = $\langle\rangle$	
	FACTS = cg1	
1	LatestMove := Ask(A,B,$q1$)	
	QUD : = $\langle q1\rangle$	Ask QUD-incrementation
2	LatestMove := Ask(B,A,$q2$)	QSPEC
	Influence($q2, q1$)	
	QUD : = $\langle q2, q1\rangle$	Assert QUD-incrementation
3	LatestMove := Assert(A,B,$p2$)	QSPEC
	(About($p2, q2$))	
	QUD := $\langle p2?, q2, q1\rangle$	Assert QUD-incrementation
4a	LatestMove := Accept(B,A,$p2$)	Accept
	FACTS := cg1 $\cup$ $\{p2\}$	
	QUD := $\langle q1\rangle$	Fact update/QUD downdate
4b	LatestMove := Assert(B,A,$p1$)	QSPEC
	(About($p1, q1$)	
	QUD := $\langle p1?, q0\rangle$	Assert QUD-incrementation
5	LatestMove := Accept(A,B,$p1$)	Accept
	FACTS := cg1 $\cup\{p1, p2\}$	
	QUD := $\langle q1\rangle$	Fact update/QUD downdate

6.2.2 The class IND

This class consists of query responses, where $q2$ is posed rhetorically, and which provide (indirectly) an answer to $q1$. In other words, $q2$ is posed in a context where an answer that resolves $q2$ can be assumed to be in FACTS – the repository of shared assumptions in the DGB, and, moreover, this answer entails a (resolving) answer to $q1$. Handling this class does not involve any additional conversational rules; it requires solely two independently needed additions to the setup described hitherto, a mechanism for rhetorical interpretation of questions and a means of accommodating *indirect* answers:

1. **Rhetorical interpretation of interrogatives**: a rhetorical use arises when an interrogative $q1$ is used in a context where the DGB contains a fact f that resolves $q1$. There are, in fact, two possible ways this can be satisfied: either the question has been discussed and a resolving answer provided; alternatively, certain answers are default values for such uses – a negative universal for wh-questions ('Who cares?', 'Who knows?'), one of the polar values for polar-questions ('Do I care?', 'Is the Pope Catholic?'). One possible treatment is proposed in Ginzburg 2012, §8.3.5: given a context in which a proposition p resolving a question q is presupposed, one postulates a root construction that assigns a clause denoting a question q the force of a reassertion of p, where p, a proposition resolving q, is a presupposition satisfied by the context.
2. **Indirect answers**: we need to allow q-specificity to include propositions that are *indirectly* about q. An explicit account of indirectness would take us too far afield here, but see e.g., Asher and Lascarides 1998 and Ginzburg 2012, §8.3.3–8.3.5 for discussion relating to questions in dialogue.

We exemplify how this works in (27): A asks the question $p0$? ('Is B's job safe?'). B responds with a reassertion of a question $q1$ ('Whose job is safe?'), which in this context reasserts the proposition $p1$ 'No one's job is safe.', which in particular resolves the question $p0$?. This explains *inter alia* why there is no need for A to respond to B's question.

(27) **A:** Is your job safe?
B: Whose job's safe?

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = ⟨⟩	
	QUD = ⟨⟩	
	FACTS = cg1	
1	LatestMove := Ask(A,B,$p0$?)	
	QUD := ⟨$p0$?⟩	Ask QUD-incrementation
2	LatestMove := ReAssert(B,A,$p1$)	QSPEC
		Resolve($p1, q1$)

We mentioned previously a subclass of IND – query responses where $q2$ *presupposes* an answer that resolves $q1$. We do not offer a detailed analysis of this subclass here, but they could, for instance, be accommodated by a slight adjustment of q-specificity which licensed responses whose content semantically presupposed a proposition p about $q1$.

6.2.3 The class FORM

This class consists of query-responses addressing the issue of the *way* the answer to $q1$ should be given. The class FORM raises interesting issues since it seems to be the sole class whose coherence intrinsically involves reasoning by the responder about the original querier's unpublicized intentions. One possible explication would be in terms similar to the relation Q-Elab (Asher and Lascarides 2003). Perhaps the simplest way to do this in the current setting would be, following Larsson (2002), to widen the definition of q-specificity so that it is relative to an information state which provides a notion of the agent's plan, decomposed into a sequence of questions to be resolved:

(28) u is q-specific relative to an information state I: an utterance whose content is either a proposition p About q or a question $q1$ on which q Depends or a question q'_1 which is a component of I.plan

One could try and collapse DP and FORM. One reason not to do this is precisely that the former does not intrinsically involve reasoning about intentions and so, in principle, its coherence should be easier to compute.

6.3 *Adversarial query responses*

Adversarial query responses are challenging for most semantic theories of questions, for reasons we discuss below. Common to all three classes is a lack of acceptance of $q1$ as an issue to be discussed. In MOTIV-type responses the need/desirability to discuss $q1$ is explicitly posed, in NO ANSW-type responses there is an implicature that $q1$ is of lesser importance/urgency than $q2$, whereas for IGNORE type responses there is an implicature that q1 as such will not be addressed.

One commonality between MOTIV and NO ANSW worth noting is that in both cases $q1$ actually needs to be added to QUD at the outset. One might think that a consequence of a responder's failure to accept q for discussion is that q will only resurface if explicitly reposed. There is evidence, however, that actually q remains in a conversational participant's QUD even when not initially adopted, as its very posing makes it temporarily DGB available. In (29), where move (2) could involve either a MOTIV query (2a), or a NO ANSW query (2b), the original question has definitely *not* been re-posed and yet B still has the option to address it, which s/he should be unable to do if it is not added to his gameboard before (29(2)).

(29) **A:** Who are you meeting next week?
B(2)**:** (2a) What's in it for you? / (2b) Who are *you* meeting next week?
A: I'm curious.
B: Aha.
A: Whatever.
B: Oh, OK, Jill.

We turn to a discussion of the coherence of each class, starting with MOTIV and NO ANSW, leaving IGNORE for later, given a certain additional complexity it embodies.

6.3.1 The class MOTIV

MOTIV utterances are an instance of metadiscursive interaction – interaction about what should or should not be discussed at a given point in a conversation, as exemplified by utterances such as (30):

(30) a. I don't know.
b. Do we need to talk about this now?
c. I don't wish to discuss this now.

A natural way to analyze such utterances is along the lines of a rule akin to QSPEC given in (24): q being MaxQUD gives (the responder) B the right to follow up with an utterance specific to the issue we could paraphrase informally as *?WishDiscuss(B,q)*.[14] Such a rule is formulated in (31), where the notation

[14] We are formulating this rule asymmetrically with respect to the interlocuters, in contrast to QSPEC, since A posing $q1$ means that A keeping the turn

$$\text{`QUD} = \left\langle \text{Max} = \left\{ \text{?WishDiscuss(B,q1)}, q1 \right\}, \text{Q} \right\rangle'$$

indicates that both ?WishDiscuss(B,q1) and $q1$ are maximal in QUD, unordered with respect to each other. The motivation for this latter is the need to integrate $q1$ in context, as per (29) above.

(31) MetaDiscussing q1

$$\left[\begin{array}{ll} \text{pre} : \left[\text{QUD} = \left\langle q1, \text{Q} \right\rangle : \text{poset(Question)} \right] & \\ \text{effects} : \left[\begin{array}{l} \text{spkr} = \text{pre.addr} : \text{Ind} \\ \text{addr} = \text{pre.spkr} : \text{Ind} \\ \text{r} : \text{Question} \vee \text{Prop} \\ \text{R: IllocRel} \\ \text{Moves} = \left\langle \text{R(spkr,addr,r)} \right\rangle \oplus \text{pre.Moves} : \text{list(LocProp)} \\ \text{c1} : \text{Qspecific(R(spkr,addr,r),?WishDiscuss(spkr,pre.MaxQUD))} \\ \text{QUD} = \left\langle \begin{array}{l} \text{Max} = \left\{ \text{?WishDiscuss(spkr,}q1\text{)}, q1 \right\}, \\ \text{Q} \end{array} \right\rangle : \text{poset(Question)} \end{array}\right] \end{array}\right]$$

In case information is accepted indicating negative resolution of ?WishDiscuss(B,$q1$), then $q1$ may be downdated from QUD. This involves a minor modification of the Fact Update/QUD Downdate rule (see (22) above).[15]

We exemplify (31) in two ways. First, with a variant of (29), where B's rejection of a question leads to the downdating of $q1$; then, with a very similar analysis of a MOTIV query response. does not wish to discuss $q1$, hence s/he accommodates ?WishDiscuss(B, $q1$)

and uttering (30b,c) would be somewhat incoherent; the status of (30a) as a follow up to $q1$ is somewhat different: in the commonest case, where a query is posed because the querier does not know the answer, (30a) is redundant and somewhat infelicitous. In cases where $q1$ is uttered in the spirit of 'Here is an interesting issue to discuss', (i) seems acceptable:

(i) *I* don't know.

Whether this should be taken to imply that (30a) and (30b,c) should be licensed by distinct mechanisms is an issue we will not try to resolve here.

[15] All that this involves is a modification of the function NonResolve which fixes the value of QUD after the fact update: in its new definition it maps a poset of questions *poset*(q) and a set of propositions P to a poset of questions *poset*$'$(q) which is identical to *poset*(q) modulo those questions in *poset*(q) resolved by

into QUD and offers an utterance concerning this issue. A accepts B's assertion, so using the new version of fact-update/qud-downdate $q1$ can be downdated and either conversationalist could introduce a new topic, as in (32):

(32) **A**(1)**:** Who are you meeting next week?
B(2)**:** No comment.
A(3)**:** I see.
A/**B**(4)**:** What are you doing tomorrow?

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = ⟨⟩	
	QUD = ⟨⟩	
	FACTS = cg1	
1	LatestMove := Ask(A,B,$q1$)	
	QUD := ⟨$q1$⟩	Ask QUD-incrementation
2	LatestMove := ⟨ Assert(B,A,$p1$) ⟩	Discussing u?
	QUD := ⟨p1? ≻ ?WishDiscuss($q1$), $q1$⟩	Assertion QUD-incrementation
3	LatestMove := ⟨ Assert(B,A,$p1$) ⟩	Accept
	QUD := ⟨⟩	Fact update/QUD downdate
	FACTS := cg1 ∪ {p1}	

We suggest that a dialogue like (33) works in a similar way: A's answer to B's question (33(2)) can satisfy B, which will lead to the question ?WishDiscuss(B, q1) being positively resolved, enabling B to downdate it from her QUD and address the question (33(1)).

---

members of P, **as well as those questions q for whom ?WishDiscuss(q) is negatively resolved.**

$$\left[\begin{array}{l} \text{pre}: \left[\begin{array}{l} \text{p} : \text{Prop} \\ \text{LatestMove} = \text{Accept(spkr,addr,p)} \\ \text{QUD} = \left\langle \text{p?,pre.QUD} \right\rangle : \text{poset(Question)} \end{array}\right] \\ \text{effects}: \left[\begin{array}{l} \text{FACTS} = \text{pre.FACTS} \cup \{\text{p}\} : \text{Set(Prop)} \\ \text{QUD} = \text{NonResolve(pre.QUD,FACTS).Q}' \\ : \text{Poset(Question)} \end{array}\right] \end{array}\right]$$

$$\text{NonResolve} \stackrel{\text{def}}{=} \text{r}: \left(\left[\begin{array}{l} \text{B} : \text{Ind} \\ \text{F} : \text{set(Prop)} \\ \text{Q} : \text{poset(Question)} \end{array}\right]\right) \left[\begin{array}{l} \text{Q}' : \text{poset(InfoStruc)} \\ \text{c1} : \text{Q}' \subset \text{r.Q} \\ \text{c2} : \forall q_0 \in \text{Q}' \neg \exists f \in \text{F} \\ \mathit{Resolve}(f, q_0.q) \\ \vee \mathit{Resolve}(f, ?\mathit{WishDiscuss}(r.B, q_0.q)) \end{array}\right]$$

B not being satisfied with A's answer is entirely similar to (32) *mutatis mutandis*:

(33) **A**(1)**:** Who are you meeting next week?
B(2)**:** Why?
A(3)**:** I need to know which refreshments to buy.

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = $\langle\rangle$	
	QUD = $\langle\rangle$	
	FACTS = cg1	
1	LatestMove := Ask(A,B,$q1$)	
	QUD : = $\langle q1\rangle$	Ask QUD-incrementation
2	LatestMove := $\langle$ Ask(B,A,$q2$) $\rangle$	Discussing u?
	QUD := $\langle q2 \succ$?WishDiscuss(B,$q1$),$q1\rangle$	Ask QUD-incrementation
3	LatestMove := Assert(A,B,$p1$)	QSPEC
	(About($p1, q2$))	
	QUD := $\langle p1? \succ q2 \succ$?WishDiscuss(B,$q1$),$q1\rangle$	Assert QUD-incrementation
4a	LatestMove := Accept(B,A,$p1$)	Accept
	FACTS := cg1 $\cup$ {$p1$}	
	QUD := $\langle q0\rangle$	Fact update/QUD downdate

6.3.2 The class NO ANSW

NO ANSW-queries can be analysed in a fairly similar fashion. The main challenge such queries pose is to consider the coherence relation between $q1$ and $q2$. Unlike IGNORE, where it seems like there is little that need connect the two questions, save for some reference to the situation associated with $q1$, for NO ANSW the questions seem to need a fairly tight link. A tentative characterization of this link is the following: $q1$ and $q2$ are not *dependent* on each other, but instead there exists a third question, $q3$, such that $q3$ depends on $q1$ and $q3$ depends on $q2$. The rationale behind this characterization is that by responding with $q2$ B provides (a) an issue that is not unconnected with $q1$, but (b) it is informationally not subservient to $q1$. Hence, given that

$q3$ is (or can be accommodated to be) the general topic under discussion, $q2$ has an arguable case to being at least as discussion worthy as $q1$:[16]

(34) a. $q1$ = what do you (B) like? $q2$ = what do you (A) like? $q3$ = Who likes what?

b. $q1$ = Why should we buy that scanner? $q2$ = Why should we not buy that scanner? ; $q3$ = Should we buy that scanner?

Based on this, we define the relation of being *unifiably coherent*:

(35) Given $q1$, $q2$: Question $q1$ and $q2$ are unifiably coherent iff

1. Neither $q1$, nor $q2$ depend on the other: $\neg$Depend($q2, q1$) $\wedge$ $\neg$Depend($q1, q2$)
2. There exists $q3$: Question which depends on both $q1$ and $q2$: Depend($q3, q1$) $\wedge$ Depend($q3, q2$)

The potential for making such queries can be captured by the conversational rule in (36). Given that $q1$ is MaxQUD, the responder may respond with $q2$, assuming it to be unifiably coherent with $q1$. The immediate effect of this is to update QUD with the issue ?WishDiscuss(B,$q1$).

[16] An anonymous reviewer for this journal points out the following exchange as problematic for our taxonomy, suggesting that it is 'fully coherent given the sequel but the pair does not seem to fit any of the schemes':

(i) **A:** Are you coming on Friday?
B: Did you ever consider quarks?
A: No.
B: Well you should for your work and Friday there will be a lecture that is just right for you. I may be there myself.

Actually, we would suggest that this example would be classified as a `NO ANSW` by the annotation criteria we offer (since B views A's question as less important to consider than his and one could eliminate B's answer at the end without affecting coherence.). Nonetheless, it calls into question our formalized definition for `NO ANSW` in that it is not clear that the $q2$ and $q1$ are *unifiably coherent*. One might use this (constructed) example to argue for weakening the unifiable coherence clause. At the same time, it seems likely that B's response would initially be viewed as incoherent by A and this should be reflected by e.g., response time, frowning etc.

(36) Challenging q1

$$\left[\begin{array}{l} \text{pre} : \left[\text{QUD} = \left\langle q1, \text{Q}\right\rangle : \text{poset(Question)}\right] \\ \text{effects} : \left[\begin{array}{l} \text{spkr} = \text{pre.addr} : \text{Ind} \\ \text{addr} = \text{pre.spkr} : \text{Ind} \\ q2 : \text{Question} \\ \text{Moves} = \left\langle \text{Ask(spkr,addr,}q2\text{)}\right\rangle \oplus \text{pre.Moves} : \text{list(LocProp)} \\ \text{c1} : \text{Unifiablycoherent(q1,}q2\text{)} \\ \text{QUD} = \left\langle \text{Max} = \left\{\begin{array}{l}\text{?WishDiscuss(B,}q1\text{)},\\ q1\end{array}\right\}, \text{Q}\right\rangle : \\ \text{poset(Question)} \end{array}\right] \end{array}\right]$$

In (37)[17] A asks $q1$, B responds with $q2$ that unifies coherently with $q1$ via, for example, the issue $q3$ = 'Should they wait?'. A responds to $q2$ and then B's second utterance can be understood as addressing $q2$. If A accepts (4), $q2$ can be downdated and, consequently $q1$ and ?WishDiscuss(B,$q1$) as well – $q1$ has also been resolved, and hence ?WishDiscuss(B,$q1$) could be taken to be resolved as well.[18]

(37) **A**(1)**:** Why won't they wait?
B(2)**:** Why should they?
A(3)**:** I waited.
B(4)**:** They have lives of their own.

