

# Round-trips with meaning stopovers

**Alastair Butler**

Institute for Excellence in Higher Education

Tohoku University

ajb129@hotmail.com

## Abstract

This paper describes taking parsed sentences, going to meaning representations (the stopover), and then back to parsed sentences (the round trip). Keeping to the same language tests the combined success of building meaning representations from parsed input and of generating parsed output. Switching languages when manipulating meaning representations would achieve translation. Transfer shortfall is seen with meaning representations built from parsed parallel corpora data, with English-Japanese as an example.

## 1 Introduction

Recent years have seen progress in the development of open-domain semantic parsers able to convert natural language input to representations that preserve much semantic content (see e.g., Schubert 2015 for an overview). This becomes relevant for translation if there is also a way back to a language string, that is, if there can also be generation from meaning representations. This paper describes a full pipeline: form (Historical) Penn-treebank parsed sentences, a semantic parser is used to create standard predicate logic based meaning representations (see e.g., Dowty, Wall and Peters 1981), which are converted to PENMAN notation (Matthiessen and Bateman 1991) to form the basis for generation, which proceeds as a manipulation of tree

structure to produce an output parsed tree which can yield a language string.

The method is illustrated by round tripping on English, so taking English parsed sentences, going to meaning representations, and then back to parsed sentences of English. It is equally possible to change the front or back end of the pipeline, e.g., calculate a meaning representation for an English sentence but use generation rules designed for Japanese. With no modification to the stopover meaning representation this arrives at a result with English words and concepts and yet Japanese parse structure. Obtaining meaning representations from parsed parallel corpora is also illustrated to form the basis for capturing data to inform the gap that remains between the meaning representations needed to generate sentences of one language from another.

The paper is structured as follows. Section 2 discusses related work. Section 3 introduces the semantic parsing to start the pipeline. Section 4 details changes for generation. Section 5 presents results of experiments carried out round tripping on English data. Section 6 discusses the open issue of what remains for translation from one language to another, with an English to Japanese example. Section 7 concludes.

## 2 Related work

There are many options for reaching what might be called meaning representations. Schubert (2015) is a recent overview of 12 distinct approaches, many with multiple implementations. Of alternatives to section

3, most closely related is the Boxer system (Bos 2008), which is also part of a pipeline taking parsed input (Boxer uses CCG derivations), and which also implements results of Dynamic Semantics, such as capturing donkey anaphora and handling quantification (see e.g., Eijck and Visser 2012; Boxer uses DRT (Kamp and Reyle 1993), rather than SCT of section 3.2). A notable difference in output is with the linking of predicates: Boxer adopts, in contrast to classical DRT, a neo-Davidsonian approach with information loss for how adjuncts are anchored, posing difficulties for transforming to the PENMAN notation used for generation in section 4. Boxer, the section 3 approach, as well as many others, aim to capture compositional sentence/discourse meaning by building representations with model theoretic embeddings, rather than aiming for a more usage directed “speaker meaning” (see e.g. Bender et al 2015 for a viewpoint against conflating sentence/speaker meaning).

With the approach of this paper, after the meaning representation is reached, much is accomplished with tree transformations. Schubert (2014) is an alternative to building meaning representations from parsed treebank data with only tree transformations, and with a different tree transforming engine (TTT; Purtee and Schubert 2012).

For semantic parsing directly to PENMAN notation, there is JAMR (Flanigan et al 2014), a semantic parser natively producing Abstract Meaning Representations (AMRs; Banarescu et al 2013). JAMR replicates the (by design) limitations of AMR (e.g., sentence outlook, absence of quantification, absence of tense information), and offers AMR advantages of developed predicate senses and semantic roles.

Generation from PENMAN notation is also a well established research area, with notably the Nitrogen system (Langkilde and Knight 1998). Nitrogen relies on a statistical component to filter results generated from a base system with phrase structure

like rules. There are other systems with generation from representations of argument structure or quasi-logical forms (e.g., Alshawi 1992, Humphreys et al 2001). The generation of this paper follows a series of transformation rules most similar to those proposed in the generative grammar literature (e.g., Chomsky and Lasnik 1993), which provides the theoretical foundation underlying the treebank annotation of section 3.1. To the knowledge of the author, the system of this paper is the first to bring together components to round trip on languages and evaluate the results based on a metric measuring semantic analysis.

