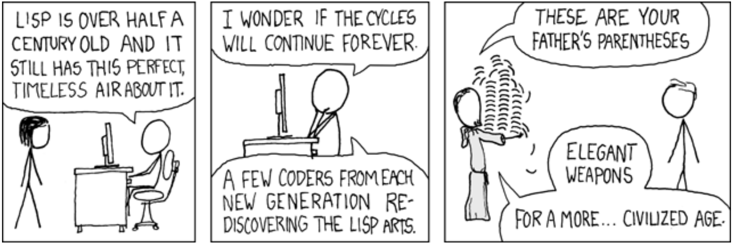
CS325 Artificial Intelligence
Chs. 9, 12 – Knowledge Representation and Inference

Cengiz Günay, Emory Univ.


LISP IS OVER HALF A CENTURY OLD AND IT STILL HAS THIS PERFECT, TIMELESS AIR ABOUT IT.
I WONDER IF THE CYCLES WILL CONTINUE FOREVER.
A FEW CODERS FROM EACH NEW GENERATION RE-DISCOVERING THE LISP ARTS.
THESE ARE YOUR FATHER'S PARENTHESES
ELEGANT WEAPONS
FOR A MORE... CIVILIZED AGE.

Spring 2013

Entry/Exit Surveys

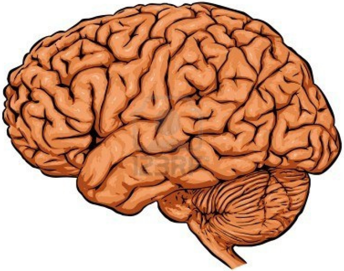
Exit survey: Logic

- Where would you use propositional vs. FOL?
- What is the importance of logic representation over what we saw earlier?

Entry survey: Knowledge Representation and Inference (0.25 points of final grade)

- What is the difference between data, information and knowledge?
- What do you think would count as a "knowledge base"?

Part I: The Variable Binding Problem



Reminder: Propositional Logic vs. First Order Logic

Propositional Logic: Facts only

First Order Logic: Objects, variables, relations

Reminder: Propositional Logic vs. First Order Logic

Propositional Logic: Facts only

First Order Logic: Objects, variables, relations

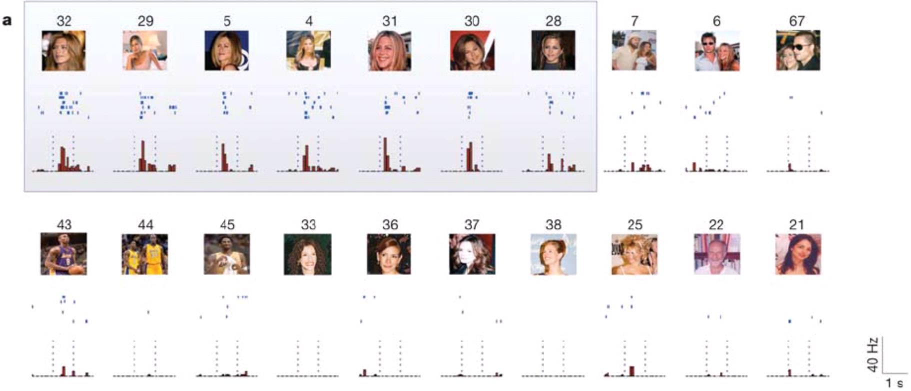
- Let's talk about my brain research!

Single neurons can represent concepts in the brain

- Human brain only takes a second to recognize an object or a person
- How this high-level representation achieved is unknown