6.4 *DGB divergence: Ignore and Clarification Requests*

Both clarification requests (CRs) and IGNORE type responses involve reasoning that requires reference to two DGBs. CRs arise due to a mismatch that occurs between what the speaker assumes her/his interlocutor's linguistic/contextual knowledge is and what it actually is;

[17] Inspired by the BNC example:
Eddie: But it's something, something in you, you have to rush don't they? Why won't they wait? Unknown: Why should they? Eddie: Why should they? Unknown: No, why should they? Eddie: I have Unknown: Take the rest of it Unknown: <*unclear*> Eddie: pleasure spending Unknown: <*unclear*> Unknown: No why, they've got lives of their own Eddie: Well Sally: let them live it, don't want saving for the children, no, they don't want nothing Eddie: Well Unknown: They've had far more than what we've ever had [KCF, 3584–3596].

[18] A general principle linking the downdating of ?WishDiscuss(B,q0) once $q0$ has been downdated should be introduced, though we will not do so here.

consequently, in the immediate aftermath of such an utterance – before the mismatch becomes manifest, the speaker updates her/his IS with the query s/he posed and the addressee updates hers/his with the clarification question s/he calculated.

Similarly, in the case of IGNOREs the initial speaker updates their information state with the query s/he posed and, ignoring this, the addressee updates hers/his with the situationally relevant question s/he has decided to pose.

6.4.1 Clarification Requests

We start by discussing CRs since they have been studied in great detail, see Ginzburg and Cooper 2004; Schlangen 2004; Purver 2006; Ginzburg *et al.* 2014; we will summarize briefly the most detailed account we are aware of, that provided in Ginzburg 2012. This will provide tools enabling us to analyse IGNORE-type responses.

Integrating clarification interaction into the DGB involves two modifications to the representations we have been using so far. One minor modification, drawing on an early insight of Conversation Analysis (Schegloff 2007), is that repair can involve 'putting aside' an utterance for a while, a while during which the utterance is repaired. That in itself can be effected without further ado by adding further structure to the DGB, specifically the field we call *PENDING*. 'Putting the utterance aside' raises the issue of *what is it that we are 'putting aside'*. In other words, how do we represent the utterance? The requisite information needs to be such that it enables the original speaker to interpret and recognize the coherence of the range of possible clarification queries that the original addressee might make. Ginzburg (2012) offers detailed arguments on this issue, including considerations of the phonological/syntactic parallelism exhibited between CRs and their antecedents, and the existence of CRs whose function is to request repetition of (parts of) an utterance. Taken together with the obvious need for PENDING to include values for the contextual parameters specified by the utterance type, Ginzburg concludes that the type of PENDING combines tokens of the utterance, its parts, and of the constituents of the content with the utterance type associated with the utterance. An entity that fits this specification is the *locutionary proposition* defined by the utterance. A locutionary proposition is a propo-

sition whose situational component is an utterance situation, typed as in (38a), and will have the form in (38b):

(38) a. $LocProp \stackrel{\text{def}}{=} \begin{bmatrix} \text{sit} & : & Sign \\ \text{sit-type} & : & RecType \end{bmatrix}$

b. $\begin{bmatrix} \text{sit} = \text{u} \\ \text{sit-type} = T_u \end{bmatrix}$

Here T_u is a grammatical type for classifying u that emerges during the process of parsing u. It can be identified with a *sign* in the sense of Head-driven Phrase Structure Grammar (HPSG) (Pollard and Sag 1994).

How then can one characterize the relevance of CRs in this setup? Corpus studies of CRs (Purver *et al.* 2001; Rodriguez and Schlangen 2004; Rieser and Moore 2005) indicate that the subject matter of CRs is, in practice, restricted to three classes: CRs requesting repetition, CRs requesting confirmation, and CRs which query the intended content of a sub-utterance. This means that the potential for CRs can be modelled in terms of a small number of schemas (*Clarification Context Update Rules* (CCURs)) of the form: "if u is the maximal element of PENDING (*MaxPENDING*) and $u0$ is a constituent of u, add the clarification question $CQ^i(u0)$ into QUD.", where '$CQ^i(u0)$' is one of the three types of clarification question (repetition, confirmation, intended content) specified with respect to $u0$.

(39) is a simplified formulation of one CCUR, Parameter identification, which allows B to raise the following issue about A's sub-utterance $u0$: *what did A mean by u0?*:

(39) Parameter identification:

$$\begin{bmatrix} \text{pre} & : & \begin{bmatrix} \text{Spkr : Ind} \\ \text{MaxPENDING : LocProp} \\ \text{u0} \in \text{MaxPENDING.sit.constits} \end{bmatrix} \\ \text{effects} & : & \begin{bmatrix} \text{MaxQUD} = \lambda x \text{Mean(A,u0,x) : Question} \\ \text{LatestMove : LocProp} \\ \text{c1: CoPropositional(LatestMove.cont,MaxQUD)} \end{bmatrix} \end{bmatrix}$$

Here *CoPropositionality* for two questions means that, modulo their domain, the questions involve similar answers: for instance

'Whether Bo left', 'Who left', and 'Which student left' (assuming Bo is a student.) are all co-propositional. More generally, the definition is given in (40):

(40) Two utterances u_0 and u_1 are *co-propositional* iff the questions $q0$ and $q1$ they contribute to QUD are co-propositional.

a. qud-contrib(m0.cont) is m0.cont if m0.cont : Question

b. qud-contrib(m0.cont) is ?m0.cont if m0.cont : Prop[19]

c. $q0$ and $q1$ are co-propositional iff there exists a record r such that $q_0(r) = q_1(r)$.

Parameter Identification, as given in (39), underpins CRs such as (41b–41c) as follow-ups to (41a). Corrections can also be dealt with, as in (41d), since they address the issue of what A meant by u.

(41) a. **A:** Is Bo here?

b. **B:** Who do you mean 'Bo'?

c. **B:** Bo? (= Who is 'Bo'?)

d. **B:** You mean Jo.

To exemplify our account of how CRs get integrated in context, we exemplify in Figure 2 how the same input leads to distinct outputs on the "public level" of information states. In this case, it arises due to differential ability to anchor the contextual parameters. The utterance $u0$ has three sub-utterances, $u1$, $u2$, $u3$, given in Figure 2 with their approximate pronunciations. A can ground her/his own utterance since s/he knows the values of the contextual parameters, which we assume here for simplicity include the speaker and the referent of the sub-utterance *Bo*. This means that the locutionary proposition associated with $u0$ – the proposition whose situational value is a record that arises by unioning $u0$ with the witnesses for the contextual parameters and whose type is given in Figure 2 – is true. This enables the "canonical" illocutionary update to be performed: the issue *whether b left* becomes the maximal element of QUD. In contrast, assume that B lacks a witness for the referent of *Bo*. As a result, the locutionary proposition associated with $u0$ which B can construct is not true. Given this, B uses the CCUR *parameter identification* to build a context appropriate for a clarification request: B increments QUD with the issue

[19] Recall from the assertion protocol that asserting p introduces p? into QUD.

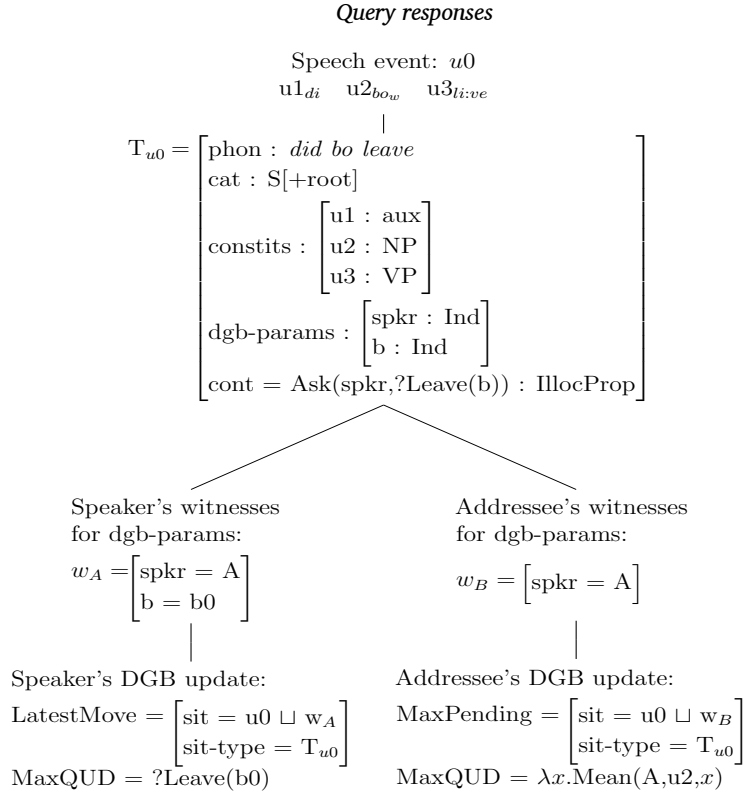


Figure 2: A single utterance giving rise to distinct updates of the DGB for distinct participants

λxMean(A,$u2$,x), and the locutionary proposition associated with $u0$ that B has constructed remains in PENDING.

6.4.2 The class IGNORE

The final class we consider is that of IGNORE-type responses. Such responses implicate that $q1$ will not be addressed, somewhat analogously to the classic Gricean floutings of relevance (A: Bob is an embarrassment B: It's very hot in here). Nonetheless, the effect such responses have is different from Gricean floutings, since these responses are situationally relevant, which appears to minimize significantly the potential impoliteness associated with ignoring $q1$. We think the difference between these two cases should be experimentally testable (e.g., response times for Gricean floutings should be significantly larger than for IGNOREs).

The conversational rule we propose allows the potential for $q2$ *and* captures the implicature concerning $q1$ being ignored. The formulation of such a rule presupposes a notion of *relevance* between the

content of an utterance ($q2$) and the current context. We assume here the notion of relevance we mentioned in Section 6.1 and define *irrelevance* as *failure of relevance*: for an utterance u being IrRelevant to an information state I amounts to: there is *no way* to successfully update I with u. At the same time we assume that $q2$ being *situationally* relevant means that the open proposition component of $q2$ is of the form $p2(\ldots a \ldots)$, with a being in the situation which concerns $q1$.

This involves positing a conversational rule along the lines of (42) – given that (the content of) MaxPENDING – the most recent utterance, as yet ungrounded, hence maximal in PENDING – is *irrelevant* to the DGB but situationally relevant to $q2$, one can make MaxPENDING into LatestMove while updating Facts with the fact that the speaker of MaxPENDING does not wish to discuss MaxQUD:

(42) Ignoring questions

$$\left[\begin{array}{ll} \text{pre}: & \left[\begin{array}{l} \text{a} : \text{IND} \\ \text{s1} : \text{SIT} \\ \text{q1} = (\text{G})\left[\begin{array}{l}\text{sit} = \text{s1} \\ \text{sit-type} = \text{T}\end{array}\right] : \text{Question} \\ \text{q2} = (\text{G1})\left[\begin{array}{l}\text{sit} = \text{s} \\ \text{sit-type} = \left[\text{c} : \text{p2(a)}\right]\end{array}\right] : \text{Question} \\ \text{In(s1,a)} \\ \text{dgb} = \left[\begin{array}{l}\text{MaxQUD} = \text{q1} : \text{Question} \\ \text{MaxPENDING}^{content} = \text{q2} : \text{Question}\end{array}\right] : \text{DGBType} \\ \text{c: IrRelevant(MaxPENDING}^{content}\text{,dgb)} \end{array}\right] \\ \text{effects}: & \left[\begin{array}{l}\text{LatestMove} = \text{pre.MaxPENDING} : \text{LocProp} \\ \text{Facts} = \text{pre.Facts} \cup \\ \left\{\neg\ \text{WishDiscuss(pre.spkr,pre.MaxQUD)}\right\}. \end{array}\right] \end{array}\right]$$

Note that this does not make the *unwillingness to discuss* the *content* of the offending utterance; it is merely an inference. Still this inference will allow MaxQUD to be downdated, via fact update/question downdate, as was discussed with respect to `MOTIV` moves and the rule MetaDiscussing q1. We exemplify this with respect to (43).

(43) **A:** Is there just one car there?
B: Why is there no parking there?

As we noted earlier, given the contextual mismatch involved, in order to describe such dialogues one needs to consider the dialogue on the basis of two distinct DGBs. One possible evolution of A's DGB is this: A utters $q1$, which becomes MaxQUD; s/he then encounters B's response; A applies the rule *Ignoring questions*, which leads to $q1$'s downdate, $q2$ becomes MaxQUD.

(44)

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = ⟨⟩	
	QUD = ⟨⟩	
	FACTS = cg1	
1	LatestMove := Ask(A,B,q1)	
	QUD : = ⟨$q1$⟩	Ask QUD-incrementation
2	LatestMove := ⟨ Ask(B,A,q2) ⟩	
	FACTS := FACTS ∪ ¬WishDiscuss(B,q1)	Ignoring questions
	QUD := ⟨ ⟩	FACTS update/QUD downdate
	QUD := ⟨ q2 ⟩	Ask QUD-incrementation

To the extent B wishes to ignore A's utterance, we do not need any additional machinery, save for a general principle needed in any case for a variety of other not necessarily linguistic events (e.g., in case one of the participants A burps, spits, or farts) – pretense that an event was not perceived. Assuming this, a possible evolution of B's DGB is as in (45): B pretends that A's utterance $u1$ did not take place, s/he utters $q2$, which relates to the situation A and B are jointly perceiving; $q2$ becomes MaxQUD:

(45)

Utt.	DGB Update (Conditions)	Rule
initial	MOVES = ⟨⟩	
	QUD = ⟨⟩	
	FACTS = cg1	
1	LatestMove := ⟨ Ask(B,A,q2) ⟩	
	QUD := ⟨ q2 ⟩	Ask QUD-incrementation

6.5 *Summary*

In this section we have shown how to characterize the relevance of the range of possible query responses $q2$ to an initial query $q1$ using DGB-based dynamics. The relevance of dependent questions is characterized in terms of QUD and the dependence relation, a relation defined on pairs of questions; `IND` uses the same contextual setup (plus mechanisms independently needed for accommodating rhetorical uses of interrogatives and indirect/presupposed answers); accommodating `FORM` involves reasoning similar to `DP`, but requires making reference to the issues constituting an interlocutor's plan; `MOTIV` and `NO ANSW` involve postulating additional conversational rules that make reference to the issue of whether $q2$'s speaker wishes to discuss $q1$, leaving this and $q1$ as issues simultaneously under discussion, hence this makes crucial use of QUD being a partially ordered set; `NO ANSW` also involves computing an additional coherence relation 'unifiable coherence' that needs to relate $q1$ and $q2$; clarification requests and `IGNORE` both require making reference to distinct DGBs for the two participants, make use of an additional buffer for ungrounded utterances, PENDING, and involve coherence relations defined at the level of utterances, not merely $q1$ and $q2$. The pre–theoretical complexity associated with each class is summarized in Table 9.

Table 9: Increasing complexity of reasoning needed to accommodate query responses

Query response type	Information state complexity
`DP`, `IND`	QUD, dependence relation
`FORM`	QUD, parametrised dep. relation
`MOTIV`	QUD as poset
`NO ANSW`	unif-coh relation, QUD as poset
`CR`, `IGNORE`	QUD, PENDING, DGB split
	non-semantic coherence

7 CONCLUSIONS

The article provides the first comprehensive, empirically based study of query responses to queries. One interesting finding here is the existence of a number of classes of adversarial responses that involve the rejection/ignoring of the original query. Indeed, in such cases the original query is rarely responded to in subsequent interaction. We

designed our taxonomy based on data from the BNC since it is a general corpus with a variety of domains and genres, but have also shown that our classification works well in a number of more specific genres and domains, which display quite different distributions of query responses. We have proposed qualitative, domain-specific explanations for the variation displayed by these distributions.

On the theoretical side, we have provided a comprehensive, information state dynamics-based characterisation of the relevance of the entire range of query response types. Our account uses the KoS framework for representing dialogue information states and its component of information arising from publicized interaction, the dialogue game board (DGB). This enables us to offer a pre-theoretical sketch of the expressive complexity of the different classes of query response types, ranging from dependent questions and `IND`, which, assuming a semantic relation of question dependence, can be accommodated in a fairly vanilla query/response setup, through `MOTIV` and `NO ANSW`, which intrinsically require the dynamic question repository QUD to be a partially-ordered set, through `IGNORE` and clarification requests, which require distinct information DGBs for the two participants, make use of an additional buffer for ungrounded utterances, PENDING, and involve coherence relations defined at the level of utterances, not merely $q1$ and $q2$.

What are the more general theoretical implications of this characterization? We believe that it offers concrete desiderata for semantic theories, more specifically for the nature of conversational context. We offer brief remarks relative to frameworks that have put forward theories of question responses, as discussed in Section 1.

Some account of question dependence can be developed by any theory of questions which supplies notions of exhaustive and partial answerhood, though it is clear that providing a more detailed empirical and theoretical account of this notion than we have given here is an important task.

