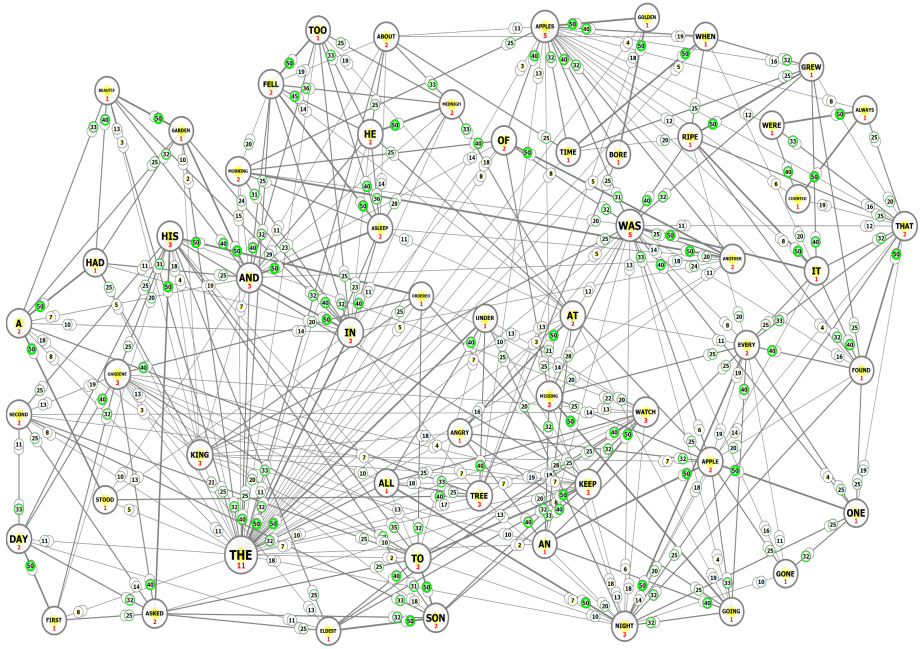
Lumped Mini-Column Associative Knowledge Graphs

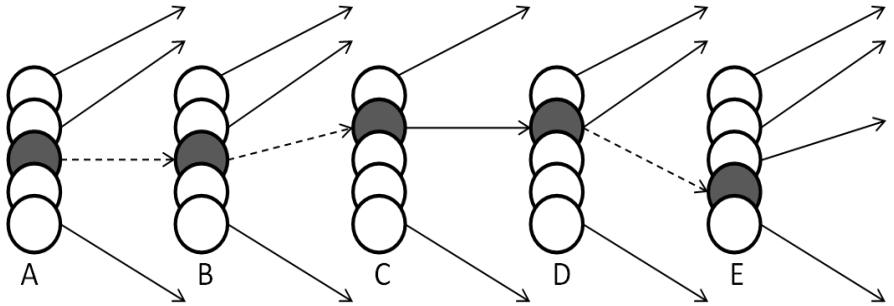
Basawaraj[1], J. A. Starzyk[1,2], and A. Horzyk[3]

[1]School of Electrical Engineering and Computer Science, Ohio University, Athens, OH, USA

[2]University of Information Technology and Management, Rzeszow, Poland

[3]Department of Automatics and Biomedical Engineering,
AGH University of Science and Technology, Krakow, Poland





Content

- Introduction
- Associative Neurons
- Semantic Memory
- Lumped Mini-Column Associative Learning Graph (LUMAKG)
- LUMAKG Organization and Principles
- LUMAKG Algorithm
- Recall Resolution
- Comparative Tests
- Computational Complexity
- Conclusion

Introduction

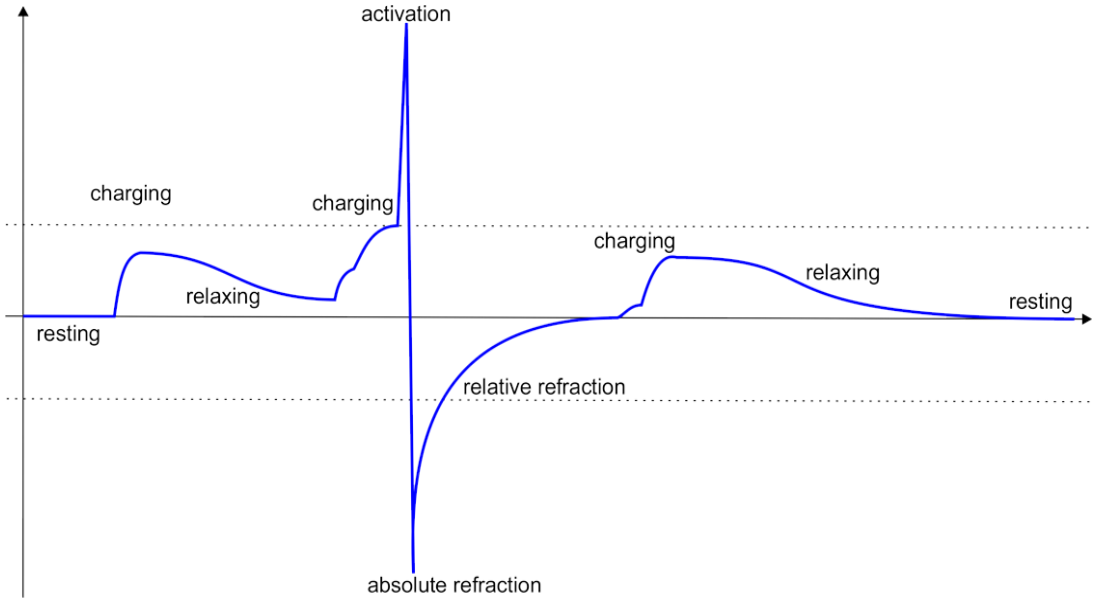
- **Intelligence** is the ability to learn, and to profit, from experience, and adapt to relatively new situations.
- **Intelligence** enables:
  - Acquire and store knowledge
  - Contextually associate knowledge with new information
  - Make generalizations and draw conclusions
  - Recall memories and apply knowledge to solve new problems
- **Intelligence** requires:
  - Sensory and motor capacity
  - Memory which can be semantic or episodic
  - Mechanism to store and associate information to form knowledge
  - Motivation to act and specialize