Single neurons can represent concepts in the brain

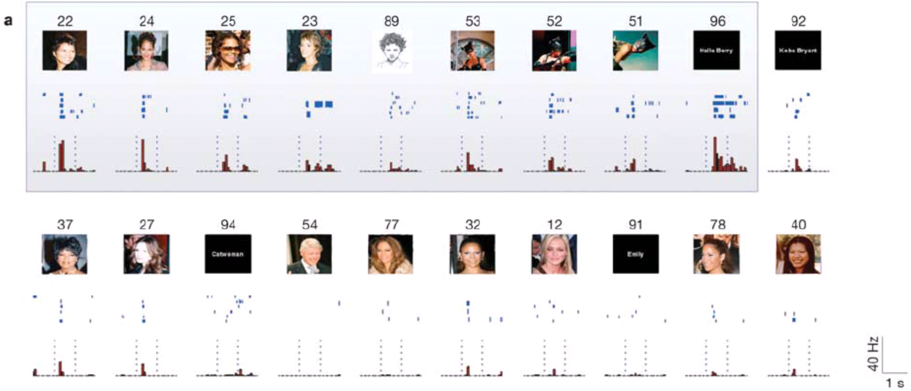
- Human brain only takes a second to recognize an object or a person
- How this high-level representation achieved is unknown
- But can find single neurons representing, e.g., actress Jennifer Aniston:



Quiroga et al. (2005)

. . . even when it is an abstraction

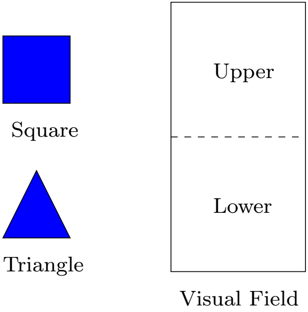
- These neurons also respond to abstract notions of the same concept (e.g., actress Halle Berry):



Quiroga et al. (2005)

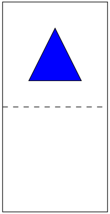
Then, are features always represented by single neurons? The Binding Problem (1)

Rosenblatt's example (1961): two shapes in two possible locations in a visual scene.

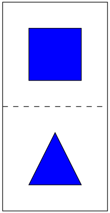


Objects can be detected individually, but not when together

If propositional representations are employed:



triangle-object $\wedge$ object-in-upper-part

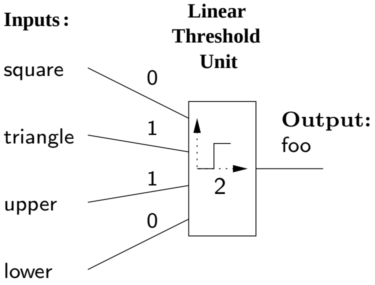


square-object $\wedge$ triangle-object$\wedge$
object-in-upper-part $\wedge$ object-in-lower-part

Both satisfies query: triangle-object $\wedge$ object-in-upper-part $\Rightarrow$ something

An LTU neuron suffers from this binding problem

This linear threshold unit (LTU) neuron exhibits the same problem:



Possible Solution (1): Combination-coding

Using a neuron for each possible configuration combination, i.e.:

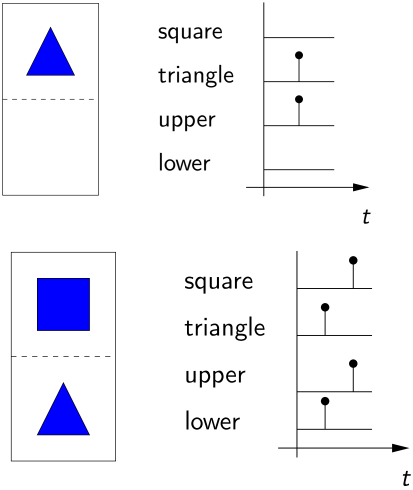
- upper-triangle, upper-square, lower-triangle, lower-square.

Drawback: Combinatorial explosion:

- Impossible that the brain has individual cells for each possible concept combination in nature (Barlow, 1972).

Possible Solution (2): Phase-coding with Temporal Binding

Bound entities are represented by temporal synchrony:



Query triangle $\wedge$ upper $\Rightarrow$ something is only satisfied by the top case!