Relations like `MOTIV` and `NO ANSW` require structure within context since they need to maintain several questions simultaneously accessible to the participants. This constitutes a challenge for views of contexts in terms of *stacks*. Such a view has been made prominent in the view of QUD due to Roberts (1996). It can also be found, for instance, in the discourse model of Farkas and Roelofsen (2011), where

a discourse context X is identified as a pair $\langle M, T \rangle$, where M is a Kripke model and T is a stack of sentences, those sentences that have been uttered so far.

The problem for stacks can be defused by adopting a distinct structure, for instance a partial order. Nonetheless, for these accounts and most other existing views of context, context is an entity shared by the conversational participants. This was also the case for the view of discourse structure in earlier work in SDRT (e.g., Asher and Lascarides 1998, 2003). In more recent work (e.g., Lascarides and Asher 2009), SDRT adopts a view advocated in KoS and also in the framework of PTT (Poesio and Traum 1998) that associates a distinct contextual entity with each conversational participant.

Given this, it seems that a framework like SDRT has potential for developing an account of question relations like IGNORE and CR which require context to 'diverge' across participants. There is one important caveat – we have argued that the notion of relevance that underpins both these question relations must make reference to non-semantic information. By contrast, in SDRT the semantics/pragmatics interface has no access to linguistic form, but only to a partial description of the content that is derived from linguistic form. This has been argued to be necessary to ensure the decidability of SDRT's glue logic (see e.g., Asher and Lascarides 2003, p. 77).

In closing, we note two questions raised by our account. The coherence follows in some cases on the basis of quite general conversational rules (e.g., QSPEC and MetaDiscussing q1) and in other cases on the basis of rather specific – though domain-independent – rules (e.g. Ignoring questions). An obvious theoretical issue is whether one can attain similar coverage on the basis of more "general" rules allied with some other very general pragmatic principles. A converse question is whether investigation of specific genres will lead to the need for genre-specific conversational rules for certain classes of question relations.

ACKNOWLEDGEMENTS

This is a much extended version of the article 'A corpus-based taxonomy of question responses' presented at the 10th International Conference on Computational Semantics (IWCS), Potsdam, March 2013

and subsequently presented in a extended version at the *Questions in Discourse* workshop in Stuttgart in May 2014.

We would also like to give our thanks to Dorota Leszczyńska-Jasion, Katarzyna Paluszkiewicz, Mariusz Urbański, Andrzej Wiśniewski, to three anonymous reviewers for the *Journal of Language Modelling*, to Elżbieta Hajnicz, to Adam Przepiórkowski, and to Carmela Chateau, for their help and insightful comments on this article.

We acknowledge support by the French Investissements d'Avenir-Labex EFL program (ANR-10-LABX-0083) and by the Disfluences, Exclamations, and Laughter in Dialogue (DUEL) project within the Projets Franco-Allemand en sciences humaines et sociales of the Agence Nationale de Recherche (ANR) and the Deutsche Forschunggemeinschaft (DFG); as well as by the Iuventus Plus grant (IP2011-031-771) and by the funds of the National Science Centre, Poland (DEC-2012/04/A/HS1/00715).

REFERENCES

Nicholas Asher and Alex Lascarides (1998), Questions in dialogue, *Linguistics and Philosophy*, 21(3):237–309.

Nicholas Asher and Alex Lascarides (2003), *Logics of conversation*, Cambridge University Press, Cambridge.

Jean Carletta (1996), Assessing agreement on classification task: The kappa statistic, *Computational Linguistics*, 22(2):249–254.

Lauri Carlson (1983), *Dialogue games*, Synthese Language Library, D. Reidel, Dordrecht.

Robin Cooper (2005), Austinian truth in Martin-Löf Type Theory, *Research on Language and Computation*, 3(4):333–362.

Robin Cooper (2012), Type Theory and semantics in flux, in Ruth Kempson, Nicholas Asher, and Tim Fernando, editors, *Handbook of the Philosophy of Science*, volume 14: Philosophy of Linguistics, pp. 271–323, Elsevier, Amsterdam.

Robin Cooper (2016), Type Theory and language: From perception to linguistic communication, `https://sites.google.com/site/typetheorywithrecords/drafts/ttl161130.pdf?attredirects=0`, book draft (access 20.03.2017).

Robin Cooper and Jonathan Ginzburg (2015), Type Theory with Records for NL semantics, in Chris Fox and Shalom Lappin, editors, *Handbook of Contemporary Semantic Theory, Second Edition*, pp. 375–407, Blackwell, Oxford,

doi:10.1002/9781118882139.ch12, http://dx.doi.org/10.1002/9781118882139.ch12.

Thierry COQUAND, Randy POLLACK, and Makoto TAKEYAMA (2003), A logical framework with dependent types, *Fundamenta Informaticae*, 20:1–21.

Donka FARKAS and Floris ROELOFSEN (2011), Polarity particles in an inquisitive discourse model, https://www.illc.uva.nl/inquisitivesemantics/papers/publications, access 20.03.2017. Manuscript, University of California at Santa Cruz and ILLC, University of Amsterdam.

Matthias GAMER, Jim LEMON, and Ian Fellows Puspendra SINGH (2012), *irr: Various coefficients of interrater reliability and agreement*, http://CRAN.R-project.org/package=irr, acess 20.03.2017, R package version 0.84.

Jonathan GINZBURG (1995), Resolving questions, I, *Linguistics and Philosophy*, 18:459–527.

Jonathan GINZBURG (2010), Relevance for dialogue, in Paweł ŁUPKOWSKI and Matthew PURVER, editors, *Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue*, Polish Society for Cognitive Science, Poznań, iSBN 978-83-930915-0-8.

Jonathan GINZBURG (2012), *The interactive stance: Meaning for conversation*, Oxford University Press, Oxford.

Jonathan GINZBURG and Robin COOPER (2004), Clarification ellipsis, and the nature of contextual updates, *Linguistics and Philosophy*, 27(3):297–366.

Jonathan GINZBURG, Raquel FERNÁNDEZ, and David SCHLANGEN (2014), Disfluencies as intra-utterance dialogue moves, *Semantics and Pragmatics*, 7(9):1–64.

Arthur C. GRAESSER, Natalie K. PERSON, and John D. HUBER (1992), Mechanisms that generate questions, in Thomas W. LAUER, Eileen PEACOCK, and Arthur C. GRAESSER, editors, *Questions and information systems*, pp. 167–187, Lawrence Erlbaum Associates, Hillsdale.

Jeroen GROENENDIJK (2009), Inquisitive semantics: Two possibilities for disjunction, in Peter BOSCH, David GABELAIA, and Jérôme LANG, editors, *Logic, Language, and Computation*, volume 5422 of *Lecture Notes in Computer Science*, pp. 80–94, Springer Berlin / Heidelberg.

Jeroen GROENENDIJK and Floris ROELOFSEN (2011), Compliance, in Alain LECOMTE and Samuel TRONÇON, editors, *Ludics, Dialogue and Interaction*, pp. 161–173, Springer-Verlag, Berlin Heidelberg.

Jacqueline C. KOWTKO and Patti J. PRICE (1989), Data collection and analysis in the air travel planning domain, in *Proceedings of the Workshop on Speech and*

Natural Language, HLT '89, pp. 119–125, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 1-55860-112-0, doi:10.3115/1075434.1075455, http://dx.doi.org/10.3115/1075434.1075455.

Jan Van KUPPEVELT (1995), Discourse structure, topicality and questioning, *Journal of Linguistics*, 31:109–147.

Staffan LARSSON (2002), *Issue based dialogue management*, Ph.D. thesis, Gothenburg University, http://www.ling.gu.se/~/sl/Thesis/thesis.pdf, access 20.03.2017.

Staffan LARSSON and David TRAUM (2003), The information state approach to dialogue management, in Jan VAN KUPPEVELT and Ronnie SMITH, editors, *Advances in Discourse and Dialogue*, Kluwer.

Alex LASCARIDES and Nicholas ASHER (2009), Agreement, disputes and commitments in dialogue, *Journal of Semantics*, 26(2):109–158.

Brian MACWHINNEY (2000), *The CHILDES project: Tools for analyzing talk*, Lawrence Erlbaum Associates, Mahwah, NJ, third edition.

Rebecca J PASSONNEAU and Bob CARPENTER (2013), The benefits of a model of annotation, in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pp. 187–195, Citeseer.

Michal PELIŠ and Ondrej MAJER (2010), Logic of questions from the viewpoint of dynamic epistemic logic, in M. PELIŠ, editor, *The Logica Yearbook 2009*, pp. 157–172, College Publications, London.

Massimo POESIO and David TRAUM (1998), Towards an axiomatization of dialogue acts, in J. HULSTIJN and A. NIJHOLT, editors, *Proceedings of TwenDial 98, 13th Twente workshop on Language Technology*, pp. 207–221, Twente University, Twente.

Carl POLLARD and Ivan A. SAG (1994), *Head-driven Phrase Structure Grammar*, University of Chicago Press and CSLI, Chicago.

Matthew PURVER (2001), SCoRE: A tool for searching the BNC, Technical Report TR-01-07, Department of Computer Science, King's College London, ftp://ftp.dcs.kcl.ac.uk/pub/tech-reports/tr01-07.ps.gz.

Matthew PURVER (2006), CLARIE: Handling clarification requests in a dialogue system, *Research on Language & Computation*, 4(2):259–288.

Matthew PURVER, Jonathan GINZBURG, and Patrick HEALEY (2001), On the means for clarification in dialogue, in *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pp. 116–125, Association for Computational Linguistics, Aalborg, Denmark, http://www.dcs.qmul.ac.uk/~mpurver/papers/pgh01sigdial.pdf.

R CORE TEAM (2013), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org/, acess 20.03.2017.

Verena RIESER and Joanna MOORE (2005), Implications for generating clarification requests in task-oriented dialogues, in *Proceedings of the 43rd Meeting of the Association for Computational Linguistics*, Michigan.

Craige ROBERTS (1996), Information structure in discourse: Towards an integrated formal theory of pragmatics, *Working Papers in Linguistics-Ohio State University Department of Linguistics*, pp. 91–136, reprinted in Semantics and Pragmatics, 2012.

Kepa RODRIGUEZ and David SCHLANGEN (2004), Form, intonation and function of clarification requests in German task-oriented spoken dialogues, in Jonathan GINZBURG and Enric VALLDUVÍ, editors, *Proceedings of Catalog'04, The 8th Workshop on the Semantics and Pragmatics of Dialogue*, Universitat Pompeu Fabra, Barcelona.

Carolyn P. ROSÉ, Barbara Di EUGENIO, and Johanna D. MOORE (1999), A dialogue-based tutoring system for basic electricity and electronics, in Susanne P. LAJOIE and Martial VIVET, editors, *Artificial intelligence in education*, pp. 759–761, IOS, Amsterdam.

Emanuel SCHEGLOFF (2007), *Sequence organization in interaction*, Cambridge University Press, Cambridge.

David SCHLANGEN (2004), Causes and strategies for requesting clarification in dialogue, in *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pp. 136–143.

Anthony J. VIERA and Joanne M. GARRETT (2005), Understanding interobserver agreement: The kappa statistic, *Family Medicine*, 37(5):360–363.

Andrzej WIŚNIEWSKI (1995), *The posing of questions: Logical foundations of erotetic inferences*, Kluwer AP, Dordrecht, Boston, London.

Andrzej WIŚNIEWSKI (2003), Erotetic search scenarios, *Synthese*, 134:389–427.

Andrzej WIŚNIEWSKI (2013), *Questions, inferences and scenarios*, College Publications, London.

Mapping theory without argument structure*

Jamie Y. Findlay
University of Oxford, Oxford, UK

ABSTRACT

Asudeh and Giorgolo (2012) offer an analysis of optional and derived arguments that does away with argument structure as a separate level of representation within the architecture of Lexical Functional Grammar in favour of encoding much of this information in a connected semantic structure. This simplifies the architecture in many ways, but leaves open the question of the mapping between thematic roles, arguments, and grammatical functions (traditionally explored under the umbrella of Lexical Mapping Theory; LMT: Bresnan and Kanerva 1989). In this paper, I offer a formalisation of these mapping relations, drawing on a modern reanalysis of traditional LMT (Kibort 2007), while also continuing Asudeh and Giorgolo's (2012) quest to evacuate as much information as possible out of individual lexical entries and into cross-categorising templates (Dalrymple *et al.* 2004; Crouch *et al.* 2012).

Keywords:
argument structure,
mapping theory,
argument linking,
LFG,
Glue Semantics

*This work has benefitted enormously from discussion and ideas which I was exposed to at the Glue Group at the University of Oxford and during Anna Kibort's illuminating lectures on argument structure in Hilary Term 2013. I would like to express my thanks to all involved, and especially to Ash Asudeh for his detailed comments on an earlier draft. Thank you also to the anonymous JLM reviewers, whose thoughtful criticism has made this a far better paper. Any remaining errors are, as ever, entirely my own. This work was completed while I was the recipient of a UK Arts and Humanities Research Council studentship (grant reference AH/K503198/1), which I gratefully acknowledge.

1 INTRODUCTION

This paper makes a contribution to the theoretical frameworks of Lexical Functional Grammar (LFG: Kaplan and Bresnan 1982; Bresnan 2001; Dalrymple 2001; Falk 2001; Bresnan *et al.* 2016; Asudeh and Toivonen 2015) and Glue Semantics (Glue: Dalrymple 1999, 2001; Asudeh 2012). Some relevant formalisms will be explained where possible, but constraints of space prevent a full introduction to the two theories here.

The main purpose of this paper will be to show that current work by Anna Kibort (Kibort 2001, 2007, 2008, 2014) on Lexical Mapping Theory (LMT) is compatible with a proposal by Asudeh and Giorgolo (2012) (hereafter A&G) to do away with argument structure as a separate level of representation in the formal architecure of LFG, and to demonstrate how the two theories can be integrated.

The paper is structured as follows. Section 2 discusses what we want from a mapping theory in general, and introduces LMT. Following this, the key points of Kibort's version of LMT are briefly sketched in Section 3, while Section 4 discusses the role of argument structure, and introduces A&G's suggestion to do without it. Section 6 contains the main proposal of the paper, namely a formalism which allows the insights of Kibort's LMT to be combined with A&G's abandonment of argument structure. This section ends with examples of how two argument alternations, the passive and the benefactive, can be treated in the new theory. Finally, Section 7 offers conclusions.

2 (LEXICAL) MAPPING THEORY

Mapping theories attempt to find general principles by which arguments and grammatical functions are related, thus avoiding repeated (and redundant) lexical stipulation. It is not a coincidence, so the theory goes, that the Agent arguments in verbs like *hit*, *select*, *put*, or many others are usually syntactically realised as subjects, while the Patient-like arguments are usually direct objects.[1]

The traditional work on this problem in LFG is Lexical Mapping Theory (LMT: Bresnan and Kanerva 1989; Bresnan 1990; Butt *et al.* 1997). However, this name may not be entirely apposite. As several

[1] At least in syntactically accusative languages.

authors have pointed out (e.g. Butt 1995; Alsina 1996), "the theory cannot apply exclusively to individual words" (Dalrymple 2001, 212), since various problems generally thought to fall under the umbrella of LMT can involve *multiple* lexemes which combine to form complex predicates in the syntax (for example, causatives are formed analytically in some languages, e.g. Romance, even if they are synthetic in others).[2] For this reason, I follow the recent trend in dropping the 'lexical' and referring to this theory simply as *mapping theory*. I will, though, continue to use the term 'LMT' when discussing researchers, like Kibort, who explicitly position their work as belonging to this tradition.

What do we expect of such a theory (whatever we call it)? If the relationship between grammatical functions and arguments were simple or straightforward, there would be nothing to a mapping theory other than a listing of the recorded correspondences for each language. However, there is no one-to-one mapping between particular roles and particular grammatical functions (GFs). There are many operations which alter the mapping between the two, such as locative inversion, the passive, the applicative, or the causative. Some, such as the passive or the applicative, are described as *morphosyntactic*, in that they do not involve a change in (truth-conditional) meaning – they merely realign participants and grammatical functions.[3] Others, such as the causative, are *morphosemantic* in that they add additional participants or change the roles of existing participants, and thus change the truth-conditional meaning of the predicate.

At the very least, mapping theory must explain the morphosyntactic alternations. Ideally, it should also offer a principled account of the morphosemantic ones: Kibort (2007), for example, suggests an extension to traditional LMT which allows it to account for morphosemantic as well as morphosyntactic alternations.

[2] Although see Ackerman *et al.* (2011) for a dissenting view on the role of syntax in predicate formation.

[3] Of course, they alter other aspects of 'meaning', in the broader sense of the word, such as information structure or pragmatics. This is not surprising, for it would indeed be strange to discover that there were truly 'gratuitous' alternations that merely added complexity to the grammar with no corresponding communicative payoff.

