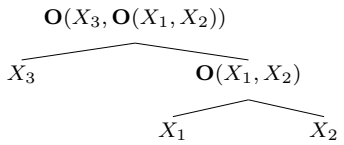
A Tensor-based Vector Space Semantics for Dynamic Syntax

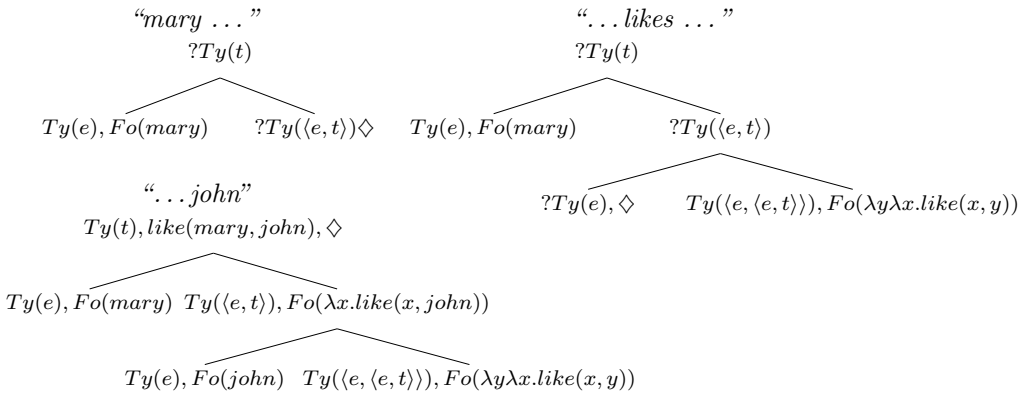
Mehrnoosh Sadrzadeh, Matthew Purver, Ruth Kempson
Queen Mary University of London

In its original form, Dynamic Syntax (DS) provides a strictly incremental formalism relating word sequences to semantic representations. Conventionally, these are seen as trees decorated with semantic formulae that are terms in a typed lambda calculus (Kempson et al., 2001, chapter 9):



In this book we will take the operation **O** *to be function application in a typed lambda calculus, and the objects of the parsing process [. . .] will be terms in this calculus together with some labels; [. . .]*

This, of course, allows us to give now-familiar analyses of the semantic output of the word-by-word parsing process, as shown below for the simple sentence *"mary likes john"*:



However, the formalism is in fact considerably more general. To continue the quotation above:

> *[. . .] it is important to keep in mind that the choice of the actual representation language is not central to the parsing model developed here. [. . .] For instance, we may take* X_1, X_2, X_3 *to be feature structures and the operation* **O** *to be unification, or* X_1, X_2, X_3 *to be lambda terms and* **O** *Application, or* X_1, X_2, X_3 *to be labelled categorial expressions and* **O** *Application: Modus Ponens, or* X_1, X_2, X_3 *to be DRSs and* **O** *Merging.*

Indeed, in some variants this generality is exploited; for example, Purver et al. (2010) outline a version in which the formulae are *record types* in Type Theory with Records (Cooper, 2005); and Hough and Purver (2012) show how this can confer an extra advantage – the incremental decoration of the *root* node, even for partial trees, with a maximally specific formula via type inference. In this paper, we show how a similar passage can be defined for vector space representations.

Vector space semantic models have a complementary property to DS. Whereas DS is agnostic to its choice of semantics, vector space models are agnostic to the choice of the syntactic system. In (Coecke et al., 2010), we show how they provide semantics for sentences based on the grammatical structures given by Lambek's pregroup grammars (Lambek, 1997); in (Coecke et al., 2013) we show how this semantics also works starting from the parse trees of Lambek's syntactic calculus (Lambek, 1958); Wijnholds (2017) shows how the same semantics can be extended to the Lambek-Grishin Calculus; and (Maillard et al., 2014; Krishnamurthy and Mitchell, 2013; Baroni et al., 2014) show how it works for Combinatory Categorial Grammar trees.

This semantics homomorphically maps the concatenation and slashes of categorial grammars to tensors and their evaluation/application/composition operations to tensor contraction. In DS terms, structures X_1, X_2, X_3 are mapped to general higher order tensors, e.g. as follows:

$$\begin{array}{lcll} X_1 & \mapsto & T_{i_1,i_2,\cdots,i_n} & \in \quad V_1 \otimes V_2 \otimes \cdots V_n \\ X_2 & \mapsto & T_{i_n i_{n+1} \cdots i_{n+k}} & \in \quad V_n \otimes V_{n+1} \otimes \cdots V_{n+k} \\ X_3 & \mapsto & T_{i_{n+k} i_{n+k+1} \cdots i_{n+k+m}} & \in \quad V_{n+k} \otimes V_{n+k+1} \otimes \cdots V_{n+k+m} \end{array}$$

The **O** operations are mapped to contractions between these tensors, formed as follows:

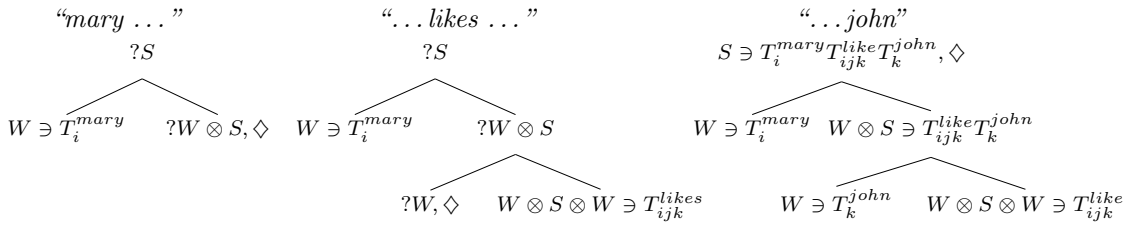
$$\begin{array}{rcl} \mathbf{O}(X_1, X_2) & \mapsto & T_{i_1 i_2 \cdots i_n} T_{i_n i_{n+1} \cdots i_{n+k}} \\ & \in & V_1 \otimes V_2 \otimes \cdots \otimes V_{n-1} \otimes V_{n+1} \otimes \cdots \otimes V_{n+k} \\ \mathbf{O}(X_3, \mathbf{O}(X_1, X_2)) & \mapsto & T_{i_1 i_2 \cdots i_n} T_{i_n i_{n+1} \cdots i_{n+k}} T_{i_{n+k} i_{n+k+1} \cdots i_{n+k+m}} \\ & \in & V_1 \otimes V_2 \otimes \cdots \otimes V_{n-1} \otimes V_{n+1} \otimes \cdots \otimes V_{n+k-1} \otimes V_{n+k+1} \otimes \cdots \otimes V_{n+k+m} \end{array}$$

In their most general form presented above, these formulae are large and the index notation becomes difficult to read. In special cases, however, it is often enough to work with spaces of rank around 3. For instance, the application between a transitive verb and its object is mapped to the following contraction:

$$T_{i_1 i_2 i_3} T_{i_3}$$

This is the contraction between a cube $T_{i_1 i_2 i_3}$ in $X_1 \otimes X_2 \otimes X_3$ and a vector T_{i_3} in X_3, resulting in a tensor T_{i_1, i_2} in $X_1 \otimes X_2$, i.e. a matrix.

We suppose $Ty(t)$ represents a sentence space S and $Ty(e)$ a word space W. Given vectors T_i^{mary}, T_k^{john} in W and the (cube) tensor T_{ijk}^{like} in $W \otimes S \otimes W$, the tensor semantic trees of the word-by-word parsing process of "*mary likes john*" become as follows:



There has been much discussion about whether sentence and word spaces should be the same or separate. In previous work, we have worked with both cases, i.e. when $W \neq S$ and when $W = S$.

DS requirements can now be treated as requirements for tensors of a particular order (e.g. $?W$, $?W \otimes S$ as above). Alternatively, we can provide an analogue to Hough and Purver (2012)'s incremental type inference procedure, by interpreting them as picking out an element which is *neutral* with regards to composition: the unit vector/tensor of the space they annotate. For the atomic S and W spaces, this is the $(1, 1)$ vector, for the verb phrase space $W \otimes S$, it is the unit matrix $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and for the transitive verb space $W \otimes S \otimes W$, it is the unit cube, which is similar to the unit matrix, with 1's on the diagonal and 0's everywhere else. This provides the desired compositionality but no new semantic information; that can arrive later as more words are parsed.

References

Baroni, M., Bernardi, R., and Zamparelli, R. (2014). Frege in space: A program of compositional distributional semantics. *Linguistic Issues in Language Technology*, 9.

Coecke, B., Grefenstette, E., and Sadrzadeh, M. (2013). Lambek vs Lambek: Functorial vector space semantics and string diagrams for Lambek calculus. *Ann. Pure and Applied Logic*, 164(11):1079–1100.

Coecke, B., Sadrzadeh, M., and Clark, S. (2010). Mathematical foundations for a compositional distributional model of meaning. *Linguistic Analysis*, 36:345–384.

Cooper, R. (2005). Records and record types in semantic theory. *J. Logic and Computation*, 15(2):99–112.

Hough, J. and Purver, M. (2012). Processing self-repairs in an incremental type-theoretic dialogue system. In *Proc. 16th SemDial Workshop*, pages 136–144, Paris, France.

Kempson, R., Meyer-Viol, W., and Gabbay, D. (2001). *Dynamic Syntax: The Flow of Language Understanding*. Blackwell, Oxford.

Krishnamurthy, J. and Mitchell, T. M. (2013). Vector space semantic parsing: A framework for compositional vector space models. In *Proc. ACL Workshop on Continuous VSMs and their Compositionality*.

Lambek, J. (1958). The mathematics of sentence structure. *American Mathematics Monthly*, 65:154–170.

Lambek, J. (1997). Type grammars revisited. In *Proc. LACL 97*. Springer.

Maillard, J., Clark, S., and Grefenstette, E. (2014). A type-driven tensor-based semantics for CCG. In *Proc. EACL Workshop on Type Theory and Natural Language Semantics*.

Purver, M., Gregoromichelaki, E., Meyer-Viol, W., and Cann, R. (2010). Splitting the 'I's and crossing the 'You's: Context, speech acts and grammar. In *Proc. 14th SemDial Workshop*, pages 43–50.

Wijnholds, G. J. (2017). Coherent diagrammatic reasoning in compositional distributional semantics. In *Proc. 24th WoLLIC Workshop*, pages 371–386.