BABAR

Knowledge System

Raymond Pierre de Lacaze
rpl@lispnyc.org

Lisp NYC
Tuesday, April 13, 2021

Foreword

Babar is a research project in Artificial Intelligence. The Babar knowledge system is named in homage to John McCarthy who not only invented the LISP programming language but also coined the term “Artificial Intelligence” in 1958.

I had the honor and pleasure of knowing John personally when I worked at the Stanford Research Institute. We had lunch and dinner on several occasions and he was also a keynote speaker at both International Lisp Conferences that I organised and chaired in 2002 and 2003.

One of the last projects John worked on was the design of a programming language called Elephant 2000 which was based on speech acts

I grew up up in a French culture in NYC and Babar the Elephant was a childhood character that I was particularly fond of.

Sadly, John McCarthy past away on October 24, 2011 which was the same year I started working on my knowledge extraction system.

I chose the name “Babar” as a way of honoring the memory of John McCarthy.

Babar Overview

- A research project in Artificial Intelligence both neural and symbolic

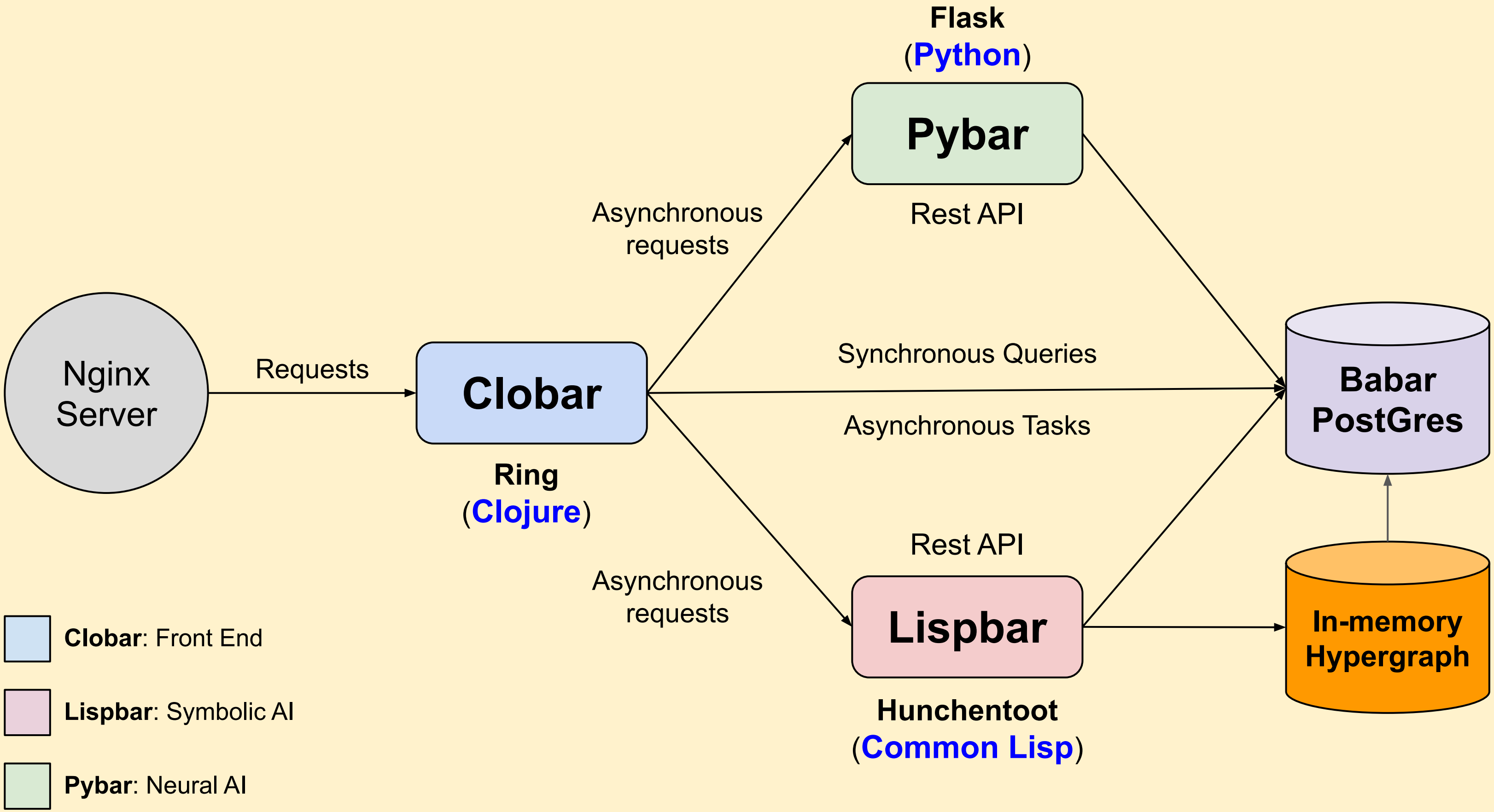
- Multi-language Implementation: CLOS, Python & Clojure

- **Lispbar**: CLOS (Common Lisp Object System)
    - AI Planning M.S. Thesis (1993)
    - In-memory Hypergraph (2008)
    - English Language Parser (2012)
    - Unsupervised Learning (2014)

- **Pybar**: Python 3.8
    - Web crawling & scraping
    - Database maintenance
    - Deep Learning

- **Clobar**: Clojure
    - Web Development
    - Server side HTML
    - CSS & Javascript
    - *Ultimately Clojurescript*

Babar Server Architecture



Presentation Overview

- Web Development
    - Clojure & Hiccup (Clobar)
    - Javascript Interface (Clobar)

- Neural NLP
    - Word Embeddings (Pybar)
    - Neural Network Classifiers (Pybar)

- Symbolic NLP
    - English Language Parser (Lipsbar)
    - Symbolic Machine Learning (Lispbar)

- Knowledge Representation
    - Clausal Form Logic (Lispbar)
    - Semantic Nets (Lispbar)

      Unsupervised Learning (Lispbar)
    - Inductive Logic Programming (LispBar)

- Applications
    - Browsing Wikipedia (Clobar)
    - Classifying Search Engine Results (Clobar)

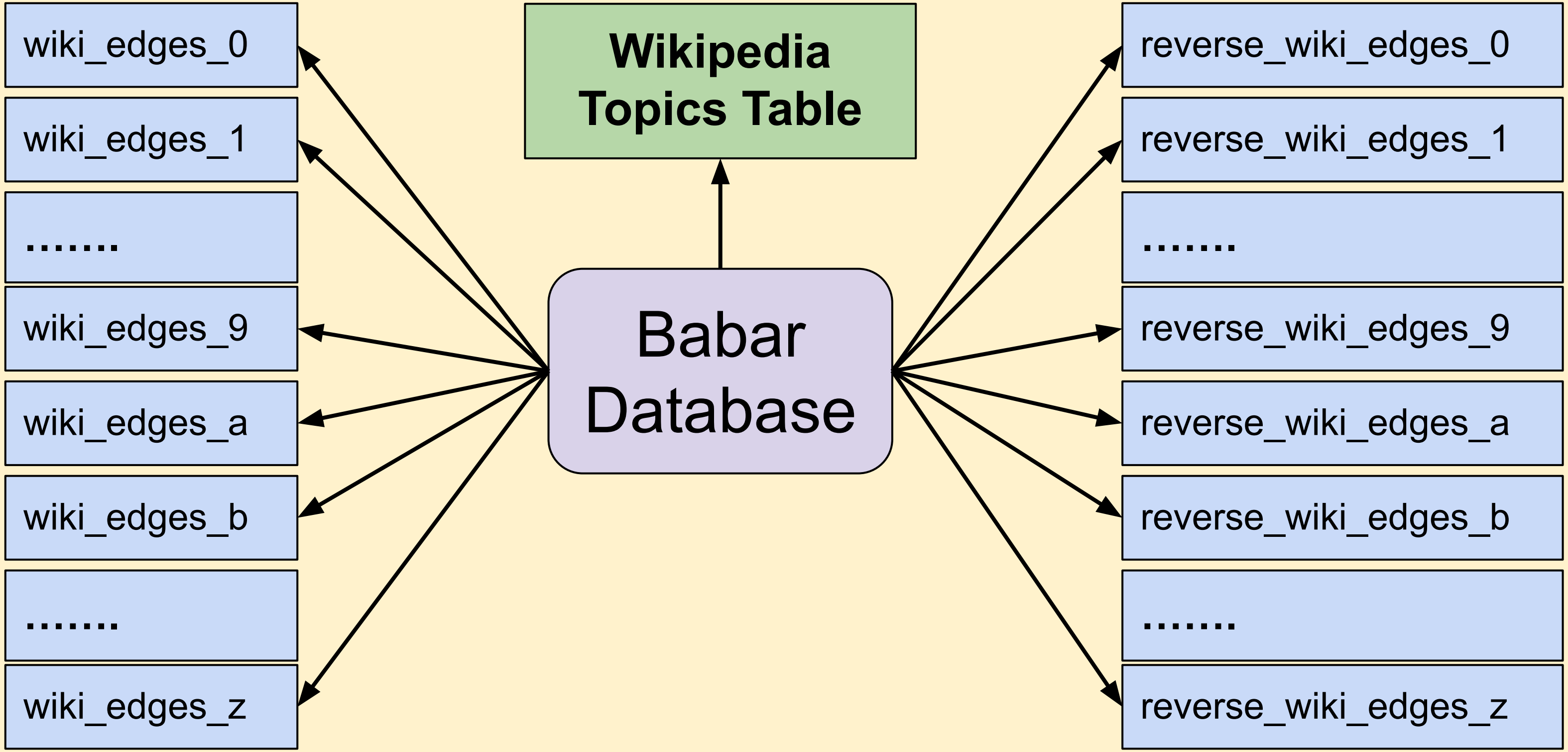
A few Notes About Wikipedia...