Let us consider an example, that of the passive, which is a morphosyntactic alternation. A transitive verb like *devour* takes two arguments: a devourer and a devourum (the thing devoured). In sentence (1), the devourer argument is associated with the SUBJect GF, and the devourum with the OBJect, while in (2), the passive, the devourum is now the SUBJ, and the devourer is either unexpressed, or realised as an OBLique *by*-phrase:

(1) Jeremy devoured the pizza.

(2) The pizza was devoured (by Jeremy).

But such alternations are not unrestricted: in English, there is no purely morphosyntactic operation which would make the devourer an object, as in (3), and none which would make the devourum an oblique, as in (4), for example:[4]

(3) a. * The pizza devoured Jeremy. [With the intended meaning.]

b. * It devoured Jeremy ((by/to/…) the pizza).

(4) * Jeremy devoured by/to/… the pizza.

Any theory of mapping must explain why the alternation in (1)–(2) is possible, while others are not. This means we need to be able to restrict the type of GF an argument can be associated with, but not simply by reducing it to one. The standard approach has been underspecification by features, to which we now turn.

[4] It may be that such alternations exist in other languages: for example, if the difference between actor voice and undergoer voice in some Western Austronesian languages is truly a voice alternation (Himmelmann 2002), then this might be an example of a morphosyntactic alternation which has the form exemplified in (3a).

As an anonymous reviewer points out, there may also be morpho*semantic* alternations which do involve such alignments. For example, (4) corresponds to the antipassive or deobjective in Slavic languages (Fehrmann *et al.* 2010, 207–208). What is more, if we consider lexical relationships, the correspondence between verb pairs like *fear* and *frighten* might be thought to realise the alternation between (1) and (3a), whereby the subject in one member of the pair corresponds to the object in the other. Such lexical relatedness goes beyond the scope of mapping theory, however.

2.1 *Grammatical functions decomposed*

In standard LMT, the four-way cross-classification of GFs given in (5) (after Bresnan and Kanerva 1989) is assumed:

(5)

	$-r$	$+r$
$-o$	SUBJ	OBL_θ
$+o$	OBJ	OBJ_θ

SUBJ, OBJ, and OBL_θ are the subject, (direct) object, and oblique functions more or less familiar from traditional grammars. OBJ_θ may be less familiar: this is the so-called secondary or restricted object, as in the second object of English dative-shifted *give*:

(6) Kim gave Colin **his book**.

The necessity of theorising such a GF has been contested, but it is still taken as standard in mainstream LFG, and so I will continue to use it here (see Kibort 2013 for a defence of the status of OBJ_θ).

The two features, $[o]$ and $[r]$, refer, respectively, to the ***object***-like properties of a GF, and to whether it is semantically ***restricted*** or not. Thus, there are two objective ($[+o]$) GFs, namely OBJ and OBJ_θ, and two non-objective ($[-o]$) ones, *viz.* SUBJ and OBL_θ. Similarly, there are two semantically restricted ($[+r]$) GFs, OBL_θ and OBJ_θ, and two non-restricted ($[-r]$) ones, SUBJ and OBJ.

With this in place, the solution to the *devour* question above becomes straightforward. In the standard theory, we simply associate each argument with a single feature, which then limits its choice of GF to two. We saw that the devourer argument could be realised as a SUBJ or as an OBL;[5] thus, in the mapping theory, it is linked with a $[-o]$ feature, and can therefore surface as a SUBJ or an OBL (but not an OBJ, for example), just as needed. Meanwhile, the devourum is marked as $[-r]$, and can thus be realised as an OBJ or a SUBJ (but not an OBL, for example), again just as observed. A separate mechanism is required to determine which argument gets priority in selecting a particular GF – this is usually explained by reference to a thematic hierarchy of some kind, although there is a lack of agreement over the

[5] For the sake of parsimony, and to avoid being drawn into a debate about exactly what information could be the realisation of θ in OBL_θ (see also fn. 18, below), I will use OBL as shorthand for OBL_θ when the exact nature of the subscript/index is unimportant.

exact form this should take (Newmeyer 2002, 65ff.; Levin and Rappaport Hovav 2005, ch. 6). In the analysis presented here, we will use a different mechanism.

2.2 *The status of the features [o] and [r]*

A natural question to raise at this stage is that of the status of these features. Certainly, they are intended to cross-classify the grammatical functions. But it would seem from the definitions that they are intended to *constitute* the GFs somehow, as well. That is, they actually contribute some information related to semantic restrictedness or objectivity – though of course these terms then raise their own definitional questions.

One possibility is that the familiar GF labels are really just abbreviations for feature structures incorporating these mapping features. This is the approach hinted at by Falk (2001, 109, fn. 12), for example. On this view, the label SUBJ is really just a shorthand way of writing the f-structure in (7), and the f-structure given in (8) is a shorthand way of writing the fully expanded f-structure in (9):

(7)
$$\begin{bmatrix} \text{R} & - \\ \text{O} & - \end{bmatrix}$$

(8)
$$\begin{bmatrix} \text{PRED} & \text{‘love’} \\ \text{SUBJ} & \begin{bmatrix} \text{PRED} & \text{‘Trevor’} \end{bmatrix} \\ \text{OBJ} & \begin{bmatrix} \text{PRED} & \text{‘Elliot’} \end{bmatrix} \end{bmatrix}$$

(9)
$$\begin{bmatrix} \text{PRED} & \text{‘love’} \\ \begin{bmatrix} \text{R} & - \\ \text{O} & - \end{bmatrix} & \begin{bmatrix} \text{PRED} & \text{‘Trevor’} \end{bmatrix} \\ \begin{bmatrix} \text{R} & - \\ \text{O} & + \end{bmatrix} & \begin{bmatrix} \text{PRED} & \text{‘Elliot’} \end{bmatrix} \end{bmatrix}$$

Now, in the standard theory, attribute-value structures such as (7) are only permitted as *values* of attributes, not as attributes themselves. F-structures are defined as functions from their attributes to their values, and the domain of those functions does not include those functions themselves. Thus, to allow structures like (9) is to alter the mathematical properties of f-structures, so that their domains no longer

include only simple atomic values, but also sets (specifically, functions). Perhaps this is what we need, but it is worth noting that it is not simply a notational variant.

Such a move also represents a departure from one of LFG's foundational theoretical principles, namely that grammatical functions are primitives in some sense. Now the features R and O are the primitives instead.[6]

In matter of fact, we do not need to answer the theoretical questions lurking behind the decompositional approach to GFs in order to take advantage of it. By appealing to these features we are making empirical claims: if it is true that there are mapping phenomena which are sensitive to the $[\pm o]/[\pm r]$ distinction, then we have determined that some pairings/alternations of GFs should be ruled out. For example, there is no way, at least not using a single feature, of describing just the pair SUBJ and OBJ_θ, or the pair OBJ and OBL, and so (purely morphosyntactic) alternations involving these pairs should be ruled out. They do not form a natural class. This is an empirical claim, and in order to describe it, it is enough to see the $[\pm o]/[\pm r]$ distinction as merely mnemonic, describing four sets of pairs which can be linked to arguments by whatever mechanism we choose to use. Thus, abstracting away from the theoretical questions, we can use disjunctions to define the following feature decompositions (suggested to me by Ron Kaplan, p.c.):[7]

(10) $\text{MINUSO} \equiv \{\text{SUBJ}|\text{OBL}_\theta\}$

(11) $\text{PLUSO} \equiv \{\text{OBJ}|\text{OBJ}_\theta\}$

(12) $\text{MINUSR} \equiv \{\text{SUBJ}|\text{OBJ}\}$

(13) $\text{PLUSR} \equiv \{\text{OBL}_\theta|\text{OBJ}_\theta\}$

[6] Butt (1995, 31) makes this claim explicitly, saying that "[w]hile it may appear that grammatical functions like SUBJ, OBJ, etc. exist as primitive notions within the theory, a given grammatical function, a SUBJ for example, is actually nothing more and nothing less than the features $[-r,-o]$. Grammatical functions thus are not independent of the features, but are instead defined and therefore also constrained by them".

[7] These are written in the regular language used in LFG *functional descriptions* (see Asudeh 2012, 64–65). The expression $\{A|B\}$ represents a disjunction between A and B.

In essence, this approach sidesteps the theoretical issues raised by the decompositional approach and simply co-opts its empirical claims.

2.3 *Optionality of grammatical functions*

One other assumption I will be making that is relevant in considering the theory of mapping presented here is that all GFs are optional: the syntactic constraints of Coherence and Completeness (see Kaplan and Bresnan 1982, 211–212, and Dalrymple 2001, 35–39, for formal definitions and discussion) are subsumed by considerations of *resource sensitivity* in a Glue-based semantics (see discussion in Dalrymple 1999; Kuhn 2001; Asudeh 2012, ch. 5). That is, the presence of all and only the arguments required by a predicate is constrained by the linear logic component of Glue: incoherence leads to resource surplus, while incompleteness leads to resource deficit. When writing f-structures, therefore, I will give PRED values as simple semantic forms in single quotation marks (e.g. 'select'), omitting the traditional GF-selection/subcategorisation information usually given inside and outside angled brackets (e.g. 'select ⟨SUBJ, OBJ⟩').[8]

3 KIBORT'S LMT

Kibort (2001, 2007, 2008, 2014) has argued for a number of modifications to LMT, most importantly for a return to the separation implied by earlier work (e.g. Bresnan 1982) between thematic roles and *argument positions*, intermediary objects standing between thematic roles and the grammatical functions which realise them. Later work collapsed this distinction, conflating thematic roles with argument positions, which then reduces the problem of mapping to that of linking thematic roles to GFs directly. If the focus of mapping theory is purely morphosyntactic operations, this is perhaps understandable, but Kibort (2007) argues for extending the scope of LMT to include

[8] The main obstacle to relegating Coherence and Completeness to the semantics is expletive arguments, i.e. those which are required by the syntax but not the semantics, and which therefore might be thought not to make any semantic contribution. Clearly, resource sensitivity will not help us if such arguments are not included in the resource accounting in the first place. This problem is not insurmountable, however: see Asudeh (2012, 113) for some suggestions about how to resolve the problem without resorting to subcategorisation via the PRED feature.

morphosemantic operations as well, and here it is important to allow participants to realign with respect to their thematic roles (more on this below).

Kibort therefore suggests that argument structure is made up of a list of argument positions, each of which has associated with it an intrinsic assignment of syntactic features (or, ultimately, a pair of GFs, as we are thinking about it), but which can be associated with different thematic roles. Predicates have open to them a universal subcategorisation frame, from which they select a certain number of arguments. The intrinsic assignments are as given in (14):[9]

(14)

<	arg_1	arg_2	arg_3	arg_4	…	arg_n	>
	$[-o]$	$[-r]$	$[+o]$	$[-o]$		$[-o]$	

These argument positions are ordered, and a predicate can select any combination of them – that is, not necessarily a contiguous subsection: a predicate could select an arg_1 and an arg_4, for example – but there can only be one of each: e.g. there cannot be two arg_2s. As the arg_n notation makes clear, there can be more than four arguments; however, all arguments above arg_4 will be of the same syntactic type as an arg_4 (namely, $[-o]$).[10,11]

[9] In the full theory, arg_1 is associated with $[-o]$ in unergative verbs and $[-r]$ in unaccusative ones; I simplify here, since the only verbs we will be looking at require $[-o]$.

[10] As a reviewer notes, this means it is, in a certain sense, possible to have 'more than one arg_4', in that there may be more than one argument position of the same syntactic type as arg_4. However, such additional arguments would be distinguished by their subscripts, so that if there are two 'arg_4's, one will in fact be an arg_5.

[11] Kibort's stance on the uniqueness of argument positions does not seem wholly consistent. In some works, argument positions are described as being "unique" (Kibort 2007, 259), while in others it is explicitly claimed that multiple arg_3s, for example, are permitted (Kibort 2008, 330). Assuming that s-structures share the same functional properties as f-structures, the proposal I give in Section 6 does not allow for multiple argument positions with the same name, which means it may not be able to handle the case of multiple applicatives discussed in Kibort (2008). However, I am concerned that Kibort's proposals to resolve this problem raise issues for the internal coherence of her own system: if there are multiple argument positions with the same name, it is not clear to me how the mapping principles are to distinguish them.

In addition to the argument positions being ordered, we can derive a partial ordering on grammatical functions from their decomposition into features, which ranks GFs from least to most marked, where being marked is equated with having more + features (Bresnan *et al.* 2016, 331):

(15) $\textsc{subj} > \textsc{obj}, \textsc{obl}_\theta > \textsc{obj}_\theta$

Mapping is then simply linking the highest arg position to the highest available GF (with appropriate restrictions such as Function-Argument Biuniqueness (Bresnan 1980) to prevent multiple arguments mapping to the same GF). Let us see a brief example of how this works.

A verb like *select* will have the following argument structure:

(16) $\textit{select} \quad < \quad \begin{matrix} \text{arg}_1 \\ [-o] \end{matrix} \quad \begin{matrix} \text{arg}_2 \\ [-r] \end{matrix} \quad >$

If there is no further specification, the highest argument position, arg_1, will then map to the highest available $[-o]$ GF, in this case the SUBJ. The next argument, arg_2, then maps to the highest available $[-r]$ GF, in this case the OBJ, which is exactly the pattern we want for an active voice transitive verb.

The passive alternation can now be easily explained as an operation which further restricts arg_1 to $[+r]$ (Kibort 2001), giving us the following argument structure:

(17) $\textit{select}_{\textsc{pass}} \quad < \quad \begin{matrix} \text{arg}_1 \\ [-o] \\ [+r] \end{matrix} \quad \begin{matrix} \text{arg}_2 \\ [-r] \end{matrix} \quad >$

The mapping now follows straightforwardly, using the same procedure. The first argument, arg_1, maps to the highest available GF which satisfies its feature requirements: in the present case, this is uniquely described, since the only GF which is both $[-o]$ and $[+r]$ is OBL. The next argument, arg_2, then maps to the highest available $[-r]$ GF, which is now the SUBJ.

Obligatorily three-place predicates like *put* will have the argument structure below:

(18) $\textit{put} \quad < \quad \begin{matrix} \text{arg}_1 \\ [-o] \end{matrix} \quad \begin{matrix} \text{arg}_2 \\ [-r] \end{matrix} \quad \begin{matrix} \text{arg}_4 \\ [-o] \end{matrix} \quad >$

In the active, this will correctly specify the three GFs as SUBJ, OBJ, and OBL. But importantly, it will also provide the correct analysis of the passive, whereby the direct object can be 'promoted' to subject, but not the (object within the) prepositional phrase, as exemplified in (19)–(20):

(19) The cup was put on the table.

(20) * On the table was put the cup./* The table was put the cup on.

The argument structure for passive *put* is as follows:

(21) put_{PASS} < $\begin{matrix}\text{arg}_1\\ [-o]\\ [+r]\end{matrix}$ $\begin{matrix}\text{arg}_2\\ [-r]\end{matrix}$ $\begin{matrix}\text{arg}_4\\ [-o]\end{matrix}$ >

If we follow the same mapping procedure as before, we can see that we obtain the correct results: arg_1 once again maps to OBL; arg_2, the next highest argument, then maps to SUBJ, thus preventing arg_4 from doing so; arg_4 maps to the highest available $[-o]$ GF, which is OBL (it is not a problem that there are two OBL arguments, since they will be distinguished by their indices, whatever these may be: for example, arg_1 might correspond to an $\text{OBL}_{\text{AGENT}}$ and arg_4 to an OBL_{GOAL}).

Kibort's analysis offers a simple and general solution to many of the traditional mapping problems, but it is obviously based in a theory where argument structure has a fundamental role. In the next section, I present evidence that we should do away with argument structure as a separate level of representation. The challenge then is to retain the advantages of Kibort's LMT in a formalism without a-structure. This is the topic of Section 6.

4 THE PROBLEM WITH ARGUMENT STRUCTURE

In the LFG conception of the architecture of the grammar, a modularity is assumed such that different components of the grammar (morphology, phonology, syntax, etc.) are treated as separate levels of structure, related by what are called correspondence functions. Of particular interest are the two levels of syntactic representation, *c(onstituent)-structure* (phrase structure) and *f(unctional)-structure* (which represents grammatical relations such as *subject of*

and *object of* in an attribute-value matrix), and the level of the syntax-semantics interface, *s(emantic)-structure*. (For more on these structures, see Dalrymple 2001, 45–68, 7–44, 230–240, respectively; on the correspondence architecture, see Dalrymple 2001, 180–182ff., and Asudeh 2012, 49–54.)

Generally, a level of *a(rgument)-structure* is also assumed, which encodes the lexical arguments of a predicate, and controls their linking to grammatical functions. However, there are questions over where exactly such a level should appear in the architecture of the grammar, or indeed whether such an independent level of representation is needed. A&G argue, based on problems caused by predicates taking optional arguments, that it is best to do away with a-structure, and relegate most of its functions to an augmented s-structure. In Section 4.1, I present their reasoning. However, we may come to the same conclusion independently, via considerations of a more abstract or meta-theoretical nature, and Section 4.2 explores these.[12]

4.1 *Optional arguments*

Certain verbs, such as *eat* or *drink*, express their Patient argument in the syntax only optionally:

(22) a. Pedro ate the cake earlier.
b. Pedro ate earlier.