### 3 Reaching meaning representations

The approach of this paper first requires a way to reach meaning representations from natural language input. Here, use is made of Treebank Semantics (Butler and Yoshimoto 2012, Butler 2015), for the ease with which it fits into the described pipeline, since it takes as input the parsed trees that will be generated as output, and for the quality of meaning representations produced.

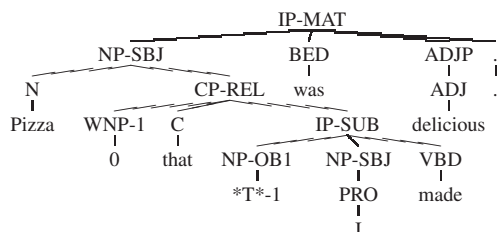
Treebank Semantics works by converting parsed constituent tree annotations into expressions of a Dynamic Semantics language (Scope Control Theory or SCT; Butler 2015) which is processed against a sequence based information state (cf. Vermeulen 2000, Dekker 2012) to return predicate logic based representations. Section 3.1 outlines the treebank annotation, while section 3.2 sketches reaching a meaning representation from an example sentence.

#### 3.1 Treebank annotation

The Treebank Semantics system accepts parsed data conforming to the *Annotation manual for the Penn Historical Corpora and the PCEEC* (Santorini 2010). This widely and diversely applied scheme forms the basis of annotations for over 600,000 analysed sentences of English (Taylor et al 2003, Kroch, Santorini and Delfs 2004,

Kroch, Santorini and Delfs 2004), French (Martineau et al 2010), Icelandic (Wallenberg et al 2011), Portuguese (Galves and Britto 2002), Ancient Greek (Beck 2013), Japanese (Butler et al 2012), and Chinese (Zhou 2015) among other languages, and has parsing systems to produce annotated trees from raw language input (e.g., Kulick, Kroch and Santorini 2014, Fang, Butler and Yoshimoto 2014).

With the annotation scheme constituent structure is represented with labelled bracketing and augmented with grammatical functions and notation for recovering discontinuous constituents. A parse in tree form for the sentence *Pizza that I made was delicious* looks like:



Every word has a word level part-of-speech label. Phrasal nodes (NP, PP, ADJP, etc.) immediately dominate the phrase head (N, P, ADJ, etc.), so that the phrase head has as sisters both modifiers and complements. Modifiers and complements are distinguished by extended phrase labels marking function (e.g., CP-REL above encodes *that I made* is a relative clause, and so a modifier of the head noun *Pizza*). All noun phrases immediately dominated by IP are marked for function (NP-SBJ=subject, NP-OBJ=direct object, NP-TMP=temporal NP, etc.). All clauses have extended labels to mark function (IP-MAT=matrix clause, CP-ADV=adverbial clause, etc.). There can be additional annotation containing scope information to ensure an unambiguous parse with respect to a default scope hierarchy.

### 3.2 Obtaining meaning representations

To obtain meaning representations, the first step is to convert a labelled bracketed tree into an expression to input to the SCT eval-

uation system. An SCT expression is built exploiting the input phrase structure by locating any complement for the phrase head to scope over, and adding modifiers as elements that scope above the head. During construction information about binding names is gathered and integrated with `fn fh =>` and `fn lc =>` acting as lambda abstractions. As a demonstration, the tree of section 3.1 converts as follows:

```
val ex1 =
  ( fn fh =>
    ( fn lc =>
      ( some lc fh "entity"
        ( relc lc "q1"
          ( pro fh "I" "arg0"
            ( arg "q1" "arg1"
              ( past "event"
                ( verb lc "event"
                  ["arg0", "arg1"] "made"))))
          ( nn lc "Pizza"))
          "arg0"
          ( some lc fh "attrib" ( adj lc "delicious")
            "attribute"
            ( past "event"
              ( verb lc "event" ["arg0"] "was"))))
          ["attribute", "arg1", "arg0"])
          ["event", "entity", "attrib"]
```

