Ellipses: Computation of

Greg Kobele

January 25, 2012

Abstract

A computational account of ellipsis should specify not only how the meaning of an elliptical sentence is computed in context, but also a description of what is being computed. Many proposals can be divided into two groups, as per whether they compute the meaning of an elliptical sentence based on the semantic or the syntactic parts of its context. A unifying theme of these proposals is that they are all based on the idea that the meaning of an elliptical sentence is determinable based on a structured representation which is transformationally related to its surface syntactic structure.

1 Introduction

A computational account of language has two parts. First is a specification of which sounds (or more generally signals) convey which meanings. This establishes what it means for an algorithm to be ‘correct’. Second is the development of algorithms which compute this specification – of particular interest here is parsing (whereby a set of possible meanings is computed from a signal).

Here we investigate the phenomenon of ellipsis, as illustrated in 1. Informally, an elliptical sentence is one which is ‘missing’ a piece, and an ellipsis site is the position in an elliptical sentence where the piece is missing (written here with an italicized e).

(1) John likes Mary, even though Bill doesn’t e.

Example 1 is easily understood to mean the same as example 2.

(2) John likes Mary, even though Bill doesn’t like Mary.

Crucially, example 1 is not understood to mean the same as example 3.

(3) John likes Mary, even though Bill doesn’t like John.

The missing information in 1 is, at least intuitively, present in this case in the broader context. The puzzle is to specify exactly what contextual information is relevant for the resolution of the meaning of an elliptical sentence, and in such a way that all and only possible meanings are associated with elliptical sentences. A straightforward idea is that the hearer reasons about what the speaker might have intended, and interprets the elliptical sentence as meaning *that*. One difficulty with this view is given by ill-formed sentences like 4 (inspired by Hankamer and Sag), where a clever hearer can nevertheless reason that the thirsty man promised not to drink the speaker’s precious water. This is to be contrasted with the well-formed example in 5.

(4) *I do not hesitate to leave my precious water in front of the thirsty man because he said that he didn’t want to e.

(5) I do not hesitate to drink my precious water in front of the thirsty man because he said that he didn’t want to e.

These examples suggest that even if we had the correct theory of how we actually reason about the intentions of others, that theory would make incorrect (i.e. overgenerous) predictions about what elliptical sentences can mean. The idea we will explore here
is that the difference between 4 and 5 is that 5 (but not 4) contains a part which the hearer can use to reconstruct the meaning of the sentence. While there could in principle be any number of ways to do this, it appears that, crucially, the work necessary is often recastable in terms of simply finding an antecedent part of the context, and identifying the ellipsis site (the $e$) with this. There are two, superficially quite different, main approaches to this problem. On a semantic approach, the material which gets filled in is a meaning representation, while on a syntactic approach, it is syntactic structure which is being reconstructed. Most theories can be grouped into one or a combination of these two approaches.

This paper is structured in the following manner. In the next section (2), we describe some of the basic elliptical phenomena which have given rise to the current state of elliptical theorizing. In §3, we present the syntactic and semantic approaches in a uniform way using the lambda calculus. Section 4 discusses the issues surrounding the algorithmic realization of these approaches in a parser. Finally, §5 is the conclusion.

2 Data

There are a few ‘differences’ between the surface form of the antecedent and how the ellipsis site could be filled in by overt material (i.e. how the elliptical sentence would have been had it not contained ellipsis) which have motivated many researchers working on the subject. In §2.1 we discuss cases where ellipsis forces one reading of an otherwise ambiguous sentence. In §2.2 we discuss difficulties raised by pronouns and referring expressions in the antecedent. In §2.3 we discuss cases where properly syntactic generalizations seem to play an important role, and in §2.4 we discuss the particularly tricky cases of split antecedents, and sloppy event identity.

The acceptability of elliptical sentences is strongly influenced by various factors, among which are discourse relations between antecedent and ellipsis site and information structure. It is an open question whether to treat the marginally acceptable cases as fully grammatical, as ungrammatical, or somewhere in between. Particular care must be taken when drawing conclusions based on unacceptable data points – they may be unacceptable for reasons orthogonal to the matter at hand!