(23) a. Amanda drank her coffee quickly.
b. Amanda drank quickly.

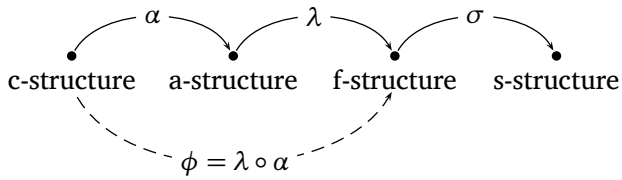
Nonetheless, this Patient argument must still be present in the verb's argument structure – it remains, after all, part of the core relation expressed by the verb – and must also be represented at semantic structure, since it is interpreted semantically – for *John ate* to be true, John must have eaten *something*. But, A&G argue, this means that the standard conception of the LFG correspondence architecture is inadequate.

Since Butt *et al.* (1997), the canonical view in LFG has been that a-structure should be treated as a separate level of representation in between c-structure and f-structure. This means that the traditional ϕ-function, which maps from c-structure to f-structure, is then seen as

[12] I thank an anonymous reviewer for their helpful observations on this point.

the composition of two new functions: the α-function from c-structure to a-structure, and the λ-function from a-structure to f-structure. The correspondence function from f-structure to s-structure remains the σ-function. This architecture is shown schematically in (24):

(24)



However, if 'optional' arguments appear at a-structure and s-structure, but not f-structure, we must posit a new correspondence function directly between a-structure and s-structure (which A&G call the θ-function) in order to bypass f-structure. This situation is shown in (25):

(25)

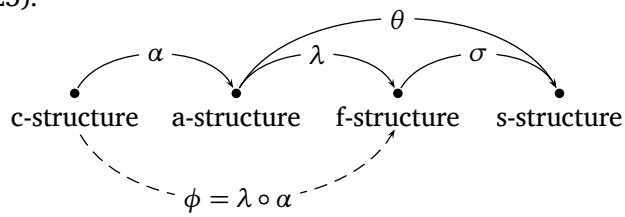


Figure 1 gives the relevant structures and correspondences for the sentence *Kim ate at noon* in this conception of the standard theory.

If we consider the PATIENT argument at a-structure, we see that it does not map to any grammatical function at f-structure. This means that we cannot reach its s-structure correspondent, p, by the normal means of composing the λ- and σ-functions, thereby passing through the f-structure – instead, we need a new, separate function, θ.

If the s-structures were not unconnected, as they are in standard LFG (wherein each of e, k and p are separate, unconnected entities, as in Figure 1), one alternative would be to pass along the outermost structures via the usual correspondence functions until one reached the semantic structure for the clause, then go from that structure, e, to the PATIENT's s-structure, p, via some internal path. However, since in the present setup there is no relation expressed at semantic structure between e and p, this is impossible.

Thus, given the standard architecture, there is no way to relate the PATIENT with its s-structure, p, except via the proposed new func-

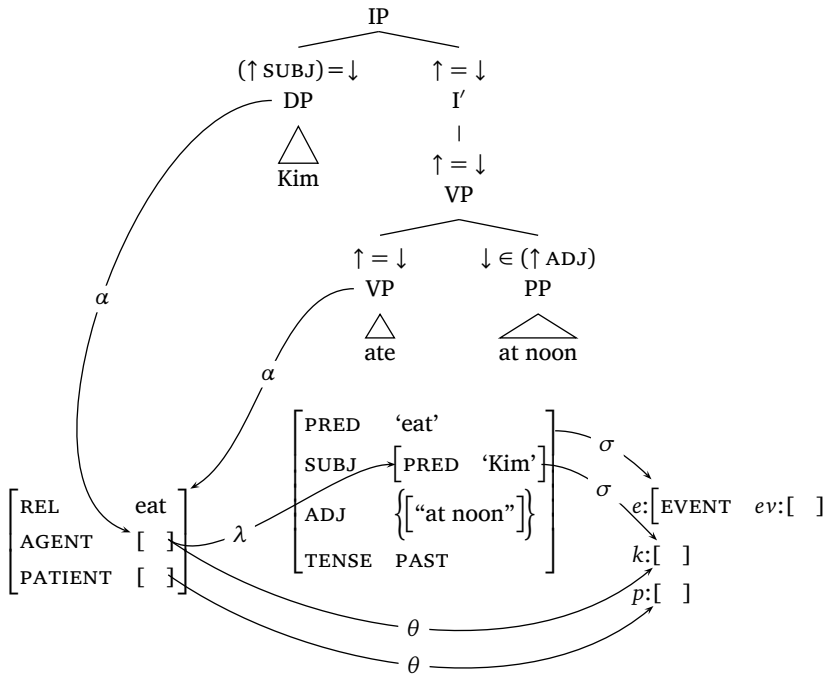


Figure 1: Relevant structures and correspondences for *Kim ate at noon* (after Asudeh and Giorgolo 2012, 70, Figure 1)

tion, θ. But, not only does making use of this new function add extra theoretical complexity, it also introduces a degree of indeterminacy into the grammar. There are now two correspondences between arguments which *are* realised syntactically (such as the AGENT argument in Figure 1) and their semantic structures, either via θ or via $\sigma \circ \lambda$. Therefore, instead of taking this option, A&G propose to make use of an architecture which does away with a-structure as a separate level of representation altogether, and with it the α-, λ-, and θ-functions (returning the ϕ-function to its former, underived, status). The information previously captured at a-structure is now encoded in a connected semantic structure. An analysis of the same sentence following this approach is given in Figure 2. A&G assume an event semantics for their meaning language, such that thematic roles are functions from events to individuals (Parsons 1990), and so avoid redundancy by using attributes like ARG_1 rather than AGENT in the semantic structure.[13]

[13] The framework suggested by A&G and elaborated on in this paper does not necessitate this treatment of thematic roles, and would be compatible with a

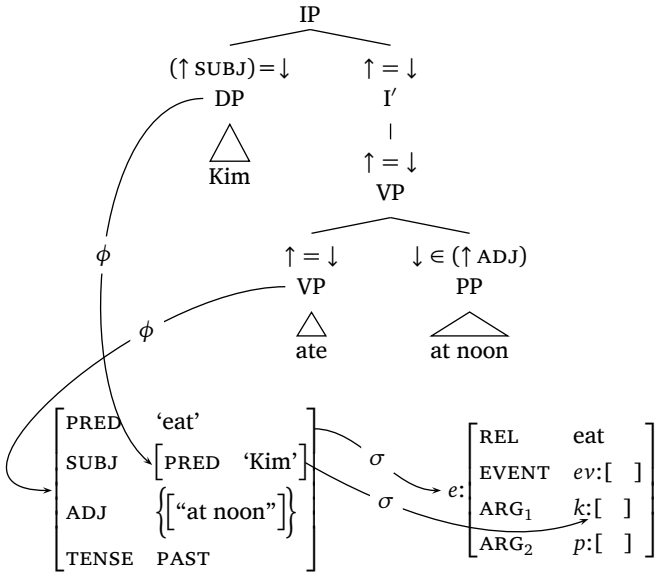


Figure 2:
Alternative analysis of *Kim ate at noon* (after Asudeh and Giorgolo 2012, 72, Figure 2)

A&G summarise the advantages that their approach brings as follows (p. 71):

1. We achieve a simplified architecture, which eliminates a separate a-structure projection, without losing information.
2. We do not lose linking relations and they are still post-constituent structure.[14]
3. We remove the non-determinacy that results from the presence of both the λ and θ correspondence functions.
4. Many of the meaning constructors for semantic composition are more elegant and simplified.

grammar that did without events in the semantics and instead treated thematic roles as e.g. attributes in s-structure (although of course appropriate modifications would be required). However, I consider it a strength of the present approach that it removes mention of thematic roles from the grammar; this is a view shared by Kibort (2007), and which I discuss further in Section 6.1.

[14] Because complex predicates can correspond to more than one node at c-structure, but to a single, complex a- or s-structure, it is important that linking relations should be post-constituent structure so that they remain many-to-one (and still functional), rather than one-to-many (and so not; see Butt 1995 and Alsina 1996).

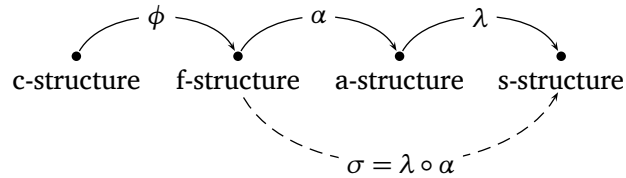
5. We regain the simple, traditional ϕ mapping from c-structure to f-structure.
6. We gain a connected semantic structure.

The form of A&G's argument is thus as follows. The location of a-structure in the correspondence architecture leads to theoretical complexity and redundancy when we consider optional arguments. One solution is to encode the information represented at a-structure somewhere post-f-structure. S-structure is post-f-structure, therefore one solution would be to encode it here. This also has the advantage of ontological parsimony: we have one less structure in our grammatical architecture.

4.2 *The role of a-structure*

Another, albeit less parsimonious, solution would be to relocate a-structure in between f- and s-structure, rather than collapsing it into the latter. That is, we might propose the architecture in (26) (here it is the σ-function which must be complexified, instead of the ϕ-function):

(26)



Aside from the problems posed by optional arguments, such a move has some independent motivation. Argument structure is generally seen as the interface between (lexical) meaning, including thematic roles, and syntax, in the form of the realisation of arguments as grammatical functions. But in the canonical architecture (in (24), above), a-structure stands between two levels of *syntax*, c-structure and f-structure, not between the syntax and the semantics. The modified architecture in (26) succeeds in remedying this situation.

However, by putting a- and s-structure in direct proximity like this, we draw attention to their potential similarities. S-structure is explicitly conceived of as the interface between syntax and semantics, acting as a syntactically-derived scaffold on which the linear logic of Glue can operate to control semantic composition. But a-structure is also an interface between syntax and semantics, relating GFs to the

roles they play in the meaning. Thus, to avoid redundancy, we might well ask whether it is possible to collapse the two structures.

Butt (1995) argues that an independent level of a-structure is needed, but her conception of a-structure is highly semantic: adapted from Jackendoff's (1990) Lexical Conceptual Structures, it includes a large amount of lexical meaning, such as aspectual information. Butt's (1995) reliance on a-structure may be an artefact of the time of writing, when the semantic component of LFG was underdeveloped – she does not discuss s-structure at all, for example. If the two levels of representation are really doing the same work, or contributing different facets of the same information, then it makes sense to collapse them.

If we want to achieve such parsimony, however, we must ensure that we are not generating additional problems at the same time as we simplify our ontology. Since mapping theories are usually reliant on a separate level of argument structure, we must be able to provide a new theory which is instead based on s-structure. The purpose of the current paper is to do just this, and to give a mapping theory which is compatible with the architecture of the grammar proposed by A&G. Before we come to this, however, I wish to discuss another motivation of their paper.

5 LEXICAL GENERALISATIONS VIA TEMPLATES

Aside from the removal of argument structure as a separate level of representation, the other major theme in A&G's paper is an attempt to abstract as much information as possible away from individual lexical entries and into *templates* (Dalrymple *et al.* 2004; Crouch *et al.* 2012; Asudeh *et al.* 2013), which are shared by multiple lexical items.

Templates are shorthand ways of abbreviating functional descriptions and other information included in lexical entries. This means that a grammar which includes templates is extensionally equivalent to one which does not, since templates serve only as abbreviations. However, templates can be used to capture commonalities and to express linguistic generalisations, which means that, while a grammar with templates may be equivalent to one without them, the former may be able to capture generalisations which the latter cannot (A&G, p. 78).

Templates can be used to name functional descriptions. For example, we might define the templates SG-SUBJ and 1-SUBJ as in (27)–(28):

(27) SG-SUBJ :=
(↑ SUBJ NUMBER) = SG

(28) 1-SUBJ :=
(↑ SUBJ PERSON) = 1

We can then build up more complex templates from these:

(29) 1SG-SUBJ :=
@1-SUBJ
@SG-SUBJ

The '**@**' symbol represents a 'call' of the following template; i.e. that line is to be expanded into the contents of the template named in the call. Thus, (29) is equivalent to (30):

(30) 1SG-SUBJ :=
(↑ SUBJ PERSON) = 1
(↑ SUBJ NUMBER) = SG

Templates can be made a little more flexible by allowing them to take arguments. For example, we can define a template PERSON, such that (28) is equivalent to (32):

(31) PERSON(X) :=
(↑ SUBJ PERSON) = X

(32) **@**PERSON(1)

We can do something similar for NUMBER, and then define a general SUBJECT template which takes two arguments, the person and the number of the predicate's subject:

(33) NUMBER(X) :=
(↑ SUBJ NUMBER) = X

(34) SUBJECT(P, N) :=
@PERSON(P)
@NUMBER(N)

Now (29) is equivalent to (35):

(35) @SUBJECT(1, SG)

Templates can also contain meaning constructors, since these are included in the functional description:

(36) FUTURE :=
$(\uparrow \text{TENSE}) = \text{FUTURE}$
$\lambda P.\exists e[P(e) \wedge \textit{future}(e)] : [(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

This template would be called by a future tense verb, and provides the relevant f-structural information about tense, as well as a meaning constructor which existentially closes the predicate's event argument and specifies that it occurs in the future.

Combining all of the above, the lexical entry for the Latin verb *bibam*, 1st person singular future tense of 'drink', would be as follows (ignoring questions of mapping for the time being):[15]

(37) *bibam* V $(\uparrow \text{PRED}) =$ 'drink'
@SUBJECT(1, SG)
@FUTURE

$\lambda y \lambda x \lambda e.\textit{drink}(e) \wedge \textit{agent}(e) = x \wedge \textit{patient}(e) = y :$
$(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

This is equivalent to the same lexical entry with all of the templates spelt out fully:

(38) *bibam* V $(\uparrow \text{PRED}) =$ 'drink'
$(\uparrow \text{SUBJ PERSON}) = 1$
$(\uparrow \text{SUBJ NUM}) = \text{SG}$
$(\uparrow \text{TENSE}) = \text{FUTURE}$

$\lambda P.\exists e[P(e) \wedge \textit{future}(e)] : [(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

$\lambda y \lambda x \lambda e.\textit{drink}(e) \ \wedge \ \textit{agent}(e) = x \ \wedge \ \textit{patient}(e) = y :$
$(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

[15] Since Latin is pro-drop, this entry should also include (i):

(i) $((\uparrow \text{SUBJ PRED}) = \text{'PRO'})$

I omit this in the text for the sake of simplicity.

The use of templates allows us to streamline lexical entries, make them more readable, and talk about commonalities across lexical entries, in terms of named, shared f-descriptions. One area in which A&G put templates to work is in evacuating as much information as possible about semantic composition from individual lexical entries into cross-categorising patterns like AGENT-PATIENT-VERB, which describes all verbs that take an Agent and a Patient argument. An example is given in (39) (A&G, p. 78, their (37)):[16]

(39) AGENT-PATIENT-VERB :=
$\lambda P \lambda y \lambda x \lambda e.P(e) \wedge \textit{agent}(e) = x \wedge \textit{patient}(e) = y :$
$[(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

This would be called by Agent-Patient verbs like *hit*, or *select*, which would have the following lexical entry:

(40) *select* V $(\uparrow \text{PRED}) = \text{‘select’}$
@AGENT-PATIENT-VERB

$\lambda e.\textit{select}(e) : (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

The only meaning that verbs contribute directly is the type of event they describe (the last line in (40)). The additional compositional work is done by the meaning constructor given in the AGENT-PATIENT-VERB template: it consumes the function contributed by the verb itself, predicates that of an event, and then provides thematic information on the meaning side, while on the linear logic side it returns a resource parallel in form to the familiar transitive verb resource (i.e. a dependency on the arguments of the verb which produces the meaning of the sentence – we now make use of the connected semantic structure positions rather than projections of grammatical functions). Asudeh *et al.* (2013) discuss this approach to composition in more detail.

In what follows, we will be able to augment these valency frame templates by including appropriate mapping information in them as well. We will also be able to describe various argument alternations

[16] A&G also stipulate various relations between grammatical functions and semantic arguments in such templates, but this is too limiting, and once we have established our theory of mapping, we can do better. As such, I omit reference to mapping from the present examples, since this is tangential to the main point under discussion.

by means of templatic material added to basic lexical entries, thus continuing the project of A&G and Asudeh *et al.* (2013) to reduce the idiosyncratic content of lexical entries as much as possible, and describe cross-categorising generalities using templates.

With all the pieces in place – Kibort's LMT, a connected semantic structure in lieu of argument structure, and the notion of lexical generalisation by template – we now turn to the main proposal of this paper.

6 MAPPING THEORY WITHOUT ARGUMENT STRUCTURE

6.1 *Preliminaries*

I want to suggest that Kibort's arg positions can be equated with the ARG attributes in A&G's connected semantic structures. This will purchase the explanatory power of Kibort's theory but without the cost of a fully fledged argument structure separate from semantic structure. One immediate advantage is that the uniqueness condition on arg positions comes for free, since the functional nature of semantic structures (assuming that they share this property with f-structures) means that there cannot be more than one attribute with the same name.