This conversion to `ex1` notably transforms into operations the part of speech tags given by the nodes immediately dominating the terminals of the input constituent tree (`some` (indefinite) `nn` (noun), `verb`, `arg` (trace), etc.), as well as triggering operations for certain construction types (e.g., `relc` occurs because there is a relative clause). Conversion also adds (i) information about local binding names (e.g., "arg0" (logical subject role), "arg1" (logical object role), "attribute"), and (ii) information about sources for fresh bindings (e.g., "event", "entity" and "attrib") for the introduction of variables of different sorts. The created operations further reduce to primitives of the SCT language as demonstrated with:

```
Hide ("event",
Close ("=", ("entity","entity"),["event", "entity", "attrib"],
...
Lam ("q1", "arg1",
  IF (fn,
    Use ("event",
      IF (fn,
        Rel ([], [], "made", [At (T ("arg0", 0), "arg0"),
          At (T ("arg1", 0), "arg1"),
          At (T ("event", 0), "event")])),
        ...
```

The SCT language primitives access and possibly alter the content of a sequence based information state that serves to retain binding information by assigning (possibly empty) sequences of values to binding names, notably: `Use` (triggers quantification introduction), `Hide` (occludes `Use`), `At` (constructs argument,role pairs), `Close` (quantificational closure), `Rel` (constructs relations), `If` (conditional to select what is evaluated), and `Lam` (shifts bindings between binding names). Evaluation of the resulting SCT expression conspires to bring about the enforcement of fixed roles on the binding names from the conversion of the parsed constituent tree annotation ("`arg0`", "`arg1`", "`attrib`", etc.).

With evaluation of `ex1`, the following meaning representation is returned:

$$\begin{aligned} &\exists z_4 x_1 A_5 e_2 e_3 ( \\ &\quad \text{past}(e_2) \wedge \\ &\quad \text{past}(e_3) \wedge \\ &\quad \text{delicious}(A_5) \wedge \\ &\quad \text{made}(e_2, z_4, x_1) \wedge \text{pizza}(x_1) \wedge \\ &\quad z_4 = I \wedge \text{was}(e_3, x_1, A_5) ) \end{aligned}$$

This assumes a Davidsonian theory (Davidson 1967) in which verbs are encoded with minimally an implicit event argument which is existentially quantified over and may be further modified. Such a meaning representation encodes truth-conditional content in a standard way, but also contains clues to assist generation. Most notably variables have sort information, thus:  $e_1, e_2, \dots$  are events,  $x_1, x_2, \dots$  are objects,  $A_1, A_2, \dots$  are attributes, etc. Also, the main predicate is the most deeply embedded right-side predicate.

## 4 Generation

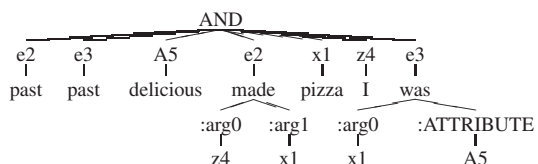
The idea behind the approach to generation is, from a meaning representation presented as a tree, to follow a series of meaning preserving transformations to arrive back at a parsed syntactic representation, that is, to a representation of the kind fed to the Treebank Semantics system at the start of the pipeline. There are two major steps. First, there is preparation, discussed in section 4.1, and subsequently there is generation, demonstrated in section 4.2 as building and transforming tree structure.