Recruitment Learning Induced by Temporal Binding

Temporal binding:

- Recent evidence of binding units in monkeys (Stark et al., 2008)
- But, only allows temporary representations (O'Reilly et al., 2003)

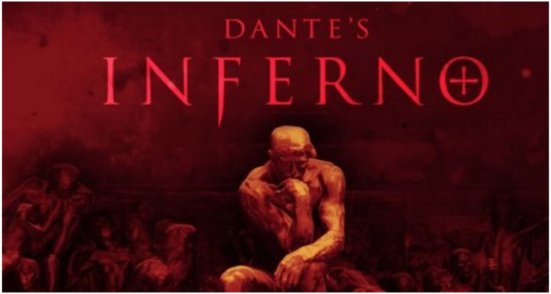
Recruitment learning (Feldman, 1982; Diederich, Günay & Hogan, 2010) forms long-term memories, which:

- Can be induced by temporal binding (Valiant, 1994; Shastri, 2001);
- Models the brain as a random graph (Wickelgren, 1979).
- Avoids combinatorial explosion by only allocating when needed (Feldman, 1990; Valiant, 1994; Page, 2000).

Brain Uses Time to Encode Variables?

- Still a valid theory
- We don't know how the brain represents binding information
- Other theories: synfire chains, synchronized oscillations

Part II: Inference



Automated Inference?

We already did it:

What we know to be True:

- $(E \vee B) \Rightarrow A$
- $A \Rightarrow (J \wedge M)$
- B

Can we infer?

T	F	?	
			E
			B
			A
			J
			M

Automated Inference?

We already did it:

What we know to be True:

- $(E \vee B) \Rightarrow A$
- $A \Rightarrow (J \wedge M)$
- B

Can we infer?

T	F	?	
		X	E
X			B
X			A
X			J
X			M

Automated Inference?

We already did it:

What we know to be True:

- $(E \vee B) \Rightarrow A$
- $A \Rightarrow (J \wedge M)$
- B

Can we infer?

T	F	?	
		X	E
X			B
X			A
X			J
X			M

- In propositional logic, resolution by forward/backward chaining

  Forward: Start from knowledge to reach query

  Backward: Start from query and go back

Automated Inference?

We already did it:

What we know to be True:

- $(E \vee B) \Rightarrow A$
- $A \Rightarrow (J \wedge M)$
- B

Can we infer?

T	F	?	
		X	E
X			B
X			A
X			J
X			M

- In propositional logic, resolution by forward/backward chaining

  Forward: Start from knowledge to reach query
  Backward: Start from query and go back

- In FOL, substitute variables to get propositions (see Ch. 9)
  - Use lifting and unification to resolve variables
- Logic programming: Prolog, LISP, Haskell

Prolog

- Most widely used logic language.
- Rules are written in backwards:
  criminal (X) :− american(X), weapon(Y), sells (X, Y, Z), hostile (Z)
- Variables are uppercase and constants lowercase.

Prolog

- Most widely used logic language.
- Rules are written in backwards:
  criminal (X) :– american(X), weapon(Y), sells (X, Y, Z), hostile (Z)
- Variables are uppercase and constants lowercase.
- Because of complexity, often compiled into other languages like: Warren Abstract Machine, LISP or C.
- Language makes it easy to contruct lists, like LISP.

Do You Have a LISP?

LISP **LIS**t **P**rocessing language: primary data structure is lists.

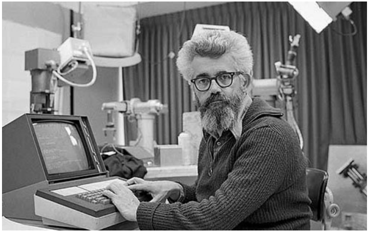
Do You Have a LISP?

LISP **LIS**t **P**rocessing language: primary data structure is lists.

- Lisp is used for AI because can work with symbols
- Examples: computer algebra, theorem proving, planning systems, diagnosis, rewrite systems, knowledge representation and reasoning, logic languages, machine translation, expert systems, . . .
- It is a *functional* programming language, as opposed to a *procedural* or *imperative* language

Functional languages

- LISP invented by John McCarthy in 1958



```
(defun factorial (n)
   (if (<= n 1)
       1
       (* n (factorial (- n 1)))))
```

Functional languages

- LISP invented by John McCarthy in 1958
- Scheme: A minimalist LISP since 1975. Introduces *lambda calculus*.