2.1 Ambiguities

Ellipsis can disambiguate otherwise ambiguous sentences. As noted by Lees (see also Sag and Williams), sentence 6 is only two ways ambiguous (both the children and the chickens are ready to eat something, or their time in the oven has elapsed). This is interesting because in the sentence without ellipsis, 7, both conjuncts can be interpreted independently. (Although they can be interpreted independently, they typically are not. See Prüst et al. and Hobbs and Kehler for references and discourse parallelism based theories of ellipsis.)

(6) The chickens are ready to eat, but are the children $e$?

(7) The chickens are ready to eat, but are the children ready to eat?

Sentences like these suggest that the representation we use for reconstructing ellipsis sites must distinguish between strings of words with different meanings.

Scopal ambiguities can also be disambiguated in elliptical contexts. While the first conjunct of 8 is ambiguous in isolation (meaning either that there is a particular individual who hit everyone, or that each person was hit, but possibly by different people), the sentence in 8 is unambiguous – there are two people, namely Bill and someone else, who hit everyone. Example 9 demonstrates that it is not the case that ellipsis simply rules out the wide scope reading of object noun phrases; here the preferred reading is that each window has hanging in front of it both a Canadian and an American flag. Still, whichever scope relation obtains in the first conjunct between subject and object must be mirrored in the second conjunct.

(8) Someone hit everyone, and then Bill did $e$.

(9) A Canadian flag was hanging in front of each window, and an American on was $e$ too.
These examples indicate that the representation we use for ellipsis must include scopal information.

2.2 Pronouns

A major puzzle for theories of ellipsis has traditionally lain in the interpretative possibilities of anaphora under ellipsis (beginning with Ross\textsuperscript{14}; see Kehler and Shieber\textsuperscript{15} and references therein). An example is in 10, which, if we take the first conjunct to mean that John likes his own mother, can be understood either as saying that Bill also likes his own mother (the ‘sloppy’ reading), or that Bill likes John’s mother (the ‘strict’ reading). The ‘missing reading’ is one where his in 10 is understood as bound by John, while ‘in the ellipsis site’ is treated as deictic. Similarly, mutatis mutandis, when his is deictic in the first clause.

\begin{align*}
\text{(10) John likes his mother, and Bill does } e \text{ too.}
\end{align*}

There have been two prominent ways of interpreting this data. By reasoning similar to that of the previous subsection, we might conclude that the representation used for ellipsis distinguishes between bound and free pronouns (as would a semantic representation where pronouns are translated as variables\textsuperscript{8}). Another alternative is to conclude that ellipsis reconstruction happens on the basis of whatever representation is input to the pronominal reference module, and that some other factor is responsible for ruling out the missing readings. To this effect, it has been noted that deaccented (as in 11) and elliptical (10) sentences have a very similar range of meanings\textsuperscript{16–19} (deaccented material is written in small capitals); on this kind of proposal, the missing readings are due to the information structural properties of elliptical sentences (i.e. that the ellipsis site must be \textit{Given}, in the sense of Merchant\textsuperscript{20}).

\begin{align*}
\text{(11) John likes his mother, and Bill \textit{likes his mother} too.}
\end{align*}

2.3 Structural Mismatches

There are a number of actually occurring (and in addition very natural sounding) cases of ellipsis where there is a significant surface mismatch between the antecedent and how we interpret the ellipsis site\textsuperscript{17;21}. In the examples below, the natural interpretation of the ellipsis site is indicated in bold.

\begin{align*}
\text{(12) Harry used to be a great speaker, but he can’t } e \text{ anymore, because he has lost his voice.} \\
& (e = \textit{speak}, \textsuperscript{17}, \text{ex. 114})
\end{align*}

\begin{align*}
\text{(13) The candidate was dogged by charges of infidelity and avoiding the draft, or at least trying to } e. \\
& (e = \textit{avoid the draft}, \text{ibid., ex. 120})
\end{align*}

\begin{align*}
\text{(14) This information could have been released by Gorbachev, but he chose not to } e. \\
& (e = \textit{release this information}, \text{ibid., ex. 131})
\end{align*}