### 4.1 Preparing for generation

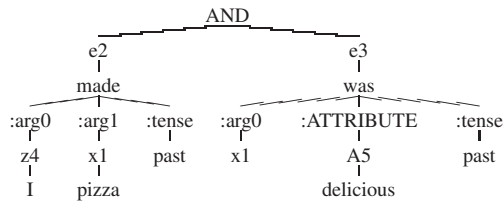
Preparation for generation involves obtaining an alternative tree-based representation of the output produced by Treebank Semantics. Rendering the meaning representation of section 3.2 as a tree with argument role information made explicit gives:



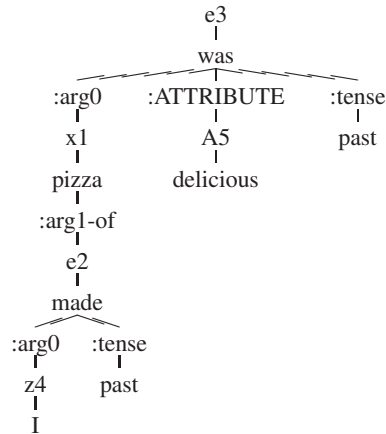
Content is further re-packaged to a tree format optimised for generation. Firstly, the binding of wide-scope existentials is made implicit with the removal of the top quantification level. Next, an argument of each predicate is promoted to become the parent of the predicate, notably: the left-hand argument of an equality relation, or an event argument if present, or the sole argument of a one-place predicate.



Next, a daughter `D` of the top level `AND` is moved inside a sister `S` when the argument name at the root of `D` is contained as an argument within `S`. Movement is to only one location (the left-most).



An internal argument is promoted to become the root of a daughter of AND if this enables further inclusion into a sister. Promotion relies on folding tree material around inverse roles from the PENMAN notation (Matthiessen and Bateman 1991), e.g., having *arg1-of* as an inverse of *arg1* (logical object) enables transformation to the following single rooted structure, and more generally compacts long distance dependencies such as are established as WH-dependencies in English:



## 4.2 Back to a parsed representation

Representations resulting from the changes of the previous section are now used as the basis for generation. This proceeds as a series of tree transformations, implemented as a *tsurgeon* script (Levy and Andrew 2006) with hundreds of transformation rules.

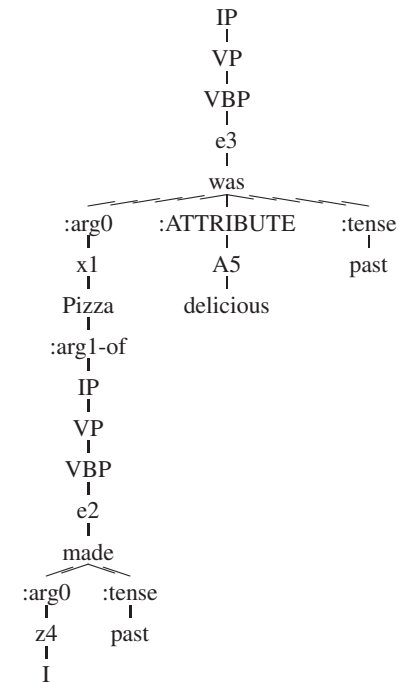
A *tsurgeon* script contains pattern/action rules, where the pattern describes tree structure and the action transforms the tree, e.g., moving, adjoining, copying or deleting auxiliary trees or relabelling nodes. Trans-

formations are repeatedly made until the pattern that triggers change is no longer matched. Thus, clause structure is built by identifying a main predicate as being headed by an event variable (so: match *e* followed by a number), and adjoining the projection of a VBP part-of-speech tag, a VP layer and an IP layer.

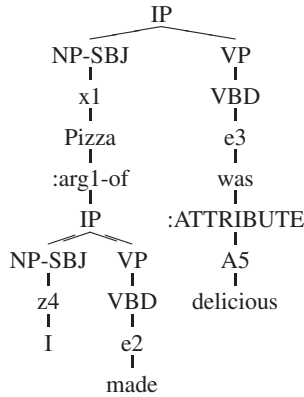
```
/^e[0-9]+$/=x !> VBP
```

```
adjoinF (IP (VP (VBP @))) x
```

