Towards a Dynamic Syntax for Language Modelling

David Tugwell

ITRI, University of Brighton

Abstract. This paper argues that a dynamic, "left-to-right" approach to modelling syntax is best suited to the demands of the language modelling task. An outline of a dynamic grammar is presented, which is based on word-by-word transitions between incomplete "prefix" semantic structures. It is argued that a further advantage of this approach is that it dispenses with the need for any notion of syntactic structure, whether based on constituents or dependencies, and is thus preferable by the argument of Occam's razor.

1 Background

Probabilistic language modelling, the prediction of the next word in a string, has obvious importance in the task of speech recognition and other domains. Important recent work ([1], [2]) has demonstrated that exploiting the information in an incrementally constructed syntactic structure has benefits for language modelling, achieving decreases in perplexity and word-error rate over existing n-gram techniques. Given the nature of language, where statistically significant relations regularly hold between words which are more than two or three words apart, such demonstrations of the usefulness of syntax are long overdue. It is argued in [1] that one reason for this may be that standard approaches to natural language syntax, based on a static syntactic structure, have obvious disadvantages when applied to the task of language-modelling and that dynamic, "left-to-right" grammars are desirable for this task.

Given this, let us consider what properties we might desire from our syntactic model from the perspective of language modelling.

1. **"Deep" rather than "surface" relations**. A little consideration will show that the most significant relations for prediction of words are essentially semantic, rather than the surface syntactic ones such as subject or object. Consider the sentence: *The dog seemed to be about to bark.* The relation of surface subject between *dog* and *seem* is of little use here, we need to construct the deeper relation that *dog* is the theme of the verb *bark*. Of course, in many cases the two relations will coincide, but a syntactic model which only constructs surface syntactic structure will consistently miss important relations with a consequent loss of predictive power.

2. **Incrementality**. It is also of paramount importance that such information can be employed as soon as possible in the left-to-right processing of the sentence, both in order to to make a good prediction of the next word in the string and also to prune sub-optimal paths as early as possible. Approaches based on syntactic constituency often do not allow this, even in cases where it is clear that human language users are in a position to make such early decisions. For example, in the sentence: *The dog barked and growled*, a constituency-based syntax would typically force us to wait until the entire VP *barked and growled* is constructed before we can add this to the NP *the dog* and recover the relation between *bark* and *dog*. Note that the same problem holds for dependency-based approaches such as [3] and even approaches based on categorial grammar with flexible constituents, such as [4].

Now let us turn our attention to how we might build a model of syntax that satisfies these conditions.

2 How to construct a dynamic model of syntax

A model of syntax must specifies the relation between form and meaning, broadly speaking the relation between a string of words (w_1, w_2, $w_3 \ldots$ w_n) and its possible interpretation ($\mathbf{M}_{w_{1..n}}$).

It has been standardly assumed since the inception of generative grammar that this relation must be specified by introducing some abstract level of syntactic structure or derivation. However, it is also possible to make use of the knowledge that not the meaning of sentence does not arise spontaneously at the end of a sentence but is built up on a word-by-word basis. So, given any sentence prefix $w_1 \ldots$ w_i) we can use as objects of our grammar the associated potential meanings of that prefix ($\mathbf{M}_{w_{1..}}$), which we will refer to as prefix meanings.

$$\mathbf{M}_{w_0} \xrightarrow{w_1} \mathbf{M}_{w_{0,1}} \xrightarrow{w_2} \mathbf{M}_{w_{0..2}} \;\ldots\ldots\; \mathbf{M}_{w_{0..n-1}} \xrightarrow{w_n} \mathbf{M}_{w_{0..n}}$$

Assuming such an approach, the whole of the grammar is reduced to specifying the change a word can make to the growing semantic/conceptual interpretation of the sentence, that is the grammar has to define all possible transitions of the form: $\{w_i,\, \mathbf{M}_j,\, \mathbf{M}_k\}$

As such transitions do not depend on previous transitions, but only the present "prefix meaning", we have a simple Markov process. Of course, since the size of prefix meanings increases with the length of the sentence, they do not belong to a finite set, and we do not have a finite-state Markov process.

