Cleaning up the lexicon

Michal Starke

2013-05

Terminological issues aside, there is near-universal agreement among linguists that the lexicon is a messy and ugly place. A place of disorder, exceptions and cacophony. This contribution is a short outline of why this consensus may prove to be wrong. More precisely, why it is vacuous on one reading, and probably wrong on the other. As an alternative, I will outline an approach with a clean, principled and restrictive lexicon, whose core is:

(1) The (syntactic) lexicon contains nothing but well formed syntactic expressions

On this view, the (syntactic) lexicon is like a museum of interesting syntactic structures; or in computational terms, a persistent database of representations generated by the syntactic engine. Nothing else. As we will see, this is very much the opposite of the standard view -- and the difference comes from the architecture of language that leads to those two views.

But first, let us see what is meant by a messy and unprincipled lexicon. On one reading, the lexicon is messy because it is full of irregularities, unpredictable facts. The putative ugliness is for instance in the fact that the concept TABLE is linked to 'stol' in some languages but 'tisch' in others. Or in the fact that the concept MONEY is assigned the value *+count* in some languages and *-count* in others. Unpredictably. Claiming that this makes the lexicon disorderly is very much like saying that the first Principles & Parameters implementation was disorderly because the value assigned to any particular parameter of any particular language was unpredictable. That theory may well have qualified as messy, but certainly not for this reason. It is the whole point of (that style of) parameters - and of lexical items - that they are containers for unpredictable choices among a range of possible values and there is nothing unprincipled about that. On this reading, the messiness claim is thus vacuous.

There is a more serious reading however. A reading which is true of current approaches to the lexicon, and which concerns the format rather than the content of the lexical entries. In the currently standard view, lexical entries are 'the stuff below syntactic terminals', stuff syntax doesn't get to manipulate or build. And hence those lexical entries are by definition non-syntactic. Syntax acts at a higher level, building phrases above the lexemes-in-terminals. The corrolary of this is that we must invent some new non-syntactic technology for the internal format of lexical entries. The accurate claim of lexical ugliness and messiness concerns that technology: it is unprincipled and unrestricted.

There is an abundance of such technology thrown into the lexicon: context-sensitive rewriting rules (eg. for allomorph selection), subcategorisation frames with their own notational and syntactic conventions, new types of mergers (for features), notations for coreference, sometimes a notational apparatus for thematic role alignment with its

own internal syntax, etc. Starting from the simplest case, consider the widespread idea that there is such a thing as 'feature bundles' in terminals/lexemes. Trivially, a 'feature bundle' is equivalent to a constituent. Enclosing elements inside square brackets is a notational variant of linking those elements under a single mother node. Feature bundles are thus trees, typically flat n-ary trees with n > 2. This means that a syntactic representation with 'feature bundles' in its terminals is composed of two types of trees, each with their own conventions: the binary branching syntactic nodes, and the n-ary branching lexical nodes at the bottom. The binary branching nodes are targets for movement, the n-ary branching ones are not. The binary branching nodes are targets for binding, the n-ary branching ones are not; etc. In other words, we just invented a second syntax and a new type of merge, for the purpose of lexical storage. A powerful move indeed. In fact, a move of unwarranted power.

Now consider subcategorisation. How is that information associated with each lexical item? There are various proposals for this in the literature, whose common thread is to invent new notations and conventions to express the content of the various arguments, how to link them to the internal and external argument slot, how to express their optionality or lack thereof, etc. This notational apparatus comes with its own conventions, i.e. its own syntax. How does that syntax relate to the flat-tree apparatus of feature-bundles? It generally doesn't, it is simply another syntax juxtaposed to conventions surrounding bundles.

A little further lies maybe the most powerful of tools in the lexicon: context-sensitive choices, typically operating on yet another data-structure (lists). An example often brought up is that of the English past tense. How do we express the fact that some verbs appear to be suffixed by *-en* instead of the regular *-ed*, to express past tense. A popular proposal is that the lexical entries include notation capable of expressing "if I am affixed to one of the following list of stems, then spell me out as EN (rather than the usual ED)", sometimes rendered as "in the context of a member of the following list, spell me out differently". This is yet another representational and computational device -- in fact a very powerful computational device. How does it relate to the technology invented for subcategorisation frames and to the technology invented for feature bundles? It doesn't, it just sits next to it. And the list goes on.

