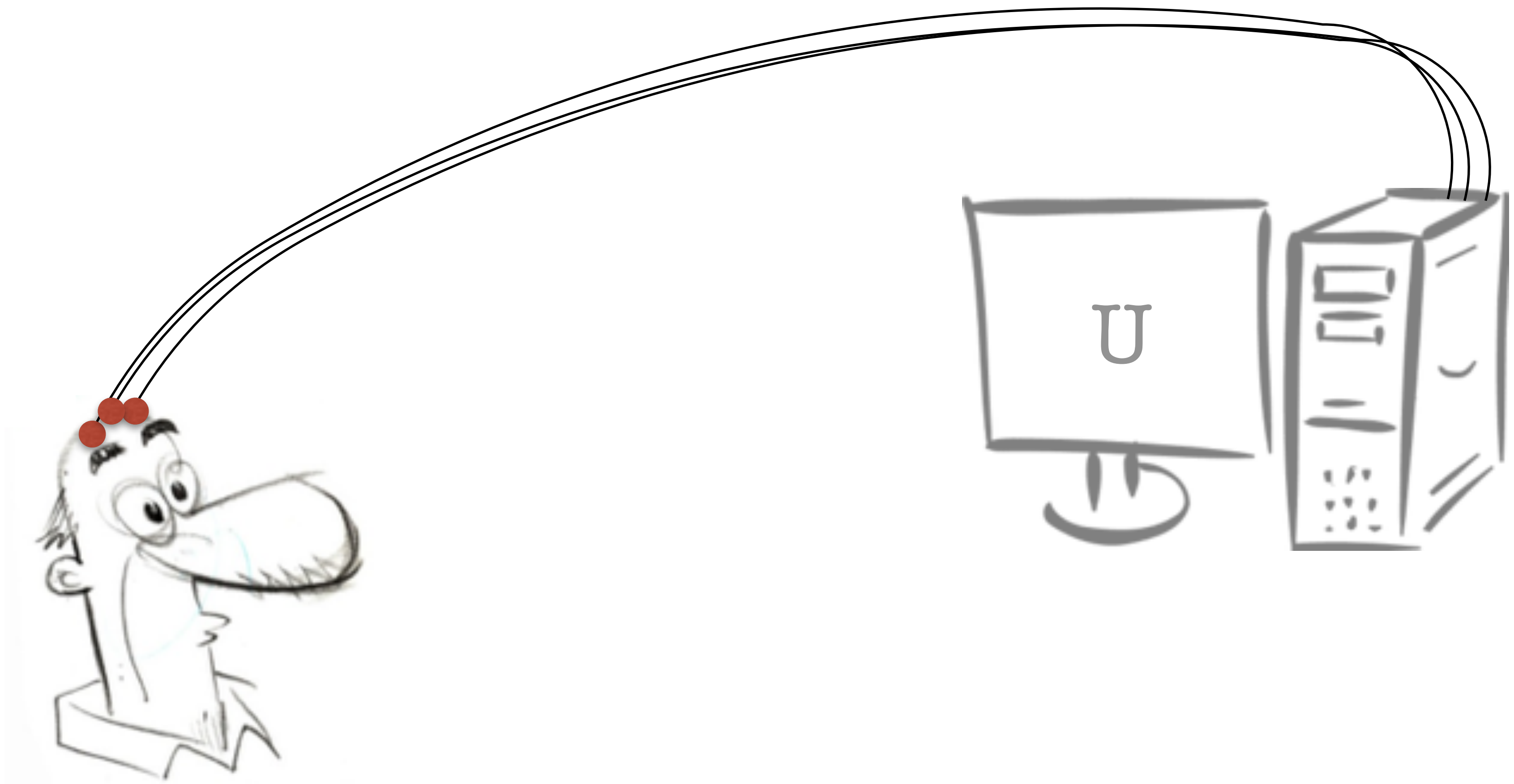
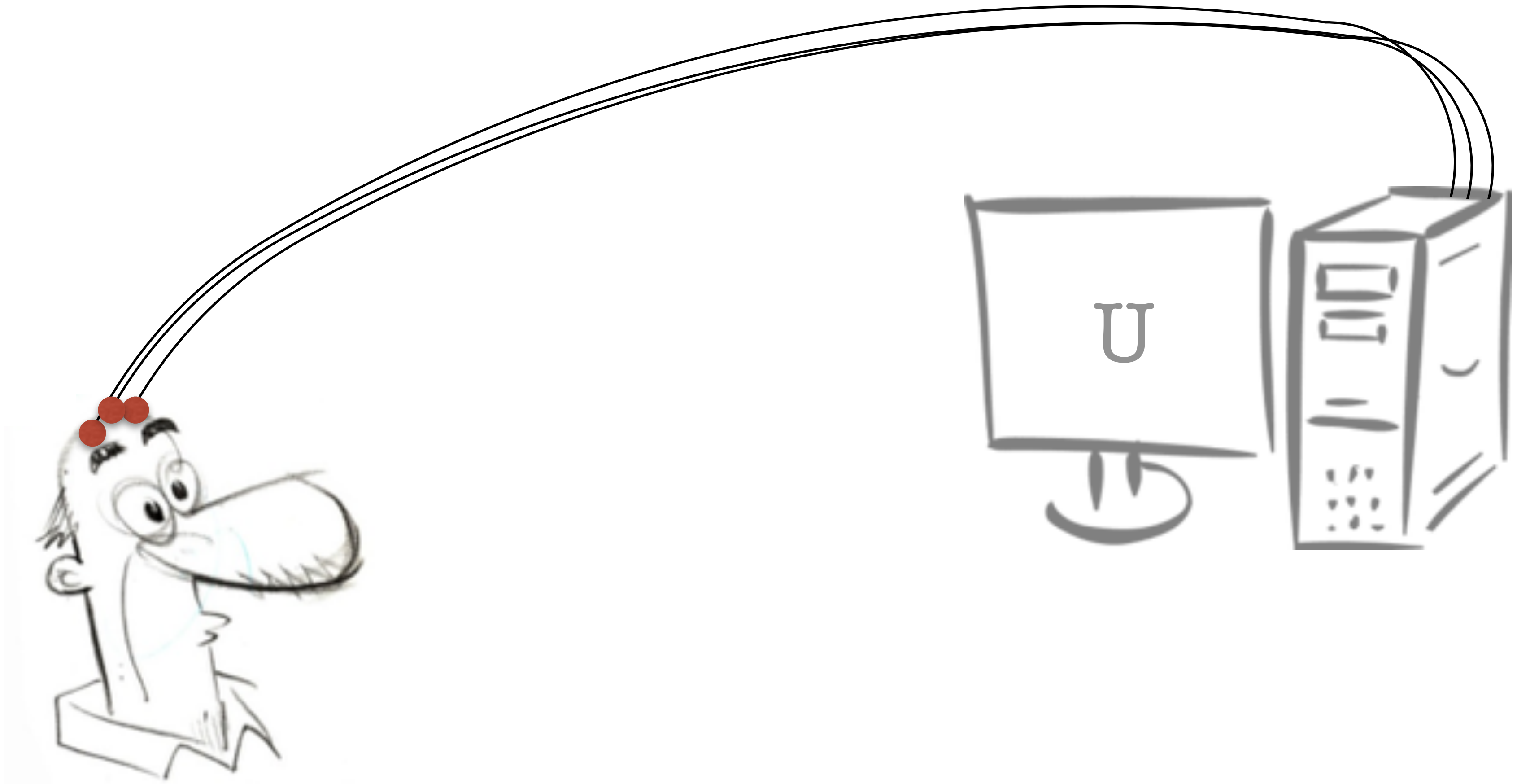
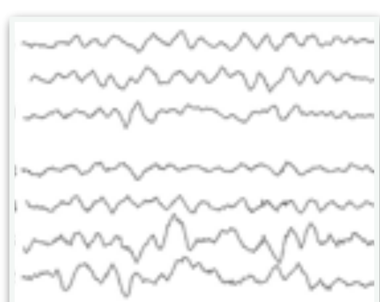
Language Models in BCI systems