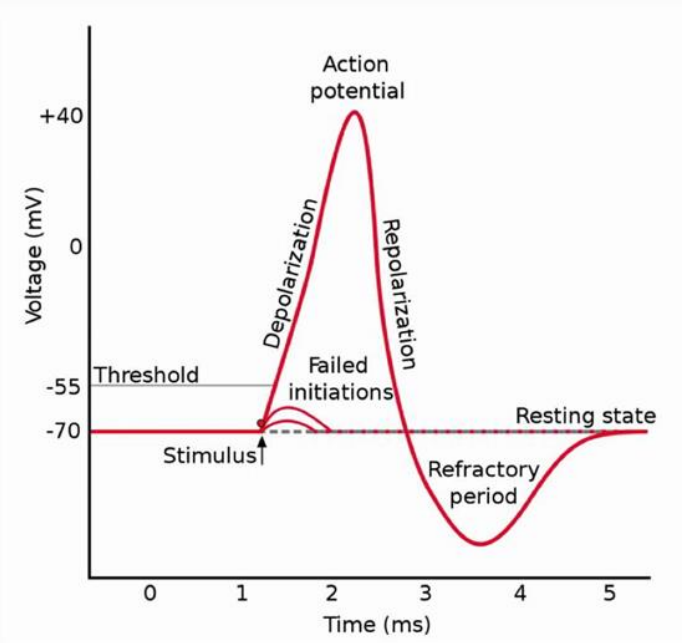
Problems Addressed in this Work

- Problem 1: Associative mechanisms needed to "form" knowledge from the "observed facts" and experiences
  - **Solution**: Use associative neural graphs!
- Why associative neural graphs?
  - Can be used to store and recall sequential memories
  - Form associations between "facts"
- Problem 2: Observations are context dependent, so we also need a mechanism to "store" contexts
  - **Solution**: Use a mini-column structure for representation!
- Why mini-column?
  - Increase contextual knowledge but not confusion

Associative Neuron

- How to model a neuron?
  - Use biological neurons as reference.
  - Why biological neurons? – Efficient and robust.
  - Reproduce the neural plasticity processes.
- Connections between biological neurons are automatically strengthened if frequent activation of one neuron leads to the activation of another one within short intervals, and are weakened if it is triggered after longer delay.





Associative Neuron Excitation Levels

- Biological neurons work continuously but the associative neurons modeled on them can be updated at discrete moments

$$X_{N_i}^{t+\Delta t} = \begin{cases} X_{N_i}^{t} + \left[\sum_{N_m \to N_i} \left(X_{N_m}^{t} \cdot w_{N_m,N_i}\right)\right] \cdot sin\left(\frac{\pi \cdot \Delta t}{2 \cdot \Delta t^{C}}\right), if \ \mathrm{t} < \Delta t \leq t + \Delta \mathrm{t}^{\mathrm{C}} \\ X_{N_i}^{t} \cdot \frac{1}{2} \cdot \left(1 + cos\left(\frac{\pi \cdot \Delta t}{X_{N_i}^{t} \cdot \Delta t^{R}}\right)\right), if \ \mathrm{t} < \Delta t \leq t + \Delta \mathrm{t}^{\mathrm{R}} \\ X_{N_i}^{t} \cdot \frac{1}{2} \cdot \left(1 + cos\left(\frac{\pi \cdot \Delta t}{\left|X_{N_i}^{t}\right| \cdot \Delta t^{F}}\right)\right), if \ \mathrm{t} < \Delta t \leq t + \Delta \mathrm{t}^{\mathrm{F}} \end{cases}$$

$\Delta \mathrm{t}^{\mathrm{C}}$ time to charge

$\Delta \mathrm{t}^{\mathrm{R}}$ time to recover/relax

$\Delta \mathrm{t}^{\mathrm{F}}$ time to refract and return to resting state

Synaptic Weight of Associative Neurons

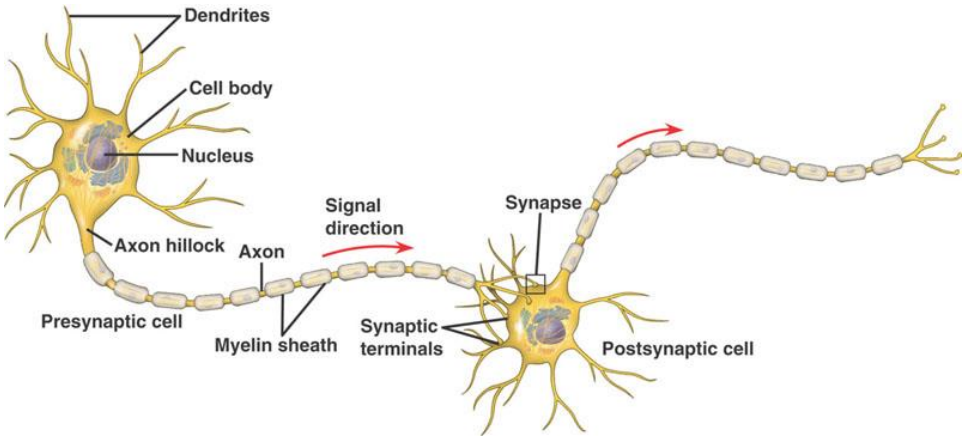
Synaptic weight of associative neurons is defined as $w = b\ c\ m$

b **- behavior factor** that determines synapse influence on postsynaptic neuron

($b = 1$ when influence is excitatory and $b = -1$ when inhibitory),

c **- synaptic permeability** and is computed using

- Linear permeability: $c = \frac{\eta}{2\eta - \delta}$
- Square root permeability: $c = \frac{\sqrt{\eta\delta}}{\sqrt{\eta\delta} + \eta - \delta}$
- Quadratic permeability: $c = \frac{\eta\delta}{\eta\delta + \eta^2 - \delta^2}$
- Proportional permeability: $c = \frac{\delta}{\eta}$
- Power permeability: $c = \left(\frac{\delta}{\eta}\right)^{\frac{1}{k}}$



http://biomedicalengineering.yolasite.com/neurons.php

where

η **number of activations** of a presynaptic neuron during training

δ **synaptic efficiency** computed for this synapse

k>1 is an integer.

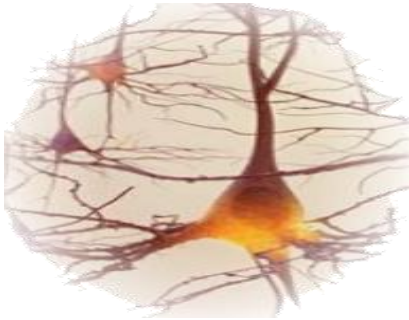
m **- multiplication factor** and is computed using: $m = \frac{\theta_{N_i}^t}{X_{N_i}^{S_1 + \cdots + S_L}} - \frac{x^{LAST}}{2}$

$\theta_{N_i^n}$ is the activation threshold of postsynaptic neuron

x^{LAST} last postsynaptic stimulation level

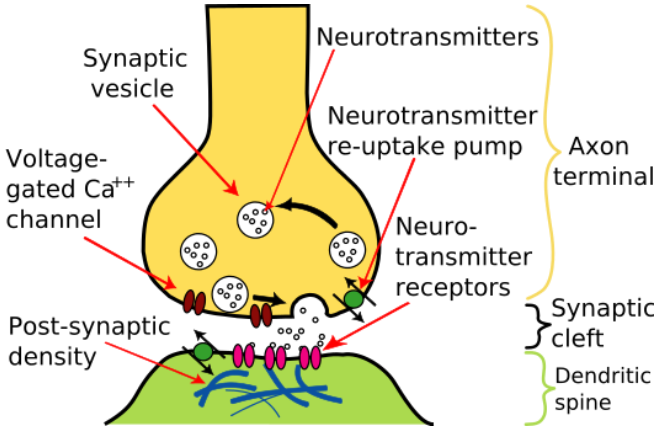
Limitation is $m \leq \theta_{N_i}^t$

Synaptic Efficiency of Associative Neurons



- Presynaptic influence is determined by the **synaptic efficiency** of a synapse between neurons $N_m \rightarrow N_i$ and is defined as:

$$\delta_{N_m,N_i} = \sum_{\{(N_m,N_i)\in S^n \in \mathbb{S}\}} \left(\frac{1}{1 + \frac{\Delta t^A - min(\Delta t^C, \Delta t^A)}{\Delta t^R}} \right)^{\gamma}$$



http://www.wikiwand.com/nl/Synaps

Δt^A time between stimulation of the synapse and activation of a postsynaptic neuron