In this sense, it is accurate to say that current theories of the lexicon are both unprincipled and unrestricted. The *format* of the entries they contain (as well as the operations performed on them, where applicable) is unprincipled and unrestricted, an unholy mix of f-constituents ('bundles'), frames, rewrite rules, etc. Much of this mess seems unavoidable if the lexicon must store information outside of the domain of syntax, living in atomic, sealed boxes inside syntactic terminals. Being outside the purvey of syntax, it cannot reuse the tools of syntax and must invent its own notations and conventions.

But all of this is avoidable in a different view of grammar, a view in which lexical items don't live below syntactic terminals. In that world, all the ad-hoc powerful apparatus described above can be dispensed with, in favor of (1), repeated here:

(2) The lexicon contains nothing but well formed syntactic expressions

To get there, I need to briefly outline this different view of grammar, and how we got there. The starting point is what I take to be one of the main results of syntactic research over the last few decades: the ingredients manipulable by syntax are much smaller and more numerous than originally thought. As syntactic theories try to become more accurate in describing word orders beyond the bare bones of subject, verbs, objects and complementisers, while keeping their explanatory power, they become sensitive to

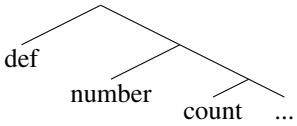
smaller and smaller entities -- quickly reaching the boundary where syntax operates on individual features. Along that path, syntactic terminals have reached and passed the point of being smaller than individual morphemes.

If the content of syntactic terminals is typically smaller than morphemes, where are the lexical items? Where are the morphemes? Where are the words? Descriptively, a single morpheme now corresponds to the conjunction of several syntactic terminals. How do we express that? We already have a tool expressing the 'conjunction of several syntactic terminals': a constituent. Once terminals become smaller than morphemes, morphemes correspond to entire phrases rather than to terminals. Hence the notion of phrasal spellout (this particular reasoning was first laid out in Starke (2002), see Caha (2009) for a friendly exposition of the technology associated to it). A morpheme is thus a syntactic constituent, with individual features as its terminals.

These syntactic constituents are what is stored in the lexicon. No feature bundles, no subcategorisation frames, no context-sensitive choice-rules. Only legal syntactic trees. And hence no messy, unprincipled, unrestricted technology is involved in the (syntactic) lexicon. On the contrary, this architecture of grammar makes the lexicon benefit from decades of research into making the syntactic apparatus more principled and minimal, thereby yielding an principled, restrictive and austere lexicon. Somewhat like a Swiss garage.

A lexical item whose syntactic content would traditionally be rendered as say [+definite, +count, +singular], eg. a third person singular personal pronoun, now comes out as:

(3)



The resulting workflow between syntax and the lexicon is the following. First syntax merges the two lowest features into a constituent, *number* and *count* in this example. Having done that, it consults the lexicon: is this newly created phrase lexicalisable? I.e. does the lexicon contain this phrase in any of its entries? If the answer is positive (i.e. not empty), syntax takes note of the lexical candidates (see below), and proceeds. At this point, the cycle of merger + lexical access repeats itself; in this case the next cycle of merger merges *definite* with the previous constituent, etc.

(Given this setup, a large part of language acquisition consists in detecting which of the constituents produced by syntax are 'interesting' enough to warrant being stored in the lexicon, paired with (at least) their corresponding sound patterns and conceptual content. Their interest typically stems from their unpredictability, but could also come from their frequency.)

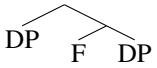
This is obviously only the barest outline of the workflow, leaving out a large number of interesting technical issues, but this is all we need for the point of this squib: phrasal spellout makes a clean and principled lexicon possible. (For various aspects of how this process of phrasal spellout works, see Starke (2009), Starke (2011), and for similar takes on phrasal spellout, Caha (2009), de Clercq (2013), Rocquet (2013), Markus (2013), etc.)

