

%%

## Hierarchical Active Inference for Language

Nested hierarchies of structures using Active inference for language recognition and reading

### Description

Each level of the hierarchy has a default structure if the lowest observable is a letter o(1)=letter a simple hierarchy can be described level 1 two factor: states = [composition of letters (e.g. word), locations of letters in the word] obs = [ letter, locations of letters in the word] than level 2 states = [composition of words (e.g. sentence), locations of words in the sentence] obs = [ word, locations of words in the sentence]

Extending the composition and the meaning of the composition it is possible to iterate this structure on N-levels

It is, also, possible to add classes for each level, that gives a context for each state recognized. For example for level 2 becomes: states = [sentence, locations of words in the sentence, context] obs = [ word, locations of words in the sentence, report]

### Getting started

Type HAI\_LANGUAGE\_pathsLoad; in main the directory to add necessary subpaths. then choose one of the main in directory "MAINS" to see a demonstration of code features

### Code

#### (1) HAI\_LANGUAGE\_DICTIONARY\_v0\_sample\_main.m

simple example with a simple dictionary 2-level hierarchy level 1 states = [ word, locations of letters in the word] obs = [ letter, locations of letters in the word] level 2 states = [sentence, locations of words in the sentence] obs = [ word, locations of words in the sentence]

DICTIONARY: Simple dictionary of 6 words composed by A B C D

#### (2) HAI\_LANGUAGE\_DICTIONARY\_v1\_sample\_main.m

2-level hierarchy level 1 states = [ word, locations of letters in the word] obs = [ letter, locations of letters in the word] level 2 states = [sentence, locations of words in the sentence] obs = [ word, locations of words in the sentence]

DICTIONARY: Simple dictionary of English words (two syllable words of six letters)

#### (3) HAI\_LANGUAGE\_DICTIONARY\_v2\_sample\_main.m

3-level hierarchy level 1 states = [syllable, locations of letters in the syllable] obs = [ letter, locations of letters in the syllable] level 2 states = [ word, locations of syllable in the sentence] obs = [syllable, locations of syllable in the sentence] level 2 states = [sentence, locations of word in the sentence] obs = [ word, locations of word in the sentence]

DICTIONARY: Simple dictionary of English words (two syllable words of six letters)

#### (4) HAI\_LANGUAGE\_DICTIONARY\_v3\_sample\_main.m

2-level hierarchy level 1 states = [syllable, locations of letters in the syllable] obs = [ letter, locations of letters in the syllable] level 2 states = [ word, locations of syllable in the sentence] obs = [syllable, locations of syllable in the sentence]

DICTIONARY: Simple dictionary of English words (two syllable words of six letters)

#### (5) HAI\_LANGUAGE\_BERT\_LOOP\_sample\_01\_main.m, HAI\_LANGUAGE\_BERT\_LOOP\_sample\_02\_main.m

LOOP of HAI Code on a DICTIONARY predicted by BERT [https://it.mathworks.com/matlabcentral/fileexchange/107375-transformer-models?s\\_tid=FX\\_rc3\\_behav](https://it.mathworks.com/matlabcentral/fileexchange/107375-transformer-models?s_tid=FX_rc3_behav) add the corresponding path to use BERT model git clone <https://github.com/matlab-deep-learning/transformer-models>

level 1 states = [ word, locations of letters in the word] obs = [ letter, locations of letters in the word] level 2 states = [sentence, locations of words in the sentence] obs = [ word, locations of words in the sentence]

### Authors and acknowledgment

Francesco Donnaruma [francesco.donnarumma@istc.cnr.it](mailto:francesco.donnarumma@istc.cnr.it)

Mirco Frosolone [mirco.frosolone@istc.cnr.it](mailto:mirco.frosolone@istc.cnr.it)

Giovanni Pezzulo [giovanni.pezzulo@istc.cnr.it](mailto:giovanni.pezzulo@istc.cnr.it)

COgnition iN ActioN Laboratory (CONAN) <https://www.istc.cnr.it/it/group/conan-0>

## License

This code is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 1, or (at your option) any later version.

This code is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. A copy of the GNU General Public License can be obtained from the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.