Δt^C time to charge and activate a postsynaptic neuron after stimulating the synapse

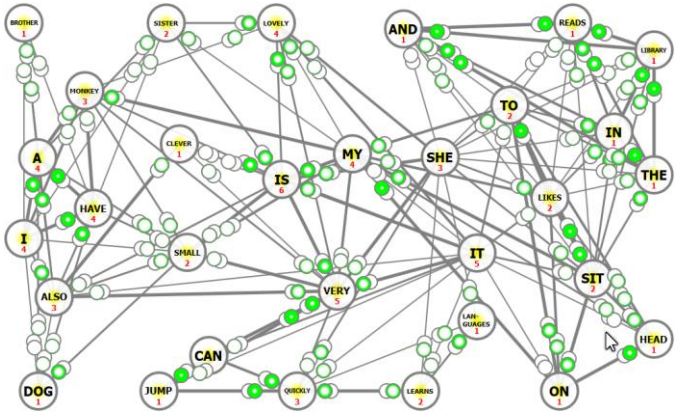
Δt^R period for a postsynaptic neuron to relax and return to its resting state

γ context influence factor changing the influence of the previously activated and connected neurons to the postsynaptic neuron N_i

$\mathrm{S^n}$ training sequence during which activations were observed

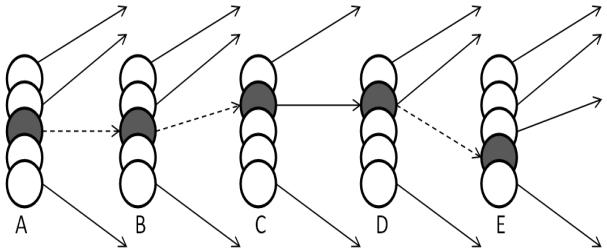
$\mathbb{S}$ the set of all training sequences used for adapting the neural network

Semantic Memory



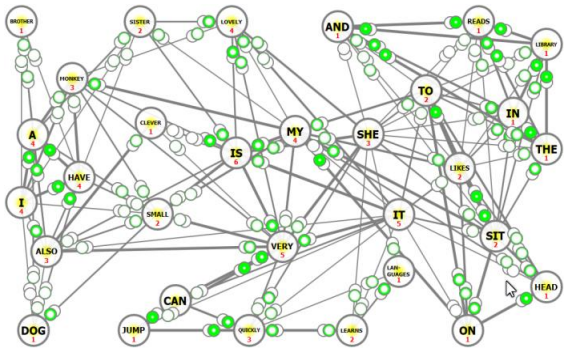
ANAKG

- A structured record of facts, meanings, concepts, and knowledge about the world.
- General factual knowledge, shared with others and independent of personal experience and of the spatio-temporal context in which it was acquired.
- May once have had a personal context, but now stand alone as general knowledge.
  - e.g. types of food, capital cities, social customs, functions of objects, vocabulary, etc.
- Abstract and relational
  - Representation is obtained through symbol grounding, associating sensory data with action and reward obtained by the system in its interaction with the environment.
- Uses an active neuro-associative knowledge graph (ANAKG) – that can represent and associate training sequences of objects or classes of objects.
- Synaptic connections are weighted and each association has its own importance.
- Can provide common sense solutions to new situations that were not experienced before (during the training/adaptation process).



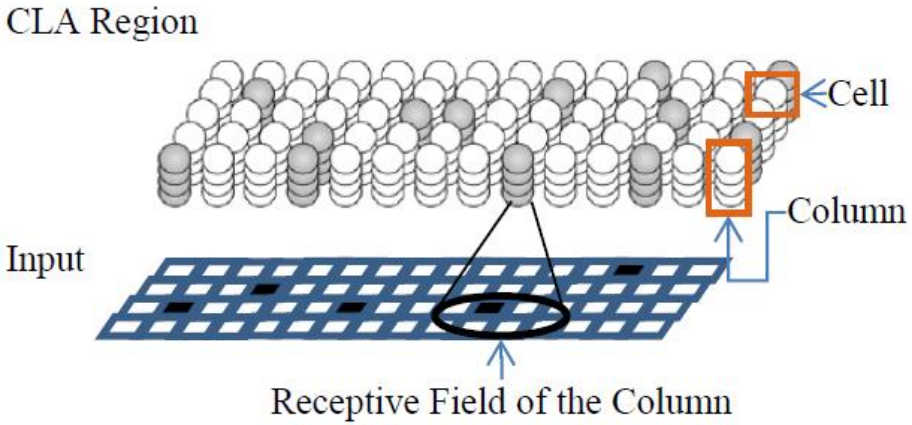
Lumped Mini-Column Associated Knowledge Graph (LUMAKG)

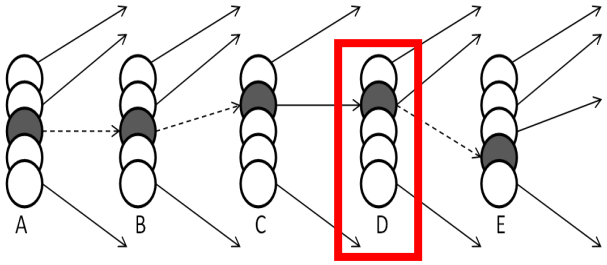
- LUMAKG – Generalization of ANAKG to mini-column form
  - Mini-column structure is better for storing spatio-temporal relations
  - Lower sensitivity to temporal noise than ANAKG



- Active Neural Associative Knowledge Graph (ANAKG) were used to build semantic memory

- Mini-column structure successfully used to build semantic memories
  - Cortical Learning Algorithms for Hierarchical Temporal Learning (HTM), general framework for perceptual learning, by Numenta.
  - HTM – sensitive to temporal noise



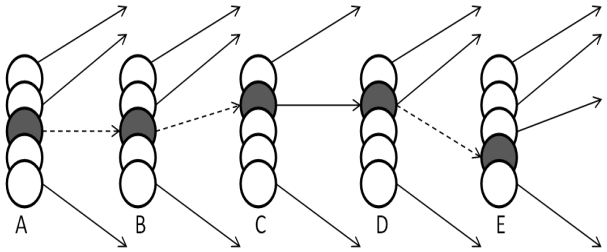


LUMAKG
Organization and Principles

- Symbolic representation was used
  - Each symbol represents a unique word
  - Duplicate each symbol five times to form an individual symbol **mini-column** consisting of five nodes (neurons).