One implication of merging the proposals in this way, though, is that the subscript numbers on the ARG features at semantic structure now actually have some significance, contra, I suspect, the intention of A&G. In other words, alongside s-structures like (41) for *select*, where there are two arguments labelled ARG_1 and ARG_2, there will also be examples like (42) for *put*, where there are discontinuities in the numberings.

(41)
$$\begin{bmatrix} \text{REL} & \text{select} \\ \text{EVENT} & [\ ] \\ \text{ARG}_1 & [\ ] \\ \text{ARG}_2 & [\ ] \end{bmatrix}$$

(42)
$$\begin{bmatrix} \text{REL} & \text{put} \\ \text{EVENT} & [\ ] \\ \text{ARG}_1 & [\ ] \\ \text{ARG}_2 & [\ ] \\ \text{ARG}_4 & [\ ] \end{bmatrix}$$

Is this a problem? Let us consider A&G's position. In their paper, they evacuate information about thematic roles out of the grammatical architecture by relegating it to the meaning language, and having empty place-holder names for semantic arguments. But without further information, this situation makes a principled theory of mapping impossible: without knowledge of which argument corresponds to which thematic role, *or* which argument corresponds to which grammatical function, we cannot know that 'John loves Mark' means *love(john, mark)*, not *love(mark, john)*, for example. To provide for this, A&G simply stipulate the mappings between GFs and ARG positions. If we want something a little more general, we will need more information. While I share A&G's desire for theoretical parsimony, I think that if they also expect a theory of mapping to provide these mapping equations without something further, they ultimately ask too much. Therefore, one of the two reductions has to be abandoned: either we return thematic role information to the grammar, or we invest the argument names with some meaning.

The first of these reductions, the move to exclude thematic roles from the grammatical architecture, is, I believe, a worthwhile one. Thematic roles are "at best a pretty obscure lot" (as Quine (1956) once said of intensions), beset by multiple theoretical issues. As many have pointed out (e.g. Gawron 1983; Dowty 1991; Ackerman and Moore 2001; Davis 2011), a satisfactory list of roles has never been given. And even when a set of roles is agreed upon, it has not proved possible to find a coherent ranking or hierarchy among them that would apply equally well to all the phenomena for which such hierarchies are adduced (Newmeyer 2002, 65ff.; Levin and Rappaport Hovav 2005, ch. 6; Rappaport Hovav and Levin 2007).

What is more, thematic roles are sometimes thought of as sets of entailments, and it would then certainly seem to make more sense to categorise them as semantic predicates which can take part in such entailments, and which can stand as abbreviations for whatever complex of 'proto-role' properties actually instantiate them (Dowty 1991; Ackerman and Moore 2001). Thus, I believe that A&G's decision to rely on an event semantics which treats thematic roles simply as unanalysed predicates is a sensible one.

But this closes one avenue to a successful mapping theory. Obviously for a verb like *eat* we want, in some sense, to say that the Agent

eats the Patient. In the syntax, this corresponds to the fact that whatever is the subject eats whatever is the object. But we cannot now say that the subject is the Agent, and that the object is the Patient, for example, since we would then be combining terms of the linear logic with terms of the meaning language.[17] Of course, the standard Glue formulation, e.g. (43), expresses the relation between thematic roles and GFs directly:

(43) $\lambda y \lambda x \lambda e.\mathit{eat}(e) \wedge \mathit{agent}(e) = x \wedge \mathit{patient}(e) = y :$
$(\uparrow \text{OBJ})_\sigma \multimap (\uparrow \text{SUBJ})_\sigma \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

But this is overly limiting in two ways. Firstly, it fails to account for morphosyntactic alternations such as passive, where the SUBJ corresponds to the Patient, not the Agent. In this case, we would have to have a different meaning constructor for passive *eat*, which seems wrong, since such alternations are supposed not to alter truth-conditional meaning and so should share the same meaning constructor. Secondly, we are faced once again with the problem of optional arguments, since there will not always *be* an OBJ, but there will always be a Patient argument. Both of these motivate understanding meaning composition in terms of semantic arguments rather than grammatical functions directly. This is where the ARG attributes at semantic structure come in.

Given the advantages of avoiding talk of thematic roles in the architecture of the grammar (a point also emphasised by Kibort as a strength of her approach), the alternative is to give up the assumption that the ARG names are devoid of significance. I do not see this as an inherent disadvantage, however. In A&G's approach, these ARG positions are the connection between syntax and semantics, inheriting this role from argument structure. It does not then seem unreasonable to me that they should in some way explain how they bear this connection. It is not enough, for example, that the two arguments of *eat* be *distinct*; we must also know which one is projected from which grammatical function. So now the question arises: What information do these argument positions encode?

[17] Interestingly, this kind of mixing *is* possible in the so-called 'Old Glue' formulation of Glue Semantics (e.g. in Dalrymple 1999), and Dalrymple *et al.* (1993) take advantage of this to implement a mapping theory very close in spirit to A&G's proposal.

For Kibort, the argument slots in her valency frame are *sui generis*; they are what argument structure consists of, and their function is simply to mediate between semantic and syntactic information. To this extent, they do not seem to *mean* anything. But by virtue of their intermediary role, they embody some information from each structure. For example, in traditional LMT, it is noted that Patient-like arguments tend to be $[-r]$. In Kibort's terms, this means that the thematic role of Patient tends to attach to arg_2 – in other words, arg_2 is in some sense associated with Patient-like properties. Similarly, Agents tend to be $[-o]$, which corresponds to arg_1. So while argument structure, under this approach, is itself technically devoid of semantic/thematic information, it still embodies certain relationships involving this information.

Of course, this is no criticism of Kibort; any theory of mapping will have to model such regularities (indeed, in many senses that is what a theory of mapping *is*). But it does suggest one way of seeing such argument roles (pointed out to me by Mary Dalrymple, p.c.): namely, that they can be thought of as embodying macro-level thematic properties. For example, arg_2/ARG_2 can be seen as grouping together some set of arguments which are 'Patient-like' in whatever way one chooses to elaborate on that concept; but that does not necessarily just mean 'Patients' *per se*. And this can be a source of cross-linguistic variation, much as Butt *et al.* (1997) propose different "intrinsic classifications" for thematic roles in different languages. For example, Goal arguments in English are often ARG_4s – in the unmarked case, they are realised by OBLiques – but in languages with morphological datives, they are often ARG_3s – being realised in the unmarked case by a restricted object, OBJ_{GOAL}.[18]

[18] It might be objected that we have not completely removed mention of thematic roles from the grammatical architecture, since we still have the semantically restricted GFs, which are indexed by thematic role (as illustrated here by OBJ_{GOAL}). However, these indices are really only for f-structure distinctness, and it doesn't especially matter what is used for that purpose. While it is true that 'mainstream' LFG has them indexed by thematic role, we can just as well use numbers, letters, or something else entirely. In fact, in the original formulation of LFG, Kaplan and Bresnan (1982) use the name of a preposition to index OBLiques (e.g. OBL_{WITH}), and we might well extend this to morphological case for restricted objects, so that the example in the text could be rewritten OBJ_{DAT} for dative case.

However, such tendencies must be just that, and nothing more concrete: the key advantage of Kibort's approach is that, like A&G's, it attempts to do without explicit thematic role information, and so any association between argument positions and thematic roles must not be too firm. This allows for what Kibort (2007, 2008) calls semantic participant re-alignment, whereby the same argument slot can have its semantic associations shifted by certain morphological processes (which allows for a better explanation of the patterns of argument-GF linking we observe in these cases). We will see an example of this in Section 6.3.2 below.

6.2 *Formalising Kibort's LMT*

The valency frames which make up Kibort's argument structure are quite esoteric objects. Let us try and formalise them a little more precisely using familiar LFG mechanisms. To clarify matters, we begin by simply rewriting Kibort's valency frame in our own terms as follows:

(44)

<	ARG_1	ARG_2	ARG_3	ARG_4	…	ARG_n	>
	MINUSO	MINUSR	PLUSO	MINUSO		MINUSO	

Kibort imposes no upper limit on the number of argument positions a verb can select, motivated by the fact that there are very many argument-adding operations such as the applicative, benefactive, causative, etc. However, we can draw a distinction, following Needham and Toivonen (2011), between *core* and *derived* arguments. Core arguments are those which are intrinsic to a verb's meaning, such as the two arguments of *devour*: a devouring event is inherently a binary relation, between the devourer and the devourum. This is in contrast to derived arguments, which can be optionally added to certain classes of verb. These include Instruments, Beneficiaries, and Experiencers, as in (45)–(47):

(45) Saint George slew the dragon **with a lance**.

(46) Kim drew a picture **for his sister**.

(47) It seems **to me** as if you don't know the answer.

Reasons of space preclude a detailed analysis of the differences between core and derived arguments here (see Needham and Toivonen 2011, especially pp. 408–413, for more), but what is interesting to

note for our purposes is that, at least in English, derived arguments are often introduced by prepositions, and therefore surface as OBLs. Notably, this corresponds to the fact that all arg positions from arg_4 and above in Kibort's valency frame are marked $[-o]$, the feature which in the unmarked case will surface as an OBL (assuming that there is usually a higher arg position which will be realised as the SUBJ). With this in mind, I propose to associate all argument slots higher than arg_4 with derived arguments. The application of the mapping theory is then restricted to the core arguments of a predicate, specifically the first four, explicitly numbered slots in Kibort's valency frame. By contrast to core arguments, derived arguments will not participate in mapping theory proper, but rather will be introduced lexically/syntactically (see Section 6.3.2 for an example). The new, compact, valency frame is given in (48):

(48) $<$ $\underset{\text{MINUSO}}{\text{ARG}_1}$ $\underset{\text{MINUSR}}{\text{ARG}_2}$ $\underset{\text{PLUSO}}{\text{ARG}_3}$ $\underset{\text{MINUSO}}{\text{ARG}_4}$ $>$

We now turn to the question of how to represent the default mapping principles in terms of the formal apparatus of LFG. Firstly, we need to associate each ARG value with its respective pair of GFs; secondly, we need to ensure that this mapping is optional, since it is always possible not to represent an argument syntactically (if it is encoded in some other way, as in e.g. the short passive, or the optional Patient arguments of *eat* and *drink*). The first task we can accomplish using a defining equation like the one in (49), for ARG_2:

(49) $(\uparrow \text{MINUSR})_\sigma = (\uparrow_\sigma \text{ARG}_2)$

Using the feature decomposition/disjunction introduced earlier, this states that the σ-projection of either the SUBJ or the OBJ maps to ARG_2. Translating all of Kibort's intrinsic assignments into this format, we have the following:

(50) a. $(\uparrow \text{MINUSO})_\sigma = (\uparrow_\sigma \text{ARG}_1)$
b. $(\uparrow \text{MINUSR})_\sigma = (\uparrow_\sigma \text{ARG}_2)$
c. $(\uparrow \text{PLUSO})_\sigma = (\uparrow_\sigma \text{ARG}_3)$
d. $(\uparrow \text{MINUSO})_\sigma = (\uparrow_\sigma \text{ARG}_4)$

For the sake of brevity/clarity, mapping information like this can be captured in a template, MAP (cf. Asudeh *et al.* 2014, 76):

(51) MAP(D, A) :=
$(\uparrow \text{D})_\sigma = (\uparrow_\sigma \text{A})$

MAP(D, A) generates the appropriate functional description to map the feature decomposition D to the argument A. So, for example, a call of MAP(MINUSR, ARG_2) means that one of the GFs in MINUSR will map to ARG_2. Thus, the generalisations in (50) can be captured more perspicuously as follows:

(52) a. MAP(MINUSO, ARG_1)
b. MAP(MINUSR, ARG_2)
c. MAP(PLUSO, ARG_3)
d. MAP(MINUSO, ARG_4)

This format also allows for lexical items to contain additional mapping entries, which augment the defaults in some way. For example, the passive rule discussed in Section 3 can be represented as (53):

(53) MAP(PLUSR, ARG_1)

This is equivalent to adding $[+r]$ to the specification of arg_1 in Kibort's theory. We will return to how the passive is implemented in Section 6.3.1 below.

The second desideratum, optionality, is a little more complicated. One suggestion might be to simply make use of the regular language of LFG's functional descriptions and indicate optionality by surrounding the expression in parentheses:

(54) $((\uparrow \text{MINUSR})_\sigma = (\uparrow_\sigma \text{ARG}_2))$

If we only required one such mapping equation per argument, this would be perfectly acceptable: the resource sensitivity of Glue Semantics would ensure that, unless something else provided the requisite mapping information or alleviated the need for a particular resource to be syntactically realised (such as the predicate being in the passive), these mapping equations would be selected.

However, one of the strengths of Kibort's approach is that morphosyntactic argument alternations can be explained in terms of additional constraints placed on particular argument positions (such as the passive, as discussed above). This means we need to be able to add

extra mapping equations. But we do not want the optionality of each mapping equation to be independent: when we say that the highest argument of a verb is $[-o]$ *and* $[+r]$ we do not mean that it can be just $[-o]$ *or* $[+r]$; if the argument is realised syntactically, it must meet *both* feature restrictions. In other words, for a particular argument, a verb must call *all* or *none* of the relevant mapping equations, not something in between. One way to enforce this is to use a disjunction:

(55) $\{@\textsc{Map}(\textsc{minuso}, \textsc{arg}_1) | (\uparrow_\sigma \textsc{arg}_1)_{\sigma^{-1}} = \varnothing\}$

The second disjunct says that nothing maps to $(\uparrow_\sigma \textsc{arg}_1)$. It does this by stating that the inverse of the σ-function applied to $(\uparrow_\sigma \textsc{arg}_1)$, which names the f-structure(s) which map(s), via σ, to $(\uparrow_\sigma \textsc{arg}_1)$, returns the empty set. In other words, there is no f-structure which maps to $\textsc{arg}_1$. Thus, the whole expression in (55) says that *either* a MINUSO GF maps to $\textsc{arg}_1$, *or* nothing does.[19]

Now, consider the situation where we have two expressions of this form:

(56) $\{@\textsc{Map}(\textsc{minuso}, \textsc{arg}_1) | (\uparrow_\sigma \textsc{arg}_1)_{\sigma^{-1}} = \varnothing\}$
$\{@\textsc{Map}(\textsc{plusr}, \textsc{arg}_1) | (\uparrow_\sigma \textsc{arg}_1)_{\sigma^{-1}} = \varnothing\}$

In this situation, if one of these disjunctions resolves to the MAP template, then the other must as well: any call of MAP which mentions

[19] I am assuming that being mentioned in a meaning constructor is sufficient for an attribute to appear at semantic structure, even in the cases where nothing explicitly maps to it. This seems to be the implication of e.g. Dalrymple's (2001, 250–253) analysis of common nouns, where the attributes VAR and RESTR appear in the semantic structure of the noun, even though nothing explicitly introduces them. If this is not the case, it may be necessary to add an extra equation to the right-hand disjunct of (55) to state that $\textsc{arg}_1$ has *some* value, even if nothing provides it lexically. The expression in (i) might be one way of doing this (suggested to me by Mary Dalrymple, p.c.):

(i) $(\uparrow_\sigma \textsc{arg}_1) = \%\textsc{a}$

This introduces a local variable (Crouch *et al.* 2012) but gives no further information about it. It is intended to be interpreted as meaning "my $\textsc{arg}_1$ has *some* value, but it doesn't matter what". The question of exactly when material appears at s-structure would seem to be an open one, to which further attention must be paid.

ARG_1 is incompatible with a constraint which states that nothing maps to ARG_1. This means that if we select the first disjunct of any of these expressions, we cannot select the second disjunct for any other expression mentioning the same ARG position. This describes exactly the situation we wanted to model: either all the mapping equations relating to a certain argument are chosen, or none are.

We are now in a position to fully encode the default mapping information for each argument position. We abbreviate them in templates, as below; for readability, we also abbreviate the second disjunct which prohibits mapping in a further template:

(57) NoMap(A) :=
$(\uparrow_\sigma \text{A})_{\sigma^{-1}} = \varnothing$

(58) a. Arg1 :=
{@Map(minuso, ARG_1)|@NoMap(ARG_1)}

b. Arg2 :=
{@Map(minusr, ARG_2)|@NoMap(ARG_2)}

c. Arg3 :=
{@Map(pluso, ARG_3)|@NoMap(ARG_3)}

d. Arg4 :=
{@Map(minuso, ARG_4)|@NoMap(ARG_4)}

With this in place, we can now augment any valency templates, such as Agent-Patient-Verb, with the appropriate argument selection templates:

(59) Agent-Patient-Verb :=
@Arg1
@Arg2

$\lambda P \lambda y \lambda x \lambda e.P(e) \wedge \textit{agent}(e) = x \wedge \textit{patient}(e) = y :$
$[(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

Which arguments a verb selects is determined by what valency template it calls, which, in turn, is constrained by the lexical semantics of the verb. Recall that the core component of a verbal lexical entry

includes a predicate which characterises the event it describes; this specification can impose restrictions on what kinds of thematic roles make sense. For example, an intransitive verb like *yawn* could not call the AGENT-PATIENT-VERB template because the nature of a yawning event is such that there can only be one entity involved.[20] We will not discuss the exact nature of such entailments here, since this would take us too far afield into the realm of lexical semantics, but see Dowty (1991), Primus (1999), and Ackerman and Moore (2001) for some discussion.