λ

```
(define-syntax let
  (syntax-rules ()
    ((let ((var expr) ...) body ...)
      ((lambda (var ...) body ...) expr ...))))
```

Functional languages

- LISP invented by John McCarthy in 1958
- Scheme: A minimalist LISP since 1975. Introduces *lambda calculus*.

λ

```
(define-syntax let
  (syntax-rules ()
    ((let ((var expr) ...) body ...)
      ((lambda (var ...) body ...) expr ...))))
```

Java implementation JScheme by Peter Norvig in 1998.

```
java jscheme.Scheme scheme-files ...
```

Functional languages

- LISP invented by John McCarthy in 1958
- Scheme: Since 1975. Introduces *lambda calculus*.
- Haskell: Lazy functional language in 90s.

```haskell
-- Type annotation (optional)
factorial :: Integer -> Integer

-- Using recursion
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

LISP Usage Areas

Famous AI applications in Lisp:

- Macsyma as the first large computer algebra system.
- ACL2 as a widely used theorem prover, for example used by AMD.
- DART as the logistics planner used during the first Gulf war by the US military. This Lisp application alone is said to have paid back for all US investments in AI research at that time.
- SPIKE, the planning and scheduling application for the Hubble Space Telescope. Also used by several other large telescopes.
- CYC, one of the largest software systems written. Representation and reasoning in the domain of human common sense knowledge.
- METAL, one of the first commercially used natural language translation systems.
- American Express' Authorizer's Assistant, which checks credit card transactions.

So What's So Special About LISP?

- First language to be **homoiconic**: data and code represented alike, can modify and execute code on the fly

So What's So Special About LISP?

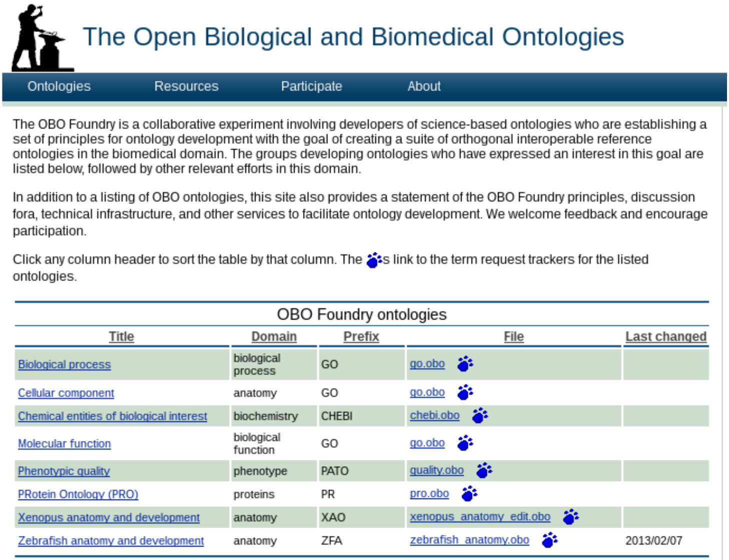
- First language to be **homoiconic**: data and code represented alike, can modify and execute code on the fly
  - Ever used Java introspection? Scripting languages like PERL and Python allow evaluating new code, too.

So What's So Special About LISP?

- First language to be **homoiconic**: data and code represented alike, can modify and execute code on the fly
  - Ever used Java introspection? Scripting languages like PERL and Python allow evaluating new code, too.
- First use of the *if-then-else* structure
- Adopted object-oriented features from language *SmallTalk*
- First use of *automatic garbage collection*

Part III: Knowledge Representation (Ch. 12)

Knowledge Ontologies



The OBO Foundry is a collaborative experiment involving developers of science-based ontologies who are establishing a set of principles for ontology development with the goal of creating a suite of orthogonal interoperable reference ontologies in the biomedical domain. The groups developing ontologies who have expressed an interest in this goal are listed below, followed by other relevant efforts in this domain.

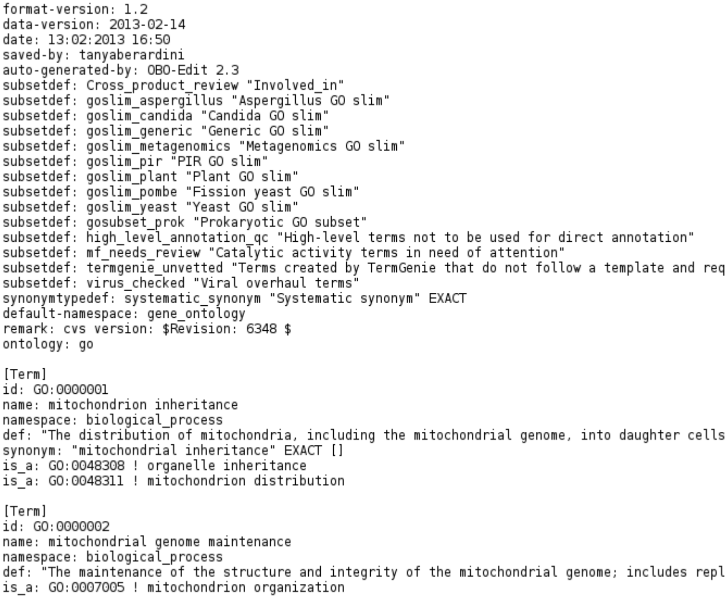
In addition to a listing of OBO ontologies, this site also provides a statement of the OBO Foundry principles, discussion fora, technical infrastructure, and other services to facilitate ontology development. We welcome feedback and encourage participation.

Click any column header to sort the table by that column. The 🐾s link to the term request trackers for the listed ontologies.

OBO Foundry ontologies

Title	Domain	Prefix	File	Last changed
Biological process	biological process	GO	go.obo 🐾	
Cellular component	anatomy	GO	go.obo 🐾	
Chemical entities of biological interest	biochemistry	CHEBI	chebi.obo 🐾	
Molecular function	biological function	GO	go.obo 🐾	
Phenotypic quality	phenotype	PATO	quality.obo 🐾	
PRotein Ontology (PRO)	proteins	PR	pro.obo 🐾	
Xenopus anatomy and development	anatomy	XAO	xenopus_anatomy_edit.obo 🐾	
Zebrafish anatomy and development	anatomy	ZFA	zebrafish_anatomy.obo 🐾	2013/02/07

Knowledge Ontologies