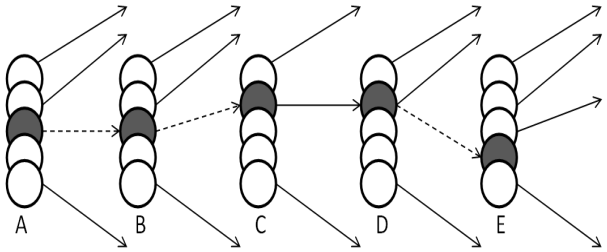
- Mini-column with 5 neurons was used for representation of each symbol
  - External stimulations activate all neurons in the mini-column.
  - Internal stimulations can activate selected neurons and switch them to the predictive mode.
  - Outputs and synaptic connections are different and distributed across the neuron.

- LUMAKG network structure is obtained dynamically
  - New mini-columns added when new symbols are observed.
  - New synaptic connections are added when new relations are observed, else existing connections are suitably modified.



LUMAKG
Activation Principles

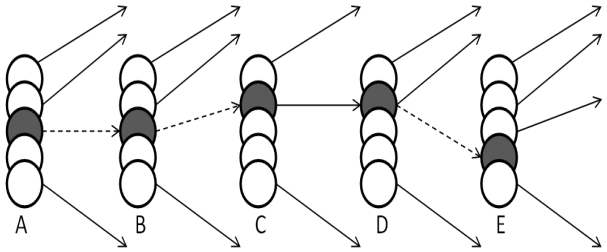
- If a node in a mini-column is **stimulated above the threshold** from associative connections then the node is switched to a **predictive mode**.
- An external input activates either **all nodes in a given mini-column that are in the predictive mode** or **the whole mini-column** if no node is in a predictive mode.
- Activated nodes that were in a predictive mode are in **predicted activation (PA)**.
- An activated mini-column without any node in a predictive mode has all nodes in **unpredicted activation (UA)**.
- **Synaptic connections weights** are changed between activated nodes in predecessor and successor mini-columns.



LUMAKG Algorithm

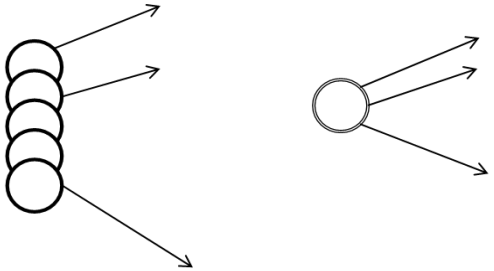
1. Read the consecutive elements of the input sequence to activate corresponding mini-columns
   - Add a new mini-column if the symbol from the input sequence is not represented, all nodes of this new mini-column are in **unpredicted activation (UA)**.

2. Establish predecessor-successor nodes in all activated mini-columns in the input sequence
   - For each consecutive activated mini-column, activate nodes in the mini-column that corresponds to the input symbol (all nodes in the predictive mode or whole mini-column if no node in predictive mode).
   - Find the first mini-column with a PA node and name this first predicted activation (FPA) mini-column
   - If no such column exists choose a node in the last mini-column with a minimum number of outgoing connections and treat it as a PA node.

