Parsed Annotation with Semantic Calculation

Alastair Butler[1]*, Stephen Wright Horn*[2]

(1) Faculty of Humanities and Social Sciences, Hirosaki University
(2) Theory and Typology Division, National Institute for Japanese Language and Linguistics

ajb129@hirosaki-u.ac.jp, horn.s.w@ninjal.ac.jp

ABSTRACT

This paper describes a corpus building program implemented for Japanese (Contemporary and Old) and for English. First, constituent tree syntactic annotations defined to describe intuitions about sentence meaning are added to the texts. The annotations undergo tree transformations that normalise the analyses while preserving basic syntactic relations. The normalisation takes the parsed data for what are very different languages to a level where language particulars have a common interface to feed a semantic calculation. This calculation makes explicit connective, predicate, argument, and operator-binding information. Such derived information reflects universal principles of language: headedness, argumenthood, modification, co-reference, scope, etc. The semantic calculation also sets some minimal conditions for well-formedness: that predicative expressions are paired with subjects; that pro-forms have retrievable referents; that a constituent is associated with at least one grammatical function, etc. Annotators confirm and correct the source annotation with the aid of a visualisation tool that integrates the calculated output as overlaid dependency links. In this way annotators ensure that their interpretation of a text is correctly represented in its annotation. Furthermore, the integration of results from the semantic calculation makes it possible to establish multiple layers of grammatical dependencies with a minimum of invested annotation work.

KEYWORDS: Parsed corpus, Sentence and discourse meaning, Normalisation, Grammatical dependencies, Discourse referents, Visualisation.

1 Introduction

This paper presents the techniques behind a parsed annotation approach that has led to the creation of corpus resources with syntactic and semantic information for languages, including:

- Contemporary English (TSPC; 7,026 trees; 87,182 words; `http://www.compling.jp/ajb129/tspc.html`),
- Contemporary Japanese (NPCMJ; 20,425 trees; 315,200 words; `http://npcmj.ninjal.ac.jp/interfaces`), and
- Old Japanese (MYS97; 159 trees; 2,549 words; `http://www.compling.jp/mys97`).

These corpus resources offer instantiations of the goal to provide fully searchable representations of texts that are described according to interpretations of the meanings of sentences. This involves analysing texts into units and accounting for how the units are composed into meaningful expressions. Units are related by either grammatical functions or semantic relations. These two types of relations are together referred to as 'dependencies'. The annotation uses unit types, structural relations, and only as a last resort, indices, to describe how dependencies are established. A full description of a sentence tells the story of how information flows through structure to compose complex meanings. In order to achieve this, the annotation uses familiar grammatical categories to name units in an economical way, following organising principles.

The principles depend on two fundamental points: (1) an adequate description of the grammar, and (2) the ability to calculate a semantic representation. In essence, we take a text, divide it up into utterances (sentences), and decide on interpretations. Next we organise the parts of the sentence to show how the parts combine in principled ways to compose the meaning we arrive at in our interpretation. This takes the form of a tree with labeled nodes (a structure). The structural assignments in the annotation are associated with a finite set of defined relationships for interpreting meaning (dependencies), so that through transformations between representations that capture the structure, we can reach alternative instantiations of the structure that can be more explicit articulations of meaning components and their comprising dependencies.

Notably, the parse annotated tree structure will be transformed into expressions of a logical language, comprised of: discourse referents, operators, predicate relations, and logical connectives. A logical expression is reached by undertaking a recursive calculation that receives normalised annotation as input. We will refer to this as the 'semantic calculation'. If the logical expression derived from a given sentence annotation corresponds to the original interpretation, this is one kind of confirmation that the structure assigned to that sentence is correct, and serves as feedback to the annotation process.

This paper is organised as follows. In section 2 the basics of the annotation schema are described: word segmentation and part-of-speech tagging, labelled bracketing of segments and phrases, and the specification of grammatical function and clause linkage type. Then in section 3 we describe the first step from a parsed annotation to reach a semantic calculation. This step is normalisation, an information-preserving structural transformation in which language-specific syntactic constructions are rewritten into configurations built out of a reduced set of structures and categories. In section 4 we briefly sketch the remaining steps taken to reach a formal expression with the properties of a Discourse Representation Structure (Kamp and Reyle 1993). In section 5 we show how the information distilled in the logical expression analysis is integrated into the parsed annotation with indexing derived from the calculated discourse referents, becoming the basis for a graphic user interface for visualising sentence and discourse level dependencies. Section 6 relates the described method to alternative approaches for corpus development. Section 7 concludes the paper.