One final piece of the puzzle is missing. Each call of the MAP template introduces a disjunction: it specifies that one of a pair of GFs is mapped to the relevant ARG position. The question now facing us is how to resolve these disjunctions.

The final instantiation of the mapping equations with particular grammatical functions will be achieved based on the ranking of the ARGs and the GFs, and crucially not by reference to any thematic hierarchy. The arguments are ordered as in Kibort's valency frame, i.e. by their subscript numbers. In other words, the following is true:

(60) ARG_m is higher than ARG_n if and only if $m < n$

We also continue to assume the partial ordering on the GFs given in (15), and repeated here:

(61) $\text{SUBJ} > \text{OBJ}, \text{OBL}_\theta > \text{OBJ}_\theta$

With this in place, the mapping procedure is the same as in Kibort's theory: the highest arguments are linked with the least marked GFs. I leave open the question of how exactly this should be implemented formally: for instance, it could make use of an Optimality-Theoretic framework (in the vein of e.g. Asudeh 2001), or of the similar but distinct approach outlined in Butt *et al.* (1997).

[20] Cognate objects, as in *She yawned a big yawn*, may be thought to pose a problem for this statement. However, unlike the understood arguments of e.g. *eat* and *drink*, they are not core arguments of the predicate, and instead behave semantically like adjuncts, adding additional information about the event which the predicate describes (Asudeh *et al.* 2014, 78–80).

To see the theory in action, let us return to the example of *devour*. The lexical entry for *devoured* will look something like (62):[21]

(62) *devoured* V $(\uparrow \textsc{pred}) = \text{‘devour’}$
@PAST
@AGENT-PATIENT-VERB

$\lambda e.\mathit{devour}(e) : (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

Unpacking the valency template, we obtain (63):

(63) *devoured* V $(\uparrow \textsc{pred}) = \text{‘devour’}$
@PAST

$\{(\uparrow \textsc{minuso})_\sigma = (\uparrow_\sigma \textsc{arg}_1) | \text{@NoMap}(\textsc{arg}_1)\}$
$\{(\uparrow \textsc{minusr})_\sigma = (\uparrow_\sigma \textsc{arg}_2) | \text{@NoMap}(\textsc{arg}_2)\}$

$\lambda P \lambda y \lambda x \lambda e.P(e) \wedge \mathit{agent}(e) = x \wedge \mathit{patient}(e) = y :$
$[(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \textsc{arg}_2) \multimap (\uparrow_\sigma \textsc{arg}_1) \multimap (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

$\lambda e.\mathit{devour}(e) : (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

Assuming that these arguments are syntactically realised (which, in the absence of some valency-reducing alternation such as the passive, they will have to be), we can extract the following mapping equations from (63), with the disjunctions spelled out in the (b) examples:

(64) a. $(\uparrow \textsc{minuso})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
b. $(\uparrow \{\textsc{subj}|\textsc{obl}_\theta\})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$

(65) a. $(\uparrow \textsc{minusr})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$
b. $(\uparrow \{\textsc{subj}|\textsc{obj}\})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

This gives four possibilities:

[21] The template PAST is just like the template FUTURE, but with appropriate changes:

(i) PAST :=
$(\uparrow \textsc{tense}) = \textsc{past}$
$\lambda P.\exists e[P(e) \wedge \mathit{past}(e)] : [(\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma$

(66) a. $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

b. $(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

c. $(\uparrow \textsc{obl}_\theta)_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{subj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

d. $(\uparrow \textsc{obl}_\theta)_\sigma = (\uparrow_\sigma \textsc{arg}_1)$
$(\uparrow \textsc{obj})_\sigma = (\uparrow_\sigma \textsc{arg}_2)$

By appealing to some version of Function-Argument Biuniqueness, we can rule out (66a). If we assume the Subject Condition (that is, that clauses must have subjects), we can also rule out (66d); notice then that the two remaining mappings are the correct ones for the active and passive respectively. However, we do not need to assume the Subject Condition. Following our mapping principles, we simply link the highest argument with the highest GF; however this is achieved formally, (66b) will be the optimal linking, since the highest argument, $\textsc{arg}_1$, is matched with the highest GF, SUBJ. The resulting mapping between f-structure and s-structure is shown in Figure 3.[22]

The meaning constructor for AGENT-PATIENT-VERB in (59) will make $\textsc{arg}_1$ the Agent and $\textsc{arg}_2$ the Patient, as shown in the Glue proof in Figure 4, and so, coupled with the mapping in (66b), we see that the SUBJect is the $\textsc{arg}_1$ which is the devourer, while the OBJect is the $\textsc{arg}_2$ which is the devourum, exactly as desired. [23]

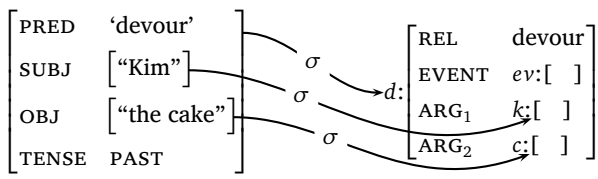


Figure 3: Structures and correspondences for *Kim devoured the cake*

[22] I abbreviate the contents of f-structures for the sake of readability.

[23] In proofs, meaning constructors have been instantiated with respect to the s-structures given in the text.

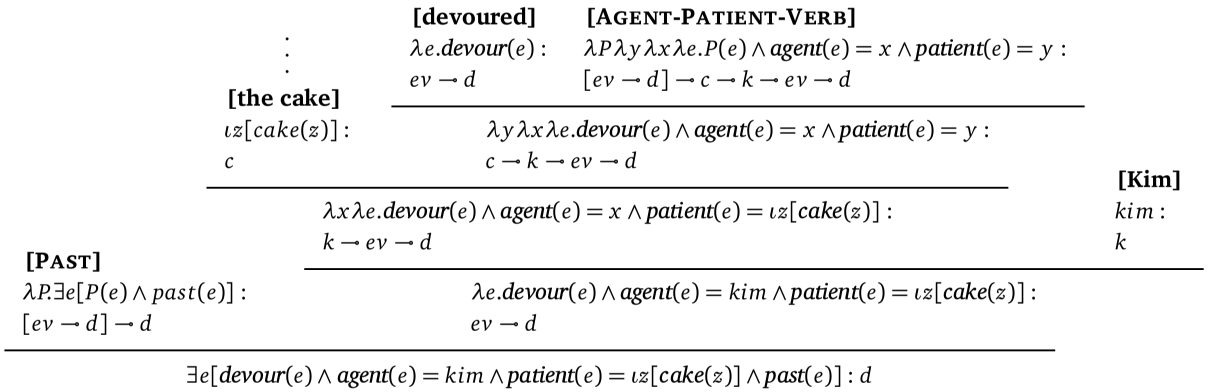


Figure 4: Proof for *Kim devoured the cake*

6.3 *Argument alternations*

The test of any mapping theory is how well it handles argument alternations. In this section, I demonstrate how the current theory handles two such processes, namely the passive and the benefactive. See Asudeh *et al.* (2014) for an example of how it can handle cognate objects, and Lowe (2015) for a compatible analysis of causatives and other complex predicates.[24]

6.3.1 The passive

As mentioned above, the passive involves restricting the mapping for the highest argument, $\textsc{arg}_1$, so that it can appear only as an $\textsc{obl}_\theta$, if it is realised syntactically at all. We can encode this and the remaining information in a template, after A&G (p. 79):

(67) PASSIVE :=
(↑ VOICE) = PASSIVE
{@MAP(PLUSR, $\textsc{arg}_1$)|@NOMAP($\textsc{arg}_1$)}

$(\lambda P.\exists x[P(x)] : [(\uparrow_\sigma \textsc{arg}_1) \multimap \uparrow_\sigma] \multimap \uparrow_\sigma)$

The first line supplies the relevant voice information for f-structure. The second line restricts $\textsc{arg}_1$ in the appropriate way (as we will see below in more detail). The third line contributes an optional meaning constructor which existentially closes a dependency on the meaning of $\textsc{arg}_1$; this will be selected in the short passive but left unused in the *by*-passive (see A&G pp. 75–76 for more detailed discussion).

The lexical entry for passive *devoured* is given in (68):

(68) *devoured* V (↑ PRED) = 'devour'
@PASSIVE
@AGENT-PATIENT-VERB

$\lambda e.\textit{devour}(e) : (\uparrow_\sigma \textsc{event}) \multimap \uparrow_\sigma$

Extracting the mapping information from the two templates, we have the following information:

[24] Lowe (2015) adopts the proposals of A&G on which this paper builds, and his analysis is wholly compatible with the elaboration of that framework presented here.

(69) a. {@MAP(MINUSO, ARG_1)|@NOMAP(ARG_1)}

b. {@MAP(PLUSR, ARG_1)|@NOMAP(ARG_1)}

(70) {@MAP(MINUSR, ARG_2)|@NOMAP(ARG_2)}

Assuming both arguments are syntactically realised, we have the following mapping equations:

(71) a. $(\uparrow \{\text{SUBJ}|\text{OBL}_\theta\}) = (\uparrow_\sigma \text{ARG}_1)$

b. $(\uparrow \{\text{OBL}_\theta|\text{OBJ}_\theta\}) = (\uparrow_\sigma \text{ARG}_1)$

(72) $(\uparrow \{\text{SUBJ}|\text{OBJ}\}) = (\uparrow_\sigma \text{ARG}_2)$

The only way to resolve the ARG_1 mapping disjunctions without contradiction is for the argument to be realised as an OBL_θ. This gives us only two options for the mapping:

(73) a. $(\uparrow \text{OBL}_\theta)_\sigma = (\uparrow_\sigma \text{ARG}_1)$
$(\uparrow \text{SUBJ})_\sigma = (\uparrow_\sigma \text{ARG}_2)$

b. $(\uparrow \text{OBL}_\theta)_\sigma = (\uparrow_\sigma \text{ARG}_1)$
$(\uparrow \text{OBJ})_\sigma = (\uparrow_\sigma \text{ARG}_2)$

Since SUBJ > OBJ on our GF hierarchy, the optimal mapping is (73a), as we require. This is shown in Figure 5.

Notice that regardless of whether ARG_1 is syntactically realised or not, the optimal mapping for ARG_2 will always, correctly, be from the SUBJ.

Assuming that passive *by* is semantically vacuous, the proof for *The cake was devoured by Kim* is identical to that given in Figure 4 (except that the **[PAST]** meaning constructor is provided by the auxiliary *was*).

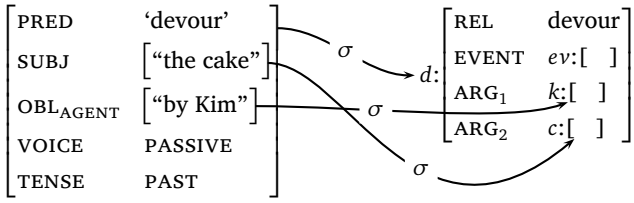


Figure 5: Structures and correspondences for *The cake was devoured by Kim*

6.3.2 The benefactive

Certain verbs in English, like *draw* or *cook*, have lexical alternants which take a core Beneficiary argument:

(74) Alicia drew New York City.

(75) Alicia drew Harry New York City.

We can treat this as zero-marked benefactive morphology, where the morphology introduces the information given in the BENEFACTIVE template below:[25]

(76) BENEFACTIVE :=
@ARG3

$$\lambda x \lambda P \lambda y \lambda e.P(y)(e) \wedge \textit{beneficiary}(e) = x :$$
$$(\uparrow_\sigma \text{ARG}_2) \multimap$$
$$[(\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap$$
$$(\uparrow_\sigma \text{ARG}_3) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$$

As per the discussion of benefactives in Kibort (2007), this adds a new ARG_3 argument to the verb's valency. In addition, the meaning constructor in (76) operationalises Kibort's notion of semantic participant re-alignment (Kibort 2007, 2008), as we will see below.

The lexical entry for regular transitive *drew* is given in (77):

(77) *drew* V $(\uparrow \text{PRED}) = \text{'draw'}$
@PAST
@AGENT-REPRESENTED-VERB

$$\lambda e.\textit{draw}(e) : (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$$

[25] Asudeh *et al.* (2014, 81) introduce the benefactive meaning constructor via an annotated c-structure rule, but Müller (2016) has pointed out various shortcomings facing such an account, including problems with coordination. In the text, I treat it as lexically introduced, thus avoiding these issues. Asudeh (2013) also uses the meaning constructor in (76), as well as the one in (83), below, to encode the requirement of animacy on the subject of the main clause; I omit this in order to simplify the analysis, but it could easily be reinstated.

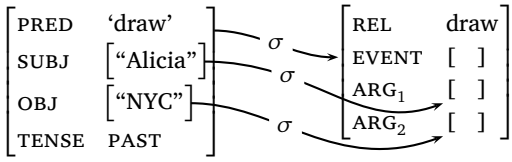


Figure 6:
Structures and correspondences for *Alicia drew New York City*

(78) AGENT-REPRESENTED-VERB :=
@ARG1
@ARG2

$\lambda P \lambda y \lambda x \lambda e.P(e) \wedge \textit{agent}(e) = x \wedge \textit{represented}(e) = y :$
$[(\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma] \multimap$
$(\uparrow_\sigma \text{ARG}_2) \multimap (\uparrow_\sigma \text{ARG}_1) \multimap (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

The mapping proceeds exactly as for *devoured*, and indeed as it would for any simple transitive verb which takes an ARG_1 and an ARG_2. We therefore obtain the structures and correspondences in Figure 6 for a sentence like *Alicia drew New York City*.

The lexical entry for benefactive *drew* is just as in (77), but with the addition of the BENEFACTIVE template:

(79) *drew* V $(\uparrow \text{PRED}) = \text{'draw'}$
@PAST
@BENEFACTIVE
@AGENT-REPRESENTED-VERB

$\lambda e.\textit{draw}(e) : (\uparrow_\sigma \text{EVENT}) \multimap \uparrow_\sigma$

There are now three arguments to be mapped. Since all of them will have to be realised syntactically, we have the following three mapping equations:

(80) $(\uparrow \text{MINUSO})_\sigma = (\uparrow_\sigma \text{ARG}_1)$

(81) $(\uparrow \text{MINUSR})_\sigma = (\uparrow_\sigma \text{ARG}_2)$

(82) $(\uparrow \text{PLUSO})_\sigma = (\uparrow_\sigma \text{ARG}_3)$

I will not list all the possibilities, but we can describe impressionistically how the mapping is determined. Firstly, the highest argument, ARG_1, is linked with the highest available MINUSO GF, namely the SUBJ. Secondly, the next highest argument, ARG_2, is linked with the highest available MINUSR GF, which is now the OBJ. Thirdly, and finally, ARG_3 is linked with the highest available PLUSO GF; since the

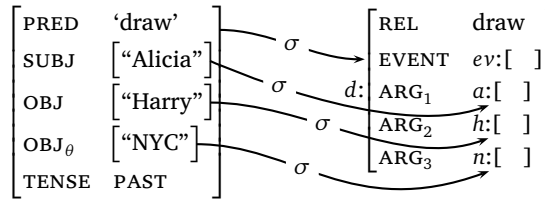


Figure 7:
Structures and correspondences for
Alicia drew Harry New York City

direct OBJ position has been taken, this is OBJ_θ. The mapping is thus as shown in Figure 7.

Notice that the OBJ/ARG_2 no longer corresponds to the drawn entity, but rather to the Beneficiary. This is what Kibort (2007, 2008) refers to as semantic participant re-alignment: in other words, the semantic role of a particular argument position has changed. We achieve this in Glue with the meaning constructor introduced by the BENEFACTIVE template. This specifies that the ARG_2 is the Beneficiary, and then modifies the main verbal meaning so that ARG_3 rather than ARG_2 is passed to it in the position of the Represented argument. This is shown in the Glue proof in Figure 8.

Lexical alternation is not the only way that English can introduce a Beneficiary argument. It can also do so syntactically, using the preposition *for*. In this case, the Beneficiary is a derived argument, and so there is no argument alternation, strictly speaking. This is evidenced in the fact that the basic mapping for the Agent and Represented arguments does not change.