In these examples, the hypothesized filled in material is not any contiguous sequence of words (or word forms) from the context, which in this case is simply the remainder of the sentence. This is most obvious in 14, where the missing material is, although present, non-contiguously distributed through the antecedent context. Although psycholinguistic experiments reveal that people consistently judge mismatching VPE examples as degraded compared to non-mismatching cases\textsuperscript{5;6} (influenced by various information structural factors\textsuperscript{4}), the fact that at least some instances of mismatches (as in 12–14; see Hardt\textsuperscript{17} for more examples) are quite natural suggests that we should treat them in the same way as we treat ‘canonical’ cases of ellipsis like 1 (and not as ‘mistakes’).

Cases of structural mismatch have been taken as arguing in favor of a semantic account of ellipsis, as opposed to a syntactic one\textsuperscript{17}, as passive and active sentences, while clearly syntactically different, are commonly taken to be truth-conditionally identical.

While verb-phrase ellipsis (as in 14) is insensitive to voice mismatches between the ellipsis site and its antecedent, other species of ellipsis\textsuperscript{22;23}.

\begin{align*}
\text{(15) Someone stole my wallet, but I don’t know who } e. \\
& (e = \textit{stole my wallet})
\end{align*}

\begin{align*}
\text{(16) My wallet was stolen, but I don’t know by whom } e. \\
& (e = \textit{my wallet was stolen})
\end{align*}
In examples 15 – 17, we see that while both passive (16) and active (15) ‘reconstructions’ of the ellipsis site e are possible, the antecedent and ellipsis site must have the same voice feature (17) in contrast to the VPE example in 14. As pointed out by Merchant and Kobele, an ellipsis-representation could account for these facts if it explicitly represented the difference between actives and passives, but in a manner where it provided a part consisting of the basic verbal argument structure which did not contain this voice information (which would be relevant in verb-phrasal ellipsis). Note that this is only relevant for a theory of ellipsis which aims to provide a uniform account for all varieties of elliptical sentences. This is not always the case; Hardt, for example, explicitly does not do this. It is an open question as to whether there exists a unified theory of ellipsis, and another as to whether it is correct.

Another kind of example which suggests that prototypical syntactic distinctions must be present in an ellipsis-representation (even in VP ellipsis) is given below. (Raising sentences (as in 18) are thought to be semantically equivalent to their non-raised counterparts (as in 19).)

\[(18) \text{Mary seems to have kissed Bob, and Susan does e too.} \quad (e = \text{seem to have kissed Bob})\]

\[(19) \text{*It seems that Mary kissed Bob, and Susan does e too.} \quad (e = \text{seem to have kissed Bob})\]

Taken at face value, these examples suggest that we need a representation which, at least sometimes, makes canonically non-semantic distinctions (such as voice, or subject-raising).

### 2.4 Miscellania

Finally, there are phenomena which suggest ways to look at how the meaning of the ellipsis site is reconstructed, instead of (as above) what it is reconstructed with.

Hobbs and Kehler classify the following sentence (attributed to Carl Pollard) as an instance of ‘sloppy event identity’, and treat it in a manner similar to cases of sloppy pronoun identity. In the reading we are interested in, \(e_1\) is reconstructed as meaning “help you”, and \(e_2\) as meaning “want me to kiss you.”

\[(20) \text{I will help you if you want me to e}_1\text{, but I will kiss you even if you don’t e}_2.\]

The interesting question is how \(e_2\) gets the meaning it does. A natural way of thinking about this is that \(e_2\) is first reconstructed as “want me to \(e_1\)”, and then the copy of \(e_1\) is reconstructed as “kiss you.”

Importantly, the ability to copy ellipsis sites is reminiscent of call-by-name evaluation or outside-in rewriting.

As noted by Webber, and expanded upon by Hardt, there are cases of elliptical sentences in naturalistic discourse where the meaning of the ellipsis site is a non-trivial combination of information from disparate subparts of the context, as in 21.

\[(21) \text{Wendy is eager to sail around the world and} \quad \text{Bruce is eager to climb Kilimanjaro, but neither of them can e because money is too tight.}\]

Hardt gives 22 as a rough approximation of the elliptical subsentence in 21.

\[(22) \text{Neither of them can sail around the world or climb Kilimanjaro.}\]

Crucially, the reconstruction of the ellipsis site involves a boolean combination of multiple antecedent VPs.