2 Basic annotation

This section presents an approach to syntactic analysis adopting parsing schemes influenced by the Penn Historical Corpora scheme (Santorini 2010) and the SUSANNE scheme (Sampson 1995). The Penn Historical Corpora scheme has informed the 'look' of the annotation, with tag labels familiar from mainstream linguistics, and the CorpusSearch format (Randall 2009). From the SUSANNE scheme, there is adoption of construction analysis, functional and grammatical information, and methods for representing complex expressions. The annotation also contains plenty that is innovative, including annotation to explicitly mark exceptional scope information (not described here for reasons of space) and information to resolve both inter and intra sentential anaphoric dependencies from a discourse perspective. Annotation is carried out with a general text editor (in practice, vi or emacs) on text files.

We can describe the overall annotation in terms of different layers of information:

1. Word segmentation and classification
2. Labelled bracketing of segments
3. Indicating grammatical function
4. Clause linkage types
5. Null elements
6. Scope marking
7. Tracking anaphora

This ordering of layers represents a loose hierarchy of informational detail, such that a 'higher' layer cannot be added unless a 'lower' (i.e. more basic) layer is already available: e.g., 2 is a pre-requisite for 3. The goal is a representation that is rich enough to allow the automatic calculation of argumenthood, modification, scope, and co-reference in discourse having little or no recourse to overt indexing. In practice, decomposing words into bindings and recasting grammatical dependencies as the management of discourse referents for those bindings can be accomplished with a surprisingly modest inventory of grammatical categories and structural configurations annotated on the source data, provided that all relations in a set of parsed trees are defined under a sufficiently developed system of calculation. The remainder of this section introduces the basics for establishing such annotation.

2.1 Word segmentation and classification

A continuous segment is a word and each word is assigned a part-of-speech (POS) tag—in practice, the label of an open bracket '(LABEL'. A POS tag can be complex, having at least a core label, but potentially including extensions separated by hyphens, and in certain cases, semantic information in curly brackets set off with a semicolon. In (1), being especially relevant for parsing Japanese, particles (P) receive extensions to mark distinct contributions:

(1)
```
P-COMP  - complementiser
P-CONN  - connective with coordination or subordination
P-FINAL - sentence final, e.g., marker of clause type
P-OPTR  - operator
P-ROLE  - marker of grammatical (syntactic or semantic) role
```

In (2), a determiner is coded for definiteness by adding semantic information:

```
(2)  (D;{DEF} sono)    "that"
```

The POS tags for a language may be sufficiently fine-grained to mark syntactic features. For example, the distinction between a common noun and a 'formal noun' used in root contexts to indicate aspect in Japanese is specified as in (3):

```
(3)  (N tokoro)        "place"
     (FN tokoro)       "occasion, instant"
```

Note that for annotating Japanese, grammatical number is not indicated by a specialised POS tag, in contrast to English seen in (4):

```
(4)  (N spoon)         singular
     (NS spoons)       plural
```

2.1.1 Multi-word expressions

A sequence of words which behaves syntactically in an idiomatic way (e.g., *ni muke te* 'with a view towards', *ga hayai ka* 'no sooner than', *sore ni shi te mo* 'irregardless') is 'chunked' into a single segment and tagged by using the word tag appropriate for the sequence as a whole, as illustrated in (5).

```
(5)  (P-ROLE nimukete)
     (P-CONN gahayaika)
     (CONJ sorenishitemo)
```

That is, such word sequences are treated as single units for the purposes of parsing.

2.2 Labelled phrase bracketing

Bracketing is a way of grouping together segments which are recognised as having a syntactic integrity as units of various kinds (sentences, clauses, phrases, words). Typically, when a $word_1$ combines with a $phrase_1$ to form a more complex $phrase_2$, $word_1$ is said to be the head of $phrase_2$. Let's consider providing syntactic analysis for (6).

(6) daijoobu ？sensei ！to odoroi ta koe de bokura wa kii ta 。
OK ? teacher ! COMP surprise PAST voice INSTR we TOP ask PAST .
' "Teacher! Are (you) OK?" we asked (him) in surprised voices.'