2. Subcategorisation. If legal syntactic trees is all the (syntactic) lexicon has, how do we express the fact that some lexical items require or admit arguments and others don't? There are two general line of thoughts about addicity. One is to invent a notation and syntax dedicated to it. The other stems from Frege's 'saturation' metaphor. Frege's intuition was that argument-taking entities intrinsically have 'hooks' which

allow arguments to snuggle into them -- they are 'unsaturated' and arguments 'saturate' them. This is typically rendered precise with the technology of functions applied to arguments. Lexemes seen as trees however give us a novel understanding of Frege's 'saturation' metaphor.

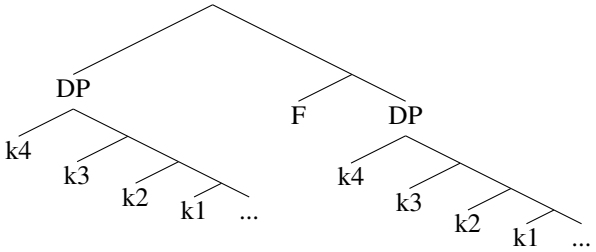
Consider the structure of an argument-taking lexeme, such as the verb *caress*. We need to express not only eventive, categorial, etc. features involved in that verb, but also the fact that it takes an internal argument, and the fact that it takes an external argument of a certain kind (eg. agentive). Let us call F the eventive, categorial, etc. features involved in *caress*, a grossly simplified representation of the syntactic structure involved is then:

(4)



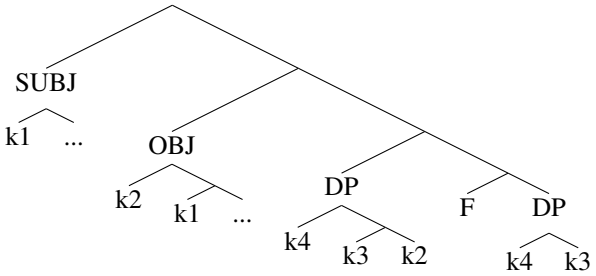
Following Caha (2009), each of the DPs starts its life with a relatively rich set of case projections:

(5)



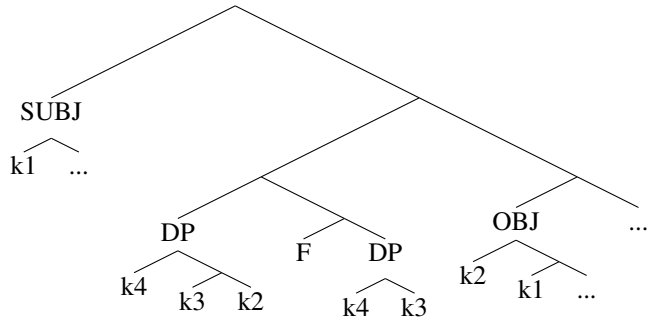
Nominative corresponds to k1, accusative to k1+k2 (genitive to k1+k2+k3, etc. -- see Caha (2009)). The external argument ends up being nominative (k1 only) despite being born with all these layers above it because the traditional movement of specVP to specTP corresponds to the movement of K1P up to specTP, stranding [k4 k3 k2] behind. Mutatis mutandis for the accusative: K2P moves out, stranding [k4 k3] behind. In an SOV language, this would be the end of the story:

(6)



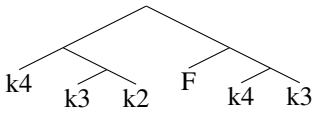
In an SVO language such as English, the remaining parts of the verbal constituent are dislocated to the left prior to the subject movement:

(7)



In both the SOV and SVO case, the verb now lexicalises the entire remnant constituent. That is, the (syntactic) lexical entry of *caress* is:

(8)



The novelty is that the upper layer of the nominal sequences are part of the lexical entry for the verb itself. They are Frege's hooks, they are the unsaturated parts of the verb. They are 'unsaturated' in the sense that building a verbal functional sequence will not allow the verb to spellout out, the verb is 'incomplete', or 'unsaturated' at that point. The only way to spellout the verb, i.e. to match the lexical entry (8) is to generate two additional noun phrases, and have them strand their upper layers to create the configuration (8). In this way, they saturate the verbal projections, making the verb full.