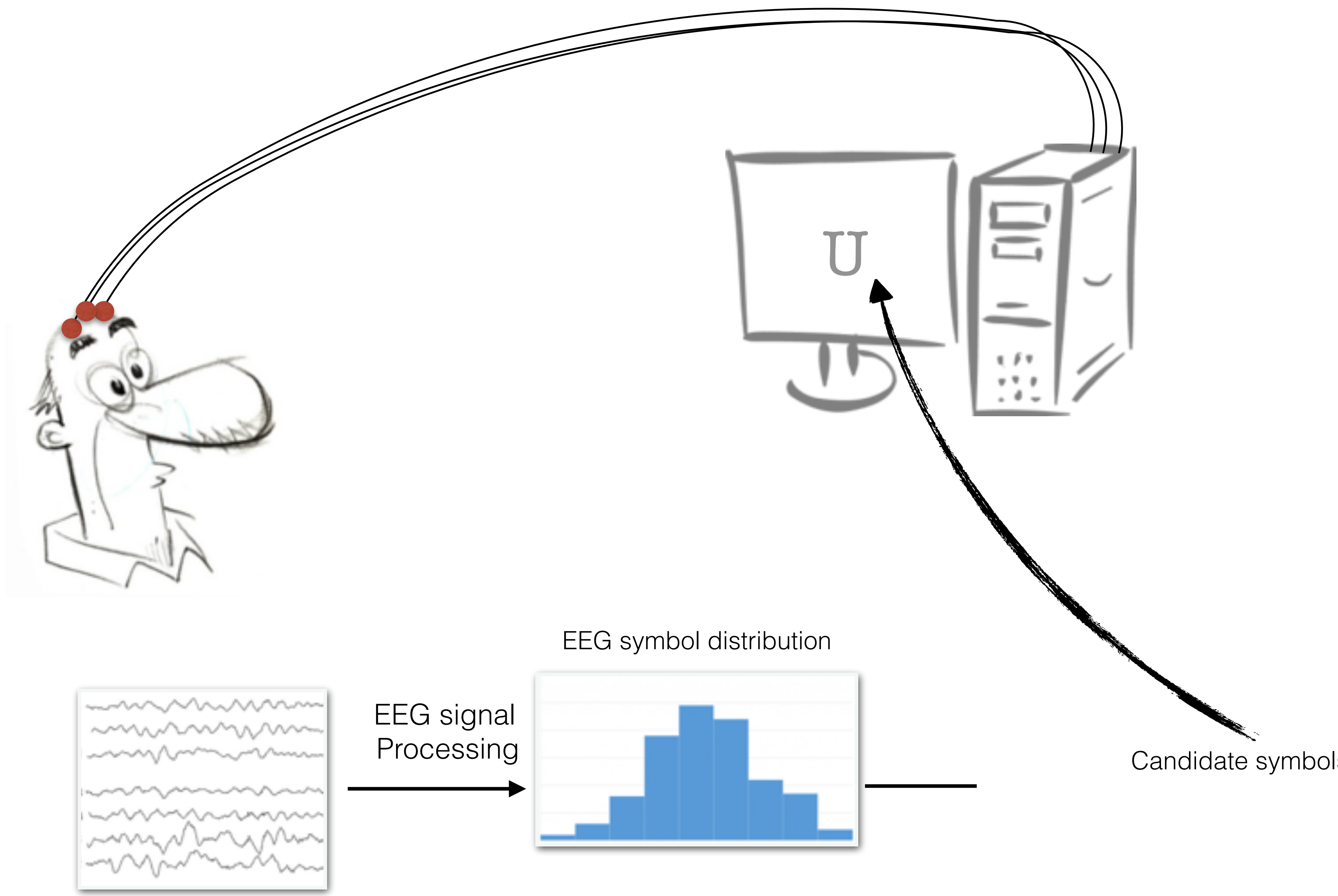
Shiran Dudy

April 23, 2019


U


U




U
EEG symbol distribution
EEG signal
Processing
Candidate symbols

Naive approach:

Signal acquisition and processing for each of the symbols in the system

Naive approach:

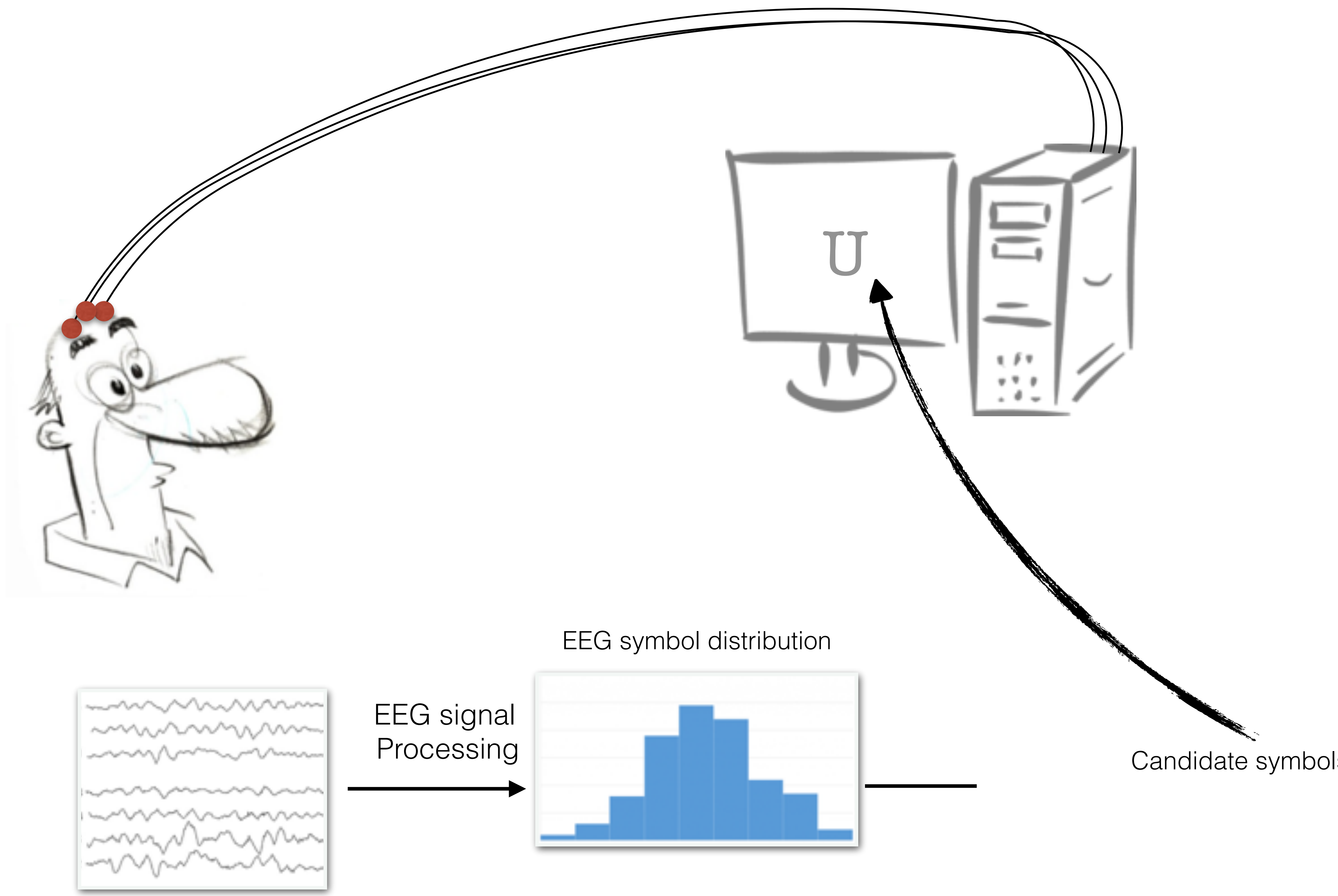
Signal acquisition and processing for each of the symbols in the system

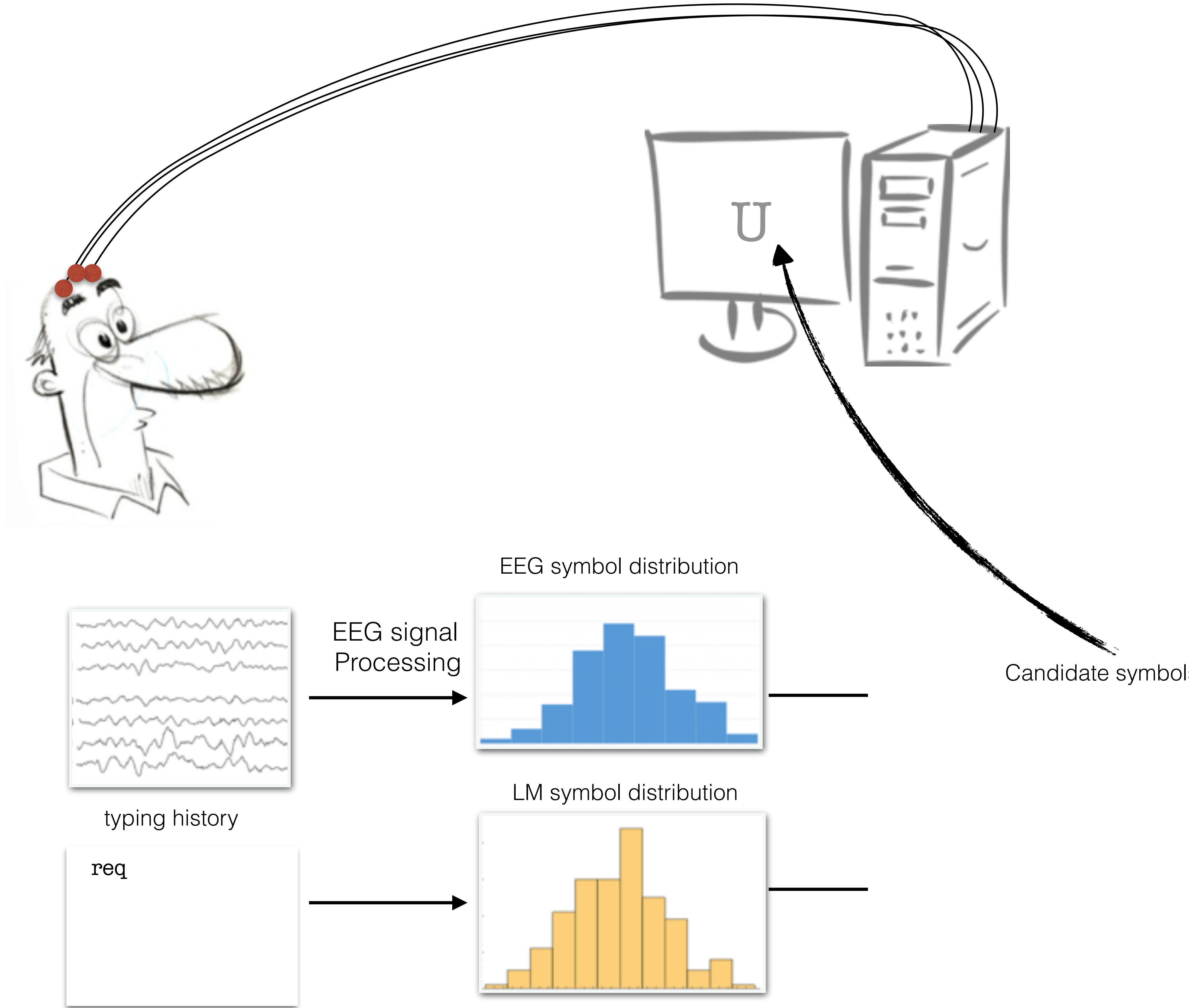
Naive approach:

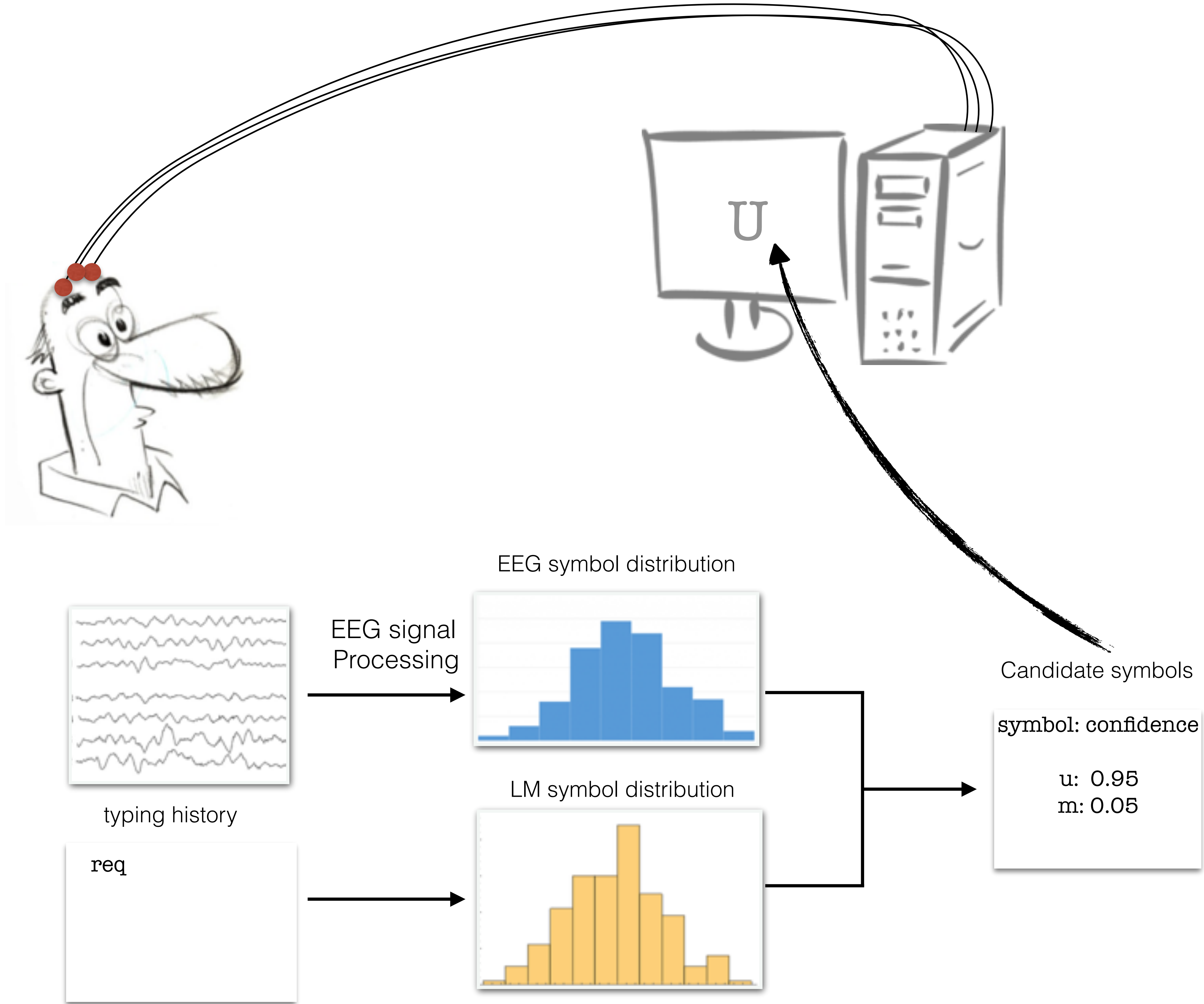
Signal acquisition and processing for each of the symbols in the system

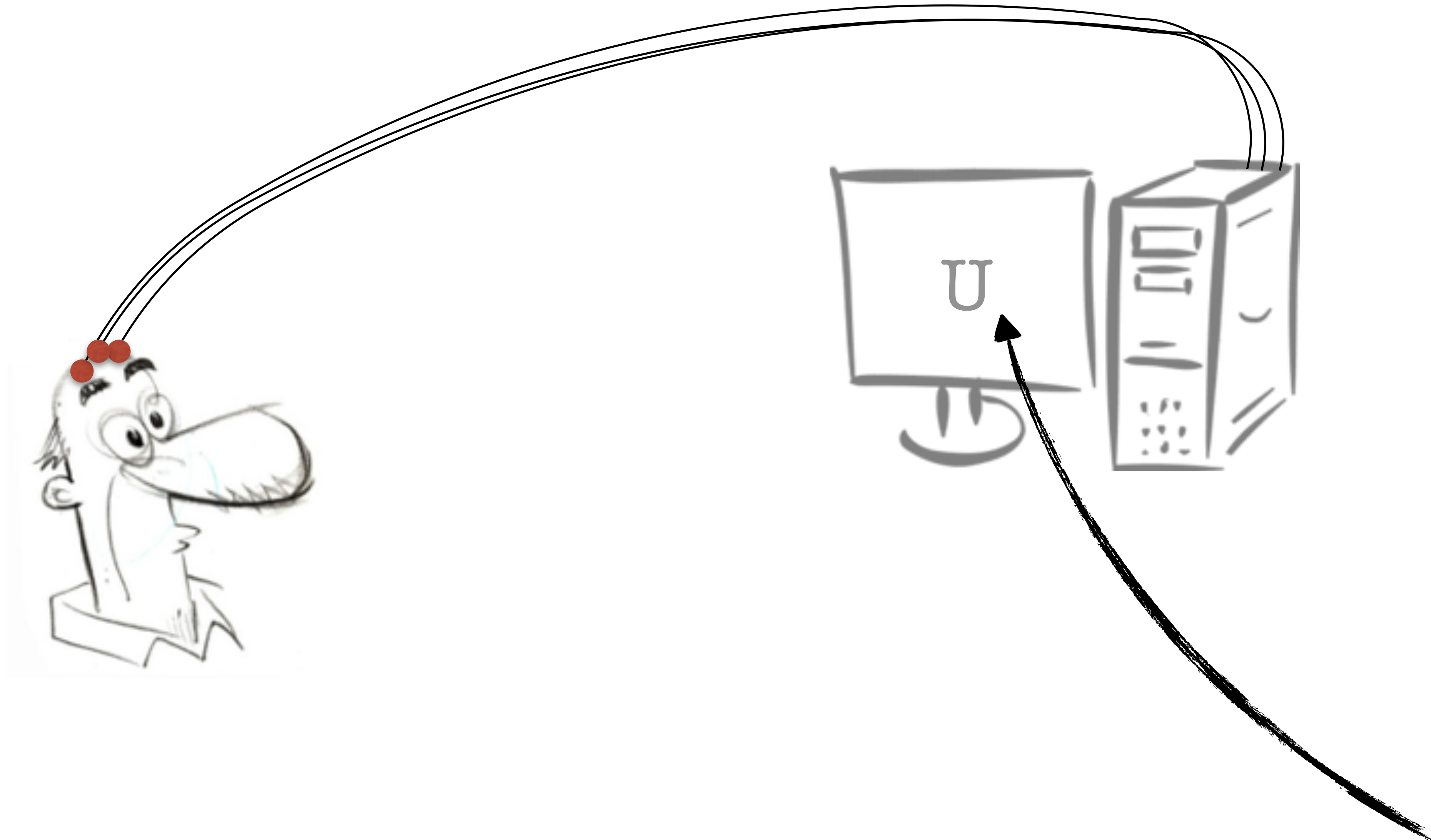
Language Model (LM) approach:

Signal acquisition and processing for fewer symbols than overall symbols using language patterns


U
EEG symbol distribution
EEG signal
Processing
Candidate symbols


U
EEG symbol distribution
EEG signal
Processing
Candidate symbols
typing history
req
LM symbol distribution


U
EEG symbol distribution
EEG signal
Processing
Candidate symbols
symbol: confidence
u: 0.95
m: 0.05
LM symbol distribution
typing history
req