- The “De Facto” Online Source of Universal Knowledge
  - Fifth most visited website in the world (6.1 billion visitors per month)
  - Mostly objective and trustworthy knowledge
  - Implicit bias wrt to subject popularity
  - Used to train GTP-3 (largest NLP model to date)

- Data vs. Knowledge
  - Knowledge as the structured organisation of data
  - Wikipedia content is for human consumption
  - Scraping Wikipedia content produces data
  - Wikipedia graph represents knowledge

- Principle source of Babar’s Knowledge System
  - Wikipedia Topics: 5,155,052
  - Wikipedia Edges: 96,260,661

Wikipedia Graph Postgres Implementation



wiki_edges_0

wiki_edges_1

....

wiki_edges_9

wiki_edges_a

wiki_edges_b

....

wiki_edges_z

Wikipedia Topics Table

Babar Database

reverse_wiki_edges_0

reverse_wiki_edges_1

....

reverse_wiki_edges_9

reverse_wiki_edges_a

reverse_wiki_edges_b

....

reverse_wiki_edges_z

CLOBAR: Graphical User Interface (GUI)

- Current System
  - Implemented in Clojure
  - Generates server side html
  - Uses Ring as the web applications library
  - Uses Hiccup for HTML
  - Uses Compojure for routing
  - Uses futures for asynchronous requests
  - Uses CSS & Javascript
  - Currently not using Node, Angular or Ajax

- Future System
  - Implemented in ClojureScript
  - Use Re-Frame & Reagent
  - MPSPA (multi-page single-page-application)

HTML Generation Example

```
(defn layout-page [title & content-section]

 (html5
  [:head
     [:meta {:content "text/html;charset=utf-8"}]
     [:title title]

     (include-css "css/grid.css")
     (include-css "css/clobar.css")

     (include-js "js/navbar.js")

   [:header (babar-navbar-section)][:br]]


  [:body {:style "background-color:lightblue"}
     (babar-logo-section)
     content-section]))
```

Interfacing Clojure & Javascript: Pie Charts

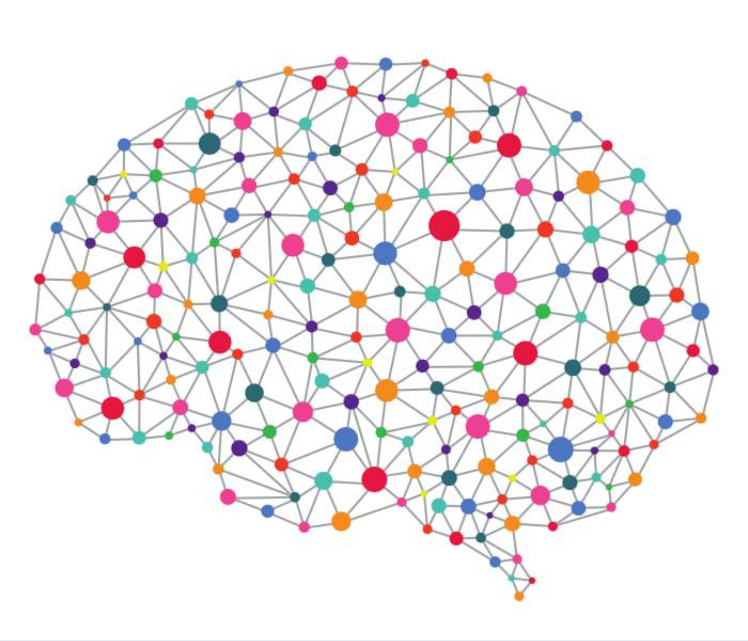
```
(defn select-pie-edges-data [rows]
  (let [erows (filter (fn [x](str/includes? (:name x) "Edges") rows)
        frows  (filter (fn [x](> (:count x) 2000000)) erows)
        labels (map data-label frows)
        data (map :count frows)
        js-data (apply str (map (fn [x y](str x ":" y ",")) labels data))]
    (str "var data = {" js-data "};")))

(defn pie-chart-section [rows]
  (let [js-str (select-pie-edges-data rows)]
    [:div {:class "row-2"}
      [:h2 "Edge Tables Distribution (96,260,661)"]
      [:script  {:type "text/javascript"} js-str]
      [:div {:id "my_dataviz"}
      [:script {:src "js/pie-chart.js"}]]]))
```

Topic Graphs

- Wikipedia as Universal Knowledge
  - 56 Million Wikipedia pages
  - Comparable to all the books in a library
  - Fifth most visited website in the world

- Topic Graph
  - Depth three crawl of a Wikipedia topic
  - By definition a subgraph of Wikipedia
  - Quickly access relevant knowledge
  - More computationally plausible
  - Provide a contextual frame of reference
  - Interactively create topic graphs
  - Persisted in Postgres



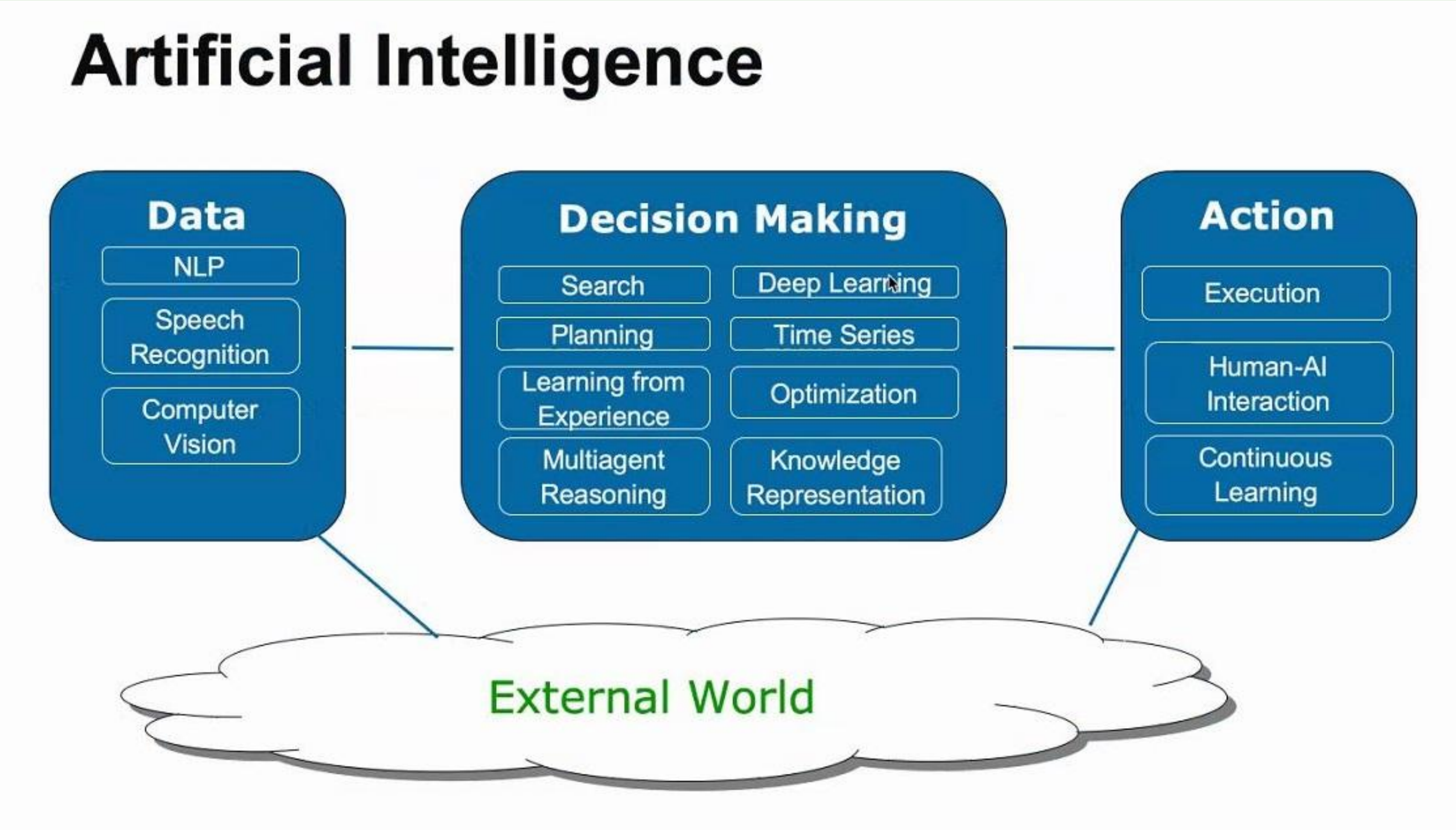
PYBAR

- Web crawling & Scraping

- Word Embeddings
  - Automatically generated
  - Provided by Pybar API
  - Gensim: Google’s Word2Vec

- Training Data
  - Automatically generated from Wikipedia
  - Based on prior human classification (i.e. knowledge)