### 3 Theory

Both syntactic and semantic approaches to the reconstruction of ellipsis can be given a uniform characterization by viewing semantic and syntactic representations as trees, and parts as (closed) lambda-terms. From a slightly more abstract perspective, both approaches can be seen to be instances of the same general strategy, which is to look at structured
representations which are related to the surface syntactic structure in a regular way. We elaborate on this briefly, beginning with the case of meanings.

In a computational setting, we typically want to deal with meaning representations, not ‘meanings’ per se, whatever the philosopher tells us they may be. (We also want to avoid dealing with ‘logical form equivalence’, as proposed by Merchant, as it is typically not computable.) In a compositional semantics, meanings (represented by sentences in some logic) are obtainable by interpreting a syntactic structure – this can be viewed in terms of transforming the syntactic representation into a semantic one. (The nature of the transformation depends on the grammar formalism. In Tree Adjoining Grammar, one uses a simple macro tree transducer, whereas in Minimalist Grammar, one uses a finite copying top-down tree transducer.) A number of different meaning representations have been investigated in the context of ellipsis; discourse representation structures in the context of Minimalist Grammar one uses a simple macro tree transducer, whereas in Minimalist Grammar, one uses a finite copying top-down tree transducer.) A number of different meaning representations have been investigated in the context of ellipsis; discourse representation structures, quasi-logical forms, and glue semantic proofs to name a few.

In the case of syntax, a distinction between the order (and hierarchy) of application of grammatical operations (tectogrammar) and the objects thus derived (phenogrammar) is present in many modern grammatical formalisms, where it is frequently cashed out in terms of derivation and derived structures. The tectostructure of an expression depends both on the kinds of operations present in the grammar formalism (for example, in Combinatory Categorial Grammar, expressions are constructed by operations like application, composition, and type raising, and in Minimalist Grammars, by the operations of merge and move), and the particular choice of linguistic analysis. Grammar formalisms necessarily assign a tectostructure to grammatical expressions (in the limiting case, as in context-free grammars, tectostructure just is phenostructure). As in the case of meaning, the phenostructure of an expression is obtained from its tectostructure by means of a tree or graph transformation.

3.1 Formal Details

The objects we are concerned with are labelled ordered trees. It will be convenient to use the linear notation for trees familiar from algebra (for more details, see Comon et al.). Each label is associated with a non-negative integer (its arity). Sometimes we write \( f^{(n)} \) to indicate that label \( f \) has arity \( n \). We require that the arity of the label of a node coincide with the number of daughters of that node. A tree with just a single node, labelled \( a \), will be written simply as \( a \). A more complex tree \( T \), with root labelled \( f \) and immediate subtrees \( T_1, \ldots, T_n \) in that order, will be written as \( f(t_1, \ldots, t_n) \), where \( t_i \) is the way we write \( t_1, t_2 \) the way we write \( T_2 \), and so on.