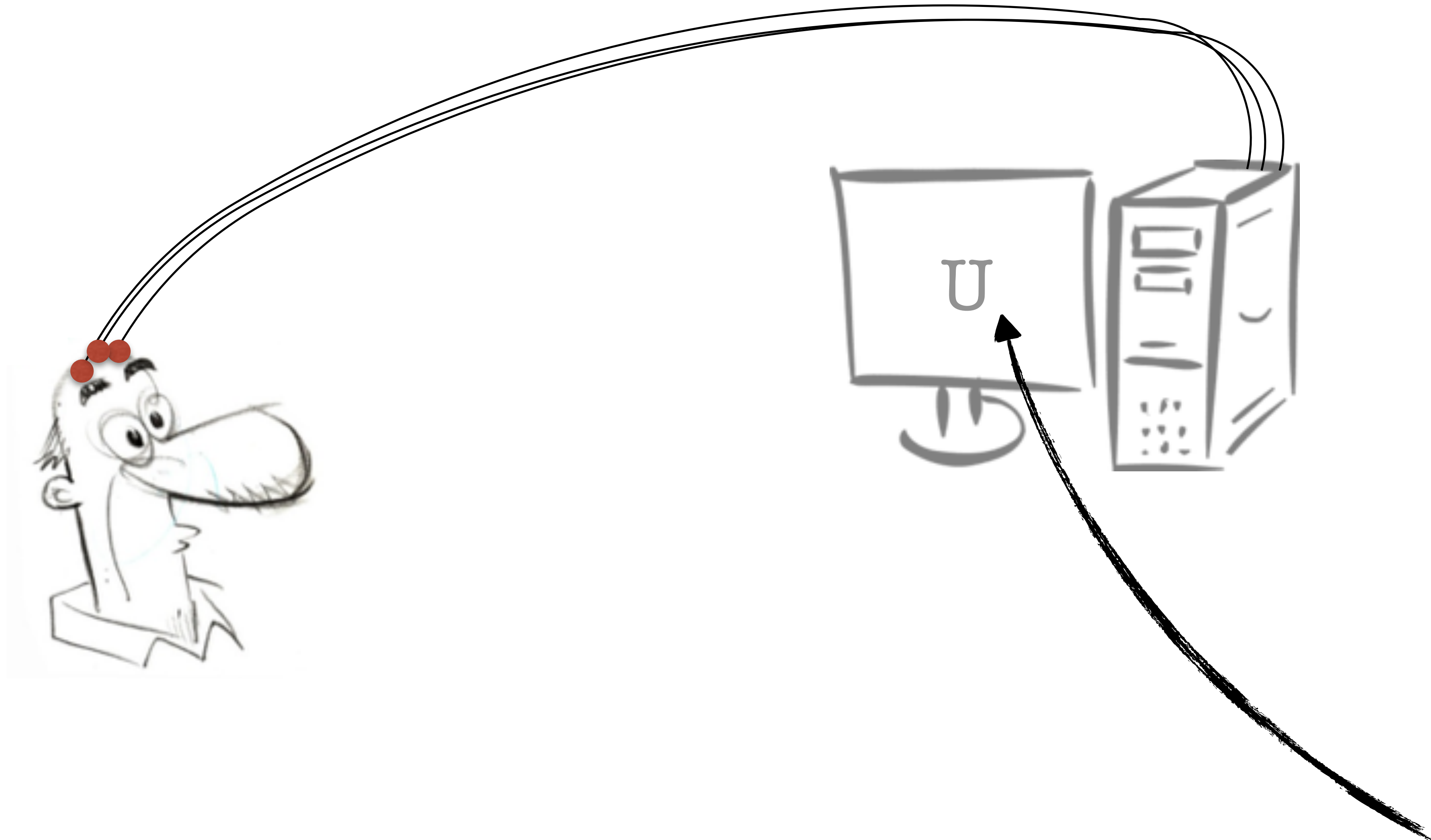


Candidate symbols

$$P(sym|EEG) = \frac{P(sym)P(EEG|sym)}{P(EEG)}$$

symbol: confidence

u: 0.95
m: 0.05



Candidate symbols

$$P(sym|EEG) \propto P(sym)P(EEG|sym)$$

symbol: confidence

u: 0.95
m: 0.05

What are Language Models?

Language Modeling Module

A statistical **language model** is a probability distribution over sequences of tokens (words is a special case)

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i \mid w_1,\ldots,w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)},\ldots,w_{i-1})$$

Language Modeling Module

A statistical **language model** is a probability distribution over sequences of tokens (words is a special case)

$$P(w_1, \ldots, w_m) = \prod_{i=1}^{m} P(w_i \mid w_1, \ldots, w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)}, \ldots, w_{i-1})$$

$$p(\text{it is simple}) = ?$$

Language Modeling Module

A statistical **language model** is a probability distribution over sequences of tokens (words is a special case)

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i \mid w_1,\ldots,w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)},\ldots,w_{i-1})$$

$$p(\text{it is simple}) = ?$$

$$p(\text{it is simple}|unigram) = p(\text{it})p(\text{is})p(\text{simple})$$

Language Modeling Module

A statistical **language model** is a probability distribution over sequences of tokens (words is a special case)

$$P(w_1,\ldots,w_m)=\prod_{i=1}^{m}P(w_i\mid w_1,\ldots,w_{i-1})\approx\prod_{i=1}^{m}P(w_i\mid w_{i-(n-1)},\ldots,w_{i-1})$$

$$p(\text{it is simple}) = ?$$
$$p(\text{it is simple}|unigram) = p(\text{it})p(\text{is})p(\text{simple})$$
$$p(\text{it is simple}|bigram) = p(\text{it}| < s >)p(\text{is}|\text{it})p(\text{simple}|\text{is})p(< /s > |\text{simple})$$

Language Modeling Module

A statistical **language model** is a probability distribution over sequences of tokens (words is a special case)

$$P(w_1,\ldots,w_m) = \prod_{i=1}^{m} P(w_i \mid w_1,\ldots,w_{i-1}) \approx \prod_{i=1}^{m} P(w_i \mid w_{i-(n-1)},\ldots,w_{i-1})$$

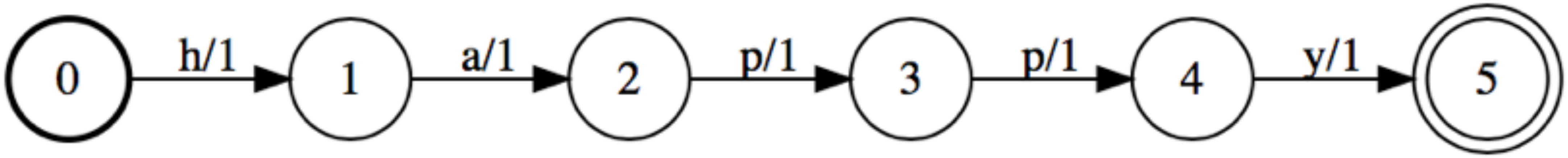
$$p(\text{it is simple}) = ?$$
$$p(\text{it is simple}|unigram) = p(\text{it})p(\text{is})p(\text{simple})$$
$$p(\text{it is simple}|bigram) = p(\text{it}| < s >)p(\text{is}|\text{it})p(\text{simple}|\text{is})p(< /s > |\text{simple})$$

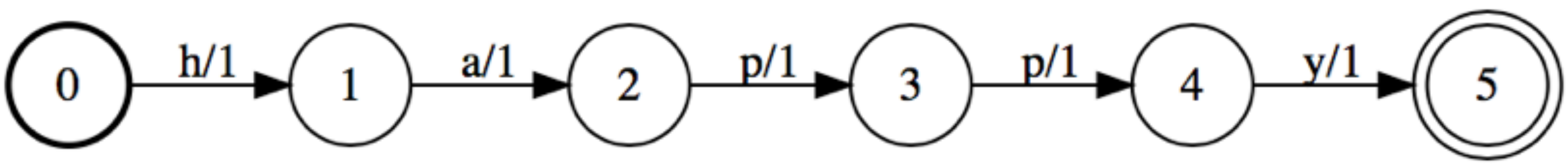
How to define probability distributions over strings?

A Toy Example

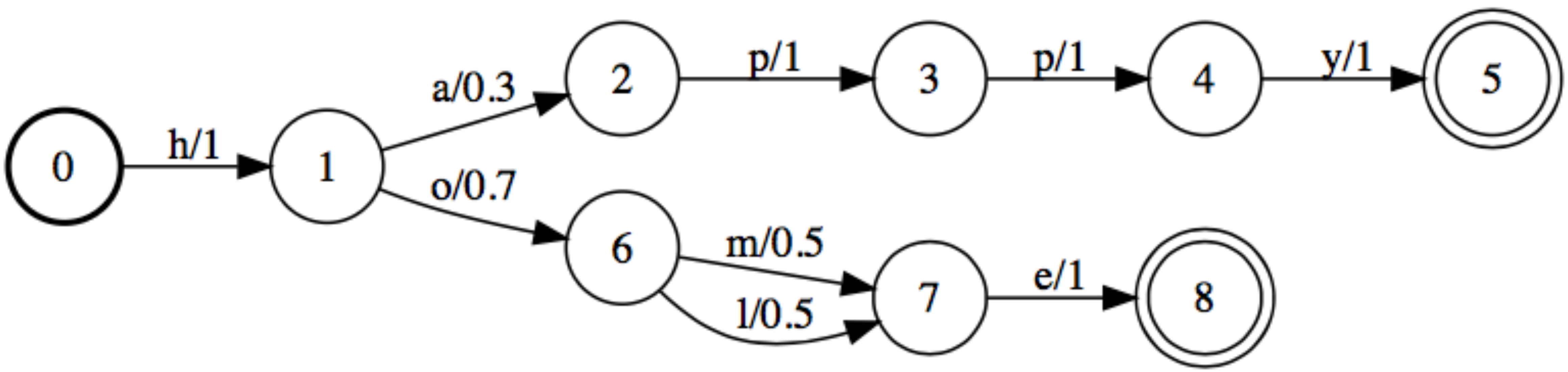
vocabulary:
happy
home
hole

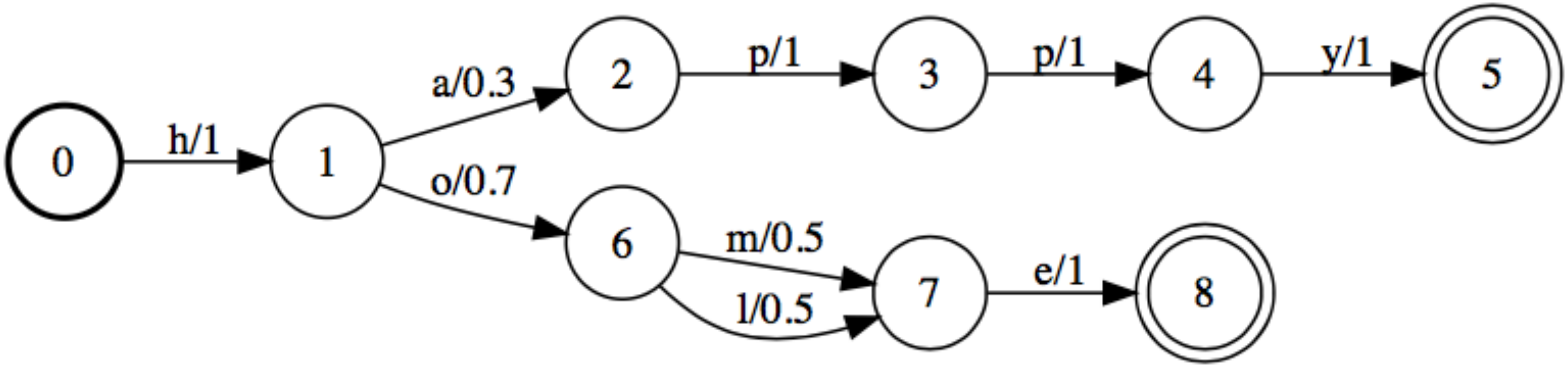
vocabulary:
~~happy~~
home
hole




0
h/1
1
a/1
2
p/1
3
p/1
4
y/1
5

vocabulary:
~~happy~~
~~home~~
~~hole~~



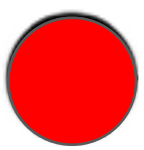


Compact representation of prob. dist. over set of strings
Defined by states, arcs, symbols, and weights

