

Outlining Probability Distributions Using the Givenness Hierarchy

Alex LaGrassa

November 23, 2015

1 Problem

Given a sentence with modifier m , determiner d , and feature vector describing an object o_i that can be any object number $n_i \in N$, find the n_i with the highest probability of being o_i

Model the robots memory as a set of mappings from an n_i to a probability mass distribution of different possible $o_i \in O$ that object could be,. A subset of those objects are in set A of activated object (objects the robot is actively working with). Call that set A and $A \subset M$. The robot has a function that maps the number(n_i) of objects present to a probability mass function

$$F_{n_i}(n_i, o_i) = f_{o_i}(o_i) = P(n_i = o_i) = P(\{o_i \in O : n_i = o_i\})$$

1.1 Example

For example, if the sentence is "the orange carrot", then $m = orange, d = the$, and *carrot* would be part of the feature vector. If the set of possible objects is $O = \{orangecarrot, redball, greenball\}$, then the o_i pointed at in the sentence is *orangecarrot*. If there are some objects where

1. $n_0 = redball$
2. $n_1 = greenball$
3. $n_2 = orangecarrot$

then the robot might have a probability distribution for each object where

$$F_{n_0}(n_0, o_0) = F_{n_0}(n_0, redball) = P(n_0 = redball) = 0.8$$

where 0.8 is some probability with uncertainty coming from the robot's feature sampling.

2 Defining the Most Probable Object

The most probable object n is the most likely object the sentence is referring to. That means it has the highest probability of being the object α such that $/m, d/ \in \text{featurevector}_{n_i}$

Thus, we are finding

$$n.P(n = \alpha) = \max(F_{n_i}(n, \alpha))$$

3 Updating the Probabilities if $\alpha \in A$

If the human and robot were working with objects on the table, it would be much more probable that the n_i where $n_i = \alpha$ was in the set of objects on the table. The set of objects on the table would then be in the activated set, A . Thus,

$$\forall n_i, P(n_i | n_i \in A) \gg P(n_i | n_i \notin A)$$

As an example, a determiner like "that" would indicate that $\alpha \in A$. It's still worthwhile to check the probabilities $\forall n_i \in M$, but if the object were found with reasonably high probability in A , checking M might not be necessary.

This way of updating the model is adapted from using the Givenness Hierarchy, (Gundel, Hedberg, and Zacharski 1993), which associates referential expressions with a "cognitive status." For example, if an object is referred to as "this", then it is in activated memory, while "that" usually refers to a familiar object that is not necessarily in activated memory.

A probability update given that $n_i \in A$ would reweight with some $w > 0.5$ so $\forall n_i \in A, n'_i = w * n_i$ and $\forall n_i \notin A, n'_i = (1 - w) * n_i$

4 Updating Probability if $\exists_{=1} \alpha \in M$

Similarly, the Givenness Hierarchy would declare an object modified with "the" as uniquely identifiable. That means the probability function F_{n_i} now must take into account that all $\forall n_j, i \neq j, F_{n_i}(n_i, o) = \prod_{n_j \in M, j \neq i} P(n_i = o | n_j \neq o)$,

which computationally becomes a much harder problem if there is conditional dependence on the other objects being α . Intuitively, it should be true that $P(n_i = o | \exists_{=1} \alpha \in M, n_j \neq i \in M \neq o) = 1$. Thus, the problem reduces to a joint probability distribution. The next question is whether to treat all other probabilities independently. If their probability of not being α depended on all other $n_j \in M$, then the joint probability would be very difficult if not impossible to compute, since it must account for dependencies on all objects.

Alternatively, this could change the algorithm by finding the n_i with the highest $P(n_i = \alpha)$ because finding "a" n_i would be less correct than finding the most probable $n_i \in M$