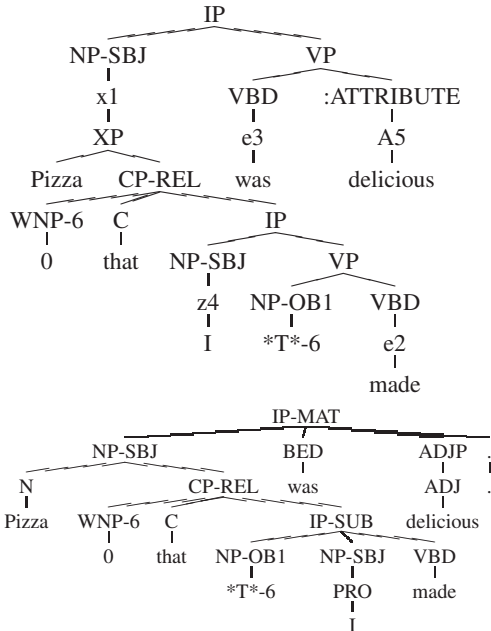
Action *adjoinF* adjoins the specified auxiliary tree into the specified target node, preserving the target node as the foot of the adjoined tree. VBP (present tense verb) may subsequently change, e.g., *tense past* triggers change to VBD (past tense verb), while *was* when generating English brings about further change to BED (past tense copula).



Subsequent changes involve moving all structure under a main predicate into the clause, starting with the creation of NP-SBJ from an *arg0* argument at the IP-level.



The inverse role `arg1-of` is the foundation for relative clause structure with an NP-OB1 (object) trace, while if pizza had been headed by an event variable, the structure would bring about a clausal embedding.



If an `arg0` argument happened to be missing, either a passive transformation may result or there may be inclusion of a subject

register	sentences	tokens	precision	recall	F-score
textbook	687	5194	0.98	0.98	0.98
newswire	121	2381	0.97	0.96	0.97
(simple) fiction	547	5241	0.96	0.96	0.96
non-fiction	97	1302	0.93	0.93	0.93

Table 1: smatch scores comparing meaning representations from original and generated sentences

expletive *it* or *there* for English. Adjunct materials can find placement based on argument role information or subtree size, e.g., vocatives (NP-VOC) are always clause initial, a temporal NP (NP-TMP) will typically be clause initial, while, for English, clause final positioning will be favoured for a heavy PP or NP (whose children reach large depths). Having arguments with the same referent can trigger the introduction of infinitival or participial clause structure to create control configurations or various types of ellipsis, such as VP ellipsis.

## 5 Experiments

In this section, the smatch metric for measuring semantic annotation agreement rates and semantic parsing accuracy (Cai and Knight 2013) is used to evaluate the success of round tripping on English. This is a metric to measure whole-sentence semantic analysis by calculating the degree of overlap between meaning representations.

The representation seen at the end of section 4.1 is essentially compatible for calculating a smatch score. This gives a meaning representation for the input sentence. A meaning representation for the output sentence is achieved by feeding the resulting output of the round trip back into the Treebank Semantics system.

Table 1 details results for 1452 annotated sentences (14,118 tokens) from four different registers that were manually selected to illustrate different levels of sentence complexity. All sentences are from the Treebank Semantics Corpus<sup>1</sup> with sentences parsed to gold standard following the annotation scheme detailed in section 3.1, and so already unambiguous for feeding to

<sup>1</sup><https://github.com/ajb129/tscorpus>

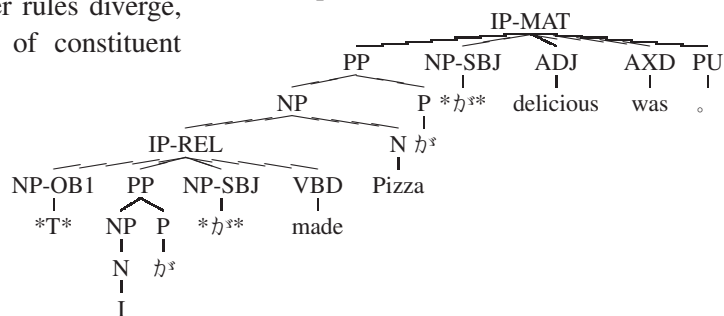
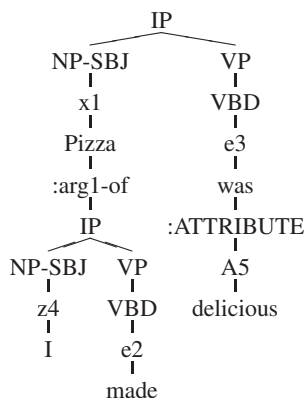
the semantic component for the creation of a gold standard meaning representation.