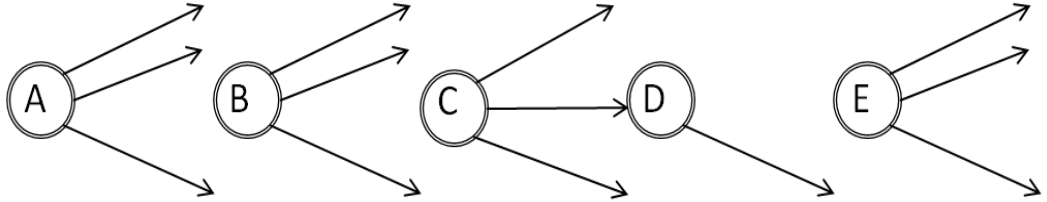


LUMAKG Algorithm

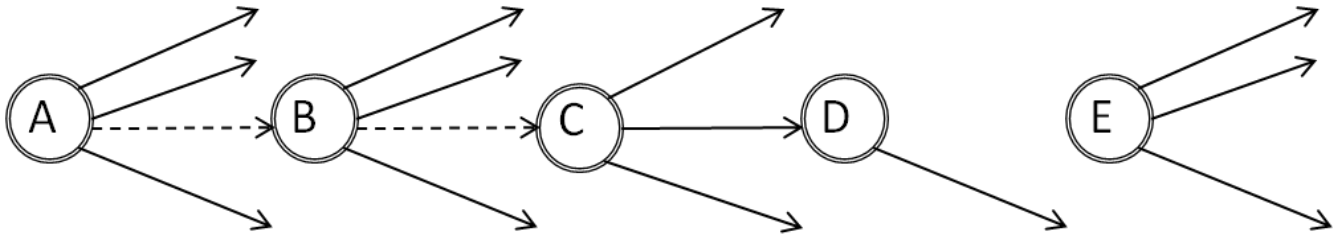
Mini-column and simplified symbol

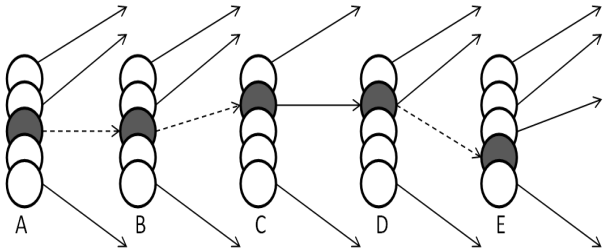


Activated mini-columns with links



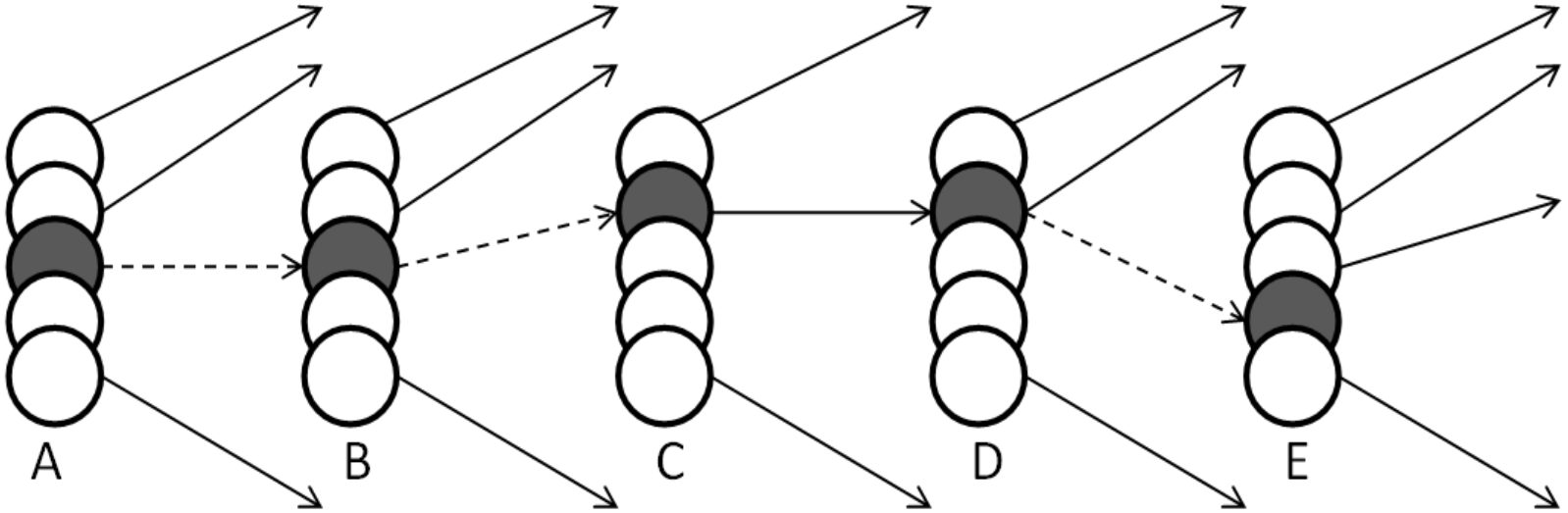
3. Starting from the predecessor mini-column to FPA:
   - Choose a node in this mini-column that has a link to the PA node in FPA and treat it as a PA node.
   - If no such node exists, choose a node with the minimum number of outgoing connections, create a connection to establish a link between the nodes, and treat it as a PA node.
4. Repeat this step for the new PA node until no predecessor mini-column is found.

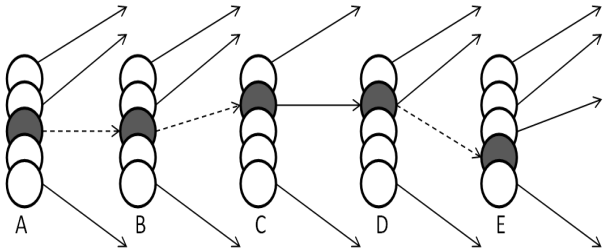




LUMAKG Algorithm

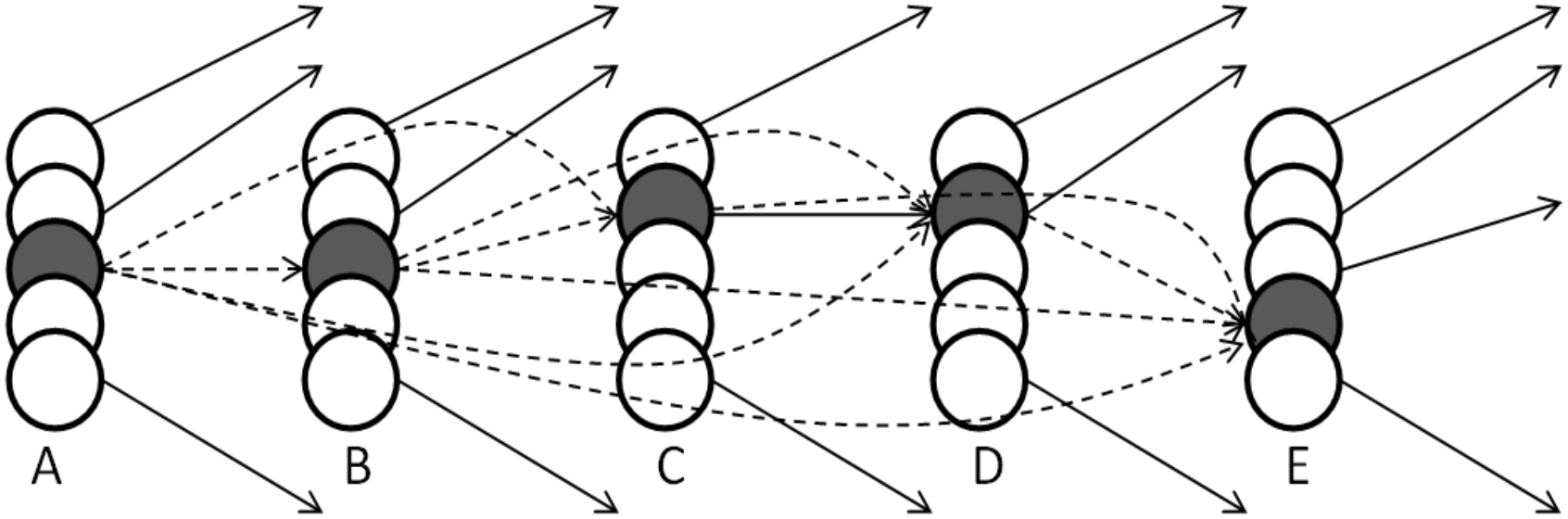
5. Starting from the successor mini-column to FPA repeat the following until no more successor mini-column is found:
   - If the successor mini-column has a PA node,
     link the two PA nodes and move to the successor mini-column.
   - If the successor is in an unpredicted activation (UA) mini-column,
     choose a node in this mini-column with the minimum number of outgoing connections and treat it as a PA node.
     Next, link the two PA nodes and move to the successor mini-column.





LUMAKG Algorithm

6. Update synaptic weights in the synaptic connections between all predecessor and successor nodes:
   - The algorithm updates all the synaptic weights between all PA nodes in predecessor and successor mini-columns according to the ANAKG algorithm.