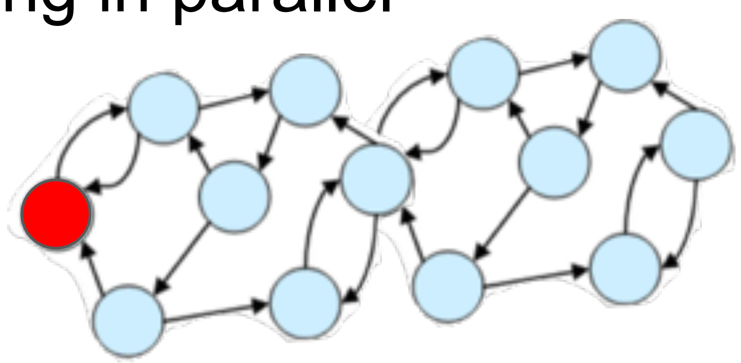
A Basic Language Model

Basic LM Module implementation

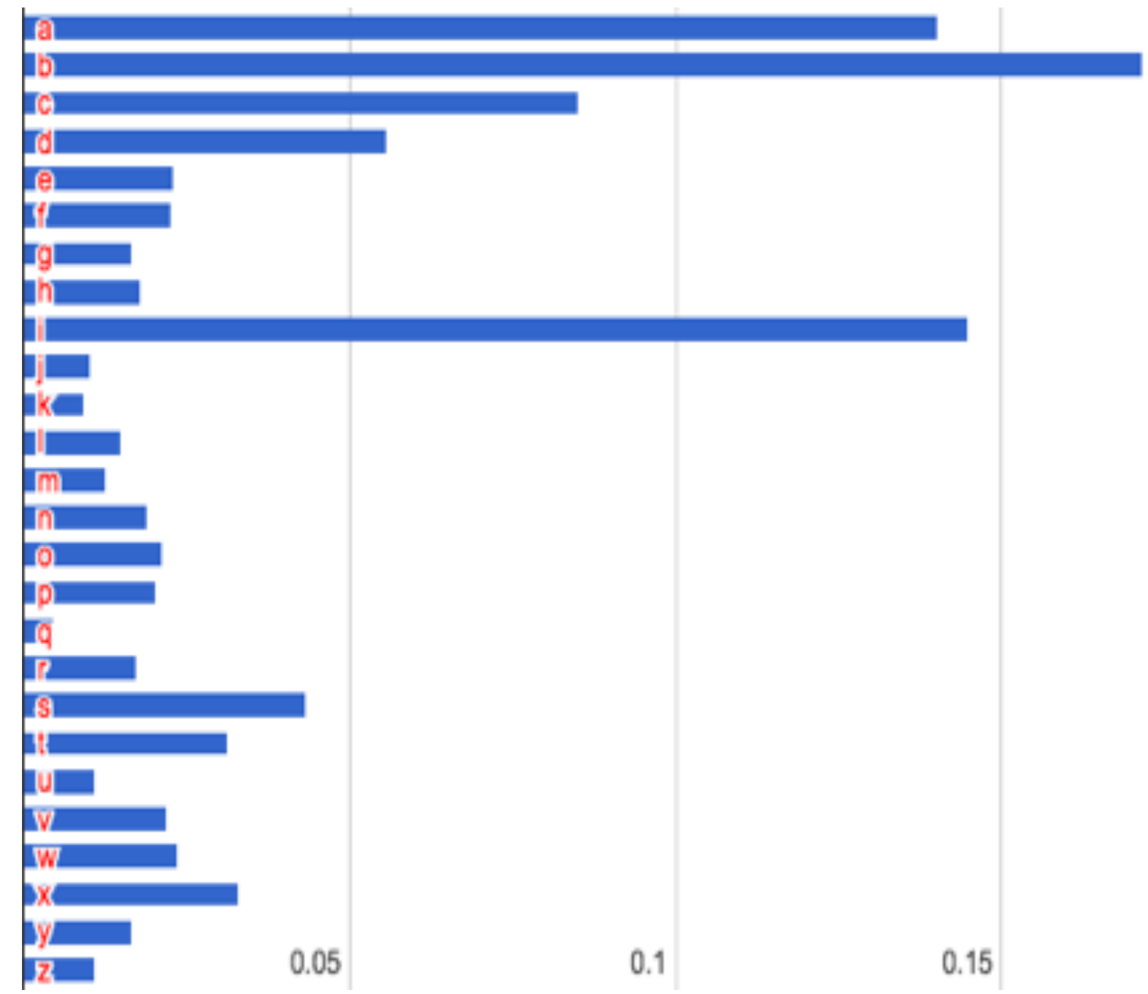
History



Intersect history and LM by walking in parallel



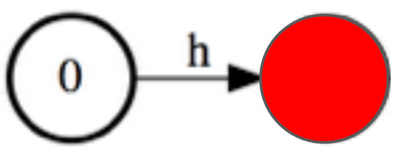
Language Model



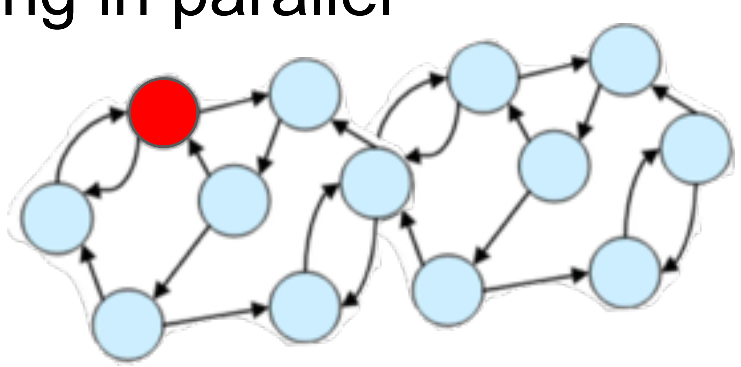
Probability of next character

Basic LM Module implementation

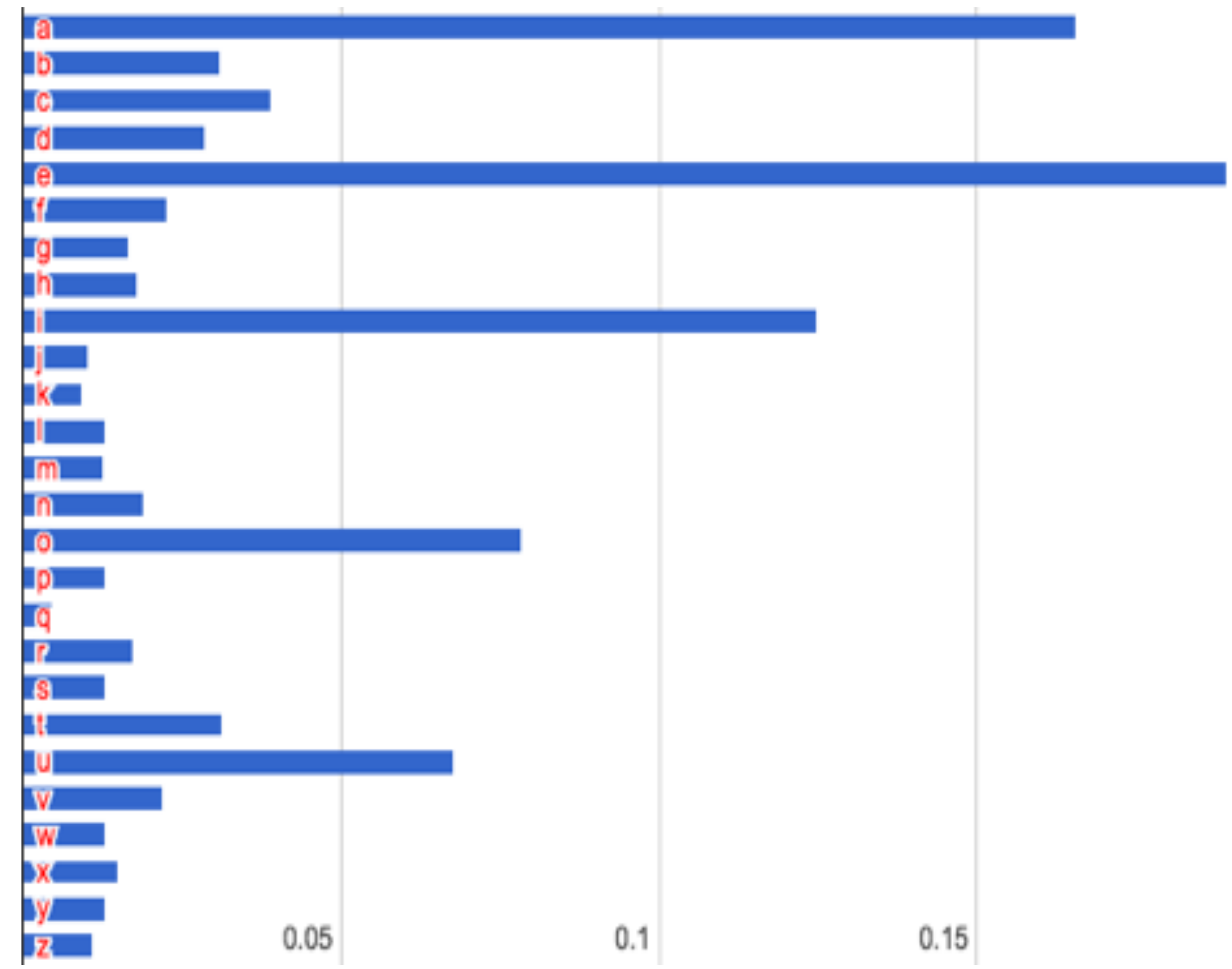
History