- Neural Net Classifiers
  - Automatically generated
  - Provided by Pybar API
  - Uses Keras and Tensorflow open source libraries
  - LSTM Based architecture
  - Future: BERT (Bidirectional Encoding Representation from Transformers)



Artificial Intelligence

Data

- NLP
- Speech Recognition
- Computer Vision

Decision Making

- Search
- Deep Learning
- Planning
- Time Series
- Learning from Experience
- Optimization
- Multiagent Reasoning
- Knowledge Representation

Action

- Execution
- Human-AI Interaction
- Continuous Learning

External World

Take away: AI is much more extensive than just Deep Learning

Word Embeddings

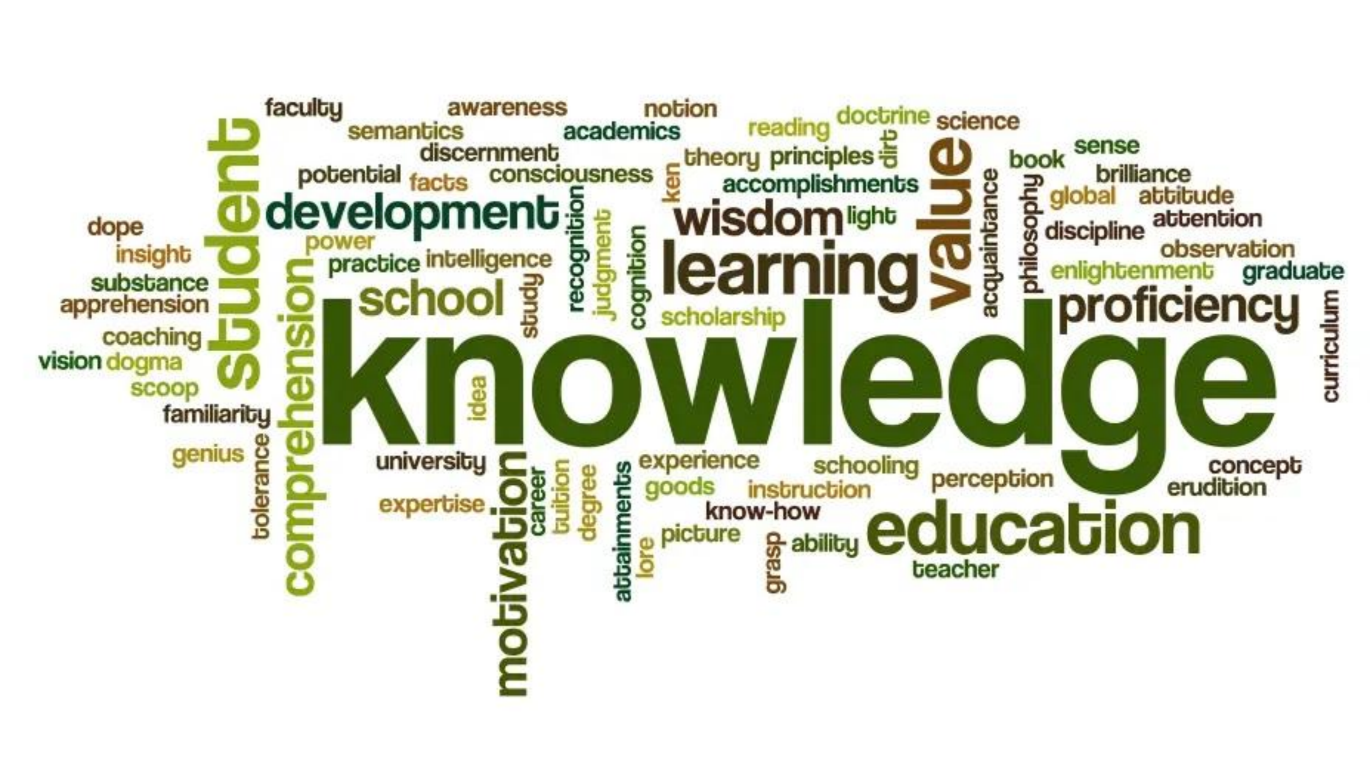
- Vectorization of words from a corpus
  - 1-Hot Encoding
  - TF-IDF Encoding
  - Word2Vec Encoding (Google)
  - Glove (Stanford)

- Spatially capture semantic similarity
  - Both distance and direction

- Babar uses Gensim
  - Gensim is popular implementation of Word2Vec
  - Interactively create embeddings from Wikipedia

- Typically first layer in an LSTM or Transformer based architecture

Word Embedding Demo



Create Word Embedding Example

Name: **Art-Science**
Topics: ['art’, ‘science']

Processing create embedding request:
Computing topic sentences…

Processing topic: Art

Related topics count: 5,910

Total Art related sentences: 378,290

Processing topic: Science

Related topics count: 5,782

Total Science related sentences: 310,797

Saving training data to:
C:\babar\data\training\Art-Science-training-data.csv

Computing word embeddings..

Saving Gensim weights to:
C:\babar\pybar\models\art_science_gensim_weights.npz

Saving Gensim vocabulary to:
C:\babar\pybar\models\art_science_gensim_vocab.json
Vocabulary Size: 68,221

Sending word embedding confirmation email to:
delaray@hotmail.com

Finished processing request.

Example Word Comparisons`

- Animals Word Embedding
- Topics: Mammal, Reptile, Bird
- Total sentences: 69,570
- Vocabulary size: 17,438

Word 1	Word 2	Value
marsupial	placental	**93.36%**
snake	lizard	**92.94%**
zebra	lizard	74.21%
lizards	nest	58.91%
birds	nest	**76.51%**
birds	bird	**69.39%**

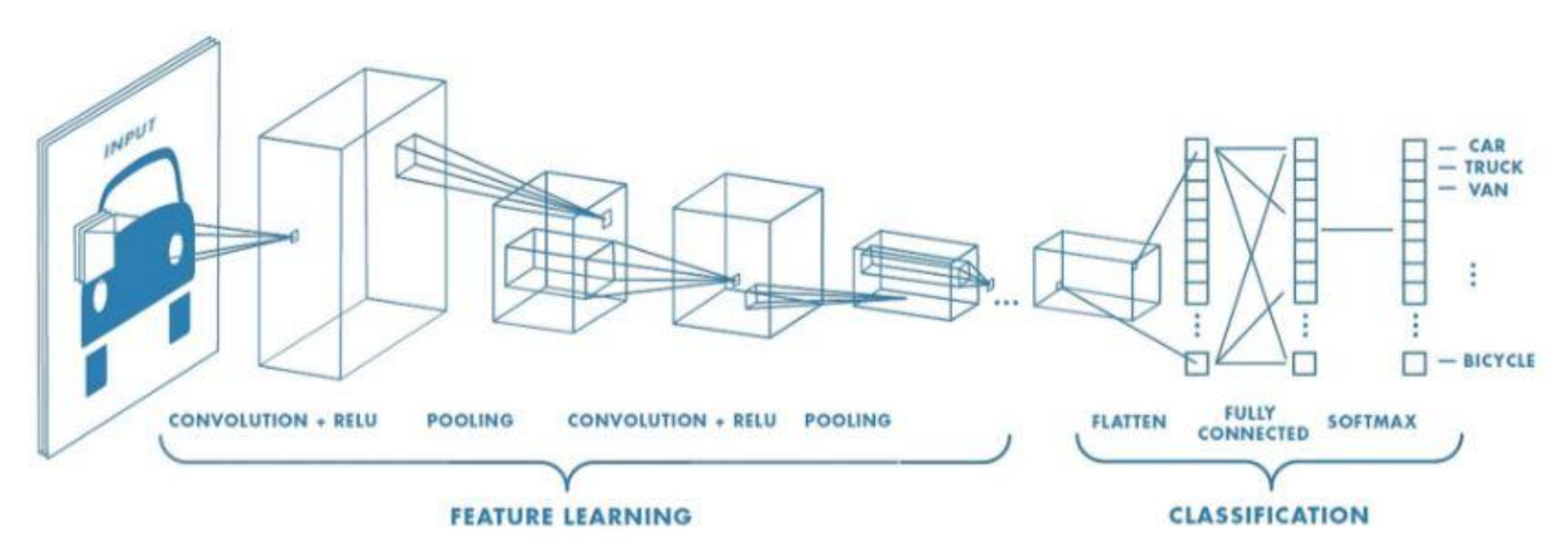
Artificial Neural Networks (ANN)

- Tremendous amount of success
  - Alexnet CNN, Krizhevsky, Sutskever and Hinton(2012)
  - ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - Object Identification & Facial Recognition
  - Language translation & generation
  - Deepmind's AlphaGo (beat Lee Sedol)
  - Deepmind's AlphaFold (3D models of proteins)
  - OpenAI’s GPT-3 (largest language model to date)

- Many types of Neural Nets
  - Convolutional Neural Networks (CNN): Computer Vision
  - Recurrent Neural Networks (RNN): Natural Language
  - Generative Adversarial Networks (GAN): Computer Vision
  - Encoder/Decoder Networks (Transformers): NLP and Image processing

- Goal of Babar
  - Leverage both Neural & Symbolic AI

Feature Learning and Classification in Deep Learning



Take away: First learn high level features, then train a neural net using those features as input.