We will view trees as special kinds of (typed) lambda terms, following de Groote. The lambda calculus provides a general way of splitting trees into pieces. The set of types we will consider is the smallest set containing \( T \) (the type of trees) and \( AB \) for every pairs of types \( A \) and \( B \). A tree is a term \( t \) of type \( T \) which does not contain any variables, and an \( n \)-ary context is a term of type \( T(T(\ldots(TT)\ldots)) \) (which we write as \( T^{n+1} \)) which uses only variables of type \( T \). When computing with lambda terms (see de Groote, and other work in the ACG tradition), there are two natural sources of complexity. The first is the order of the lambda term: an eta-expanded lambda term (i.e. one where all arguments are abstracted over) has order one higher than the maximal order of any of its bound variables, or order one, if it has no bound variables. (Thus a lambda term of order two can be thought of as taking trees as arguments to build another tree.) The second is whether the term duplicates its arguments (‘linear’ vs ‘non-linear’), and if so, whether these duplicated arguments are all first order (‘almost linear’) or higher order. We are therefore particularly interested in linear lambda terms. Non-linear terms implement a form of structure sharing. Given a tree \( t \), it is immediately decomposable into linear terms \( t_1, t_2, \ldots, t_n \) just in case \( t'(t_1) \cdots (t_n) \) beta-reduces to \( t \). For example, the term in 23 represents the right branching binary branching tree of height 3, with nodes labeled from \( a \) to \( e \) in a breadth first order.
The expressions in 24 and 25 represent the same tree as in 23 (they are equivalent modulo beta reduction). While 24 represents the tree as being composed of parts which consist of the top ‘half’ of the tree (\(\lambda x_T. a(b, x))\), and the subtree \(c(d, e)\), 25 breaks the tree into much more abstract parts – the part \(\lambda f_{TT}. a(b, f(d))\) consists of the top half of the tree together with the leaf \(d\) inside the rest of the tree, and \(\lambda x_T. c(x, e)\) is the bottom part of the tree, minus the \(d\). As the examples illustrate, as the order of a term increases, it corresponds to a more and more abstract (i.e. discontinuous) part of a tree. Having recourse to very abstract parts allows us to ‘find’ objects corresponding to elided material in even very difficult cases. On the other hand, as we need to specify not only which parts can be used in ellipsis, but also which ones cannot, making use of very abstract parts puts us in danger of wild overgeneration. In other words, the more parts we have available to us, the more difficult it becomes to delimit the class of parts which actually allow for ellipsis.

For instance, consider the example of voice mismatch in ellipsis in 14, which has been used to argue against a (surface) syntactic account of ellipsis. A third order term, \(\lambda f_{TT}. f(\lambda x_T. \text{VP}({\text{release}}, x))(\text{NP}({\text{this}}, \text{info}))\), expresses the relevant shared part in a natural surface tree for example 14! (This is the part that something has when it contains both a VP headed by ‘release’ with something in the object position (a trace, or an NP), as well as (somewhere) an NP ‘this info’.) However, just this part is shared between the antecedent clause and the sentence “this information released a monkey”, which does not provide an appropriate antecedent for the elliptical part of 14. Thus, the debate about the proper representation over which to reconstruct ellipsis can be understood as an attempt to find a representation where we have a natural and restrictive characterization of the kinds of parts that are available to us.

4 Algorithm

From a cognitive perspective, parsing is the model of our ability to make sense of heard (or read) utterances. As the linguist’s grammar provides a compact description of possible interpretations of well-formed sentences, parsing is the task of assigning a grammatical description to a sentence. Parsing algorithms transform grammars into procedures for determining which structures (if any) they assign to strings of words. In the case of ellipsis, however, it is not always enough to recover a syntactic representation for a sentence – we want in addition to resolve the meanings of the ellipses. Accordingly, we adopt the broader cognitive perspective, and view parsing as the assignment of meaning representations to strings.

Algorithms for parsing elliptical sentences have a common ‘two-step’ structure. In the first step, a syntactic structure is assigned to a sentence. This is possible in many grammar formalisms by simply treating ellipsis sites (what we have been notating with an \(e\) in our examples) as actual lexical items with no phonetic content. (Or (what amounts to the same thing) by assigning heads which can appear before an ellipsis site a second ‘elliptical’ lexical entry, as do Hardt and Jäger.) (In some grammar formalisms the partial parse tree containing the ellipsis site might have many \(e\) nodes, where intuitively there is only a single ellipsis site.) A parser is thus able to ‘guess’ whether it has just seen an ellipsis site. (Hardt and Jäger demonstrate that the determination of the correct antecedent can be made with reasonable accuracy.) The result of a parse is a sentence with (perhaps) multiple \(e\) elements. In order to obtain from such an intermediate structure a meaning representation, we need to ‘flesh out’ the \(e\)s. It is not necessarily the case that every such intermediate structure can be fleshed out as a meaning representation; in sentence 26, there is no appropriate fleshing out of \(e_1\) but as \(e_2\), and none of \(e_2\) but as \(e_1\) (assuming an empty context).

(26) John did \(e_1\), but Bill didn’t \(e_2\).