Intersect history and LM by walking in parallel



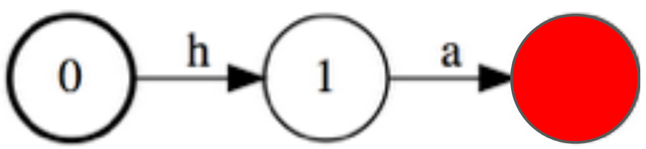
Language Model



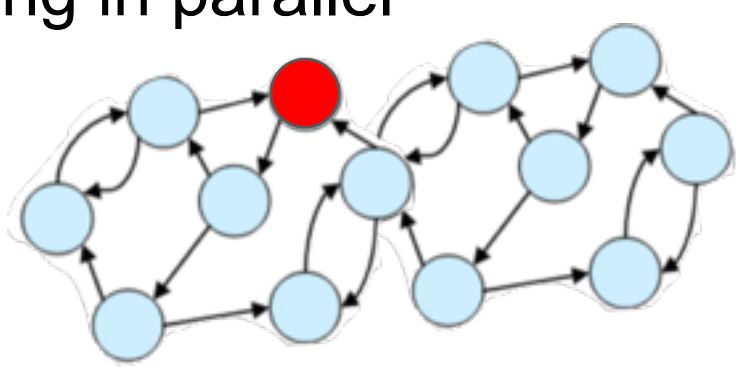
Probability of next character

Basic LM Module implementation

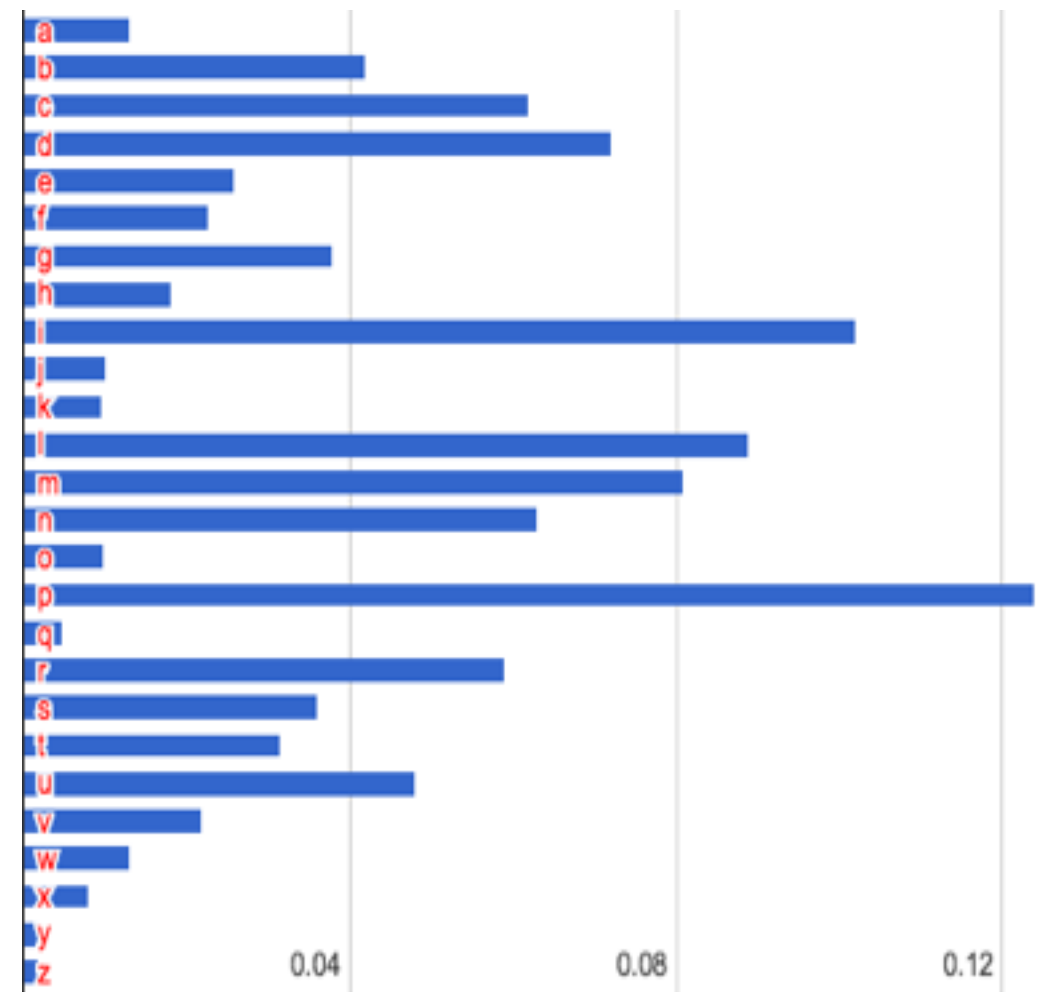
History



Intersect history and LM by walking in parallel



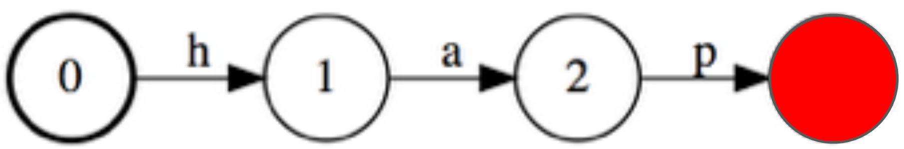
Language Model



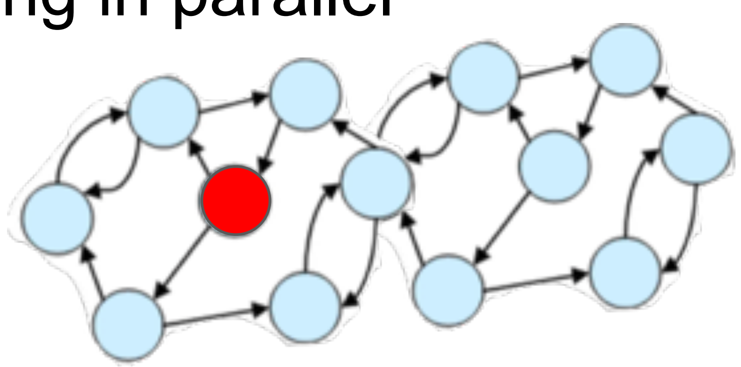
Probability of next character