Babar Uses Long Short-Term Memory (LSTM)

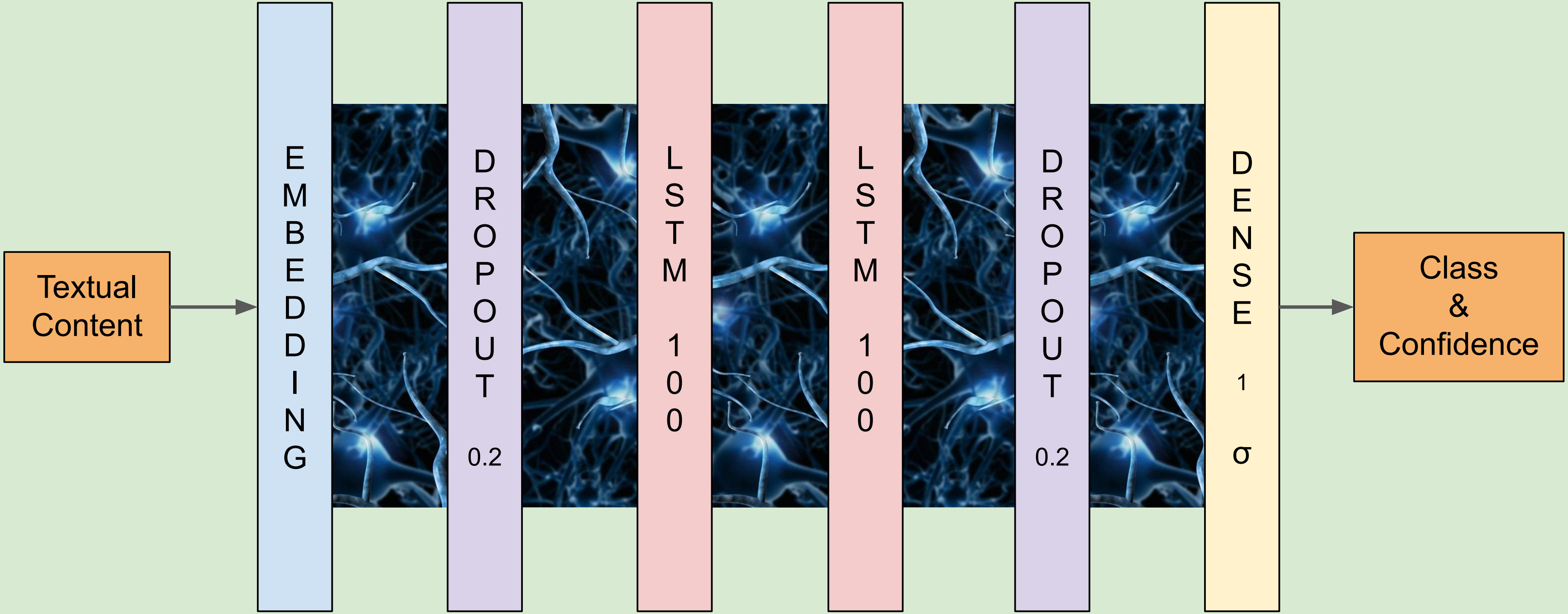
- Type of Recurrent Neural Network (RNN)
  - Based on gated neural nets
  - Trained used Backpropagation Algorithm

- Solve vanishing and exploding gradient issue
  - Problem multi-layered networks

- Well suited to handle sequences of inputs
  - Textual Language
  - Speech Processing

- Becoming somewhat obsolete
  - Transformers (Attention mechanism)
  - BERT (Bidirectional Encoder Representation from Transformers)
  - GPT-3 (OpenAI): Largest language model to date.
    - 175 billion machine learning parameters (variables)

Babar LSTM Classifier Based Architecture



Art/Science Classifier Demo

Classifier Prediction Example: Art vs. Science

```
In [93]: classifier = cl.load_category_classifier("Art-Science")


In [94]: s1 = "John went to the Louvre museum for the Monet exhibit."

In [95]: pr.predict_category(classifier, s1)

Out[95]: ('Art', 0.99781513)


In [96]: s2 = "The technology in computers uses advanced circuits."

In [97]: pr.predict_category(classifier, s2)

Out[97]: ('Science', 0.7981771)
```

Prediction Ambiguity: Art vs. Science

```
In [21]: classifier = cl.load_category_classifier("Art-Science")

In [22]: s = "Lucy went to the Zoo with her dad to see the elephants.”


In [23]: pr.predict_category(classifier, s)

Out[23]: (‘Art’, 0.58)
```

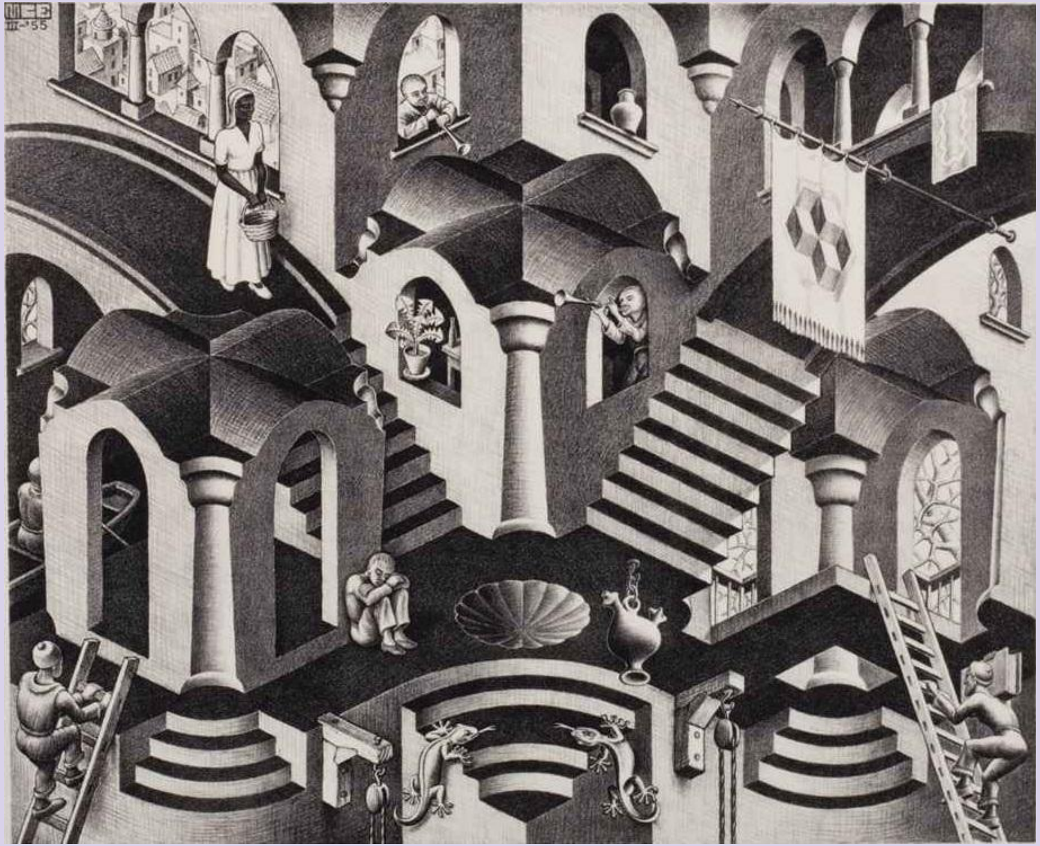
- Importance of context
  - Maybe Lucy wanted to paint various exotic animals (Art)
  - Maybe Lucy wanted to work on her science project (Science)

A Moment of Discontinuity

BRAINS

Neurons

Python



MINDS

Thoughts

LISP

LISPBAR (Symbolic AI)

- Knowledge Representation
  - Clausal Form Logic
- English Language Parser
  - Layered Design: Simple, Generic, Domain, English
  - Symbolic Machine Learning
- Semantic Nets
  - In-memory hypergraph
  - Implemented Sparse matrices
  - Knowledge Base
- Clustering
  - Unsupervised Learning
- Inductive Logic Programming (ILP)
  - *FOIL: Learning sets of Horn clauses*

Knowledge Representation (KR)

- Wikipedia is Knowledge for humans
- Need for a language for representation
- Mathematics is the language of Science
- E.g. Ideal Gas Law: PV=nRT

- Logics for Artificial Intelligence
  - Propositional Calculus (PC)
  - First Order Predicate Calculus (FOPC)
  - First order Logic
    - **∃ x ∀ y | x+y=y** *(existence of a neutral element)*
    - **∀ x ∃ y | x+y=0** *(existence of a inverse element)*
  - Non-monotonic Reasoning
  - Modal Logics *(Believes Jane (Stole John money))*

- Clausal Form Logic (CFL)
  - First Order Logics: (<predicate> <atomic-1> <atomic-2>...)
    - (mother Jane Peter)
    - (grandparent ?x ?y) ⇐ (parent ?x ?z)(parent ?z ?y)
  - Second Order Logics
    - (told Jeff Pierre (believes Zoe (ISA Arthur party-animal)))
  - HORN Clauses
    - Disjunction of conjuncts with at most one positive literal