In example (6), represented with bracketing in (7) below, pronoun *bokura* 'we' is a segment that, being unmodified, forms a phrase by itself. As a phrase, it can combine with a head, in this case topic particle *wa*. Particle *wa* projects a topic particle phrase *bokura wa*, which relates as a whole to the verb *kii* 'ask'. In contrast, *sensei* 'teacher', is a vocative phrase, relating to an immediately preceeding sentence, *daijoobu* ？ 'Are you OK?' by virtue of co-reference. These two units combine to form an utterance marked by a complementiser *to*. The resulting unit appears at the same structural level as the particle phrase *bokura wa*, where it too relates to the verb *kii*. The outermost brackets indicate that the whole complex utterance is a unit. Thus the verb *kii* 'ask' is part only of the whole utterance in example (7), and as such is the head of the largest sentence unit.

```
(7)  ( ( ( ( daijoobu) ？
           ( sensei) ！) to)
       ( ( ( odoroi ta)
           koe)
         de)
       ( bokura wa)
       kii ta
       。)
```

This layer of bracketing indicates the composition of non-terminal syntactic units (or constituents) such as Noun Phrase, labelled NP, Preposition/Postposition Phrase, labelled PP, Clause, labelled IP, and Complementiser Phrase, labelled CP, etc. For example, adding phrase-level labels to (7) above yields the labelled analysis of (8):

```
(8)  (IP (CP (CP (IP daijoobu) ？
                 (NP sensei) ！) to)
           (PP (NP (IP odoroi ta)
                   koe)
               de)
           (PP (NP bokura) wa)
           kii ta
           。)
```

For phrase-level annotation in Japanese, the following basic categories are assumed:

- Clause unit (which can be the utterance-level) (IP, CP, FRAG)
- Noun phrase (NP)
- Postposition phrase (PP)
- Adverb phrase (ADVP)

Annotation for English adds one further phrase type:

- Adjective phrase (ADJP)

Including the POS labels for terminal nodes completes the basic structure of a parsed tree:

```
(9)  (IP (CP (CP (IP (ADJN daijoobu)) (PU ？)
                 (NP (N sensei)) (PU ！))
             (P-COMP to))
           (PP (NP (IP (VB odoroi)
                       (AX ta))
                   (N koe))
               (P-ROLE de))
           (PP (NP (PRO bokura)) (P-OPTR wa))
           (VB kii)
           (AXD ta)
           (PU 。))
```

2.3 Indicating grammatical function

The grammatical function which relates a unit to the element with which it combines is identified by either:

1. the combination of the head plus the structural position of that unit, (e.g., a P-ROLE heading a PP may specify the function of that PP with respect to a predicate),
2. the node description of that unit having hyphenated extensions with information about grammatical function, such as -SBJ for subjects, -LOC for locational adjuncts, etc., or
3. having 1 supplemented by 2.

Example (10) illustrates scenario 1 with a PP headed by instrumental case marker `(P-ROLE de)` at the matrix clause level (directly below IP-MAT). The information supplied by the terminal node `de` together with the part of speech tag P-ROLE is considered sufficient for determining the grammatical function of the constituent.

Example (10) illustrates scenario 2 with its PP-SBJ at the clause level. As an operator particle, the 'toritate' particle *wa* can mark topichood, contrast, or focus of negation, for example, but not grammatical role, so the function of the PP with respect to the head of the whole sentence is indicated by the combination of (i) the sibling relationship between the PP and the head of the whole sentence, and (ii) the extension -SBJ.

(10)
```
(IP-MAT (CP-THT (CP-QUE (IP-SUB (ADJN daijoobu))
                        (PU ？)
                        (NP-VOC (N sensei))
                        (PU ！))
                (P-COMP to))
        (PP (NP (IP-EMB (VB odoroi)
                        (AX ta))
                (N koe))
            (P-ROLE de))
        (PP-SBJ (NP (PRO bokura))
                (P-OPTR wa))
        (VB kii)
        (AXD ta)
        (PU 。))
```

At the clause level, all units locally relating to (i.e., sibling to) the head must have grammatical function information. Grammatical function information can be either syntactic (e.g., -SBJ for subject) or semantic (e.g., -LOC for location). In contrast, as a complement within a PP, an NP need not have an extension. For utterances that consist of a question, the tag CP-QUE is used. In (10) this unit combines with the `(P-COMP to)` to form a complementiser phrase CP-THT. This in turn combines with `(VB kii)`. Note that the heads of sentences (i.e., predicates) can be analytic. Here the past tense morpheme `(AXD ta)` forms one part of a complex verbal syntagm.

