Minimum Viable AGI Test “Animals1”

Goal:

1. If possible, create/design a cognitive architecture solving the problem or/and demonstration of such working solution (Jupyter/Colab notebook, web demo, video).
2. If not possible, prove why the problem can not be solved by human or/and point out why the problem statement is incorrect so human can not solve it.

Terms:

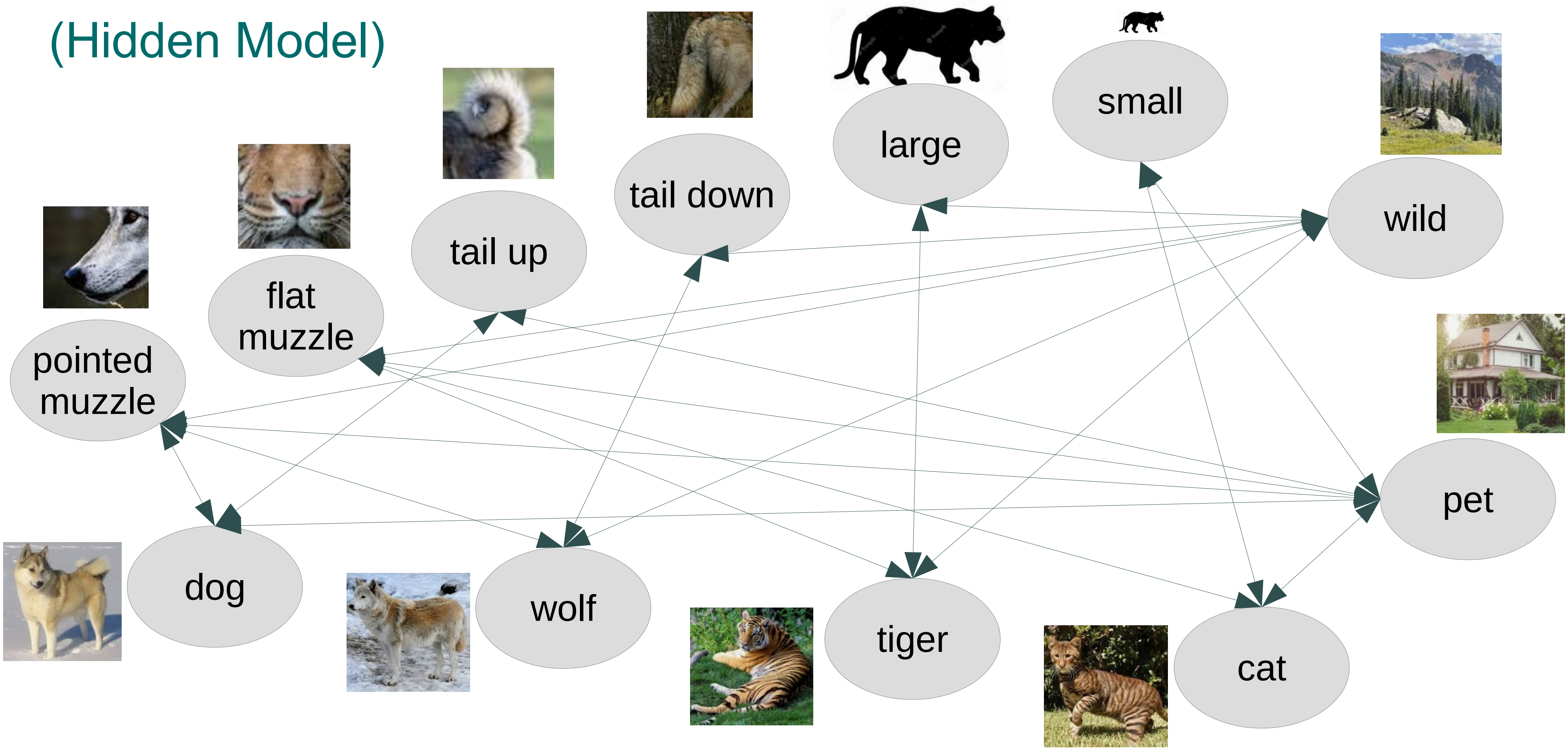
1. Have 12 variables (stimuli) which can be in state of either “present” and “unknown”.
2. First, system can be trained with any combinations (contexts) of stimuli (set to “present”) that are mutually associated according to the “hidden model” (which is not known to the system being trained). System may not be given any combination of “present” stimuli not being associated in the “hidden model”.
3. Next, system can be tested with any combination of stimuli (set to “present”) as an iinput coontext, so that when being tested,