```
format-version: 1.2
data-version: 2013-02-14
date: 13:02:2013 16:50
saved-by: tanyaberardini
auto-generated-by: OBO-Edit 2.3
subsetdef: Cross_product_review "Involved_in"
subsetdef: goslim_aspergillus "Aspergillus GO slim"
subsetdef: goslim_candida "Candida GO slim"
subsetdef: goslim_generic "Generic GO slim"
subsetdef: goslim_metagenomics "Metagenomics GO slim"
subsetdef: goslim_pir "PIR GO slim"
subsetdef: goslim_plant "Plant GO slim"
subsetdef: goslim_pombe "Fission yeast GO slim"
subsetdef: goslim_yeast "Yeast GO slim"
subsetdef: gosubset_prok "Prokaryotic GO subset"
subsetdef: high_level_annotation_qc "High-level terms not to be used for direct annotation"
subsetdef: mf_needs_review "Catalytic activity terms in need of attention"
subsetdef: termgenie_unvetted "Terms created by TermGenie that do not follow a template and req
subsetdef: virus_checked "Viral overhaul terms"
synonymtypedef: systematic_synonym "Systematic synonym" EXACT
default-namespace: gene_ontology
remark: cvs version: $Revision: 6348 $
ontology: go

[Term]
id: GO:0000001
name: mitochondrion inheritance
namespace: biological_process
def: "The distribution of mitochondria, including the mitochondrial genome, into daughter cells
synonym: "mitochondrial inheritance" EXACT []
is_a: GO:0048308 ! organelle inheritance
is_a: GO:0048311 ! mitochondrion distribution