At this point we can begin tracking how the syntactic information is ultimately interpreted by comparing the interpretation encoded in the annotation (here, specifically, the intuition that `bokura wa` 'we' functions as the subject of the predicate `kii_ta` 'ask') with the formula output of the semantic calculation (11):

(11)
```
1  ∃ TEACHER[8] STUDENTS[5] EVENT[7] EVENT[9] de[6] THT[1].(STUDENTS[5] = bokura
2    ∧ to(THT[1],QUEST ∃ TEACHER[3] EVENT[4] TEACHER[2].(sensei(TEACHER[2])
3            ∧ TEACHER[3] = TEACHER[2]
4            ∧ daijoobu(EVENT[4],TEACHER[3])))
5    ∧ koe(de[6],odoroi_ta(EVENT[7],STUDENTS[5]))
6    ∧ TEACHER[8] = *pro*
7    ∧ past(EVENT[9])
8    ∧ kii_ta(EVENT[9],STUDENTS[5],_,TEACHER[8],THT[1]) ∧ de(EVENT[9]) = de[6])
```

In (11) we see a sorted discourse referent `STUDENTS[5]` (that is, referent `[5]` of sort `STUDENTS`) under existential quantification (discourse closure) and equated with pronoun `bokura` in line

1. In addition STUDENTS[5] is one of the arguments in a binding with predicate kii_ta in line 8. In addition, STUDENTS[5] appears as the argument of odoroi_ta in line 5.

Note that the basic annotation in (10) is not yet a complete mark-up of (6) with regard to the data model for the approach. More layers of information have to be added before a formula such as (11) can be generated. A complete annotation for (6) can be seen in (12).

(12)
```
(IP-MAT (CP-THT (CP-QUE (IP-SUB (NP-SBJ;{TEACHER} *pro*)
                                (ADJN daijoobu))
                        (PU ？)
                        (NP-VOC;{TEACHER} (N sensei))
                        (PU ！))
                (P-COMP to))
        (PP (NP (IP-EMB (VB odoroi)
                        (AX ta))
                (N koe))
            (P-ROLE de))
        (NP-OB2;{TEACHER} *pro*)
        (PP-SBJ;{STUDENTS} (NP (PRO bokura))
                           (P-OPTR wa))
        (VB kii)
        (AXD ta)
        (PU 。 ))
```

The need to feed a semantic calculation places requirements on the annotation. Trees are connected graphs with the first labelled branching node taking its label from a set of utterance types (Exclamation CP-EXL, Imperative IP-IMP, Fragment FRAG, etc.). In general an IP is defined as a nexus of a predicate (e.g., a verb, adjective, or copular expression) and at minimum a subject. The meaning of the predicate conditions the realization of arguments and modifiers. Elements that are not expressed overtly in the text may be assigned nodes with null elements, or they may be left empty to inherit the contribution of a non-local argument position. Elements combining with predicates must be specified for grammatical or semantic role. Pronominal elements (such as *bokura* in (10)) must have their reference resolved with 'sort' information. For example, an annotator will add '`;{STUDENTS}`' to the overt pronoun in (12). Floating quantifiers must be associated with target constituents. Relative clauses must contain a trace.

In general, constructions are assigned structures using a limited inventory of annotation conventions (labels, and indices). The structures are defined so that language-specific details can be differentiated, while at the same time basic dependencies can be extracted. In practice, an elaborated system of defaults is needed in order to cover the data in natural language to some degree. The data model employed here sets enough requirements to allow the semantic calculation to arrive at unambiguous results for argumenthood, modification, and anaphoric relations. Achieving this density of dependencies allows the generation of an output that can be examined and corrected for accuracy.

3 Normalisation

So far we have seen how applying the annotation produces a language-specific syntactic analysis of the source text. We now turn to the question of how to feed the information encoded in the annotated tree into the semantic calculation mechanism (sketched in section 4) to output a form in which grammatical relations are expressed as logical relations. This first step consists of rewriting language-specific structures so that information is preserved about the dependencies and the lexical material while taking a generalised form that can apply to any language. This

is called 'normalising' the annotation. We illustrate the process with (6), using its completely annotated form (12) from the previous section as the starting input.

Normalisation reduces the number of node labels, replaces items denoting basic grammatical roles with offset elements headed by labels specifying those roles, migrates some label information into terminal nodes as predicates denoting grammatical categories, and concatenates any terminal nodes which have had the information from their part of speech information discharged in generated predicates. The normalisation of (12) is shown in (13).