Basic LM Module implementation

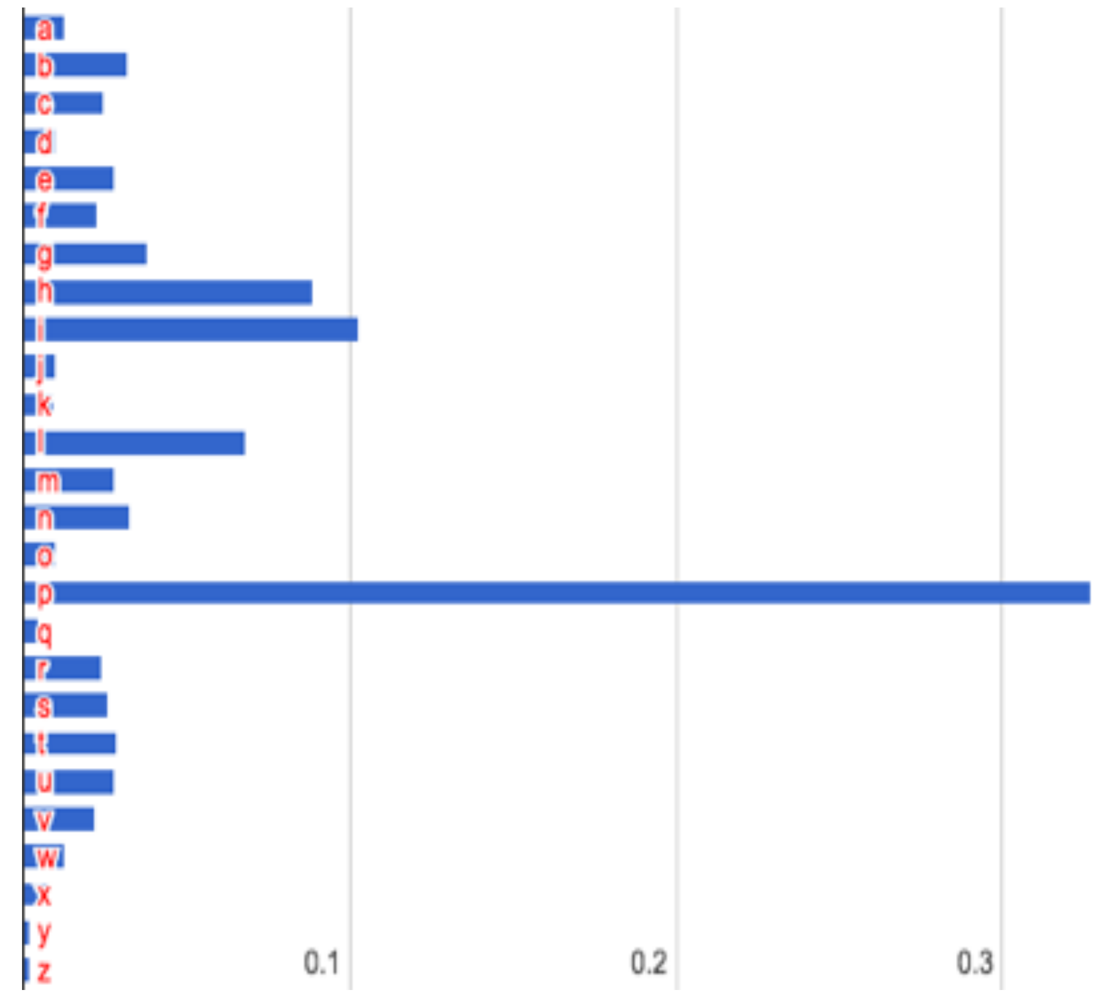
History



Intersect history and LM by walking in parallel



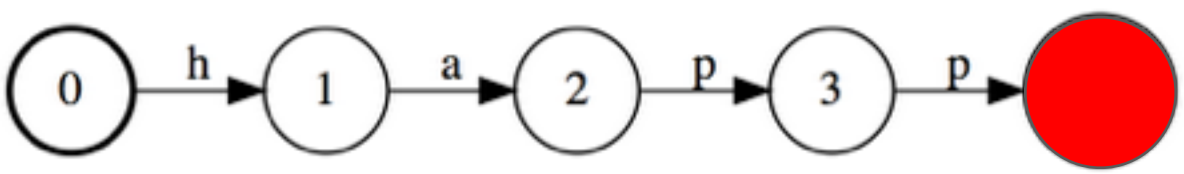
Language Model



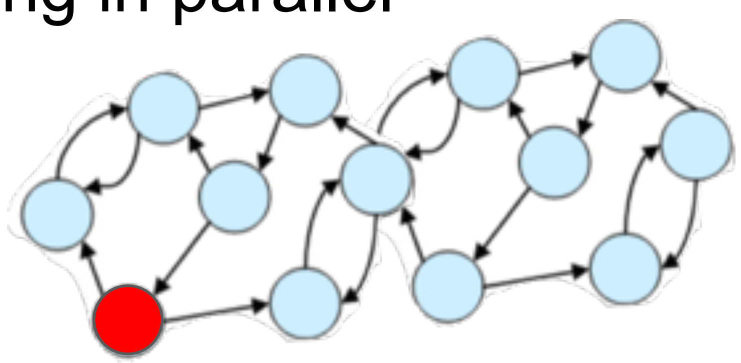
Probability of next character

Basic LM Module implementation

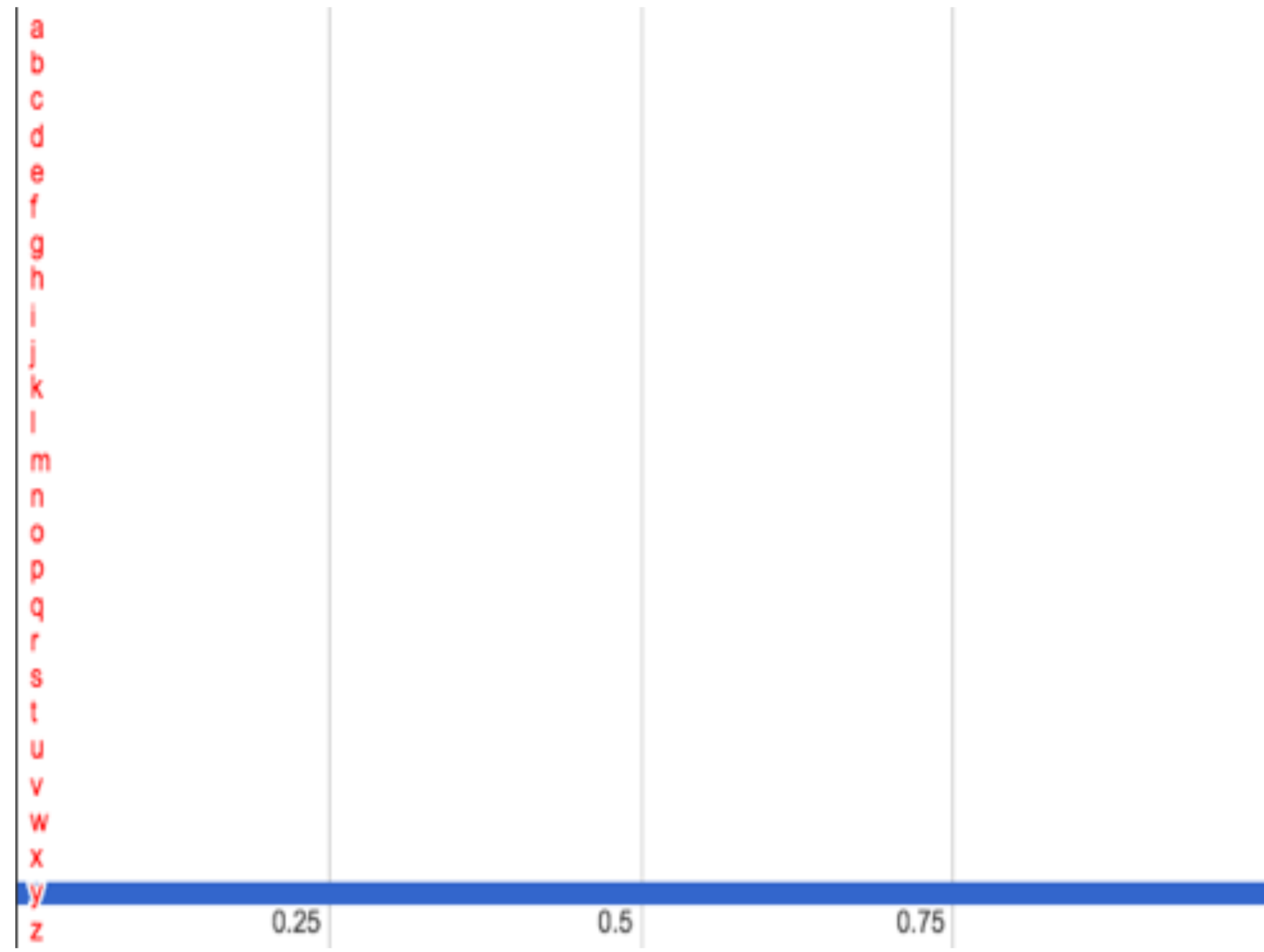
History



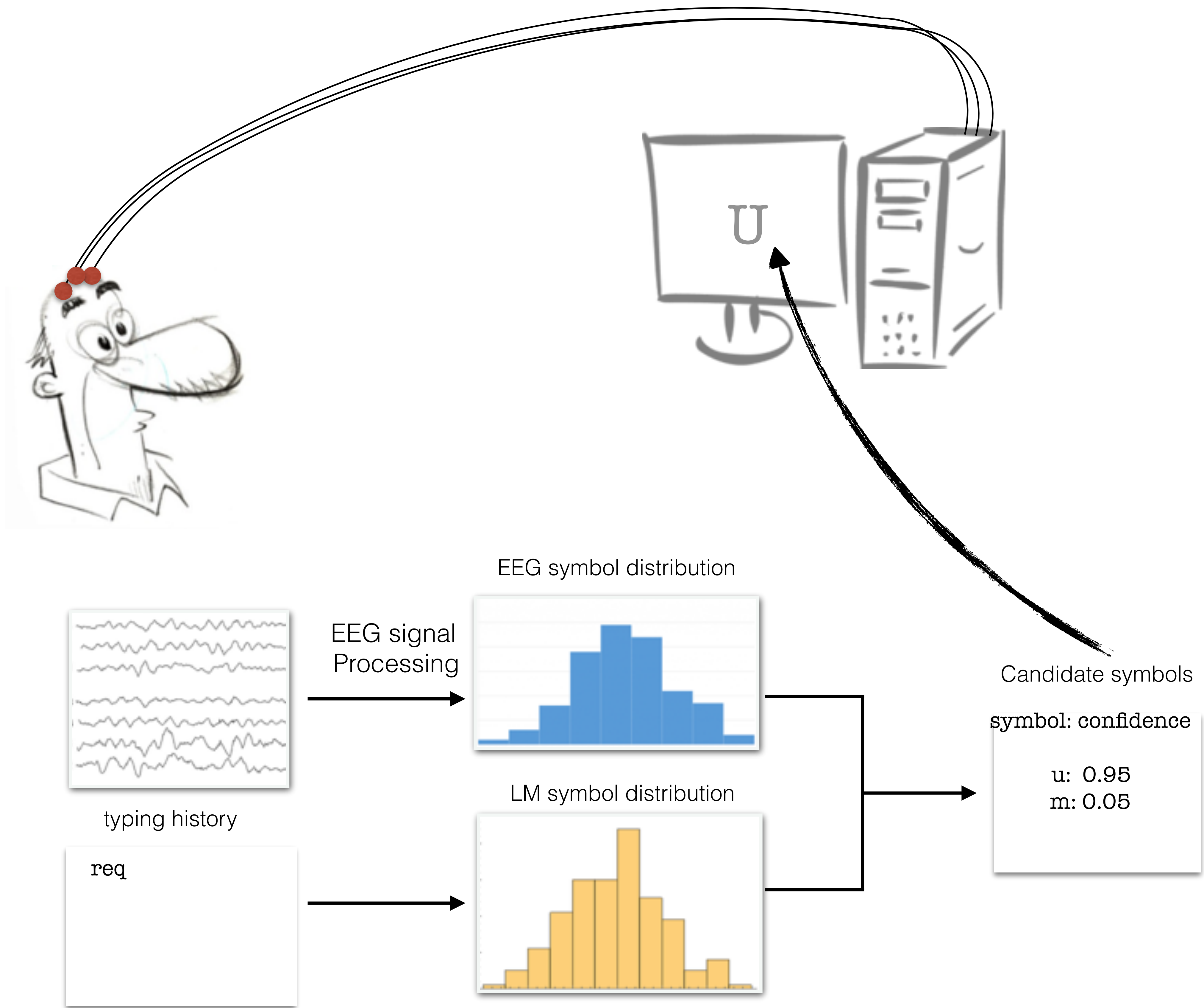
Intersect history and LM by walking in parallel