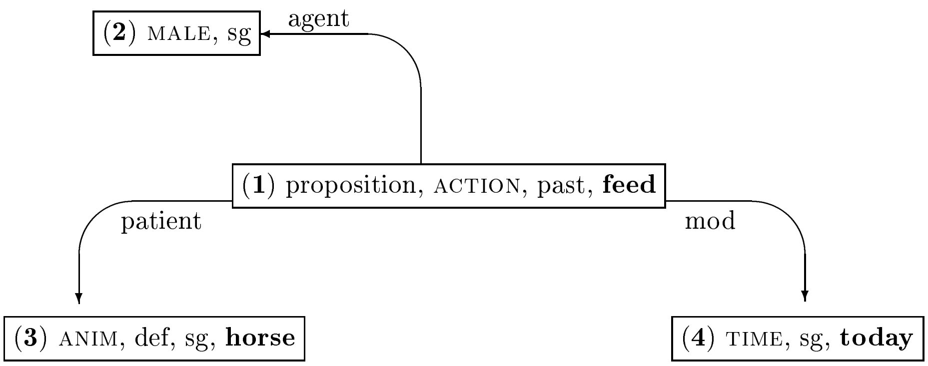
It should be stressed that this is a declarative grammar, not a model of processing. The above represents a *possible* derivation of the meaning of the sentence. At any word there may be multiple transitions possible and it is not the job of the grammar to decide which path or paths to pursue. The distinction is thus preserved between a competence grammar and the use of this grammar in a processing system. This is point often leads to confusion since the grammar is referring to objects (prefix meanings) that in a typical architecture are outside the realm of grammar and have to be specified and constructed by the processor. By putting these objects in the grammar, we are in effect reslicing the competence/performance cake. It is argued that such a redistribution not only simplifies the grammar in that it only deals with objects which are conceptually necessary[1], but that it also greatly simplifies the extra work that the processor has to do in employing the grammar, reducing it to specifying which path or paths to take, when to backtrack and so on.

Although the relation between grammar and processor is therefore much more transparent than in approaches based on constituent structure, it clearly separates the two. This contrasts the approach with models such as ATN's and deterministic parsers such as that described in [5], where with hindsight the unclear relation between grammar and parser was held to be a major drawback to their usability.

To see how such a dynamic grammar might look in practise, we must first look at how the building blocks of the grammar, the prefix meanings, are to be represented.

2.1 Semantic structures

We shall assume a psychologically-real, representationalist conception of semantic structures, akin to that of [6], [3] and others, and that we can represent these in complex hierarchical networks of conceptual constituents linked by semantic/thematic relations. To illustrate, suppose that we can represent important elements of the meaning of the simple sentence *he fed the horse today* in the following network:



[1] Unlike syntactic structures, for which it is generally admitted there is little direct evidence.

We are using here a relatively "flat" representation of the semantic network, which allows us to focus on individual elements: events, entities, states, propositions and so forth. This will be important when we consider how syntactic rules, specifying transitions between such structures, are to be formulated.

2.2 How to specify transition rules

The first problem facing us in formulating transition rules is that prefix meanings are (potentially vast) hierarchical structures. Clearly we need some way to pick out from such structures a small subset of salient information. To this end, we assume that the semantic constituents, in addition to being related by thematic relations with other constituents, are at the same time subject to a salience ordering, that is an ordering of constituents on what we will term an *active stack*, which is employed in the following way:[2]

- Incomplete constituents are placed on the active stack.
- Newly-completed constituents may be removed from this stack.
- The topmost constituent on the stack is termed the *active constituent*.

Syntactic transition rules will then add new structure ($\approx$ the semantic content of the present word), conditional on the present active constituent and the syntactic category of the word. To illustrate this, let us consider the word-by-word derivation of our earlier example: *he fed the horse today*.

	Active stack	Inactive constituents
$\mathbf{M}_{w_0}$:	[1, proposition]	
he		
$\mathbf{M}_{w_{0,1}}$:	[1, proposition, [+2]]	[2, MALE, sg]
fed		
$\mathbf{M}_{w_{0..2}}$:	[1, prop, **feed**, past, agent:2, patient: , [+$\underline{2}$]]	[2, MALE, sg]
the		
$\mathbf{M}_{w_{0..3}}$:	[3, def] [1, prop, **feed**, past, agent:2, patient:3, [+$\underline{2}$]]	[2, MALE, sg]
horse		
$\mathbf{M}_{w_{0..4}}$:	[1, prop, **feed**, past, agent:2, patient:3, [+$\underline{2}$]]	[3, def, sg, **horse**] [2, MALE, sg]
today		
$\mathbf{M}_{w_{0..5}}$:	[1, prop, **feed**, past, agent:2, patient:3, mod:4, [+$\underline{2}$]]	[4, TIME, **today**] [3, def, sg, **horse**] [2, MALE, sg]

[2] This has an obvious correlation with the parse stack of familiar left-to-right parsing algorithms. As discussed above, in the present approach prefix meanings are objects of the grammar and not relegated entirely to the processor.

It will be seen that the first transition (for *he*) adds a constituent which cannot yet be related semantically to the active constituent (the empty proposition). Until such time as it can be interpreted (in this case at the next word) the constituent address is placed on the *store* of the proposition and marked with the *syntactic relation* of subject ("+"). The store will be used in such a way for all temporarily uninterpretable items, not only subjects, but also topics and *wh*-constituents.