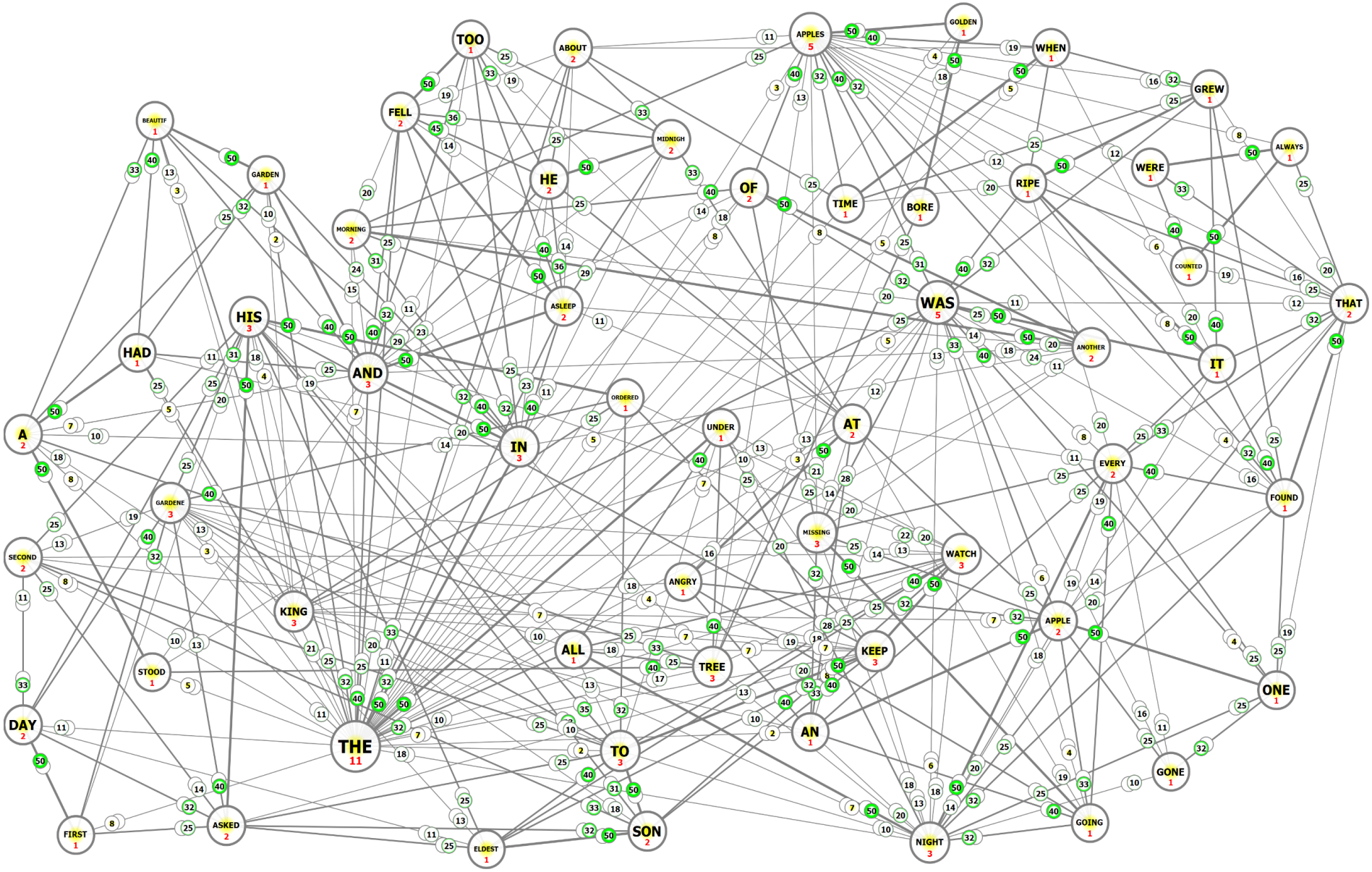


Comparative Testing: LUMAKG vs ANAKG

- Training sequences (taken from Grimm fairies):
  - The king had a beautiful garden, and in the garden stood a tree.
  - The tree bore golden apples, apples that were always counted.
  - About the time when the apples grew ripe, it was found that every night one apple was gone.
  - The king was angry at an apple going missing every night.
  - The king ordered his gardener to keep watch all night under the tree.
  - The first day the gardener asked his eldest son to keep watch.
  - About midnight he fell asleep, and in the morning another of the apples was missing.
  - The second day the gardener asked his second son to keep watch.
  - At midnight he too fell asleep, and in the morning another of the apples was missing.

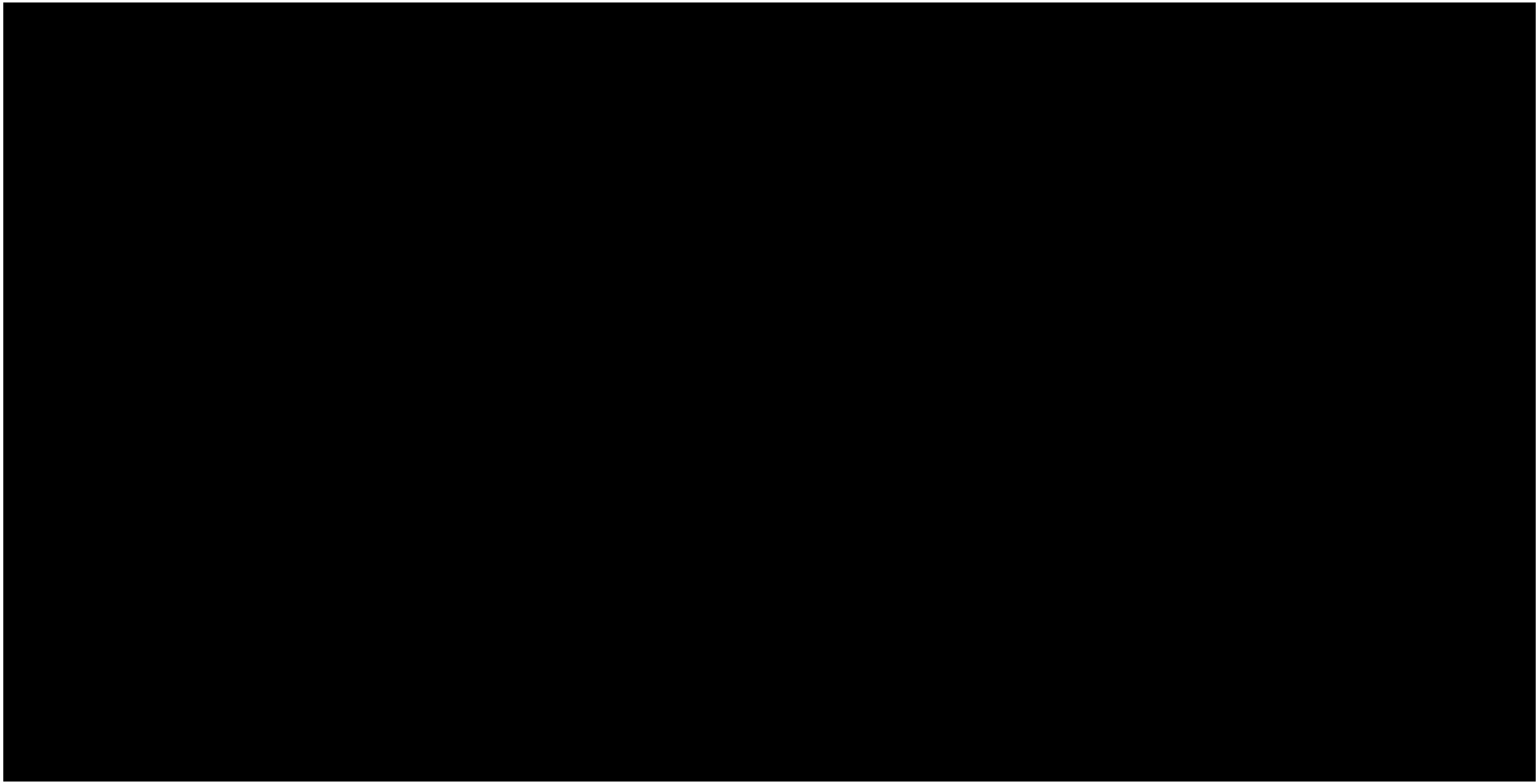
ANAKG NEURAL NETWORK STRUCTURE

- The ANAKG graph structure created for the above-mentioned training data set.