[Term]
id: GO:0000002
name: mitochondrial genome maintenance
namespace: biological_process
def: "The maintenance of the structure and integrity of the mitochondrial genome; includes repl
is_a: GO:0007005 ! mitochondrion organization
```

How to Define an Ontology?

Ontological language must represent:

Categories: Groups

Composition: *PartOf(Bucharest, Romania)*
Can define hierarchical **taxonomy**

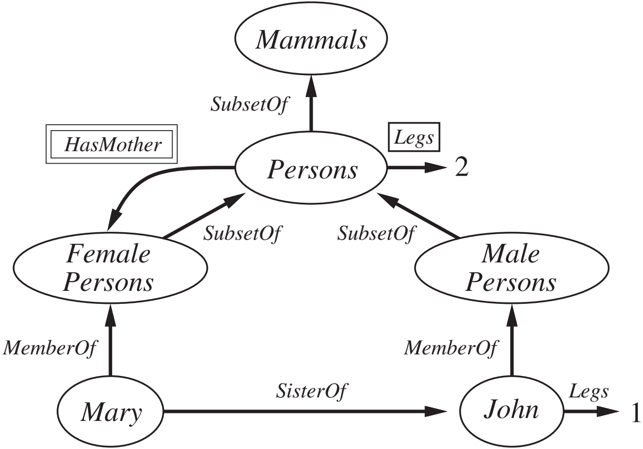
Relations: Between objects

Events, Processes: *Happens(e, i)*

Quantities: *Centimeters(*3.81*)*
(Continuous values are problematic)

Time: *Interval(i)*, *Time(Begin(*1987*))*

Semantic Nets



Never took off, better write it with **description logic**

Semantic Web

Standards for machine readable knowledge for the web exist:

OWL:	Description logic
RDP:	Relational logic

Semantic Web

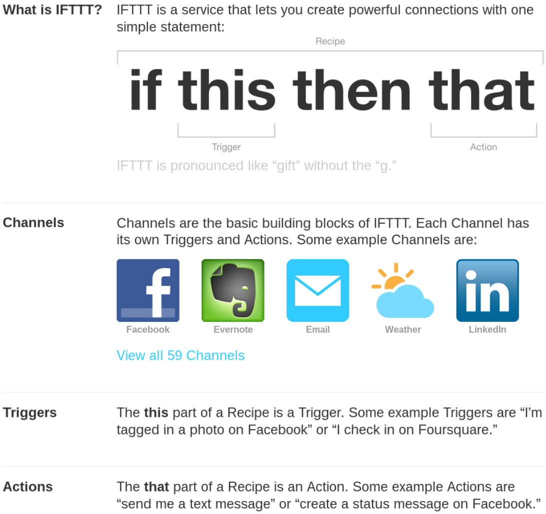
Standards for machine readable knowledge for the web exist:

OWL:	Description logic
RDP:	Relational logic

- But they are not widely used (except for knowledge bases)
- Other web agents are emerging

Web Agents

- Crawlers, IFTTT, Yahoo pipes



What is IFTTT? IFTTT is a service that lets you create powerful connections with one simple statement:

Recipe

if this then that

Trigger

Action

IFTTT is pronounced like "gift" without the "g."

Channels Channels are the basic building blocks of IFTTT. Each Channel has its own Triggers and Actions. Some example Channels are:

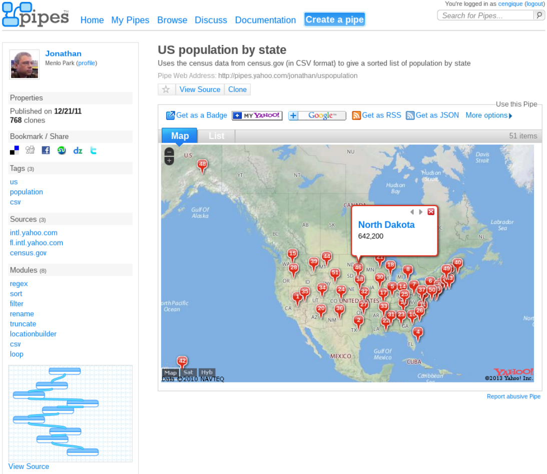
View all 59 Channels

Triggers The **this** part of a Recipe is a Trigger. Some example Triggers are "I'm tagged in a photo on Facebook" or "I check in on Foursquare."

Actions The **that** part of a Recipe is an Action. Some example Actions are "send me a text message" or "create a status message on Facebook."

Web Agents

- Crawlers, IFTTT, Yahoo pipes



pipes™ Home My Pipes Browse Discuss Documentation Create a pipe

You're logged in as cengique (logout)

Search for Pipes...

Jonathan