Babar Knowledge Extraction *(WIP)*

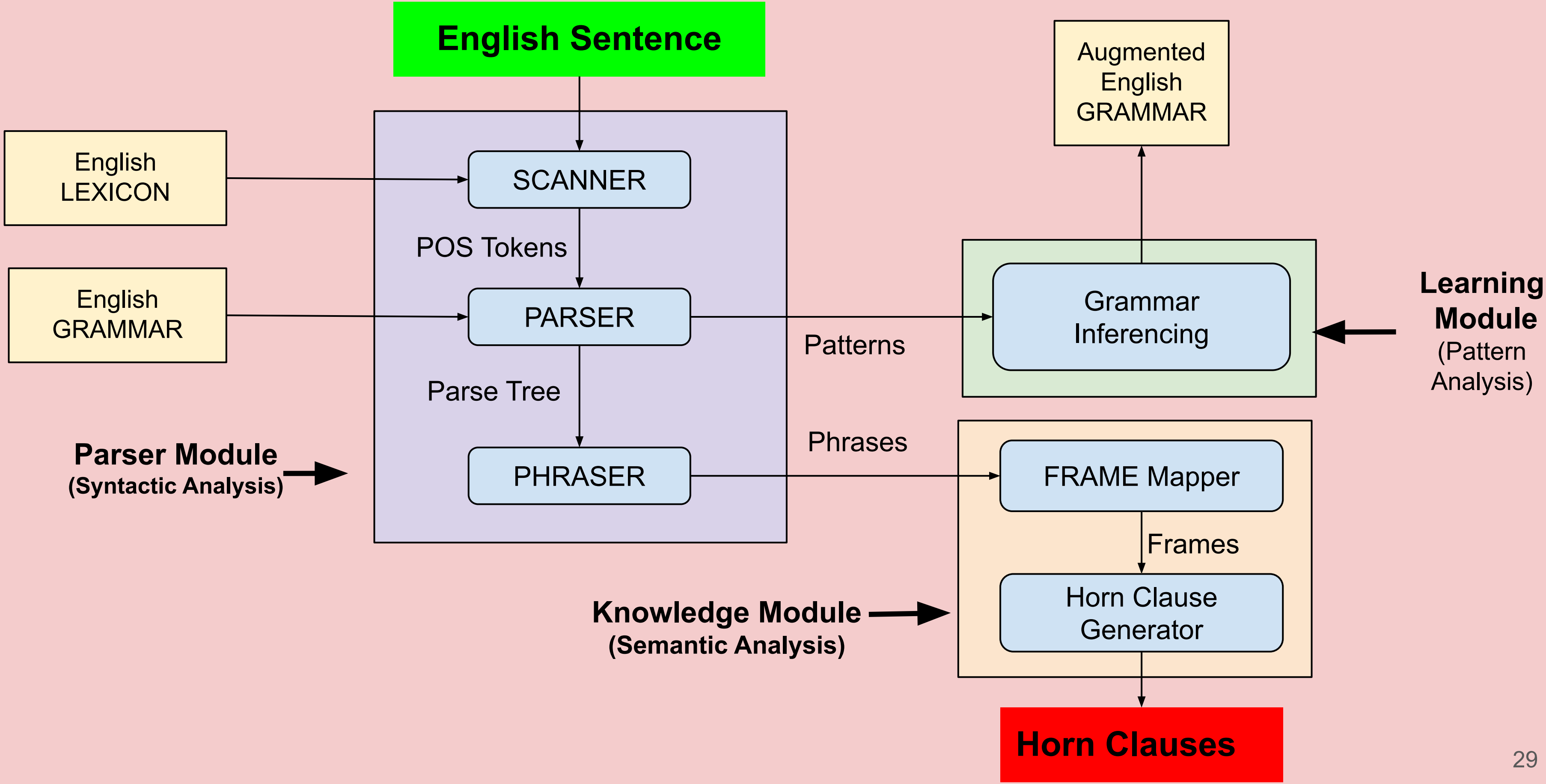
- Convert human knowledge to clausal form logic

- A Horn clauses
  - A disjunction of literals with at most one positive literal (Alfred Horn, 1951)
  - A a literal is a predicate symbol applied to some terms. e.g. (isa cat mammal)
  - $(a_1 \wedge a_2 \wedge \ldots a_n) \Rightarrow b \;\Leftrightarrow\; \sim(a_1 \wedge a_2 \wedge \ldots a_n) \vee b \;\Leftrightarrow\; (\sim a_1 \vee \sim a_2 \vee_{\ldots} \vee \sim a_n \vee b)$

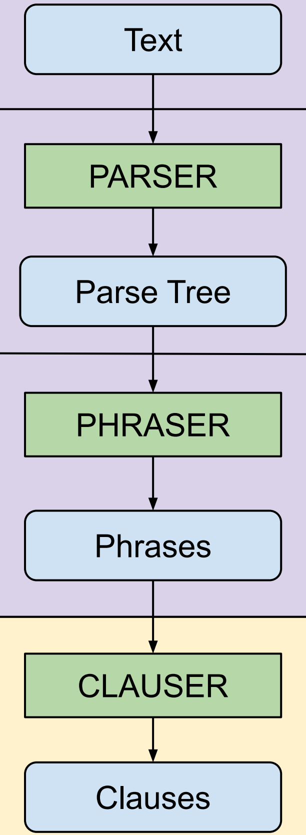
- Parser Module
  - Scanner
  - Parser
  - Phraser

- Knowledge Module
  - Frames (Marvin Minsky, 1974)
  - Horn Clauses

Knowledge Extraction in Babar



Knowledge Extraction Example



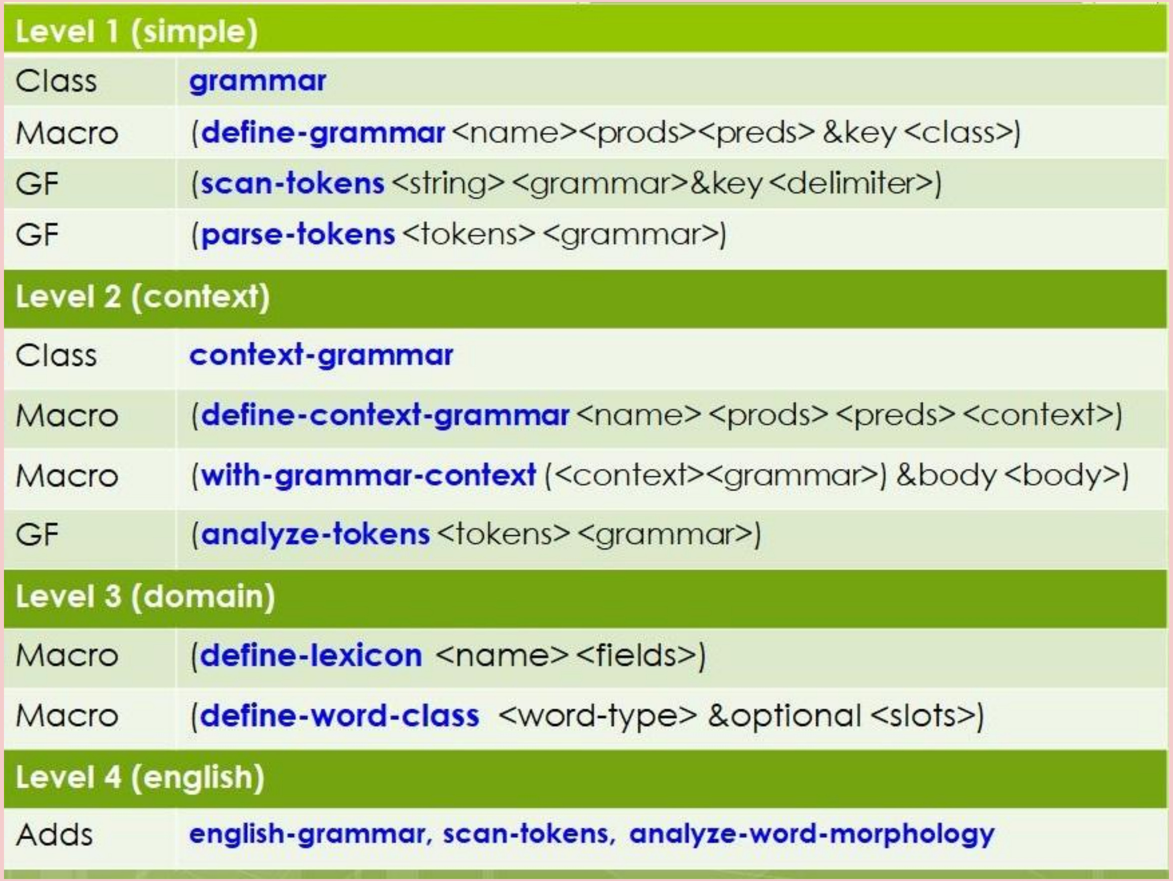
"A cat has four legs"

```
NAME    "A cat has four legs"
VALUE: (#<PARSE-TREE: A cat has four legs>)
WORDS:  (#<ARTICLE: a> #<NOUN: Cat> #<VERB: has>
         #<NOUN: Four> #<NOUN: legs>)
```

```
((#<PARSE-NODE: :NP> (#<ARTICLE: a> #<NOUN: Cat>))
 (#<PARSE-NODE: :VP> (#<VERB: has>))
 (#<PARSE-NODE: :NP> (#<NOUN: Four> #<NOUN: legs>)))
```

```
(KB::HASA #<NP: A cat> #<NP: four legs>)
```