The results show that in round tripping with English, so building a meaning representation A and generating back to an English sentence and then building a meaning representation B from the generated sentence, and then comparing A with B, it is possible to retain the bulk of semantic content with high precision and recall.

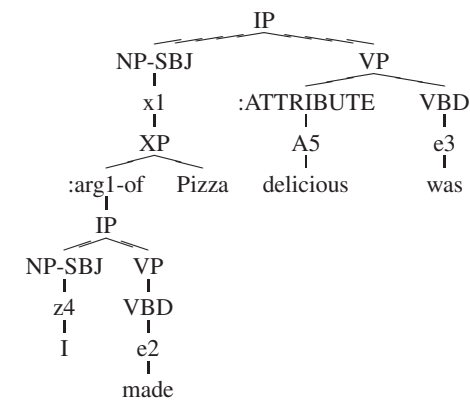
The results also reflect that performance starts to decline on more challenging data. In particular there is a notable reduction in F-score with the non-fiction data (from a technical manual describing the IBM 1401 Programming System). Weaknesses revealed typically involve complex interactions, such as happen with coordination, or stem from constructions that are difficult to provide a generalisable semantic analysis, such as comparatives. On the generation side, improvements are possible with more construction and lexical specific pattern/action rules, reordering existing rules, or arranging for existing rules to be retriggered.

## 6 Towards translation

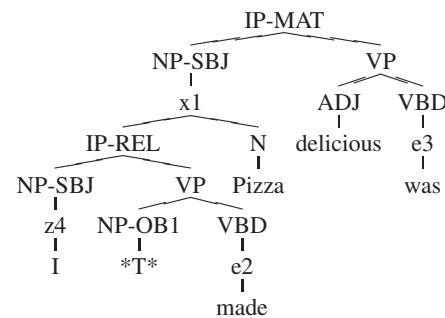
In this section, generation rules for Japanese are demonstrated. Consider starting with the same meaning representation input as section 4.2, and first projecting VP, IP structure. Thereafter rules diverge, differing mostly in terms of constituent placement.



This has demonstrated generation to Japanese parse structure from a meaning representation with English words and concepts. In the case of this illustrative example there is a close match with the

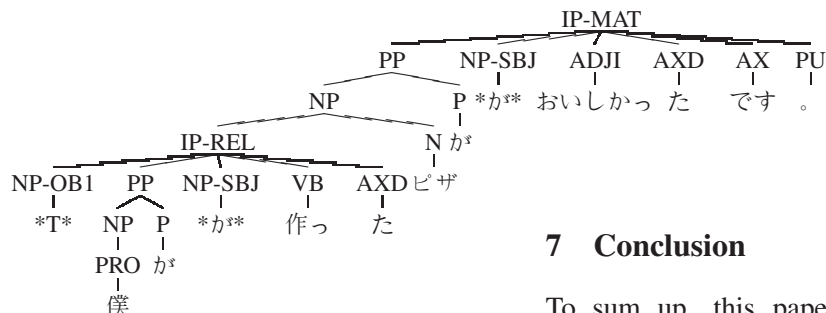


Projection of relative clause structure is again triggered, only for Japanese there is projection of an IP-REL layer to the left side of the head noun.



Generation is completed with the addition of case particles.

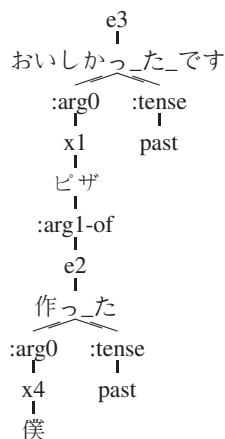
corresponding Japanese version 僕が作ったピザがおいしかったです, seen annotated below. However, for the general translation task, substantial transformation and lexical substitution of the meaning representation used for generation will be required.



