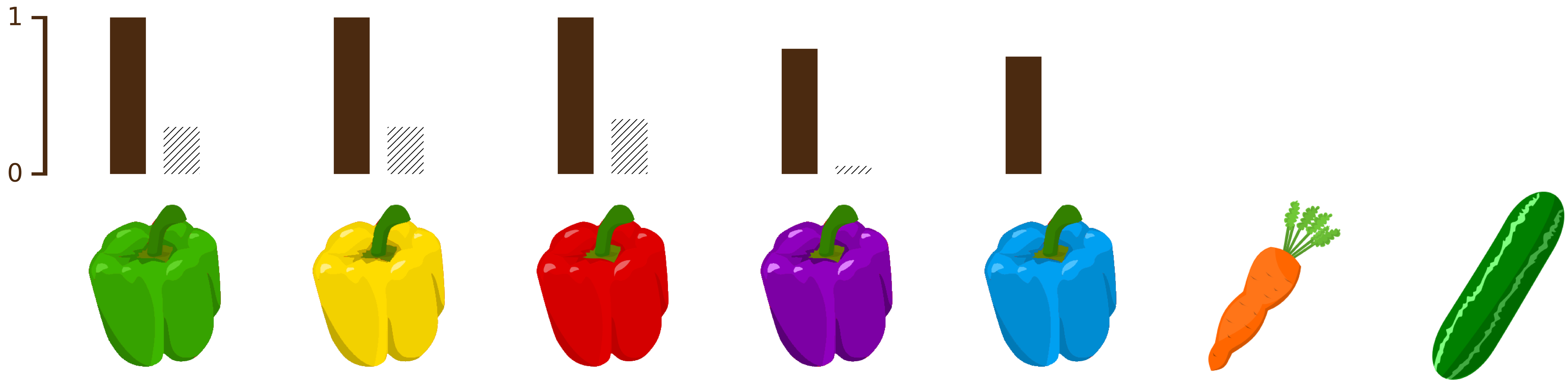
Functional Distributional Semantics

Guy Emerson and Ann Copestake
University of Cambridge
{gete2,aac10}@cam.ac.uk

Semantic Functions

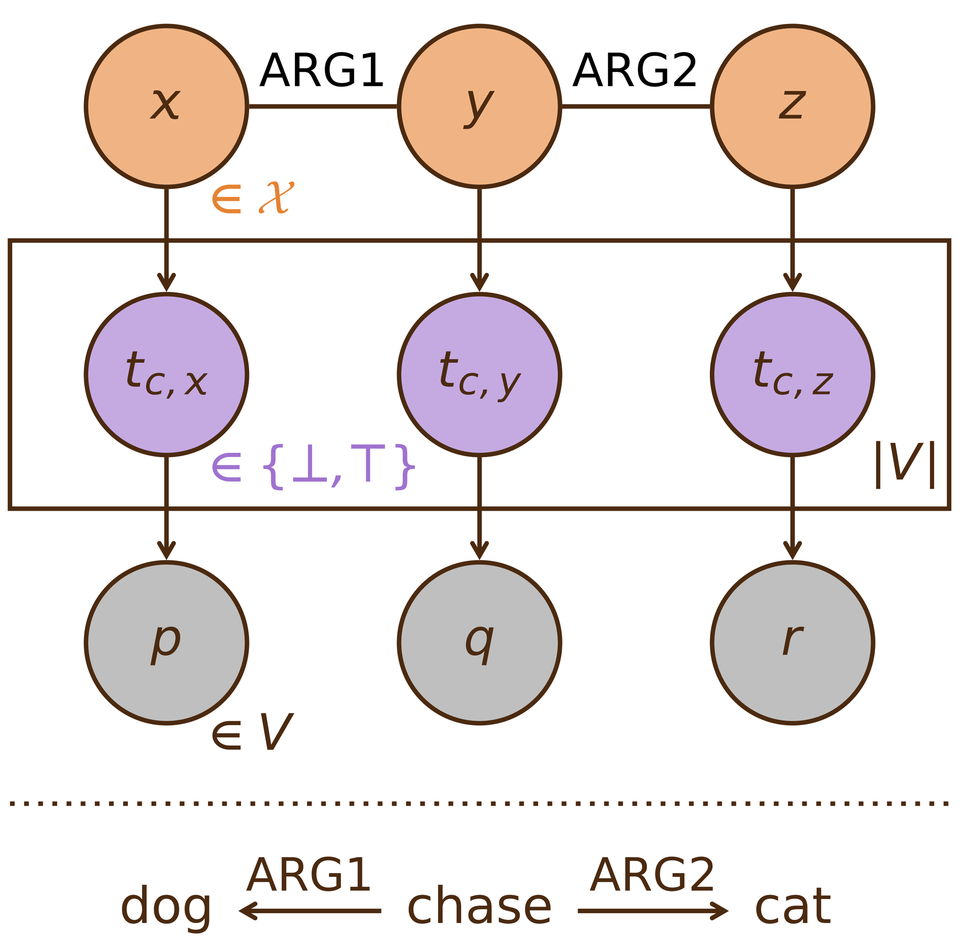
Vectors do not provide natural stucture for various semantic operations, including composition and inference. We propose representing the meaning of a predicate as a *function*, mapping entities to probabilistic truth values. The aim is to build a framework for distributional semantics which maintains strong links with formal semantics.