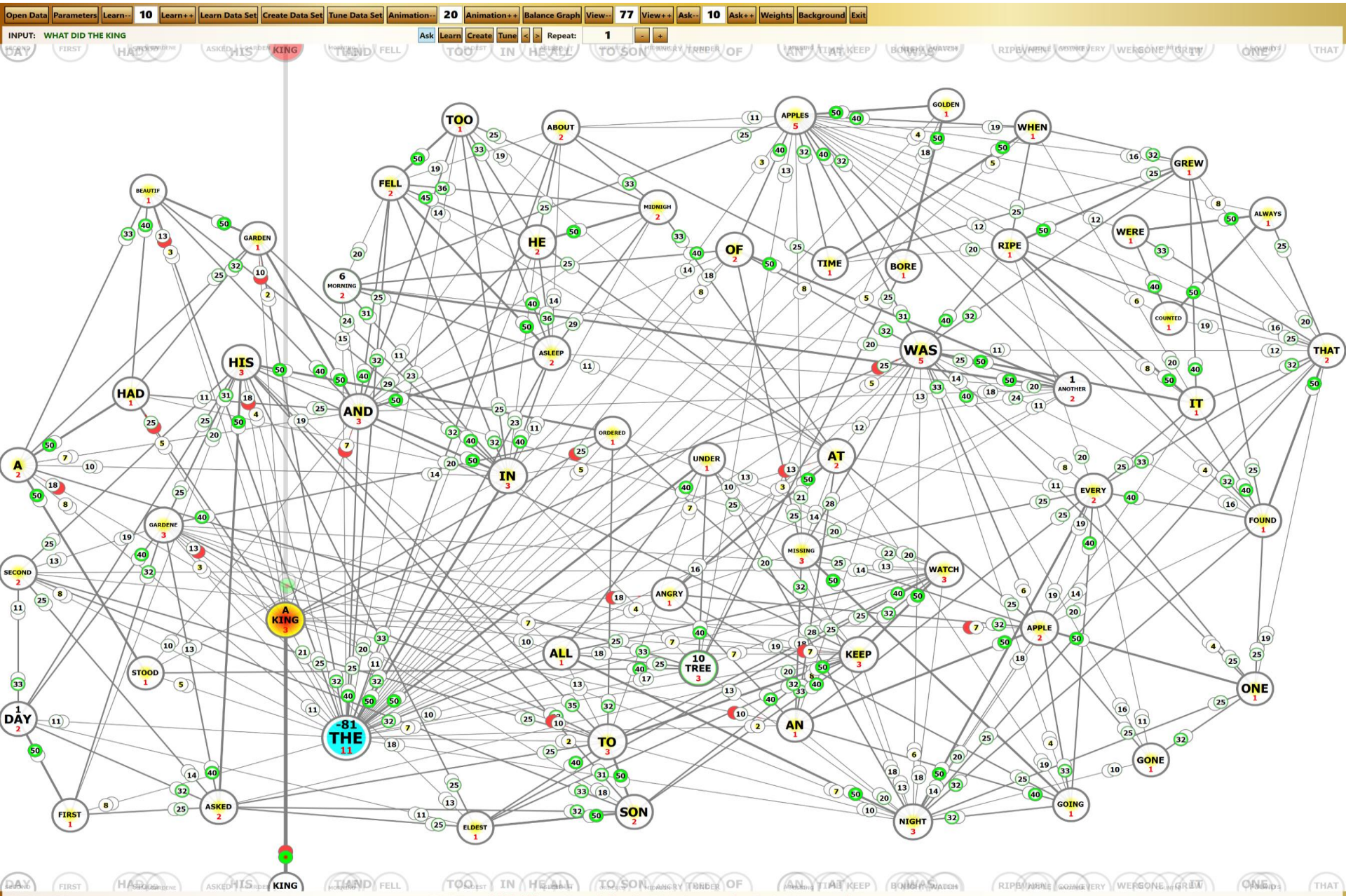
CREATION OF ANAKG NEURAL NETWORK STRUCTURE

- The ANAKG graph structure is gradually developed for the training sequence data set.



- The animation of the slowed down development of the ANAKG for training data set.
- As a result, we achieve the complex graph of neurons contextually connected according to the input context coming from the training sequences, i.e. the sequences of words.

STIMULATED ANAKG NEURAL NETWORK



Recall Resolution – Test I: Small Training Set

- Sequences related to training sequences used to test recall abilities.
- Results show that LUMAKG provides more meaningful answers than ANAKG.

Input sequence	ANAKG memory output	LUMAKG memory output	Desired output
What did the king had?	The king had a beautiful garden	The king had a beautiful garden	The king had a beautiful garden
What stood in the garden?	**Stood a in the tree the garden**	**Stood in the garden stood a tree**	**In the garden stood a tree**
Why was the king angry?	**Was the king angry at an apple missing**	**Was the king was angry at an apple missing**	**The king was angry at an apple going missing**
What were always counted?	Were always counted	Were always counted	Apple that were always counted
What did the king order his gardener?	**the king his gardener**	**The king was his gardener to keep watch**	**The king ordered his gardener to keep watch**
What was missing in the morning?	**Was missing in the morning another of the apple was**	**Was missing in the morning another of the apple was missing**	**In the morning another of the apple was missing**

Recall Resolution – Test II: Larger Training Set

- Test repeated for:
  - 78 training sentences containing over 2500 words, with over 500 unique words

Input sequence	ANAKG memory output	LUMAKG memory output	Desired output
What did the king had?	**What did the king had**	**What did the king had a beautiful garden**	**The king had a beautiful garden**
What stood in the garden?	**What stood in the garden**	**What stood in the garden stood a tree**	**In the garden stood a tree**
Why was the king angry?	**Why should was the king angry at**	**Why should was the king angry at an apple**	**The king was angry at an apple going missing**
What were always counted?	What were always counted	What were always counted	Apple that were always counted
What did the king order his gardener?	**What did the king his gardener**	**What did the king his gardener to keep watch**	**The king ordered his gardener to keep watch**
What was missing in the morning?	**What was missing in the morning**	**What was missing in the morning another of the apple**	**In the morning another of the apple was missing**

Levenshtein Distance Quality Measure

- Levenshtein distance between two string *a* and *b* of lengths *u* and *v* respectively, is given by

$$d_{a,b}(i,j) = \begin{cases} max(i,j) & if\ min(i,j) = 0 \\ min \begin{cases} d_{a,b}(i-1,j) + 1 \\ d_{a,b}(i,j-1) + 1 \\ d_{a,b}(i-1,j-1) \\ d_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & otherwise \end{cases}$$

$d_{a,b}(i,j)$ – distance between first *i* and *j* elements of string *a* and *b*

Input sequence	ANAKG		LUMAKG	
	Test I	Test II	Test I	Test II
What did the king had?	0	5	0	2
What stood in the garden?	6	5	1	2
Why was the king angry?	3	8	2	6
What were always counted?	2	2	2	2
What did the king order his gardener?	4	6	1	3
What was missing in the morning?	3	8	2	5

Recall Resolution – Test I: Small Training Set

- Desired output shares words/symbols with test sequence
  - Consequence of forming grammatically valid sentences
  - Multiple possible grammatically valid responses possible
  - Consider only output symbols not in input sequence