(13)
```
( (IP-MAT (ACT past)
          (CP-THT (C to)
                  (CP-QUE (PU ？)
                          (PP (P-ROLE VOC)
                              (NP (SORT *TEACHER*)
                                  (N sensei)))
                          (PU ！)
                          (IP-SUB (PP (P-ROLE ARG0)
                                      (NP (SORT *TEACHER*)
                                          (PRO *pro*)))
                                  (VB daijoobu))))
          (PP (P-ROLE ARG0)
              (P-OPTR wa)
              (NP (SORT *STUDENTS*)
                  (PRO bokura)))
          (PP (SORT de)
              (P-ROLE de)
              (CP-ADV (C koe)
                      (IP-SMC (VB odoroi_ta))))
          (PP (P-ROLE ARG2)
              (NP (SORT *TEACHER*)
                  (PRO *pro*)))
          (VB kii_ta)
          (PU 。))
  (ID 6_misc_discourse_3;JP))
```

By comparing (13) with (12), it can be seen that the NP-SBJ tag in the source annotation is changed into an NP tag which is placed under a PP projection headed by a `(P-ROLE ARG0)` element. All arguments are converted to PP projections with role information appearing as a terminal string (effectively to function as the information for how the argument's discourse referent will be linked to the predicate the argument serves to bind in a resulting logical expression). For underdetermined elements such as `(PP (NP (IP-EMB (VB odoroi) (AX ta)) (N koe) (P-ROLE de))`, the information carried by the particle is copied into the terminal node position of a `SORT` element. Part-of-speech tags are regularised in other ways as well. For example, `daijoobu` is a 'na-adjective' (a category peculiar to Japanese), but its ADJN tag is regularised to VB, which is the category for all clause level predicates after normalisation. Some information about basic grammatical categories is migrated from node labels into offsets. For example, the part of speech label for the past tense morpheme AXD triggers the creation of an ACT tag with a predicate `past` to retain the tense information. (Note that the past tense morpheme, stripped of its part of speech tag, is concatenated to the free morpheme immediately preceding it.) For pronouns, the 'sort' information that resolves their reference is made into an off-set element as well.

Normalisation renders all the information relevant to the domain of discourse in forms that are defined under an intermediate formal language. The next step is to pass the normalised tree to a script that decomposes the lexical items into bindings, and recasts structural relations

as instructions about how to manipulate them. The intermediate language, then, is basically a set of interrelated rules that use the normalised information to generate predicates, binding names, and the sequences of discourse referents that binding names are assigned, plus various operations over and relations between these elements.

4 The semantic calculation

At the start of the calculation, there is a collection of 'discourse referents' which are read from the normalised input based on the occurrences of SORT nodes. Discourse referents are ultimately destined to be bound by operations of closure (e.g., ∃) in the overall resulting logical expression. The operations of closure are themselves either reflexes of quantification instructions from the normalised input, or arise because there is discourse closure. During the runtime of the calculation, the collected discourse referents are released as content for argument slots of predicates that populate the resulting logical expression. The exact makeup of argument slots (valencies) for a given predicate is typically left unspecified by the input. At the point in the calculation when the predicate is reached, the availability of accessible discourse referents is established. The predicate's sensitivity to what is accessible determines the arguments for the predicate. More information about the semantic calculation is available at: `http://www.compling.jp/ajb129/ts.html`. As a result of calculation, (13) is turned into the formula analysis seen as (11) of section 2.3.

5 Treebank Semantics Visualisation Tool

With a formula rendering, such as (11) of section 2.3, relationships are expressed by sorted discourse referents appearing in multiple contexts, or relating to other discourse referents (e.g., as being identical to other discourse referents). Such relationships can be re-expressed through indices shared between nodes in a tree structure. Such indices, being derived from the source annotation in the first place, are in principle a redundant notational variant rather than an addition of information. Nevertheless, they are convenient as loci for adding information about dependencies between constituents in tree structures. The dependencies derived from the annotation of (6), rewritten as indices, can be embedded back into the source phrase structure tree annotation to yield (14):