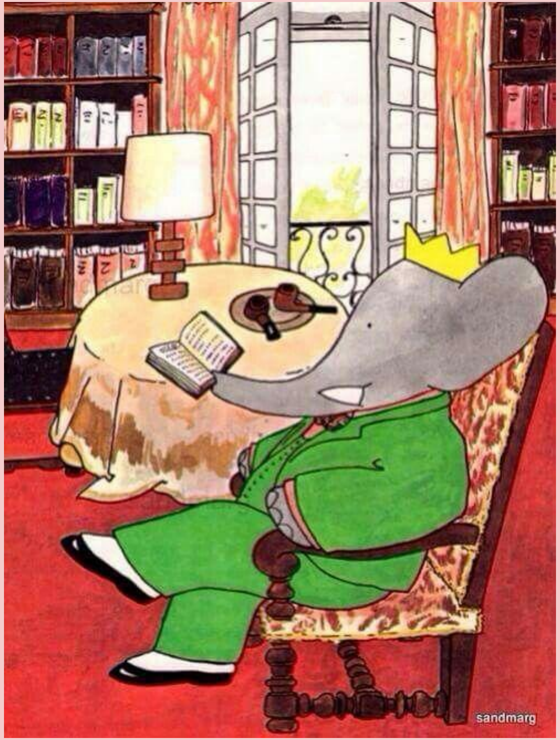
Parser Module Layers: Scanner, Analyzer & Parser



Level 1 (simple)	
Class	**grammar**
Macro	(**define-grammar** <name><prods><preds> &key <class>)
GF	(**scan-tokens** <string> <grammar>&key <delimiter>)
GF	(**parse-tokens** <tokens> <grammar>)
Level 2 (context)	
Class	**context-grammar**
Macro	(**define-context-grammar** <name> <prods> <preds> <context>)
Macro	(**with-grammar-context** (<context><grammar>) &body <body>)
GF	(**analyze-tokens** <tokens> <grammar>)
Level 3 (domain)	
Macro	(**define-lexicon** <name> <fields>)
Macro	(**define-word-class** <word-type> &optional <slots>)
Level 4 (english)	
Adds	**english-grammar, scan-tokens, analyze-word-morphology**

Lexicon: Books That Babar has Read (WIP)

Megan and the Princess of Death
The Adventures of Goat, Dog and Cow
Nancy Drew and the Hidden Staircase
Agatha Christie: The Mysterious Affair at Styles
Jules Verne: 20,000 Leagues Under the Sea
Edgar Allan Poe: Tales of Mystery and Macabre
Arthur Conan Doyle: Adventures of Sherlock Holmes
Gabriel G.Marquez: One Hundred Years of Solitude
Charles Dickens: Great Expectations
Leo Tolstoy: War and Peace
Charles Darwin: On the Origin of the Species
The Dictionary of Biology
Many, many, many Wikipedia pages



Take away: Vocabulary of 260,000+ words with parts of speech and definitions.

Scanner: Primitive Token Mixin Classes

- Scanner methods are executed first and create primitive token classes
- Analyzer methods analyze tokens and assign parts of speech
- English Parser adds domain specific scanner & analyzer methods
- E.g. Analyzer ⇒ *Alpha-special* “library.” ⇒ Split into “library” and “.”
- Token Mixin Classes: *Alphabetic Mixin, Numeric Mixin, Special Mixin*

Alphabetic Class
Numeric Class
Special Class
Alpha-Numeric Class
Alpha-Special Class
Numeric-Special
Alpha-Numeric-Special

Primary methods *(Domain specific knowledge)*

Before methods *(Logging & printing patterns)*

After methods *(Database persistence)*

Around methods *(Domain specific knowledge)*

- **Take away**: Considerable domain knowledge application prior to parsing phase

Scanner & Analyzer: Merriam-Webster Parts of Speech

- Merriam-Webster as a lexicon
- Part of speech tagging
- Penn Treebank Project: 36 POS tags & 12 Other tags

```
('3ps-present', 19),
('abbreviation', 114),
('adjective', 945),
('adjective combining form', 4),
('adjective suffix', 6),
('adverb', 216),
('auxiliary verb', 1),
('combining form', 21),
('conjunction', 9),
('idiom', 2),
('imperative', 1),
('imperative verb', 1),
(interjection', 18),
('intransitive verb', 41),
('noun', 1738),
('noun combining form', 1),
('noun phrase', 3),
(noun suffix', 5),
('past-participle', 69),
('past-tense', 69),
('person', 14),
('place', 10),
('plural', 72),
('plural noun', 6),
('prefix', 14),
('preposition', 39),
('present-participle', 71),
('pronoun', 13),
('pronoun, plural in
construction', 2),
('pronoun, construction', 1),
('proper-noun', 41),
('suffix', 1),
('symbol', 17),
('trademark', 3),
('transitive verb', 129),
('verb', 1489),
('verb suffix', 1)]
```

DEFINE-ENGLISH-WORD-CLASS

```
(defmacro DEFINE-ENGLISH-WORD-CLASS (word-class word-type &optional slots)
 (let* ((predicate-name  (intern (string-upcase (concatenate 'string (symbol-name word-class) "-P"))  :parser)))
   `(progn

     (defclass ,word-class (ENGLISH-WORD)
       (,@slots))

     (defmethod INITIALIZE-INSTANCE :after ((word ,word-class &rest args)))
       (setf (english-word-type word)  ,word-type))

      (defmethod PRINT-OBJECT ((obj ,word-class) stream)
       (cond (*print-readably*
              (call-next-method))
             (t
              (format stream "#<~a: ~a>" (english-word-type obj) (util::object-name obj)))))

      (defmethod ,predicate-name ((word DOMAIN-WORD)(context T))
       (typep word ',word-class))
```

Take away: Homogeneity. Defined new behavior without overwriting methods or any breaking changes.

English Language Parser (Weak AI)

- Classical Recursive Descent Parser
  - Top-down parsing approach

- Non-determinism is handled by backtracking
  - Three ways of handling non-determinism in Computer Science
  - Backtracking, Parallelism & Probability Theory
  - Could optimize using Earley algorithm (CNF)

- **Novel Aspect**: Parser Never Fails
  - Always returns a **forest of parse trees**
  - Creates ***unparse trees*** for sequences of unparsed tokens

- **Novel Aspect:** Adaptive Parser (*machine learning*)
  - Logs parser forest patterns
  - Learns new grammar productions (rules)
  - Terrifies functional programming folks

- Model-based vs. Model-free AI
  - Normally this qualifies as a model-based reductionist approach
  - Peter Norvig: “ABC as Easy as 123”, shift in Natural Language Processing paradigm.
  - Unparse trees & the adaptive aspect allow for emergence (model-free)

English Parser: Recursive Descent Parser

- Classical Parsing Technology (Martin & Jurafski)
- A context-free grammar describes an AND-OR Tree
- Nested disjunctions of conjunctions

- Non-determinism is handled with backtracking ($O(2^n)$)
- Can be optimized with Earley Algorithm ($O(n^3)$)

- Babar returns the first successful parse tree.
- Easily extended to return all parse trees.

- **Parser can be viewed as a *predicate function***
- **Compilation Error:**
  - Expecting semicolon but found the string “Arthur”
- **Babar: Purpose is *knowledge extraction***

- **Babar: Returns a forest of parse trees**
- **Eg. “A cat and a dog” ⇒ (*noun-phrase unparse-tree noun-phrase*)**

Babar Parser Demo

Knowledge Sideline : Consistency of Human Knowledge

- **Question**: Can birds fly?
- Answer: **Yes**

- **Question**: Are penguins birds?
- Answer: **Yes**

- **Question**: Can penguins fly?
- Answer: **No**

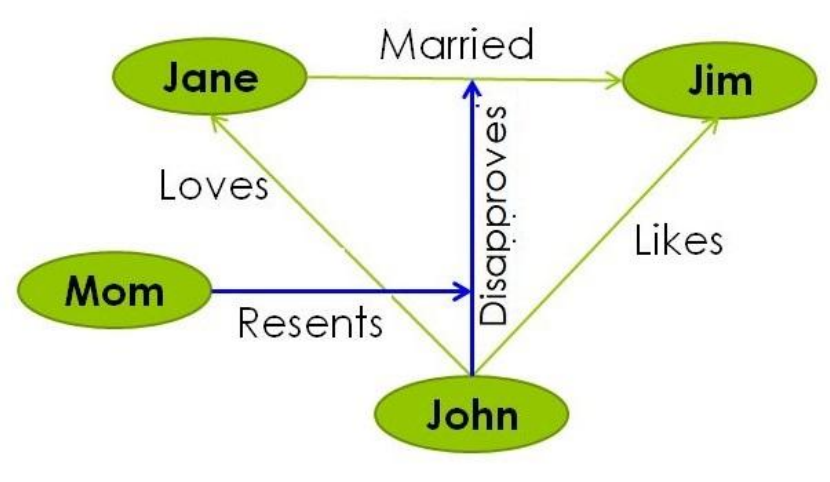
Take away: Always question the rationality of *Homo Sapiens.*

Knowledge Representation & Reasoning (WIP)

- Clauses in a Semantic network
  - In-Memory Hypergraph
  - Mary, Jane, John & Mom
  - Persisted in PostGres

- Generalization (FOIL)
  - Cats, dogs, rabbits have four legs
  - Cats, dogs, rabbits are mammals
  - ⇒ All mammals have four legs

- Inferencing *(syllogisms)*
  - All animals have four legs
  - Arthur is a party animal
  - ⇒ Arthur has four legs



Knowledge Extraction Examples (WIP)

(HASA "Female African Elephants" "large tusks")

(ISA "Elephants" "large land mammals")

(ISA "Elephants" "herbivores")

(HASA "African Elephants" “three nails")

(HASA "Indian Elephants" "four nails")