Menlo Park (profile)

Properties

Published on **12/21/11**

768 clones

Bookmark / Share

Tags (3)

us

population

csv

Sources (3)

intl.yahoo.com

fl.intl.yahoo.com

census.gov

Modules (8)

regex

sort

filter

rename

truncate

locationbuilder

csv

loop

View Source

US population by state

Uses the census data from census.gov (in CSV format) to give a sorted list of population by state

Pipe Web Address: http://pipes.yahoo.com/jonathan/uspopulation

☆ View Source Clone

Use this Pipe

Get as a Badge MY YAHOO! + Google Get as RSS Get as JSON More options ▸

Map List 51 items

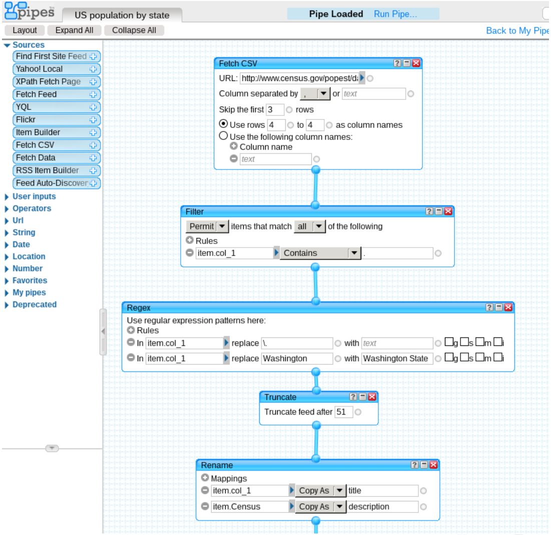
North Dakota

642,200

Report abusive Pipe

Web Agents

- Crawlers, IFTTT, Yahoo pipes



Fetch CSV

URL: http://www.census.gov/popest/d

Column separated by , or text

Skip the first 3 rows

● Use rows 4 to 4 as column names

○ Use the following column names:

Column name

text

Filter

Permit items that match all of the following

Rules

item.col_1 Contains .

Regex

Use regular expression patterns here:

Rules

In item.col_1 replace \. with text □g □s □m □i

In item.col_1 replace Washington with Washington State □g □s □m □i

Truncate

Truncate feed after 51

Rename

Mappings

item.col_1 Copy As title

item.Census Copy As description

References

Diederich J, Günay C, Hogan J (2010). *Recruitment Learning*. Springer-Verlag

Feldman JA (1982). Dynamic connections in neural networks. *Biol Cybern*, 46:27–39

O'Reilly RC, Busby RS, Soto R (2003). Three forms of binding and their neural substrates: Alternatives to temporal synchrony. In Cleeremans A, ed., *The Unity of Consciousness: Binding, Integration and Dissociation*. Oxford University Press, Oxford

Quiroga R, Reddy L, Kreiman G, et al. (2005). Invariant visual representation by single neurons in the human brain. *Nature*, 435(7045):1102–1107

Stark E, Globerson A, Asher I, et al. (2008). Correlations between Groups of Premotor Neurons Carry Information about Prehension. *J Neurosci*, 28(42):10618–10630