By feeding the Japanese version of the example sentence into the Treebank Semantics system a meaning representation is built:

$$\begin{aligned} &\exists x_4 x_1 e_2 e_3 ( \\ &\quad \text{past} (e_3) \wedge \\ &\quad \text{past} (e_2) \wedge \\ &\quad x_4 = \text{僕} \wedge \\ &\quad \text{作っ}_\text{た} (e_2, x_4, x_1) \wedge \\ &\quad \text{ピザ} (x_1) \wedge \\ &\quad \text{おいし}_\text{かつ}_\text{た}_\text{です} (e_3, x_1) ) \end{aligned}$$

Such a representation can be modified, as in section 4.1, to form the basis for generation, exactly as with the English example.



Having the above meaning representation and the meaning representation for the cor-

responding English sentence in section 4.1, together with meaning representations for other sentences of parallel corpora, is a basis for extracting rules for a full translation system.

## 7 Conclusion

To sum up, this paper has described a complete pipeline for taking parsed sentences, going to meaning representations (initially to standard Davidsonian predicate logic based meaning representations, then to PENMAN notation), and then back to parsed sentences (the round trip). Keeping to the same language tests the combined success of building meaning representations from parsed input and of generating parsed output. Using the smatch metric reveals that the bulk of semantic content is retained with high precision and recall on a range of data.

Results show that, while there is no explicit flagging in a conventional Davidsonian predicate logic meaning representation, as seen in section 3.2, of what is a verb, noun, adjective, relative clause, passive, control relation, etc., much information is found to facilitate generation when there is sort and argument role information and when there is subsequent re-packaging of content, as in section 4.1, guided by the aim to form single rooted structures.

The future direction for this research is to show relevance for translation in being able to switch languages at the point of manipulating meaning representations. Current transfer shortfall is seen with meaning representations built from parsed parallel corpora data.



## Acknowledgements

This paper benefitted considerably from the comments of three anonymous reviewers, as well as discussions with Pascual Martínez-Gómez, Masaaki Nagata, Kei Yoshimoto and Zhen Zhou, all of which is very gratefully acknowledged. This research is supported by the Japan Society for the Promotion of Science (JSPS), Research Project Number: 15K02469.

## References

- Alshawi, Hiyun. 1992. *The Core Language Engine*. Cambridge, Mass.: MIT Press.
- Banarescu, L., C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider. 2013. Abstract Meaning Representation for Sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186.
- Beck, Jana E. 2013. Annotation manual for the Penn parsed corpora of Historical Greek. Tech. rep., Department of Computer and Information Science, University of Pennsylvania, Philadelphia.
- Bender, Emily M., Dan Flickinger, Stephan Oepen, Woodley Packard, and Ann Copestake. 2015. Layers of interpretation: On grammar and compositionality. In *Proceedings of the 11th International Conference on Computational Semantics*, pages 239–249. London, UK: Association for Computational Linguistics.
- Bos, Johan. 2008. Wide-coverage semantic analysis with Boxer. In J. Bos and R. Delmonte, eds., *Semantics in Text Processing. STEP 2008 Conference Proceedings*, Research in Computational Semantics, pages 277–286. College Publications.
- Butler, Alastair. 2015. *Linguistic Expressions and Semantic Processing: A Practical Approach*. Heidelberg: Springer-Verlag.
- Butler, Alastair, Zhu Hong, Tomoko Hotta, Ruriko Otomo, Kei Yoshimoto, and Zhen Zhou. 2012. Keyaki treebank: phrase structure with functional information for Japanese. In *Proceedings of Text Annotation Workshop*.
- Butler, Alastair and Kei Yoshimoto. 2012. Banking meaning representations from treebanks. *Linguistic Issues in Language Technology - LiLT* 7(1):1–22.
- Cai, Shu and Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. In *Proc. of the ACL 2013*.
- Chomsky, Noam and Howard Lasnik. 1993. The theory of Principles and Parameters. In J. Jacobs, ed., *Syntax: An International Handbook of Contemporary Research*. Berlin: Walter de Gruyter.
- Davidson, Donald. 1967. The logical form of action sentences. In N. Rescher, ed., *The Logic of Decision and Action*. Pittsburgh: University of Pittsburgh Press. Reprinted in: D. Davidson, 1980. *Essays on Actions and Events*. Clarendon Press, Oxford, pages 105–122.
- Dekker, Paul. 2012. *Dynamic Semantics*, vol. 91 of *Studies in Linguistics and Philosophy*. Dordrecht: Springer Verlag.
- Dowty, David, Robert Wall, and Stanley Peters. 1981. *Introduction to Montague Semantics*. Dordrecht: Kluwer.
- Eijck, Jan van and Albert Visser. 2012. Dynamic Semantics. In E. N. Zalta, ed., *The Stanford Encyclopedia of Philosophy*. Winter 2012 edn.
- Fang, Tsaiwei, Alastair Butler, and Kei Yoshimoto. 2014. Parsing Japanese with a PCFG treebank grammar. In *Proceedings of the Twentieth Annual Meeting of the Association of Natural Language Processing*, pages 432–435. Sapporo, Japan.
- Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer, and Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. In *Proc. of the ACL 2014*.
- Galves, Charlotte and Helena Britto. 2002. *The Tycho Brahe Corpus of Historical Portuguese*. Department of Linguistics, University of Campinas. Online publication, first edition.
- Humphreys, Kevin, Mike Calcagno, and David Weise. 2001. Reusing a statistical language model for generation. In *Proceedings of the 8th European workshop on Natural Language Generation - Volume 8*, pages 1–6. Stroudsburg, PA: Association for Computational Linguistics.
- 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.
- Kroch, Anthony, Beatrice Santorini, and Lauren Delfs. 2004. *The Penn-Helsinki Parsed Corpus of Early Modern English (PPCEME)*. Department of Linguistics, University of Pennsylvania. CD-ROM, first edition.
- Kulick, Seth, Anthony Kroch, and Beatrice Santorini. 2014. The Penn parsed corpus of Modern British English: First parsing results and analysis. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 662–667. Baltimore, Maryland, USA: Association for Computational Linguistics.