Input sequence	ANAKG memory output	LUMAKG memory output	Desired output
What did the king had?	a beautiful garden	a beautiful garden	a beautiful garden
What stood in the garden?	a tree	a tree	a tree
Why was the king angry?	at an apple missing	at an apple missing	at an apple going missing
What were always counted?			Apple that
What did the king order his gardener?		was to keep watch	to keep watch
What was missing in the morning?	another of apple	another of apple	another of apple

Recall Resolution – Test II: Larger Training Set

Input sequence	ANAKG memory output	LUMAKG memory output	Desired output
What did the king had?		a beautiful garden	a beautiful garden
What stood in the garden?		stood a tree	a tree
Why was the king angry?	should at	should at an apple	at an apple going missing
What were always counted?			Apple that
What did the king order his gardener?		to keep watch	to keep watch
What was missing in the morning?		another of apple	another of apple

Levenshtein Distance Quality Measure

Input sequence	ANAKG		LUMAKG	
	Test I	Test II	Test I	Test II
What did the king had?	0	3	0	0
What stood in the garden?	0	2	0	1
Why was the king angry?	1	5	1	3
What were always counted?	2	2	2	2
What did the king order his gardener?	3	3	1	0
What was missing in the morning?	0	3	0	0

Reciprocal Word Position

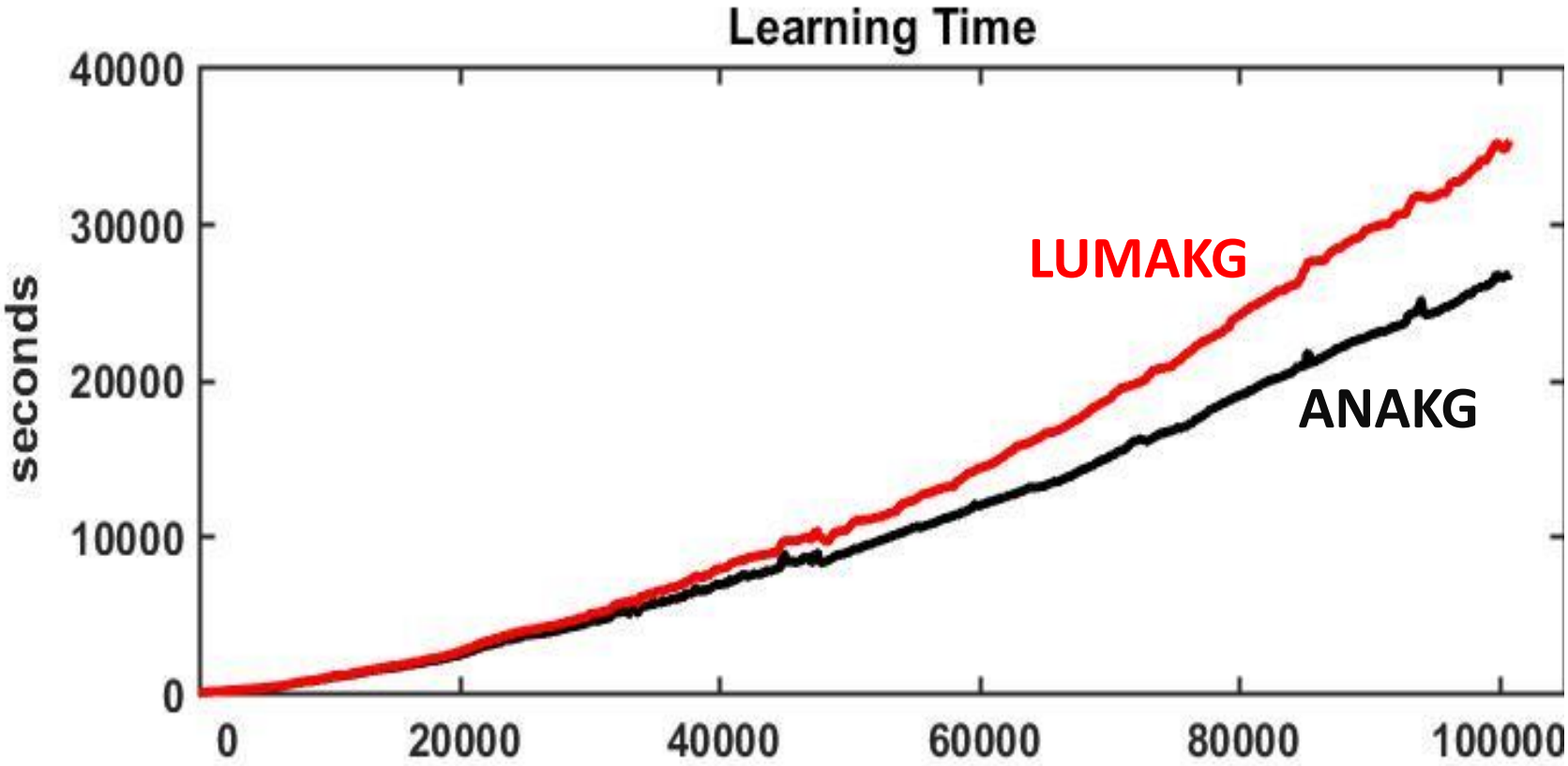
- Reciprocal Word Position (RWP) measures user's effort in extracting the desired response from the output generated by the semantic memory.
- RWP is calculated on the basis of the comparison of the positions of all the words in the desired output to those in the actual memory output:
  - if the positions are the same the word gets a weight of 1;
  - if the positions are different by 'n' words the word gets a weight of 1/(n+1);
  - if a word does not exist it gets a weight of 0.
  - RWP equals to the sum of the weights of all the words in the desired sequence divided by the maximum of the number of words in the desired and actual outputs.
  - RWP has values always between 0 and 1, where 1 means the best result.
  - The higher the value of RWP the better match to the desired output.

Comparison of Reciprocal Word Positions

Input sequence	ANAKG		LUMAKG	
	Test I	Test II	Test I	Test II
What did the king had?	1	0	1	1
What stood in the garden?	1	0	1	1/3
Why was the king angry?	7/10	1/10	7/10	3/10
What were always counted?	0	0	0	0
What did the king order his gardener?	0	0	3/8	1
What was missing in the morning?	1	0	1	1

Computational Complexity

- The third type of tests was performed to determine computational complexity of LUMAKG memory in comparison to ANAKG memory.
- Measured time needed to create the associative memory as a function of the number of objects.
- Computational cost for LUMAKG is between 30-40% higher than for ANAKG.



Conclusions:

- We developed and tested a mini-column based ANAKG structure, called LUMAKG.
- LUMAKG shows better ability to recall sequences stored in this semantic memories than ANAKG to which it was compared.
- Levenshtein distance and RWP measure show that LUMAKG memory has higher capacity and better resolution for a short term memory recall.

Future work:

- Extend LUMAKG to a distributed representation of all symbols stored in the memory, to significantly increase memory storage capacity.
- Use larger training and test data sets to obtain a better assessment of the network properties.
- Study the effect of the varying number of neurons in mini-columns.

Thank you for your attention