3.1. it may respond with any stimuli (set to “present”) so that all responded stimuli are correctly associated with each other and all of the input stimuli (so that union of the input and output stimuli is not associating anything which is not associated within the “hidden model”);

3.2. it should not respond with any stimuli (see “N/A” in example) in case if any of the input stimuli is not associated with any other in context of the “hidden model”.

Meanings of Variables and their True Associations

(Hidden Model)



pointed muzzle

flat muzzle

tail up

tail down

large

small

wild

pet

dog

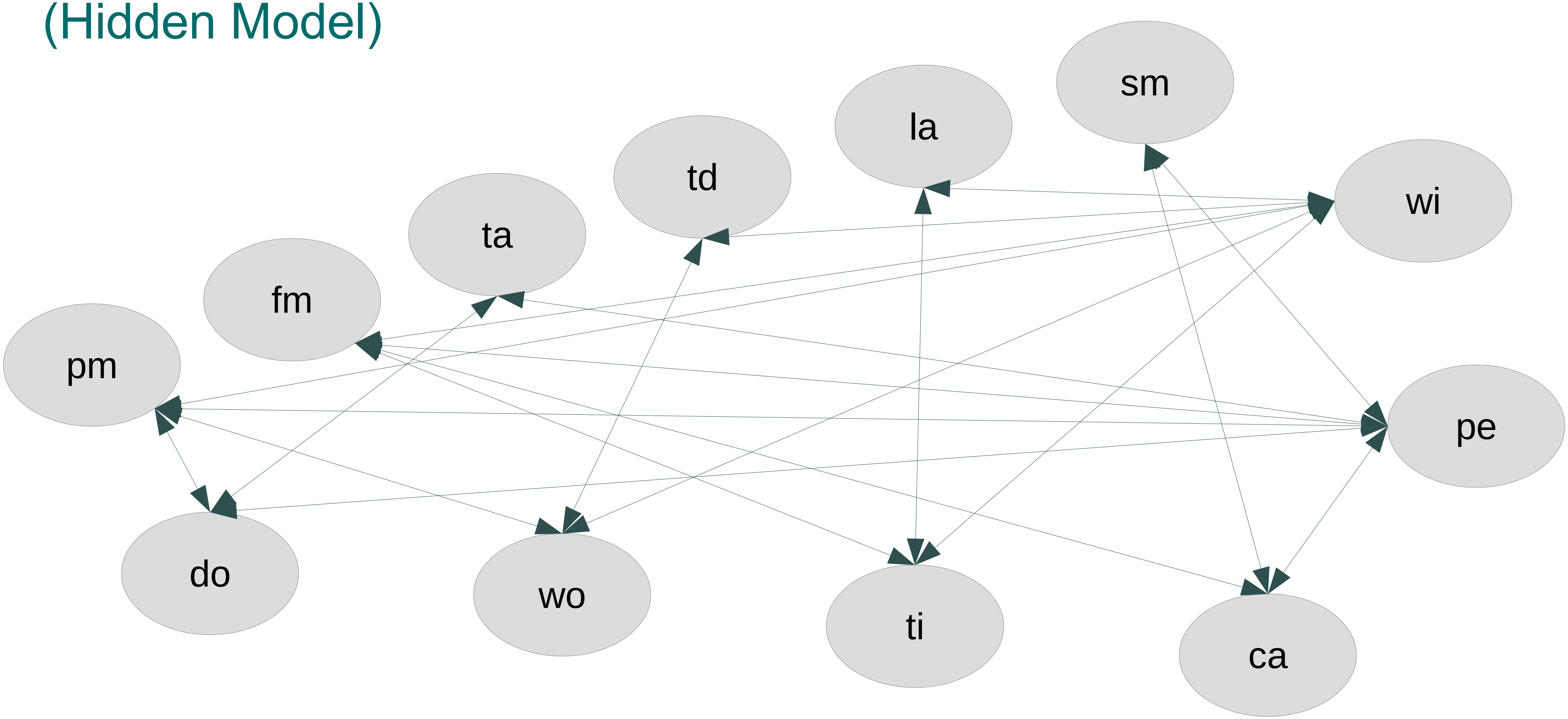
wolf

tiger

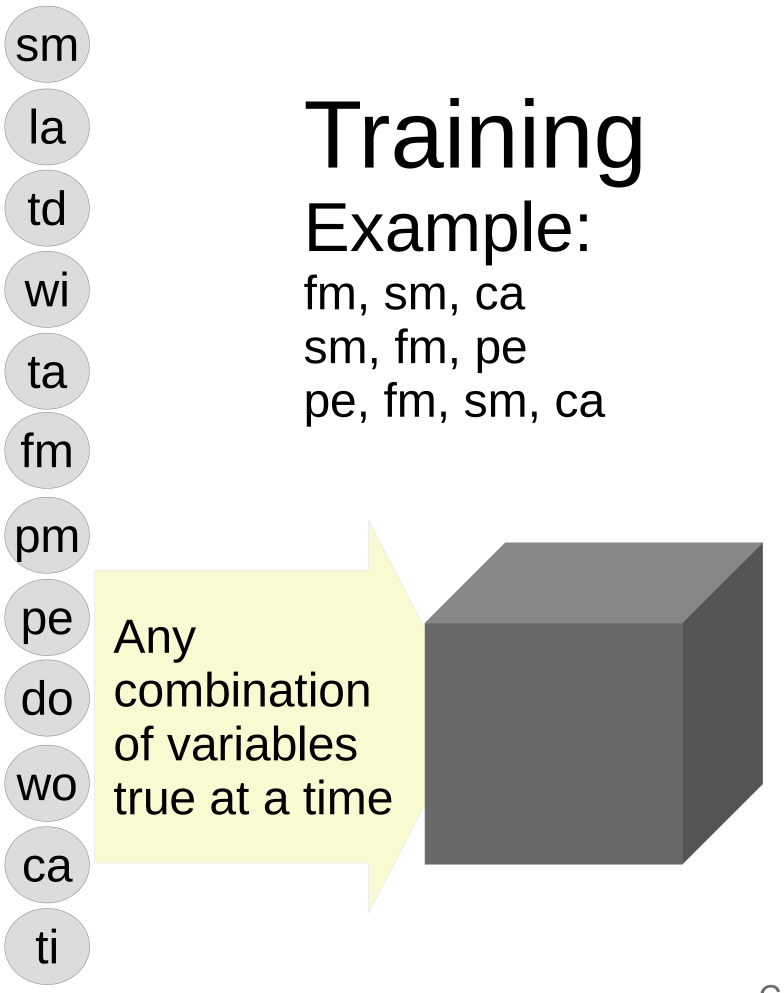
cat

Obfuscated Variables and their True Associations

(Hidden Model)