(14)

```
(IP-MAT (CP-THT;<THT[1]> (CP-QUE (IP-SUB (NP-SBJ;<TEACHER[3]>
                                             (PRO pro;{,TEACHER[2],}))
                                         (ADJN;<,TEACHER[3]@ARG0,EVENT[4]@EVENT,>
                                             daijoobu))
                                 (PU ？)
                                 (NP-VOC;<TEACHER[2]>
                                         (N;<,TEACHER[2]@h,> sensei))
                                 (PU ！))
                           (P-COMP;<,THT[1]@FACT,> to))
        (NP-OB2;<TEACHER[5]> (PRO pro;{,}))
        (PP-SBJ;<ENTITY[6]> (NP (PRO bokura;{,})) (P-OPTR wa))
        (VB;<,ENTITY[6]@ARG0,TEACHER[5]@ARG2,THT[1]@THT,EVENT[7]@EVENT,> kii)
        (AXD ta)
        (PU 。))
```

This 'indexed' view gives a view of the tree structure with indexing information that specifies

argument relationships and antecedence relationships. Argument dependencies are marked on the label of the target predicate as sets of index/grammatical function pairs. Antecedence relationships (including big PRO, as the terminal `*PRO*;{GROUP[1]}`) are spelled out using discourse referents typed according to 'sort' information.

This 'indexed' view explicitly encodes grammatical dependencies that the original annotation had left implicit. The indexing makes the following kinds of contributions:

- Indexing given the form '`<discourse_ref>`' marks a node that serves as an argument for a predicate.
- The arguments that a predicate takes are marked on the pre-terminal node for the predicate with a '`<,...,discourse_ref@role,..,>`' format, with '`discourse_ref`' providing information to locate the argument and '`role`' stating the argument role.
- Control and trace information is presented with the format '`{discourse_ref}`', that is, specifying a single obligatory antecedent.
- Pronominal information is presented with the format '`{,discourse_ref,...,}`', that is, specifying potentially multiple antecedents.

In this 'indexed' view the dependencies that obtain between constituents within a given tree are fairly easy to read, but for checking discourses over multiple trees, display in a graphic user interface is preferable. To this end, a web browser based tool was developed with a tree-drawing program and numbered nodes called the Treebank Semantics Visualisation Tool (TSVT). The components that make up the tool are available at: `http://www.compling.jp/ajb129/view.html`. The derived dependencies are displayed in various ways depending on their type. Figure 1 below presents the GUI image for the indexed view of (6) in the original Japanese script.

Local dependencies are indicated by integer/role pairs underneath the target predicate. For example, in figure 1 below, the predicate node `(VB 聞い)` —romanised *kii* 'ask'— has below it a set of pairs specifying constituents and their roles respective to *kii* 'ask':

- `243-arg2` (node no. 243, indirect object)
- `230-で` (node no. 230, instrumental particle 'de')
- `239-arg0` (node no. 239, subject)
- `217-tht` (node no. 217, clausal complement)

Note that ATB extraction relationships are drawn with the same notational convention as local relationships.

Control relationships are indicated by dotted lines connecting the node number of an antecedent constituent with a shared secondary node number for a derived subject position. For example, in figure 1 below the subject argument node `(PP-SBJ (NP (PRO 僕ら)) (P-OPTR は))` —romanised *bokura wa* 'we'— is in a control relationship with a derived subject position in a noun-modifying clause IP-EMB. The derived subject position is node no. 233, but carries a secondary node number matching that of its antecedent: no. 239, and connecting with it via a dotted line.

Anaphoric relationships are drawn with the same notational convention as control relationships. For example, in figure 1 below, the null subject in IP-SUB is co-referential with the post-posed vocative phrase `(NP-VOC (N 先生))` —romanised *sensei* 'teacher'. The node number of the antecedent (no. 225) is connected to a matching secondary number marked on the pronominal

constituent. In this respect the role of context in the interpretation of meaning becomes apparent. For example, (PP-SBJ (NP (PRO 僕ら)) (P-OPTR は)) (no. 239) and (NP-OB2 (PRO pro)) (no. 243) carry secondary numbers from a discourse referent fragment introduced at the beginning of a longer discourse (no.s 7 and 3, respectively). Additionally, an exophoric antecedent preceding (6) supplies the reference for a pronoun in a tree following (6), showing up as a dotted line passing uninterrupted through the tree in figure 1.