- Examples of Clausal Form Logic (CFL)
- Well suited for a Graph Database (i.e. triples)
- Still handling False-Positives and Nonsense
- Form of KR that is very different from ANN

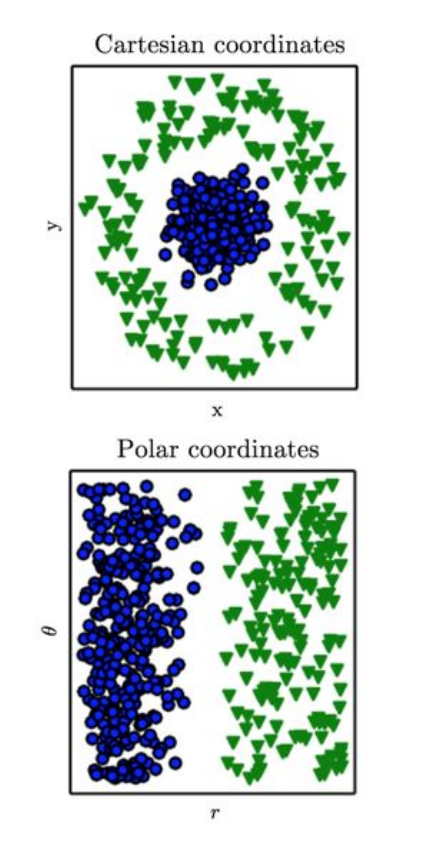
Clustering as Model-free Knowledge Representation

- Unsupervised Learning (mostly model-free)
- Typically applied to unlabeled data
- A form of categorization and generalisation

- Many Different Clustering Algorithms
  - K-Means (spheroids)
  - Hierarchical Agglomerative Clustering (Dendrogram)
  - Big Data Clustering
    - Distributed algorithms (Spark ML)
    - Representative elements (e.g. CURE)

- Fails to semantically label the clusters
  - Labels are an important: of semantic generalization
  - Provide conciseness if not preciseness, e.g. birds can fly.

- Babar Uses SR-Clustering



SR Clustering

- **Simple-Ray Clustering**
    - Proprietary Algorithm
    - Sort of like a flattened version of hierarchical agglomerative clustering

- **Basic Algorithm**
    - ***For each data point, place it in the correct cluster and you’re done.***
    - Provably correct by definition
    - If it doesn’t belong to any cluster, create a new cluster with that single data point

- **Cluster Membership Predicate**
    - Within a ***proximity threshold*** of every data point in that cluster.

- **Proximity Metric**
    - The ***Jaccard Index***
        - Norm of the intersection divided by the norm of the union
        - I.e. Percentage overlap of two sets
    - Babar uses the ***outbound links*** of each Wikipedia topic being compared

Clustering Metric: Jaccard Index (WIP)

Compute the Jaccard Index of pairs of topics using the related topics of each topic as the sets to be compared.

	African	Asian	Indian	Babar	Horton	War
African	100.00	38.46	21.05	4.35	6.82	7.94
Asian	38.46	100.00	37.74	4.00	6.25	20.00
Indian	21.05	37.74	100.00	6.90	7.14	24.39
Babar	4.35	4.00	6.90	100.00	28.57	7.14
Horton	6.82	6.25	7.14	28.57	100.00	7.41
War	7.94	20.00	24.39	7.14	7.41	100.00

NB: An optimal proximity threshold can be determined automatically by maximizing the threshold such that there is no overlap between the individual clusters.

Example Clustering Results for Subtopics of Elephant

Cluster 1

Asian_elephant(49)
African_elephant(60)

Cluster 2

Babar_the_Elephant(7)
Horton_the_Elephant(5)
Elmer_the_Patchwork_Elephant(4)

Cluster 3

Asian_elephant(49)

Indian_Elephant(18)
Sri_Lankan_elephant(12)
Sumatran_Elephant(11)
Borneo_pygmy_elephant(3)

Cluster 4

War_elephant(22)
Spotting_the_elephant (7)
Shooting_the_elephant (15)
Execution_by_elephant(5)

Cluster 5

Year_of_the_Elephant(8)

Cluster 6

Dwarf_elephant(24)

Cluster 7

White_elephant(10)

Take away: Strictly based on Graph connectivity. Exemplifies the importance of relations.

Human Knowledge Acquisition Process

- Can be a two decade long process !!!
  - Kindergarten – Post Doctoral work
  - Reptiles are born as mini adults

- Grade school topics as default knowledge categories.
  - Science, History, Literature, Art, etc…

- Goal: Assign categories to subtopic clusters
  - Create topic graphs for each knowledge category
  - Use Jaccard Index to identify the category
  - Automatically generate semantic cluster labels
    - E.g. Babar = Literature_Elephant
    - E.g. Asian Elephant = Science_Elephant

```
(((( :SCIENCE 0.47666672) (:HISTORY 0.44666672))
  (#<Concept(49): Asian_elephant> #<Concept(60): African_elephant>))

 ((( :SCIENCE 0.39) (:GEOGRAPHY 0.37800002))
  (#<Concept(3): Borneo_pygmy_elephant> #<Concept(49): Asian_elephant>
   #<Concept(18): Indian_Elephant> #<Concept(12): Sri_Lankan_elephant>
   #<Concept(11): Sumatran_Elephant>))

 ((( :ART 0.33333334) (:GEOGRAPHY 0.30666667))
  (#<Concept(7): Babar_the_Elephant> #<Concept(5): Horton_the_Elephant>
   #<Concept(4): Elmer_the_Patchwork_Elephant>))

 ((( :HISTORY 0.6) (:GEOGRAPHY 0.46)
  (#<Concept(8): Year_of_the_Elephant>))

 ((( :GEOGRAPHY 0.72333336) ( :HISTORY 0.43666664))
  (#<Concept (7) Spotting_the_elephant> #<Concept (15) Shooting_the_elephant>
    #<Concept(5): Execution_by_elephant> #<Concept(22): War_elephant>
   #<Concept(4): Crushing_by_elephant>))
```

Examples of Clustering topics not containing “elephant”

((#<Concept(60): African elephant>
#<Concept(49): Asian_elephant>)
(**#<Concept(10): Elephant intelligence**>
#<Concept(103): Animal cognition>
#<Concept(4): Elephant tusk>
#<Concept(102): Proboscidea>
#<Concept(876): Mammal>)

(#<Concept22): War_elephant>
#<Concept(41): Spotting_the_elephant>
#<Concept(13): Shooting_the_elephant>
#<Concept(37): Execution_by_elephant>
#<Concept(55): Ivory>)

(**#<Concept(24): Dwarf elephant**>
#<Concept(66): Mammoth>
#<Concept(25): Mastodon>
#<Concept(62): Afrotheria>
#<Concept(86): Gestation>
#<Concept(8): Gomphotherium>
#<Concept(27): Tooth>
#<Concept(8): Tooth_development>))

(#<Concept(7): Babar_the_Elephant>
#<Concept(5): Horton_the_Elephant>
#<Concept(4): Elmer_the_Patchwork_Elephant>
#<Concept(6): List_of_fictional_elephants>)

Take away: Model-Free Emergence of concepts knowledge from relations

Afterthoughts on Clustering Results & KR in General

- In 2012: There were 36 subtopics of Elephant on Wikipedia
- In 2021: There 105 subtopics of of Elephant on Wikipedia

- Indiscriminately selecting neighbor-based metrics
  - Need a natural way of filtering relenat topics

- Weighted approach
  - Strongly Related Topics
  - Triangles in the Graph

- Addressing Wikipedia Popularity Bias
  - External references
  - Other sources in general

- Factor in content based Deep Learning paradigms (*Word Embeddings*)
- Most importantly: Need to maintain a model-free approach
- Basically anything that enables natural emergence and evolution
- E.g. In 2012 there was no Covid-19 Wikipedia page of FB group.

Inductive Logic Programming (WIP)

- Yet another form of symbolic machine learning

- Learning sets of Horn clauses
  - “Learning Logical Definitions From Relations” (Quinlan, 1990)
  - FOIL Algorithm (FOIL-Gain)
  - Supports learning recursive rules
  - E.g. Learning “Ancestor” relation
  - Learning Prolog programs
  - Relationship with Decision Tree Learning (Information Gain)

- ILP in Babar (WIP)
  - Symbolic Generalization: E.g. (hasa mammals four-legs)
  - Concept identification: E.g. Prehistoric animals
  - Concept emergence: E.g. Fictional elephants
  - Concept creation (Imagination)
    - *Hotel Elephant*: An elephant that hangs out at hotel bars drinking Piña Coladas with Arthur
    - *Elephant Hotel*: A type of hotel that accommodates such elephants and party animals

The FOIL Algorithm (Inductive Logic Programming)

```
FOIL (target-predicate, predicates, examples)
pos = those examples for which target-predicate is true
neg = those examples for which target-predicate is false
learned-rules  {}
while pos is not empty do
    learn a new rule
    new-rule  = rule that predicts target predicate with no preconditions.
    new-rule-neg  = neg
    while new-rule-neg do
        add a new literal to specialize new rule
        candidate-literals  = new literal candidates based on predicates
        best-literal  = argmax (FOIL-Gain (literal, new-rule)) over candidate-literals
        add best-literal to the preconditions of new-rule
        new-rule-neg = subset of new-rule-neg which satisfy new-rule preconditions.
     learned-rules  = learned-rules + new-rule
     pos  = pos – (members of pos covered by new-rule)
return learned-rules
```

Classification & Clustering Remarks

- **A function of the Classes and/or the Data**
  - Classes are subject to data coverage limitations
  - E.g. Black sheep vs. white sheep

  - Classes are subject to subjectivity
  - E.g. It’s a bird. Is it a plane. No, it’s Batman.