As a consequence, the recognition problem – construed as the question “does sentence \(s\) get assigned
a meaning by the grammar? – becomes in general more difficult (we can no longer reduce this problem to testing for the emptiness of the intersection of the grammar with a string \(^{[46,47]}\)).

Dalrymple et al. \(^{[48]}\) (see also Shieber et al. \(^{[49]}\)) frame the problem of fleshing out \(e\) elements in terms of solving equations with \(e\) elements as the unknowns (using higher order unification\(^{[50]}\)). For example, in a sentence like 27, they ask whether there is a value for \(e\) such that \(\text{like}(\text{golf, dan})\) is equal to \(e(\text{dan})\). This value of \(e\) (namely, \(\lambda y.\text{like}(\text{golf, }y)\)) is then taken to be the meaning of the ellipsis site.

(27) Dan likes golf, and George does \(e\) too.

Gardent et al. \(^{[51]}\) extend the above method to rule out certain undesired ‘solutions’ (such as \(\lambda y.\text{like}(\text{golf, dan})\)) and to determine which elements in the antecedent are parallel to which elements in the sentence containing the ellipsis site in a principled way. Prié et al. \(^{[10]}\) use a method similar to that of Dalrymple et al. \(^{[48]}\) to determine discourse structure, making the interpretation of ellipsis a by-product of this procedure. Egg et al. \(^{[52]}\) propose to incorporate constraints on acceptable solutions (lambda expressions), and propose a method for finding solutions satisfying such constraints. While these articles use meaning representations as the objects reconstructed in elliptical contexts, they are by no means restricted to this perspective, and can be equally well applied to syntactic representations. Indeed, Thompson \(^{[53]}\) proposes a hybrid theory according to which first the verbal spine of an ellipsis site is syntactically reconstructed, and then a higher order unification procedure is applied over the semantic representation of the antecedent and the syntactically incomplete elliptical sentence to reconstruct the meaning of the ellipsis site.

Although parsing elliptical sentences has a logical two step structure, it is natural to interleave the two steps, instead of performing them serially\(^{[54,55]}\). A natural way of doing this is to maintain a data structure of possible parts, which can then be accessed during parsing. It has been observed that ellipsis resolution is similar to pronominal reference resolution\(^{[17]}\). A natural idea is to use related data structures for ellipsis parts, and for pronominal references – Hardt\(^{[56]}\) implements this idea in a dynamic semantics. A more general, continuation based framework for pronoun resolution\(^{[57]}\) makes clear that even a syntactic approach to ellipsis is compatible with the insights of Hardt\(^{[17]}\) regarding the proform-like behaviour of ellipsis.

Another means of maintaining a data structure of possible parts takes advantage of parsing techniques. Different syntactic parsing strategies can be viewed as different ways to traverse a tree\(^{[58]}\). A tree traversal visits nodes in a particular order; at any point, the set of nodes visited (and its complement) form parts of a tree. Thus the choice of parsing strategy affects which parts are available. Most intuitively, a bottom-up parser breaks a tree up into its constituent subtrees and their linear tree contexts. In grammar formalisms such as minimalist grammars or tree adjoining grammars, the tree traversed is the derivation tree, not the surface tree. This has the effect that derivational constituents needn’t correspond to a contiguous series of words in the surface string (which makes possible an account of sentences like 14). Lavelli and Stock\(^{[59]}\) suggest that the parser, upon encountering an ellipsis site, should simply reuse a previously derived constituent (which is maintained in a data structure called the chart). This idea has been reproposed in the context of tree-adjoining grammars\(^{[60]}\), and in minimalist grammars\(^{[5]}\).

5 Conclusion

While other approaches to ellipsis have been explored\(^{[61]}\), it is fair to say that the majority are based on identifying parts in a structure related transformationally to the surface syntactic structure of an expression. The major issue in identifying such a representation is whether it provides a restricted enough set of parts to correctly identify just the range of meanings an elliptical sentence may have in a discourse. This can take the form of a formal restriction on the shape of allowable lambda-terms, or as a statement about the kind of parsing strategy employed. While ellipsis resolution procedures are typically presented in the context of a particular representation, they are often logically separable from this latter.


References