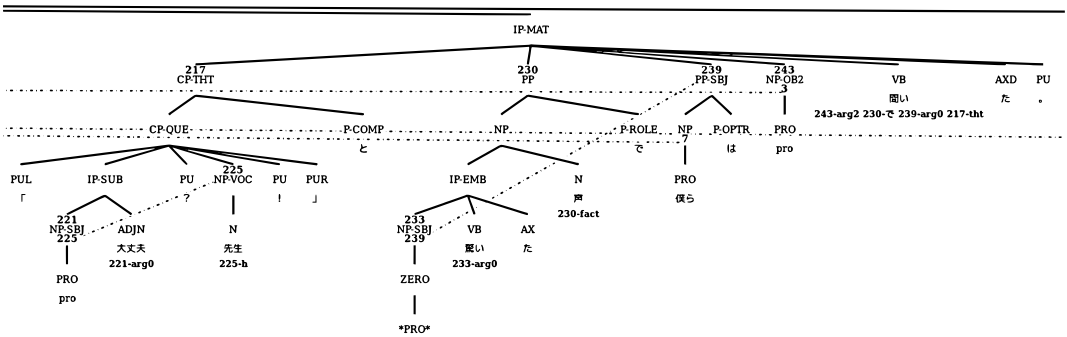

Figure 1: TSVT visualisation of (6)

The TSVT is able to provide an output at the point where the density of dependencies encoded in the source parse supply the minimum necessary information to account for the requirements/contributions of lexical items appearing in the source parse. Once this critical mass of information is reached, the tool can now assist in the establishment of additional dependencies that, while not stipulated by the grammar, are nonetheless crucial to meaning interpretation. Annotators can modify bracketed trees and see the results of the semantic calculation instantly using the TSVT. Corrections that have impact on the semantic calculation typically include changing attachment sites for constituents, changing the specification for clause linkage, the addition or subtraction of null pronouns, and the addition of 'sort' information to noun phrases. Overt indices are rarely used in the annotation, the structural assignments given are usually common patterns, and exceptions are typically listed in the annotation documentation. In short, the predictable behaviour of the semantic calculation allows annotators to use a finite set of strategies to represent all the basic dependencies involved in the interpretation of sentence meaning in discourse context.

6 Comparison with other approaches

The techniques described in this paper are complementary to the classic treebank methods of building parse trees as developed at Lancaster (e.g., Garside, Leech and Sampson 1987, Sampson 1995) and Penn (e.g., Marcus, Santorini and Marcinkiewicz 1993, Kroch, Santorini and Diertani 2010). To this, the approach adds a semantic calculation dimension. In this regard, Butler and Yoshimoto (2012) and Kiselyov (2018) are closely related approaches.

Alternative wide-coverage systems with semantic components have been developed by other means. One prominent branch has involved boot-strapping from tree based corpora to produce resources within categorial grammar frameworks, notably, CCGbank (Hockenmaier and Steedman 2005) and TLGbank (Moot 2015). The CCGbank approach has been extensively developed with the Groningen Meaning Bank (Basile et al. 2012) and its multi-language variant,

the Parallel Meaning Bank (Abzianidze et al. 2017), where there is improvement of source CCG annotation to allow for an interleaving of grammar development and corpus annotation in a beneficial cycle. In this regard the major difference with the current approach is the base input ('Bits of Wisdom' versus constituent trees), as both approaches create Discourse Representation Theory analysis (Kamp and Reyle 1993).

With corpus annotation relying on a beneficial cycle of analysis results, there is a clear link with approaches to corpus production that have emerged out of full grammar development programs, e.g., DeepBank (Flickinger et al. 2012) based on HPSG, and NorGramBank (Dyvik et al. 2016) based on LFG. Such programs have led to the creation of high coverage corpora, and have included semantic levels of analysis, and have been applied cross-linguistically. In comparison, the current approach is significantly 'lighter' in the sense of not requiring the development of a full grammatical system for the annotation to make progress.

7 Conclusion

Linking structure and semantics sets constraints on how the whole corpus is designed. Starting from the semantic requirements, at a very basic level, discourse referents need to be introduced as parts of resulting logical expressions, and there must also be the means for introduced discourse referents to find their way to the appropriate argument slots of predicates, where predicates are used to instantiate the contributions of nouns, verbs, adjectives, adverbs, etc. Such introductions and their subsequent management have to be linked to the structures in the corpus. Needing to establish such links can actually simplify the structure of the corpus. This is a reflection of the fact that languages in general have fixed ways of keeping track of language components. In grammar, we see these facts as the reach (or lack of reach) of an argument dependency through structure, the marking of definiteness and specificity to project scope, accessibility conditions on anaphoric reference, etc.