In traditional model-theoretic semantics, *entities* (objects in our model) are distinguished from *predicates* (which are true or false of each entity). We treat truth values as random variables, allowing us to apply Bayesian inference. We assume entities lie in a semantic space $\mathcal{X}$, and we take the meaning of a predicate to be a function from $\mathcal{X}$ to values in the interval $[0,1]$, denoting how likely a speaker is to judge the predicate applicable to the entity. We can also view such a function as a classifier, which ties in with a view of concepts as abilities, as proposed in both philosophy and cognitive science.

The vegetables above form a simple discrete space $\mathcal{X}$. We are interested in the truth t of the predicate for *bell pepper* for an entity $x \in \mathcal{X}$. **Solid bars:** the semantic function $P(t|x)$ represents how much each entity is considered to be a pepper, and is bounded between 0 and 1; it is high for all the peppers, but slightly lower for atypical colours. **Shaded bars:** the distribution $P(x|t)$ represents our belief about x, if all we know is that the predicate for *bell pepper* applies; the probability mass must sum to 1, so is split between the peppers, skewed towards typical colours, and excluding colours believed impossible.

Generative Model



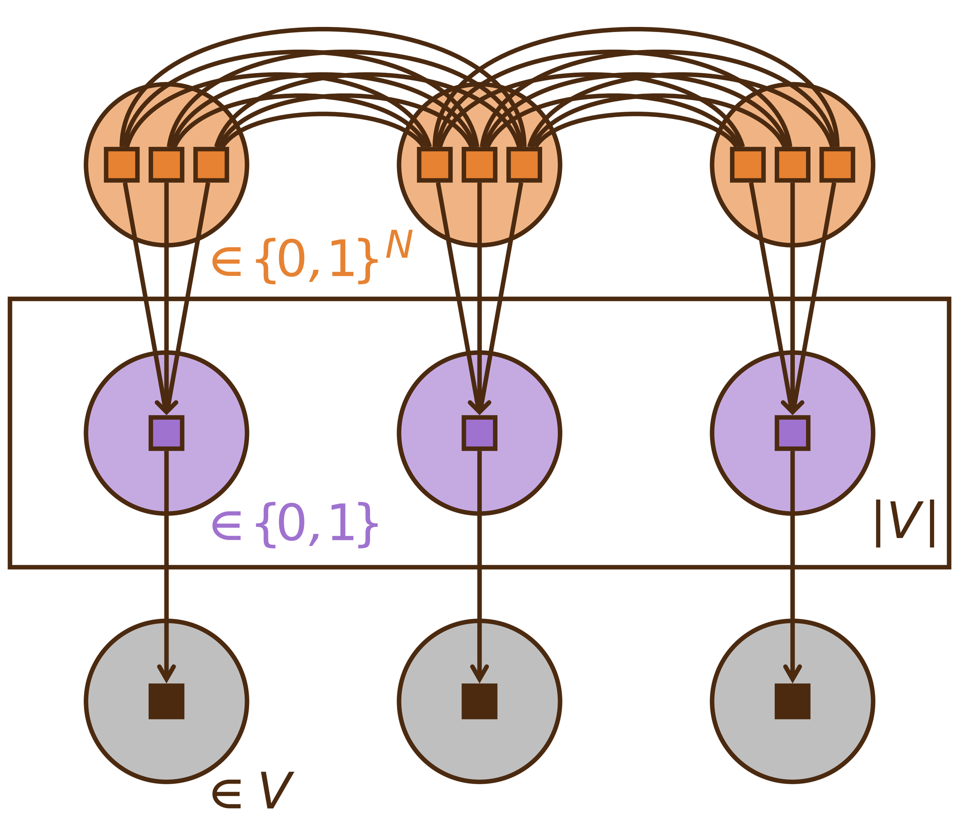
Top row: entities forming a *situation*, jointly distributed as an undirected graphical model, where the edges are semantic dependency links. **Middle row:** each predicate $c \in V$ is true or false of each entity, with probability according to the predicate's semantic function. **Bottom row:** a single true predicate is chosen for each entity. This is all we observe in distributional semantics. **Below:** example subject-verb-object (SVO) graph.

Training