Rephrasing this in traditional generative terms, this gives a novel interpretation to 'theta role assignment'. What is going on is not so much 'assignment' of a thematic role by a verb, but rather the verb itself spelling out the thematic layers of the noun. This expresses the relational aspects of thematic roles without the need for the technology of assignment: the thematic information starts its life as a property of the noun, qua high functional projections in the nominal sequence, and ends it life as part of the verb, once the derivation has stranded those high nominal functional projections near the low verbal projections, and they get spelled out as the verb.

We thus have lexical entries which express both bundles and subcategorisation, without anything more than familiar syntactic entities: binary constituents and movements.

3. Irregular forms. An important part of the lexicon is that it stores 'irregular' forms. Forms that seem to fall outside of the regular patterns. Much of the traditional messy technology is invented in order to deal with just this. It is thus one of the nicest fallout of a phrasal lexicon that irregularity (and semi-regularity) can be brought back to perfect regularity. The core difference between so called 'irregular' forms and their regular cousins is that irregular lexemes happen to store a tree of a different size wrt. the regular forms.

To see this, let's walk through one traditional example: the irregular English plural *mice*. The logic of the situation is simple: what needs to be expressed is the fact that English has a special morpheme dedicated to express '*mouse* + plural', whereas in the general case, nouns get spelled out on their own, and the plural gets spelled out on its own. Phrasal spellout allows us to implement just that: *mice* is the spellout of a bigger tree which contains both the noun and the number features, whereas the regular

pattern is to store two smaller trees, one for the noun and one for the plural suffix. Here are stylised entries for *rats* and *mice*, assuming that countable nouns are a countP constituent whose internal structure is irrelevant for present purposes:

(9) a. *rat* countP
b. *-s* pluralP
c. *mice* [countP pluralP]

Again, I am skipping many relevant technical details, to concentrate on the logic of the situation: once the lexicon is phrasal, we can throw away all the messy and dangerously powerful tools in our lexical artillery, and reuse the patiently crafted results of decades of syntactic research instead. In this case, an entire category of 'irregularities' vanishes in favour of a much simpler statement: one lexical item contains an unusually big syntactic constituent.

There is however one piece of technology that I do want to walk you through, at least in its outline. This is because it offers an important insight into the internal structure of the lexicon itself. To introduce it, let us look at a potential problem. The syntactic entry for *mice* in (9) is *[countP pluralP]*. This entry however matches any plural count noun, not just *mice*. And hence we run the risk of mispredicting that every plural count noun in English comes out as *mice*. Put differently, the lexical entry (9c) expresses 'the plural of a countable noun is *mice*', but what we want to express is 'the plural of *mouse* is *mice*'.

The solution to this issue lies in the spellout workflow outlined above. Recall that after a merger, a lexical lookup is performed, the chosen lexeme is remembered by syntax who then proceeds with the next merger, etc. In our case, the relevant merger is that of *count* with its complement. For concreteness, let us assume that the complement of *count* is the feature *entity* (interpreted as the common ground between *count* and *mass*, and in opposition to a pure *property*). The features *count* and *entity* thus get merged and the lexicon is consulted for the resulting *countP*, which yields numerous candidates (rat, mouse, horse, elephant, table, chair, etc.). At this point, the speaker's cognitive apparatus makes a choice between those equivalently apt candidates [1]. In our case, *mouse* is chosen, and we can informally describe the result of the lexical access as:

(10) countP{mouse}

The next step is to merge this with *plural*, yielding (ignoring issues of linearisation, possible remnant movements, etc):

(11) [countP{mouse} plural]