The semantic calculation, together with the application that allows its results to be displayed (the TSVT), make it possible to establish multiple layers of grammatical dependencies with a minimum of annotation. The ability to re-integrate derived relationships into parsed trees makes it possible, for example, to do statistical analyses on pronominalisation patterns and co-valuation (for example, topic persistence, donkey anaphora, some types of periphrasis, etc.). When paired with an exhaustive analysis of grammatical relations in context, adding an additional layer of semantic roles is informative not only for lexical semantics, but also for studies on the interaction of semantic and grammatical roles (e.g., case frame theory). With the semantic calculation, fleshed out lexical profiles for phrasal heads can be generated. These can be used, for example, to identify null positions, which in turn can add to the possibilities for annotating cohesive relations in texts. With the systematic enrichment of annotation capturing the intuitions of native speakers, a data-driven description of language at a high level of abstraction becomes a real possibility.

Acknowledgements

We are grateful to three anonymous reviewers who gave extremely valuable feedback to improve the paper. Support was received from JSPS Grant-in-Aid for Scientific Research KAKENHI grants 15K02469 and 18K00560. Support also came from the "Development of and Research with a Parsed Corpus of Japanese" collaborative project based at the National Institute for Japanese Language and Linguistics. We thank all project members for their help and advice.

References

Abzianidze, Lasha, Johannes Bjerva, Kilian Evang, Hessel Haagsma, Rik van Noord, Pierre Ludmann, Duc-Duy Nguyen, and Johan Bos. 2017. The parallel meaning bank: Towards a multilingual corpus of translations annotated with compositional meaning representations. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages Valencia, Spain. 242–247.

Basile, V., J. Bos, K. Evang, and N.J. Venhuizen. 2012. Developing a large semantically annotated corpus. In *Proceedings of the 8th Int. Conf. on Language Resources and Evaluation*. Istanbul, Turkey.

Butler, Alastair and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT* 7(1):1–22.

Dyvik, Helge, Paul Meurer, Victoria Rosén, Koenraad De Smedt, Petter Haugereid, Gyri Smørdal Losnegaard, Gunn Inger Lyse, and Martha Thunes. 2016. NorGramBank: A 'Deep' Treebank for Norwegian. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3555–3562. Paris, France: European Language Resources Association (ELRA).

Flickinger, Dan, Valia Kordoni, and Yi Zhang. 2012. DeepBank: A dynamically annotated treebank of the Wall Street Journal. In *Proceedings of TLT-11*. Lisbon, Portugal.

Garside, Roger, Geoffrey Leech, and Geoffrey Sampson, eds. 1987. *The Computational Analysis of English: a corpus-based approach*. London: Longman.

Hockenmaier, Julia and Mark Steedman. 2005. CCGbank: User's manual. Tech. Rep. MS-CIS-05-09, Department of Computer and Information Science, University of Pennsylvania, Philadelphia.

Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic: Introduction to Model-theoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: Kluwer.

Kiselyov, Oleg. 2018. Transformational Semantics on a Tree Bank. In S. Arai, K. Kojima, K. Mineshima, D. Bekki, K. Satoh, and Y. Ohta, eds., *JSAI-isAI 2017*, vol. 10838 of *Lecture Notes in Computer Science*, pages 241–252. Heidelberg: Springer.

Kroch, Anthony, Beatrice Santorini, and Ariel Diertani. 2010. *The Penn-Helsinki Parsed Corpus of Modern British English (PPCMBE)*. Department of Linguistics, University of Pennsylvania. CD-ROM, second edition, (http://www.ling.upenn.edu/hist-corpora).

Marcus, Michell, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

Moot, Richard. 2015. A Type-Logical Treebank for French. *Journal of Language Modelling* 3(1):229–265.

Randall, Beth. 2009. CorpusSearch 2 Users Guide. (http://corpussearch.sourceforge.net/CS-manual/Contents.html).

Sampson, Geoffrey R. 1995. *English for the Computer: The SUSANNE Corpus and Analytic Scheme*. Oxford: Clarendon Press (Oxford University Press).

Santorini, Beatrice. 2010. Annotation manual for the Penn Historical Corpora and the PCEEC (Release 2). Tech. rep., Department of Computer and Information Science, University of Pennsylvania, Philadelphia. (http://www.ling.upenn.edu/histcorpora/annotation).