Training requires a corpus of dependency graphs – we used WikiWoods, an automatically parsed version of the English Wikipedia. We aim to maximise the likelihood of observing the predicates c, integrating over latent entities x. In the gradient below (where E is energy), the first two terms give updates for the distribution over situations, and the second two terms for the semantic functions. To approximate the expectations, we sample an entity for each token, as well as 'negative' entities and 'negative' predicates.

$$\begin{aligned}\frac{\partial}{\partial\theta}\log P(c) = {} & \mathbb{E}_{x|c}\left[\frac{\partial}{\partial\theta}\left(-E^b(x)\right)\right] \\ & - \mathbb{E}_{x}\left[\frac{\partial}{\partial\theta}\left(-E^b(x)\right)\right] \\ & + \mathbb{E}_{x|c}\left[\left(1-t_c(x)\right)\frac{\partial}{\partial\theta}\left(-E^p(x,c)\right)\right] \\ & - \mathbb{E}_{x|c}\left[\mathbb{E}_{c'|x}\left[\left(1-t_{c'}(x)\right)\frac{\partial}{\partial\theta}\left(-E^p(x,c')\right)\right]\right]\end{aligned}$$

Implementation



We model entities as binary vectors, and situations using a Restricted Boltzmann Machine – so each link label has a weight matrix, e.g. representing which kinds of subjects and objects occur with which kinds of events.

We model each semantic function as a probabilistic feedforward network – so each predicate has a weight vector, representing what kinds of entity that predicate is true of.

Evaluation

Model	SL Nouns	SL Verbs	WS Sim.	WS Rel.
Word2Vec (10-word)	.28	.11	**.69**	.46
Word2Vec (2-word)	.30	.16	.65	.34
SVO Word2Vec	.44	.18	.61	.24
Sparse SVO Word2Vec	**.45**	**.27**	.63	.30
Semantic Functions	.26	.14	.34	**.01**

flood / water (related verb, noun)	.06
flood / water (related nouns)	.43
law / lawyer (related nouns)	.44
sadness / joy (near-antonyms)	.77
happiness / joy (near-synonyms)	.78
aunt / uncle (differ in one feature)	.90
cat / dog (differ in many features)	.92

Above: rank correlation of similarity for SimLex-999, and for similarity and relatedness subsets of WordSim-353. **Left:** similarity for thematically related words, versus co-hyponyms.