The lexical entry for beneficiary-*for* is given in (83) (after Asudeh 2013):

(83) *for* P $(\uparrow \text{PRED}) = \text{'for'}$
$(\uparrow \text{OBJ})_\sigma = ((\text{OBL}\ \uparrow)_\sigma\ \text{BENEFICIARY})$

$\lambda x \lambda P \lambda e.P(e) \wedge \textit{beneficiary}(e) = x :$
$(\uparrow \text{OBJ})_\sigma \multimap$
$[((\text{OBL}\ \uparrow)_\sigma\ \text{EVENT}) \multimap (\text{OBL}\ \uparrow)_\sigma] \multimap$
$((\text{OBL}\ \uparrow)_\sigma\ \text{EVENT}) \multimap (\text{OBL}\ \uparrow)_\sigma$

This does several things. In the second line, using an inside-out expression (Halvorsen and Kaplan 1988; Dalrymple 2001, 143–146), it maps its object to a new, idiosyncratically named argument position, BENEFICIARY, in the main clause's semantic structure. Since derived arguments do not take part in LMT proper, the attribute names ARG_1–ARG_4 are reserved for core arguments, and derived arguments

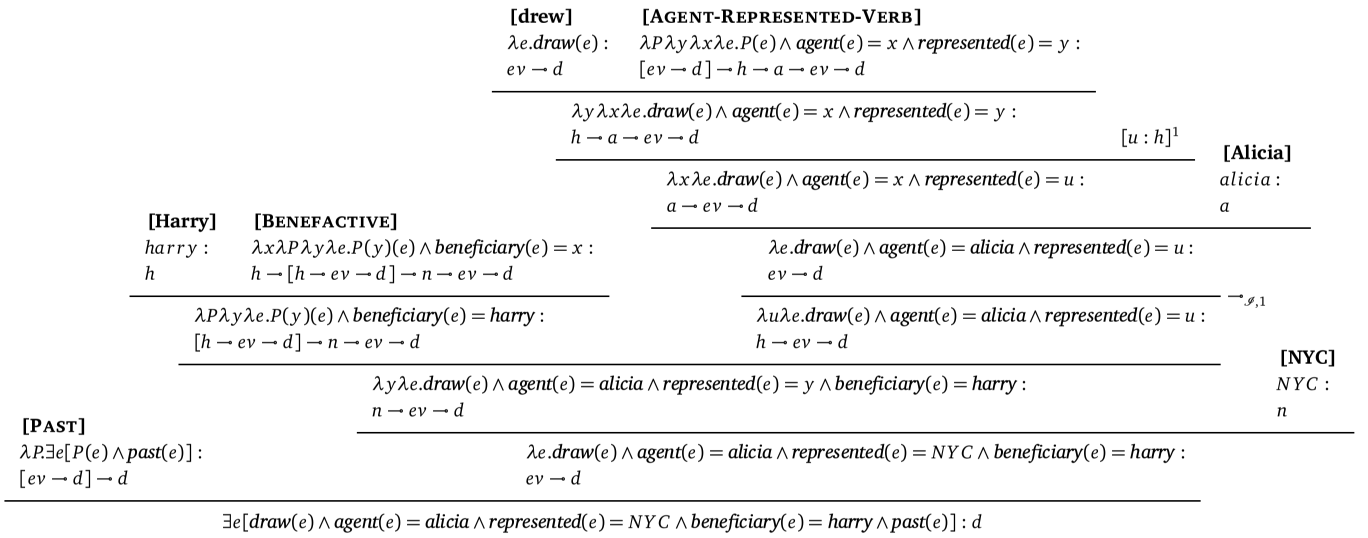


Figure 8: Proof for *Alicia drew Harry New York City*

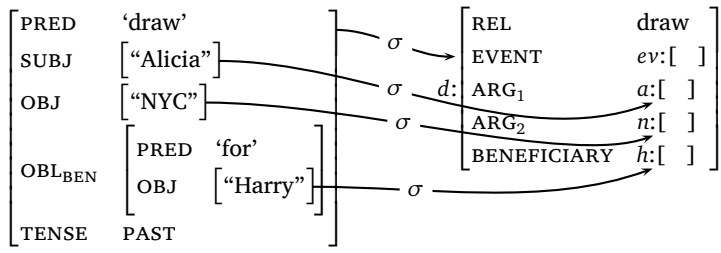


Figure 9: Structures and correspondences for *Alicia drew New York City for Harry*

are instead given mnemonic names at s-structure for (a) distinctiveness and (b) uniqueness, the latter being enforced by the functional nature of s-structure, as discussed above. The mappings for *Alicia drew New York City for Harry* are shown in Figure 9 (the ARG_1 and ARG_2 mappings will proceed as discussed above for regular transitive *drew*).

The third line of *for*'s lexical entry is a meaning constructor which introduces the appropriate Beneficiary meaning. Using the lexical entry for simple transitive *drew* given above, the Glue proof in Figure 10 shows this in action.

7 CONCLUSION

A&G's proposal, to do away with argument structure as a separate level of representation, promises major advances in theoretical parsimony, as well as additional explanatory power. Our grammar is ontologically simpler, and we have a whole new connected structure with internal relations that can be exploited in semantic analyses. However, in the absence of a satisfactory theory of the mapping between arguments and grammatical functions, we lose a great deal of the explanatory power that an a-structure-based mapping theory granted us. In this paper, I hope to have shown that such a theory can be developed, and have chosen to base my approach on recent work in LMT by Kibort. One of the things which sets her proposal apart from earlier versions of LMT is that it argues for a separation of thematic role information and argument structure, which makes it eminently compatible with the A&G proposal, since these authors advocate a very similar position. It is surely encouraging that independent strands of research should have converged in this way. By demonstrating that it is possible to formalise Kibort's theory in terms compatible with the approach of

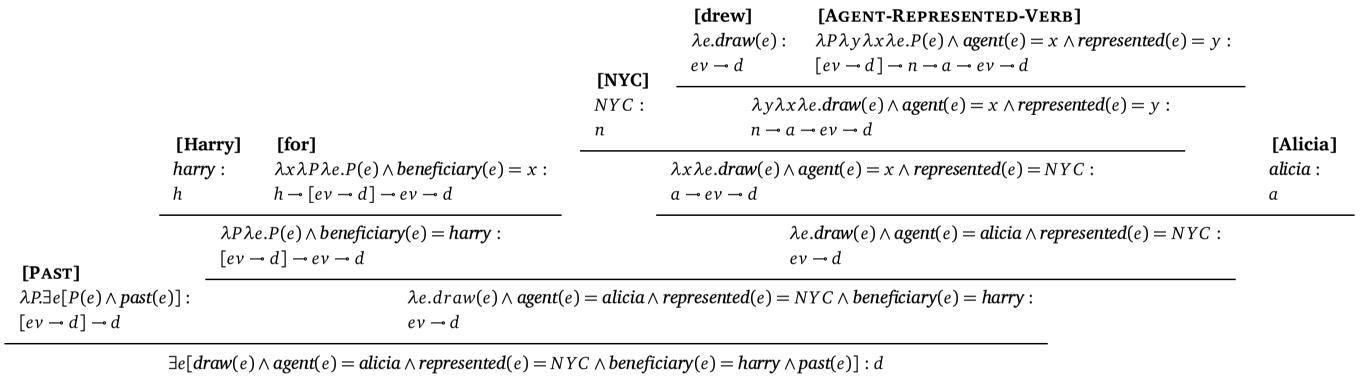


Figure 10: Proof for *Alicia drew New York City for Harry*

A&G, I hope to have lent additional support to both proposals, and laid the foundations for further fruitful work which takes advantage of the strengths of both.

REFERENCES

Farrell Ackerman and John Moore (2001), *Proto-properties and grammatical encoding*, CSLI Publications, Stanford, CA.

Farrell Ackerman, Gregory T. Stump, and Gert Webelhuth (2011), Lexicalism, periphrasis, and implicative morphology, in Robert D. Borsley and Kersti Börjars, editors, *Non-transformational syntax: Formal and explicit models of grammar*, pp. 325–358, Wiley-Blackwell, Oxford, UK.

Alex Alsina (1996), *The role of argument structure in grammar: Evidence from Romance*, CSLI Publications, Stanford, CA.

Ash Asudeh (2001), Linking, optionality, and ambiguity in Marathi, in Peter Sells, editor, *Formal and empirical issues in Optimality-Theoretic syntax*, pp. 257–312, CSLI Publications, Stanford, CA.

Ash Asudeh (2012), *The logic of pronominal resumption*, Oxford University Press, Oxford, UK.

Ash Asudeh (2013), Flexible composition and the argument/adjunct distinction in LFG + Glue, paper presented at the LLI Lab, Institute of Cognitive Science, Carleton University.

Ash Asudeh, Mary Dalrymple, and Ida Toivonen (2013), Constructions with Lexical Integrity, *Journal of Language Modelling*, 1(1):1–54, `http://jlm.ipipan.waw.pl/index.php/JLM/article/view/56/49`.

Ash Asudeh and Gianluca Giorgolo (2012), Flexible composition for optional and derived arguments, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG12 Conference*, pp. 64–84, CSLI Publications, Stanford, CA, `http://www.stanford.edu/group/cslipublications/cslipublications/LFG/17/papers/lfg12asudehgiorgolo.pdf`.

Ash Asudeh, Gianluca Giorgolo, and Ida Toivonen (2014), Meaning and valency, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG14 Conference*, pp. 68–88, CSLI Publications, `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14asudehetal.pdf`.

Ash Asudeh and Ida Toivonen (2015), Lexical-Functional Grammar, in Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis* (2nd edn.), pp. 373–406, Oxford University Press, Oxford, UK, `http://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780199544004.001.0001/oxfordhb-9780199544004-e-017`.

Joan BRESNAN (1980), Polyadicity, in Teun HOEKSTRA, Harry VAN DER HULST, and Michael MOORTGAT, editors, *Lexical Grammar*, pp. 97–121, Foris., Dordrecht, NL, in revised form in Joan Bresnan (ed.) 1982. *The mental representation of grammatical relations*, 149–172. Cambridge, MA: MIT Press.

Joan BRESNAN (1982), The passive in lexical theory, in Joan BRESNAN, editor, *The mental representation of grammatical relations*, pp. 3–86, MIT Press, Cambridge, MA.

Joan BRESNAN (1990), Levels of representation in locative inversion: A comparison of English and Chichewa, manuscript, Stanford University and Xerox Palo Alto Research Center.

Joan BRESNAN (2001), *Lexical-functional syntax*, Blackwell, Oxford, UK.

Joan BRESNAN, Ash ASUDEH, Ida TOIVONEN, and Stephen WECHSLER (2016), *Lexical-functional syntax* (2nd edn.), Wiley-Blackwell, Oxford, UK.

Joan BRESNAN and Jonni M. KANERVA (1989), Locative inversion in Chichewa: A case study of factorization in grammar, *Linguistic Inquiry*, 20(1):1–50.

Miriam BUTT (1995), *The structure of complex predicates in Urdu*, CSLI Publications, Stanford, CA.

Miriam BUTT, Mary DALRYMPLE, and Anette FRANK (1997), An architecture for linking theory in LFG, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG97 Conference*, CSLI Publications, Stanford, CA, `http://web.stanford.edu/group/cslipublications/cslipublications/LFG/LFG2-1997/lfg97butt-dalrymple-frank.pdf`.

Dick CROUCH, Mary DALRYMPLE, Ron KAPLAN, Tracy KING, John MAXWELL, and Paula NEWMAN (2012), *XLE documentation*, Palo Alto Research Center (PARC), Palo Alto, CA., `http://www2.parc.com/isl/groups/nltt/xle/doc/xle.html`.

Mary DALRYMPLE, editor (1999), *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*, MIT Press, Cambridge, MA.

Mary DALRYMPLE (2001), *Lexical Functional Grammar*, number 34 in Syntax and Semantics, Academic Press, Stanford, CA.

Mary DALRYMPLE, Angie HINRICHS, John LAMPING, and Vijay SARASWAT (1993), The resource logic of complex predicate interpretation, in *Proceedings of the 1993 Republic of China Computational Linguistics Conference (ROCLING)*, Hsitou National Park, Taiwan, `http://www.aclclp.org.tw/rocling/1993/K01.pdf`.

Mary DALRYMPLE, Ronald M. KAPLAN, and Tracy Holloway KING (2004), Linguistic generalizations over descriptions, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG04 Conference*, pp. 199–208, CSLI Publications, Stanford, CA, `http://www.stanford.edu/group/cslipublications/cslipublications/LFG/9/lfg04dkk.pdf`.

Anthony R. Davis (2011), Thematic roles, in Claudia Maienborn, Klaus von Heusinger, and Paul H. Portner, editors, *Semantics: an international handbook of natural language meaning*, volume 1, pp. 399–420, Mouton de Gruyter, Berlin, DE.

David Dowty (1991), Thematic proto-roles and argument selection, *Language*, 67(3):547–619.

Yehuda N. Falk (2001), *Lexical-Functional Grammar: An introduction to parallel constraint-based syntax*, CSLI Publications, Stanford, CA.

Dorothee Fehrmann, Uwe Junghanns, and Denisa Lenertová (2010), Two reflexive markers in Slavic, *Russian Linguistics*, 34(3):203–238.

Jean Mark Gawron (1983), Lexical representations and the semantics of complementation, doctoral dissertation, University of California, Berkeley.

Per-Kristian Halvorsen and Ronald M. Kaplan (1988), Projections and semantic descriptions in Lexical-Functional Grammar, in *Proceedings of the International Conference on Fifth Generation Computer Systems*, pp. 1116–1122, Institute for New Generation Systems.

Nikolaus P. Himmelmann (2002), Voice in western Austronesian: an update, in Fay Wouk and Malcolm Ross, editors, *The history and typology of western Austronesian voice systems*, pp. 7–16, Pacific Linguistics, Canberra, AU.

Ray Jackendoff (1990), *Semantic structures*, The MIT Press, Cambridge, MA.

Ronald M. Kaplan and Joan Bresnan (1982), Lexical-Functional Grammar: A formal system for grammatical representation, in Joan Bresnan, editor, *The mental representation of grammatical relations*, pp. 173–281, MIT Press, Cambridge, MA.

Anna Kibort (2001), The Polish passive and impersonal in Lexical Mapping Theory, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG01 Conference*, pp. 163–183, CSLI Publications, Stanford, CA, `http://www.stanford.edu/group/cslipublications/cslipublications/LFG/6/lfg01kibort.pdf`.

Anna Kibort (2007), Extending the applicability of Lexical Mapping Theory, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG07 Conference*, pp. 250–270, CSLI Publications, Stanford, CA, `http://www.stanford.edu/group/cslipublications/cslipublications/LFG/12/papers/lfg07kibort.pdf`.

Anna Kibort (2008), On the syntax of ditransitive constructions, in Miriam Butt and Tracy Holloway King, editors, *Proceedings of the LFG08 Conference*, pp. 312–332, CSLI Publications, Stanford, CA, `http://www.stanford.edu/group/cslipublications/cslipublications/LFG/13/papers/lfg08kibort.pdf`.

Anna KIBORT (2013), Objects and Lexical Mapping Theory [Abstract], in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG13 Conference*, CSLI Publications.

Anna KIBORT (2014), Mapping out a construction invetory with (Lexical) Mapping Theory, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG14 Conference*, pp. 262–282, CSLI Publications, Stanford, CA, http://web.stanford.edu/group/cslipublications/cslipublications/LFG/19/papers/lfg14kibort.pdf.

Jonas KUHN (2001), Resource sensitivity in the syntax-semantics interface and the German split NP construction, in W. Detmar MEURERS and Tibor KISS, editors, *Constraint-based approaches to Germanic syntax*, CSLI Publications, Stanford, CA.

Beth LEVIN and Malka RAPPAPORT HOVAV (2005), *Argument realization*, (Research Surveys in Linguistics Series), Cambridge University Press, Cambridge, UK.

John J. LOWE (2015), Complex predicates: an LFG + glue analysis, *Journal of Language Modelling*, 3(2):413–462.

Stefan MÜLLER (2016), Flexible phrasal constructions, constituent structure and (cross-linguistic) generalizations: a discussion of template-based phrasal LFG approaches, in Doug ARNOLD, Miriam BUTT, Berthold CRYSMANN, Tracy Holloway KING, and Stefan MÜLLER, editors, *Proceedings of the Joint 2016 Conference on Head-driven Phrase Structure Grammar and Lexical Functional Grammar*, pp. 457–477, CSLI Publications, Stanford, CA, http://web.stanford.edu/group/cslipublications/cslipublications/HPSG/2016/headlex2016-mueller.pdf.

Stephanie NEEDHAM and Ida TOIVONEN (2011), Derived arguments, in Miriam BUTT and Tracy Holloway KING, editors, *Proceedings of the LFG11 Conference*, pp. 401–421, http://www.stanford.edu/group/cslipublications/cslipublications/LFG/16/papers/lfg11needhamtoivonen.pdf.

Frederick J. NEWMEYER (2002), Optimality and functionality: A critique of functionally-based Optimality-Theoretic syntax, *Natural Language and Linguistic Theory*, 20(1):43–80.

Terence PARSONS (1990), *Events in the semantics of English*, MIT Press, Cambridge, MA.

Beatrice PRIMUS (1999), *Cases and thematic roles: ergative, accusative and active*, Niemeyer, Tübingen, DE.

Willard V. QUINE (1956), Quantifiers and propositional attitudes, *The Journal of Philosophy*, 53(5):177–187.

Malka RAPPAPORT HOVAV and Beth LEVIN (2007), Deconstructing thematic hierarchies, in Annie ZAENEN, Jane SIMPSON, Tracy Holloway KING, Jane

GRIMSHAW, Joan MALING, and Chris MANNING, editors, *Architecture, rules, and preferences: Variations on themes by Joan W. Bresnan*, pp. 385–402, CSLI Publications, Stanford, CA.