The actual transitions in the derivation may be seen to be instances of particular transition rules, which are outlined in the following table. Here the top of the active stack is shown before and after the transition: **X** stands for the information in the old active constituent and **N** for the new information supplied by the lexical entry for the current word.

rule	applies at	lexical entry	top of stack (S_n)	top of stack (S_{n+1})
add-subject	*he*	{nominative}	(X, prop)	(N) (X, prop, [+N]
finite-verb	*fed*	{verb, fin}	(X, prop, [+α])	(X ∪ N, prop, arg1: α)
add-arg	*the*	{det}	(X, arg:)	(N) (X, arg:N)
add-head	*horse*	{ noun}	(X)	(X ∪ N)
modify	*today*	{adv}	(X)	(N) (X, mod:N)

A manageably small number of such transition rules may be produced to cover a significantly large range of English constructions, including *wh*-movement, coordination, raising and others.

2.3 Does dynamic syntax fit the data?

Another constraint on how useful a grammar is, and how interesting from a theoretical point of view, is how faithfully it models the data, here the relationship between strings of words and possible meanings. It has been extensively argued[3] that the dynamic approach offers clear advantages in a number of syntactic phenomena, such as problems of non-constituent coordination, binding, scrambling and man others. From consideration of this range of constructions, it will be seen that the key advantage of the dynamic approach is that it avoids the unnecessary problems that are introduced by the supposition that there is some level of representation (syntactic structure) relating the levels of form and meaning and therefore containing elements of both. The dynamic approach introduces no such level.[4]

[3] For example, in [7], [8], [9], [10] and [11]

[4] A thought-provoking modern take on Occam's razor is given as: "Whatever happens, try all possible ways of explaining it in terms of things we already know about, before introducing aliens or dark matter." Mike Holderness, New Scientist 3rd June, 2000

3 Constructing a probabilistic language model

If we have a dynamic grammar of the type described with a reasonable coverage of grammatical constructions, how can we extend this into a language model?

The syntactic rules and the lexicon will provide us with possible transitions $\{w_i, S_j, S_k\}$, but we need to estimate the probability of such transitions occurring $\mathrm{P}(S_k|w_i, S_j)$. The basic factors to be used in such calculations will be collocations between the head of a constituent and the head of its arguments or modifiers. So in the example *the stray dog was barking loudly*, we will need estimates for the collocations: *bark'(theme=dog')*, *bark'(manner=loudly')* and *dog'(quality=stray')*.

If we factor in estimates of the probability of such relations we are then in the position to make predictions of the next word. Luckily, such counts can be obtained from large corpora to a usable degree of accuracy with simple regular expression techniques. To counter problems of data sparseness there will be a need for smoothing, possibly using similarity metrics.

The final requirement for a processor is a suitable control structure, determining which potential paths of the derivation to pursue and which to reject and when.The processor in [1] employs the beam search algorithm, where a number of paths are pursued in parallel subject to them satisfying some threshold criteria.

4 Conclusion

This paper has outlined the architecture of a dynamic model of syntax. Although very much in their infancy, such models offer the enticing prospect of a single approach being used both for the practical task of language modelling for speech recognition, as well as for addressing some outstanding problems in theoretical linguistics.

References

[1] Chelba, Ciprian & Frederick Jelinek: Exploiting syntactic structure for language modelling. COLING-ACL (1998) vol I 225–231

[2] Wu, Jun & Sanjeev Khudanpur: Combining nonlocal, syntactic and n-gram dependencies in language modelling. Proceedings of Eurospeech'99 vol 5 (1999) 2179–2182.

[3] Hudson, Richard: *English word grammar*. Oxford: Blackwell (1990)

[4] Steedman, Mark: *The syntactic process*. MIT Press/Bradford Books (2000)

[5] Marcus, Mitchell P.: *A theory of syntactic recognition for natural language*. MIT Press (1980)

[6] Jackendoff, Ray: *Semantic structures*. MIT Press (1990)

[7] Hausser, Roland: *Foundations of computational linguistics*. Berlin: Springer (1999)

[8] Kempson, Ruth, Wilfried Meyer-Viol & Dov Gabbay: *Dynamic Syntax: The Flow of Language Understanding*. Oxford: Blackwell (to appear)

[9] Milward, David: Dynamic Dependency Grammar. Linguistics and Philosophy **17** (1995) 561–615

[10] Philips, Colin: Order and Structure. PhD thesis, MIT (1996)

[11] Tugwell, David: Dynamic Syntax. PhD thesis, University of Edinburgh (1999)