- Langkilde, Irene and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of the ACL/COLING-98*. Montreal, Québec.
- Levy, Roger and Galen Andrew. 2006. Tregex and tsurgeon: tools for querying and manipulating tree data structure. In *5th International conference on Language Resources and Evaluation*.
- Martineau, France, Paul Hirschbühler, Anthony Kroch, and Yves Charles Morin. 2010. *Corpus MCVF (parsed corpus), Modéliser le changement : les voies du français*. Département de français, University of Ottawa. CD-ROM, first edition.
- Matthiessen, Christian and John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from English and Japanese*. Pinter Publishers.
- Purtee, A. and Lenhart K. Schubert. 2012. TTT: A tree transduction language for syntactic and semantic processing. In *EACL 2012 Workshop on Applications of Tree Automata Techniques in Natural Language Processing (ATANLP 2012)*. Avignon, France.
- 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.
- Schubert, Lenhart K. 2014. From treebank parses to episodic logic and commonsense inference. In *Proc. of the ACL 2014 Workshop on Semantic Parsing*, pages 55–60. Baltimore, MD.
- Schubert, Lenhart K. 2015. Semantic representation. In *29th AAAI Conference (AAAI15)*, pages 55–60. Austin, TX.
- Taylor, Ann, Anthony Warner, Susan Pintzuk, and Frank Beths. 2003. *The York-Toronto-Helsinki Parsed Corpus of Old English Prose (YCOE)*. Department of Linguistics, University of York. Oxford Text Archive, first edition.
- Vermeulen, C. F. M. 2000. Variables as stacks: A case study in dynamic model theory. *Journal of Logic, Language and Information* 9:143–167.
- Wallenberg, Joel, Anton Karl Ingason, Einar Freyr Sigurðsson, and Eiríkur Rögnvaldsson. 2011. *Icelandic Parsed Historical Corpus (IcePaHC)*. Department of Linguistics, University of Iceland. Online publication, version 0.9.
- Zhou, Zhen. 2015. *Parsing Chinese for Semantic Processing: Focusing on Serial Verb Constructions and Resultative Constructions*. Ph.D. thesis, Tohoku University, Sendai, Japan.