I have noted above that the acquisition process now largely consists of detecting which syntactic phrases are 'interesting' and storing them in the lexicon. Suppose that the learner successfully detects that there is something interesting about (11) and decides to store it. The solution to our problem resides in the fact that there are two ways of storing (11). Either the learner decides to store this as a general case, applicable to all [countP plural] configurations, in which case they will store it ignoring the annotation for the previous round of lexical access:

[1] This is the narrow space where free choice enters the picture, and is the pure-late-insertion equivalent of creating a numeration, except it is restricted by the built-up syntactic context instead of being unconstrained as it is in numeration-based approached.

(12) mice: [countP plural]

Or they decide that this is an entry that this applicable only to that particular derivation, with that particular lexical choice, and they will store the entire configuration (11) as is:

(13) mice: [countP{mouse} plural]

In other words, the way cyclic syntax works gives the lexicon the opportunity to store entries with or without references to prior lexical choices. In this case, (13) is clearly the more correct choice. Our problem is now gone, and the door is opened to handle another class of exceptions: idioms. An idiomatic expression such as the proverbial *kick the bucket* is nothing else than a higher level constituent being stored in the lexicon, together with its prior lexical choices, in this case *kick* and *bucket*.

A welcome side-effect of our setup is thus to unify the treatment of word-level irregularities such as *mouse/mice* with the analysis of idioms such as *kick the bucket*. To see what underlying technology made this solution possible, let's take a step back and look at the overall picture. The traditional view of irregular morphology is so to speak 'horizontal': one morpheme makes a statement about its sister morpheme (eg. "if I am next to this root/stem, pronounce me as EN instead of ED", or vice-versa). As we have seen, this leads to the introduction of powerful (and unnecessary) technology, but it also makes it hard to unify such cases with idioms. Phrasal spellout allows us to shift to a "vertical" approach to irregularities: what makes two sisters irregular is that their mother node is itself stored in the lexicon, with a specification that she wants particular daughters. This shift from a horizontal statement to a vertical approach allows us to generalise the solution to many kinds of irregularities and semi-regularities, at various levels of structure.

4. Conclusion. The aim of this short paper has been to illustrate that *phrasal spellout enables a clean and principled lexicon*. Every (syntactic) lexical entry has the same restricted and principled format: it is a well formed syntactic structure produced by the syntactic engine. There is no need for a new constituency type under the guise of 'feature bundles', the technology of 'subcategorisation frame' or the apparatus of context-sensitive decision-making, among others. The availability of principled structured representations in the lexicon allows us to express bundling, subcategorisation and irregular shape-choices with nothing more than lexical lookups operating on standard syntactic (sub-)trees. Some particulars of this process are scattered across the nascent Nanosyntactic literature (see eg. Caha (2009), Pantcheva (2011), de Clercq (2013), Markus (2013), Rocquet (2013)), this squib brings those threads together into a unified vision of a principled, restrictive and austere lexicon.

References

Caha, Pavel. 2009. The nanosyntax of case. phd, University of Tromsø. URL http://ling.auf.net/lingbuzz/000956.

de Clercq, Karen. 2013. The upward path of negation. phd, University of Ghent.

Markus, Andrea. 2013. Building hungarian monoargumental verbs. phd, University of Tromsø.

Pantcheva, Marina. 2011. Decomposing path: The nanosyntax of directional expressions. phd, University of Tromsø. URL http://ling.auf.net/lingbuzz/001351.

Rocquet, Amélie. 2013. Split o and you get ϕ. phd, University of Ghent.

Starke, Michal. 2002. The day syntax ate morphology.

Starke, Michal. 2009. Nanosyntax - a short primer to a new approach to language. *Nordlyd* 36:1–6. URL http://ling.auf.net/lingbuzz/001230.

Starke, Michal. 2011. Towards elegant parameters: Language variation reduces to the size of lexically stored trees. In *Linguistic variation in a minimalist framework*, ed. C. Picallo. Oxford University Press. URL http://ling.auf.net/lingbuzz/001183.