  - Classes not always mutually exclusive
  - E.g. Arthur: Party animal, Land mammal, Bar(n) animal

  - Seriously: Taxonomies are subject to their usage
  - E.g. Classifying web sites and web pages

- **Meta Knowledge & Context**
  - Why are we asking the question?
  - What knowledge does it provide?
  - Is the question reasonable or useful?

Combining Supervised and Unsupervised Learning

- Semi-supervised Learning
  - Small set of labeled data
  - Large set of unlabeled data
  - Train a classifier on the labeled data
  - Use classifier to label the unlabeled data

- Example: Identifying people on Wikipedia
  - Start with a database of about 30,000 scraped famous people
  - Generate labeled data for the corresponding Wikipedia pages
  - Sample Wikipedia in order to generate examples of non-people
  - Train a person page classifier on the labeled data
  - Use the classifier to predict and label unlabeled pages
  - Finally, train a classifier on the larger set of labeled data
  - Application: Identifying people or news articles about people (NER)

Neuro-Symbolic Computing

- Integrates in a principled way neural network-based learning with symbolic knowledge representation and logical reasoning

Possible Real World Applications of Babar (WIP)

- **Clustering Search Engine Results**
  - Model-based Approach
    - Pre-select categories of interest
    - Train a classifier on those categories
    - Predict category of each Google Search result
    - Group results by pre-defined category
  - Model-free Approach
    - Analyze Google Search results and vocabulary
    - Allow categories to naturally emerge
    - Group results according to those categories

- **Research Engine**
  - Browsing the Web more effectively
  - Researching topics in general
  - Hours vs. Milliseconds

- **Content Generation**
  - Topic Summaries
  - Advertisement content

Scalability Issues (WIP)

- Handling 56 million wikipedia topics
  - How does Wikipedia do do it?
- Handling billions of edges
  - How social media platforms do it?
- Handling millions of users
  - Standard practices?
- Partitioned Tables vs. Multiple Tables
  - Are other options?
- Distributed Databases
  - CAP Theorem
- Graph Databases
  - Neo4j
  - AllegroGraph
  - Datomic

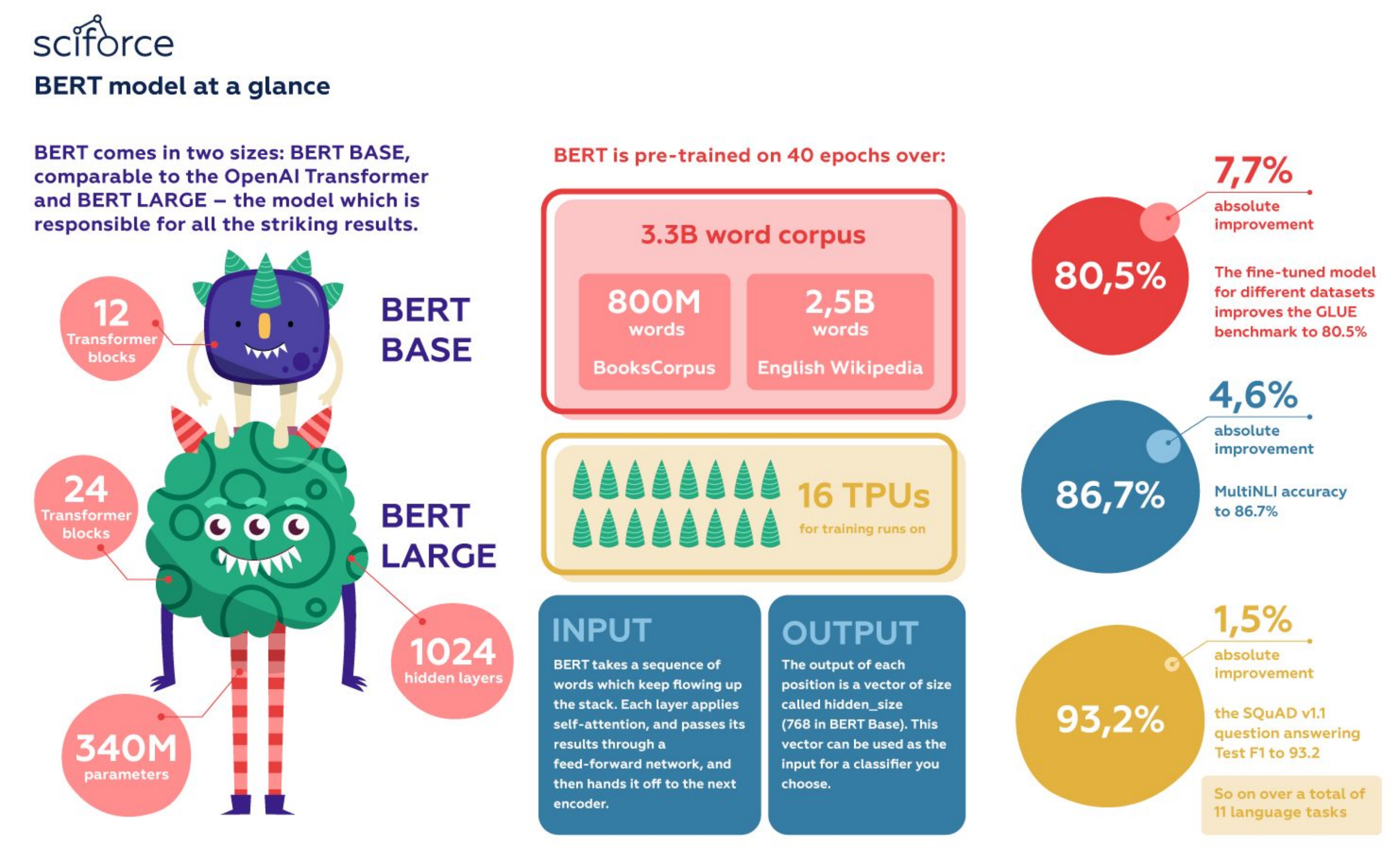
References

- Books
    - Artificial Intelligence: A Modern Approach, 4th Ed., Russell and Norvig, 2019
    - Speech and Language Processing, 3rd Ed., Jurafsky, and Martin, 2020
    - Dive into Deep Learning, Zhang, Lipton, Li, and Smola, 2021
    - Deep Learning, Goodfellow, Bengio and Courville, 2016
    - Mining of Massive Datasets, Leskovec, Rajaraman and Ullman, 2010
    - Pattern Recognition and Machine Learning, Christopher Bishop, 2006
    - Machine Learning, Tom Mitchell, 1997

- Papers
    - Attention is All You Need, Vaswanie, Shazeer, Parmar, Uszkoreit, Dec. 2017
    - Neurosymbolic AI: The 3rd Wave, Artur d’Avila Garcez and Luís C. Lamb, Dec. 2020
    - Learning Logical Definitions from Relations, J.R. Quinlan, 1990
    - Elephant 2000: A Programming Language Based on Speech Acts, John McCarthy, 2000
    - A Framework for Representing Knowledge, Marvin Minsky, 1974

That's all Folks!

Appendix A: BERT (NLP State of the Art)



sciforce

BERT model at a glance

BERT comes in two sizes: BERT BASE, comparable to the OpenAI Transformer and BERT LARGE – the model which is responsible for all the striking results.

BERT is pre-trained on 40 epochs over:

3.3B word corpus

800M words	2,5B words
BooksCorpus	English Wikipedia

16 TPUs for training runs on

INPUT

BERT takes a sequence of words which keep flowing up the stack. Each layer applies self-attention, and passes its results through a feed-forward network, and then hands it off to the next encoder.

OUTPUT

The output of each position is a vector of size called hidden_size (768 in BERT Base). This vector can be used as the input for a classifier you choose.

80,5% — 7,7% absolute improvement

The fine-tuned model for different datasets improves the GLUE benchmark to 80.5%

86,7% — 4,6% absolute improvement

MultiNLI accuracy to 86.7%

93,2% — 1,5% absolute improvement

the SQuAD v1.1 question answering Test F1 to 93.2

So on over a total of 11 language tasks

Appendix B: Programmer Ethics in CLOS

- Primary methods do a call-next-method
  - Ensures complete inheritance
  - Ensures complete behavior

- Around methods conditionally do a call-next-method
  - Heavy handed by definition
  - Typically do a call-next-method
  - Allow deviation in method dispatching (before part)
  - Allow customization of returned value (after part)

- Before and After methods are used for side effects
  - Logging information
  - Persisting data in a database

Appendix C: Programmer Responsibility in CLOS

- Do not change existing functionality (no breaking changes)
  - Inheriting partial attributes is a bad idea
  - Inheriting partial behavior is a bad idea

- Create new classes to extend existing functionality
  - Guitarist inherits from Person, adds tour dates and a mailing list

- Create new classes to add functionality
  - Department of Airborne Vehicles

- Add methods on on existing classes to augment functionality
  - Teleportation ID & associated methods

Appendix D: Business Issues

- Legal Concerns
  - Copyright violations: Wikipedia & Merriam-Webster
  - Extracting content in general
- Cost Issues
  - Cloud computing is expensive
  - Ideally one server per customer
- Security Issues
  - Proprietary customer data
  - Login Info
  - Payment Info