Training and Testing Environments & Examples

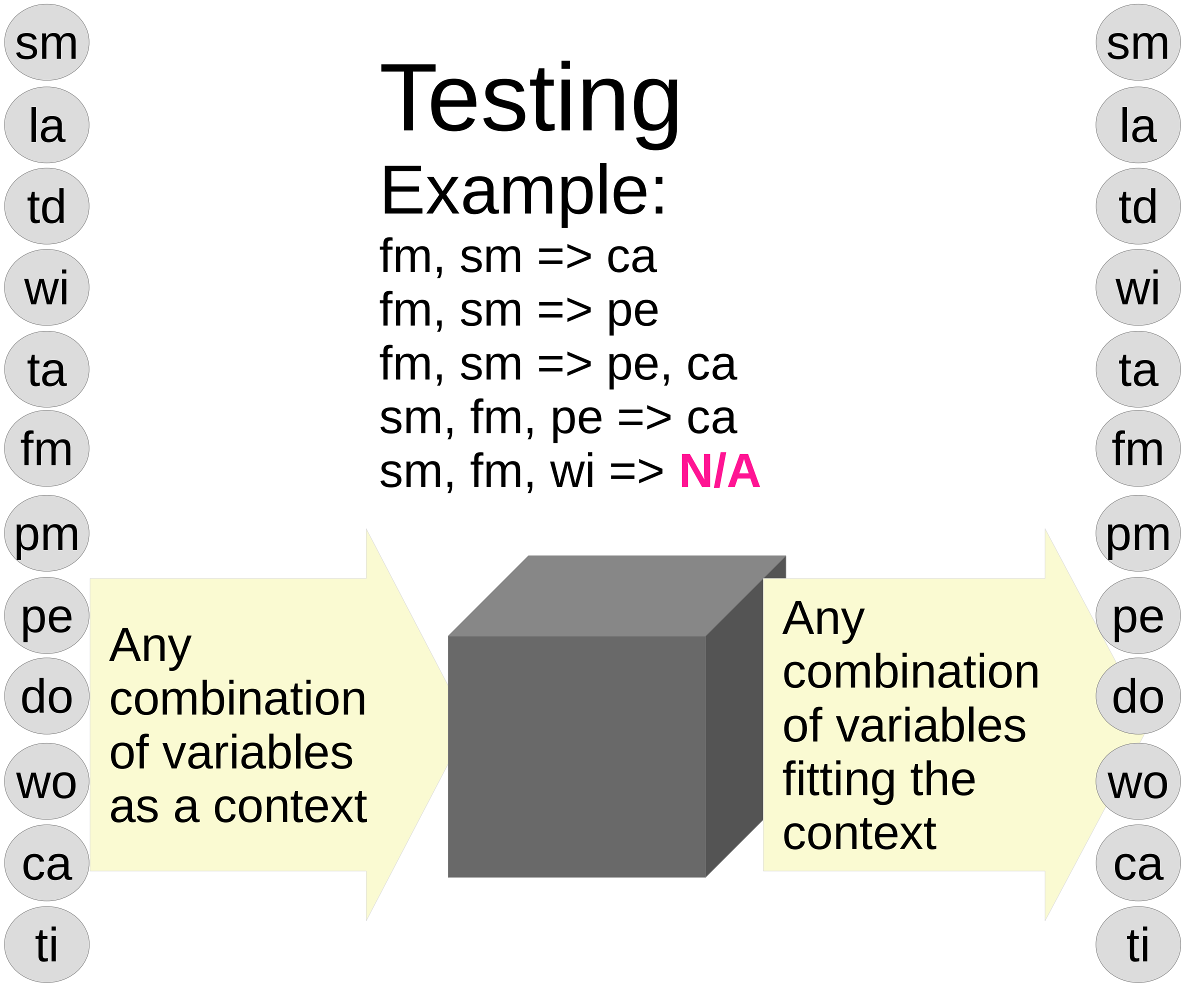


Training

Example:

fm, sm, ca
sm, fm, pe
pe, fm, sm, ca

Any combination of variables true at a time



Testing

Example:

fm, sm => ca
fm, sm => pe
fm, sm => pe, ca
sm, fm, pe => ca
sm, fm, wi => **N/A**

Any combination of variables as a context

Any combination of variables fitting the context

Disclaimers

It is not about:

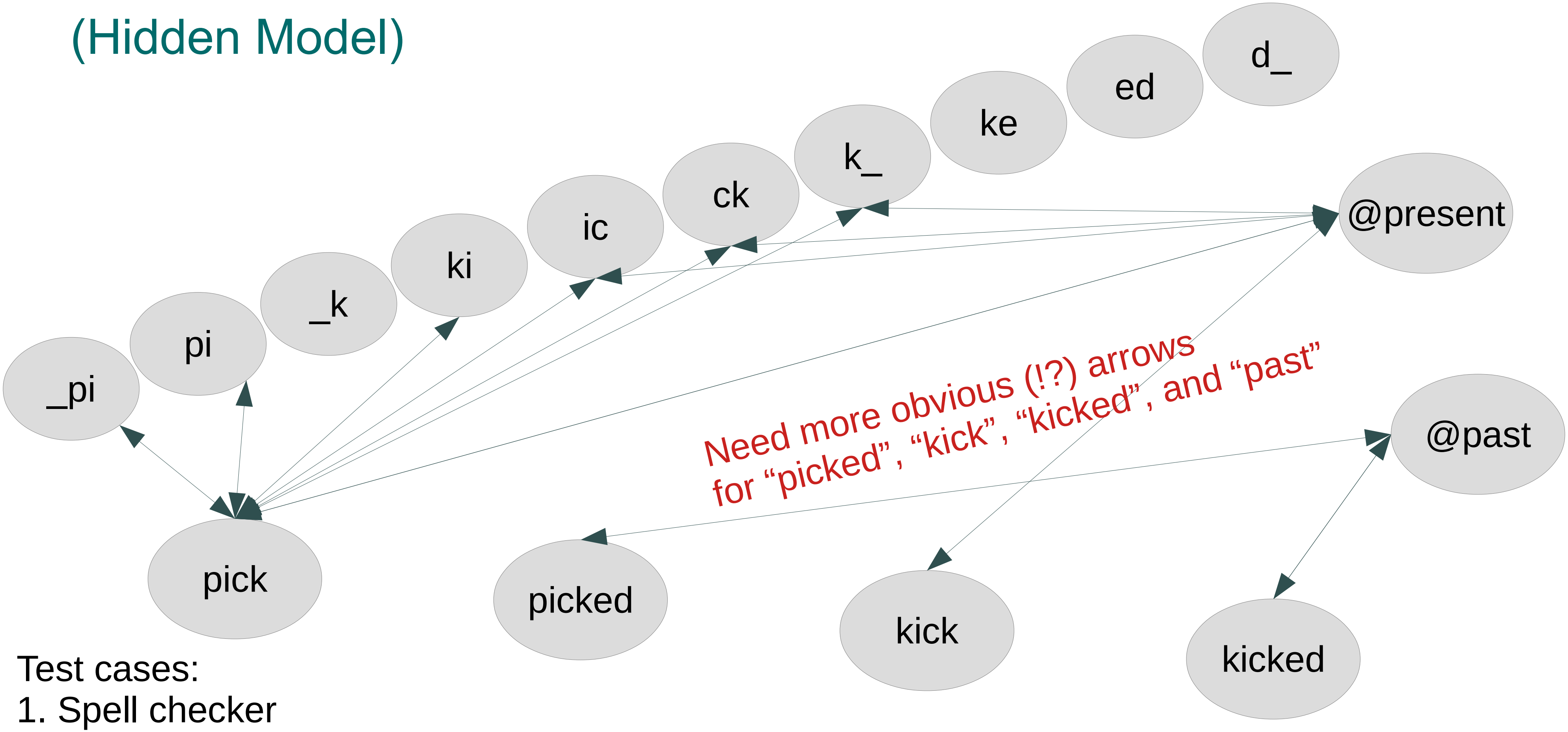
1. neural networks in Python (can use them as solution but the terms are not bound to them);
2. boolean logic [true,false];
3. ternary logic [true,unknown,false];
4. involving negations explicitly at the input/output layers (although negations can be used by the system internally, based on the design);
5. a case where you need to specify a list of the output variables to check when testing (so the outputs may be random or excessive but must comply to the terms), though it may be extended in a future version of the test;
6. fitting the model explicitly, because we unsupervisedly train first and validate the model next;
7. reinforcement learning where we can give a positive/negative feedback, though it may be extended to that in a future version of the test.

It is about:

1. stimuli which can be present or absent, so that if stimuli is present, we do know that there is a cause in the "hidden model" for that but when the stimuli is not present, we just do not know anything;
2. being able to deal with insufficient and uncertain information where decisions should be made given the potential lack of full information;
3. being able to generate different test cases for the same model given alternative "hidden models" still keeping the system's ability to provide correct responses on the previously learned "hidden models";
4. being able to extend the system in the future to deal with conflicting inputs on redundant data and provide probabilistic assessments on the output in such case.

Word-Letter Variables and their True Associations

(Hidden Model)



Need more obvious (!?) arrows for “picked”, “kick”, “kicked”, and “past”

Test cases:
1. Spell checker
2. POS tagger