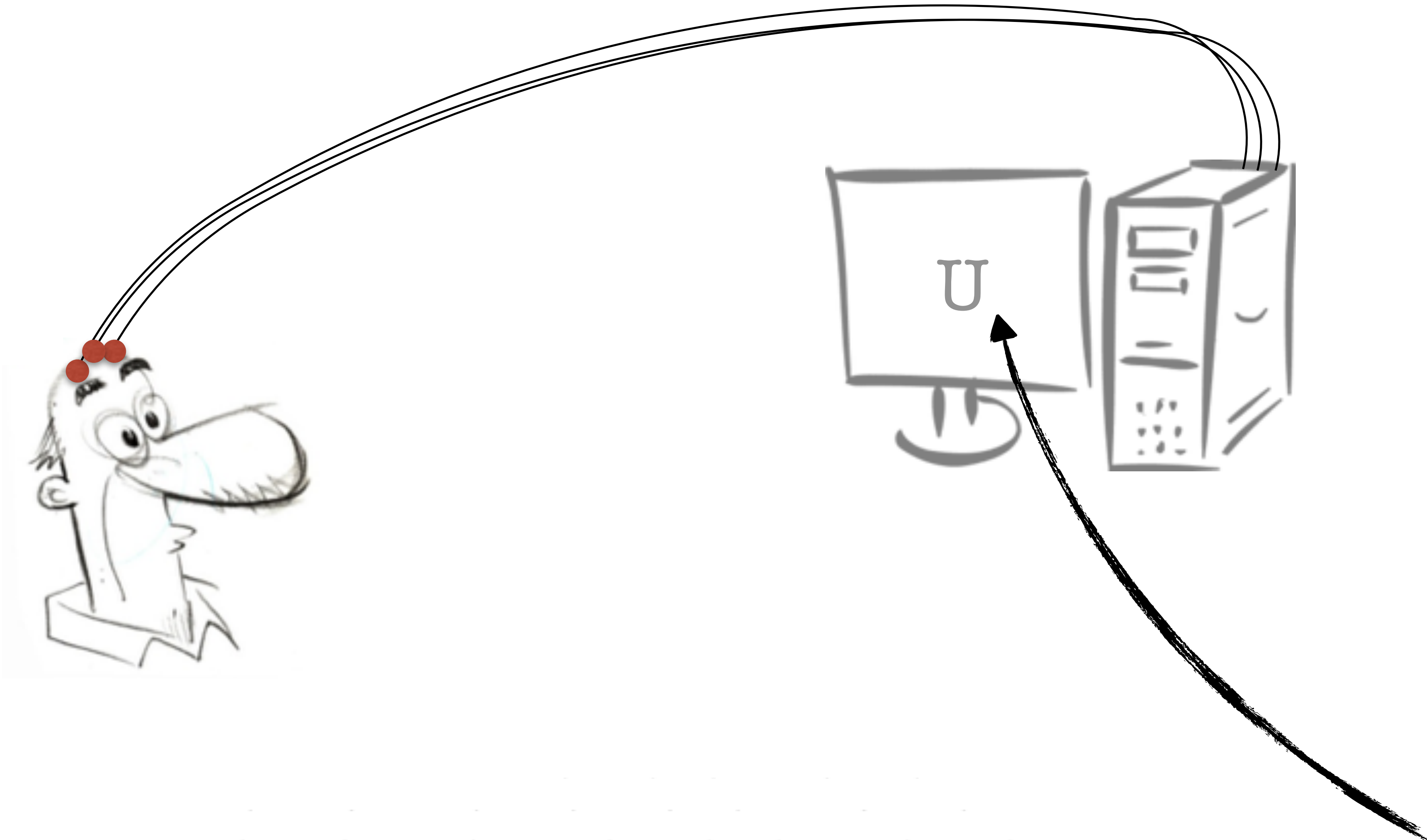


Language Model



Probability of next character


U
EEG symbol distribution
EEG signal
Processing
Candidate symbols
symbol: confidence
u: 0.95
m: 0.05
typing history
req
LM symbol distribution



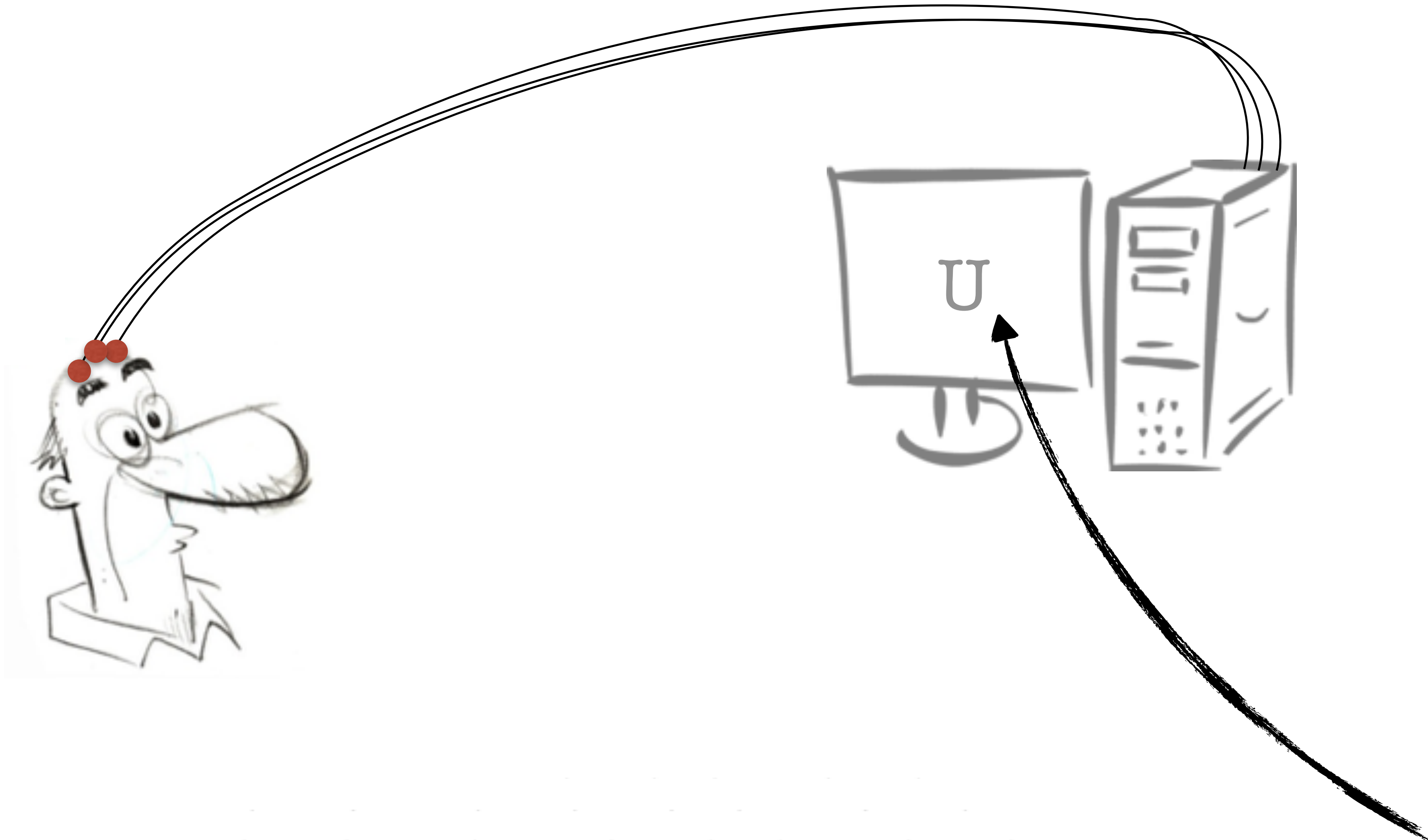
$$P(sym|EEG) \propto P(sym)P(EEG|sym)$$

$$P(sym) = P(sym|X_m)$$
$$X_0 = \{sym_{t-i}\}, i > 0$$
$$X_1 = \{word_{t-i}\}, i > 0$$
$$X_2 = \{EEG_{t-i}\}, i \geq 0$$

Candidate symbols

symbol: confidence

u: 0.95
m: 0.05



$$P(sym|EEG) \propto P(sym)P(EEG|sym)$$

$$P(sym) = P(sym|X_m)$$

✓ $X_0 = \{sym_{t-i}\}, i > 0$

$X_1 = \{word_{t-i}\}, i > 0$

$X_2 = \{EEG_{t-i}\}, i \geq 0$

Candidate symbols

symbol: confidence

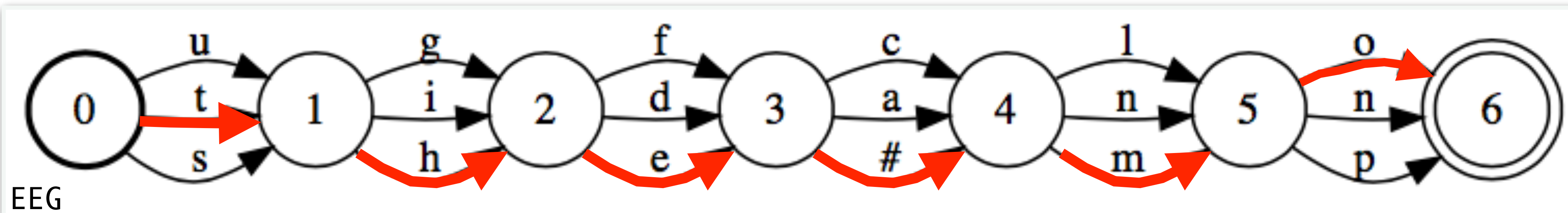
u: 0.95
m: 0.05

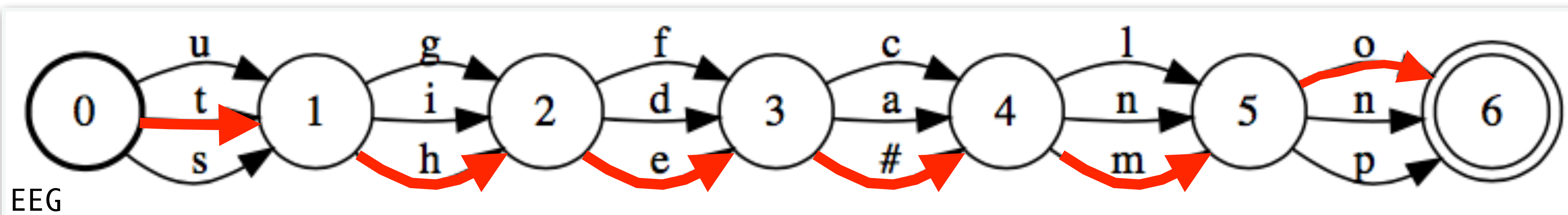
Issues with the basic LM module

In reality the EEGs are a distribution of symbols. While the basic model assumes a deterministic history.

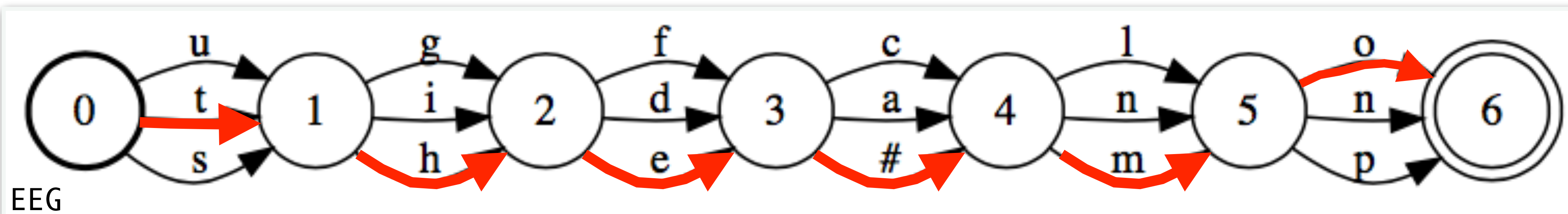
While the basic model learns probabilities from n-grams, it does not incorporate word level information.

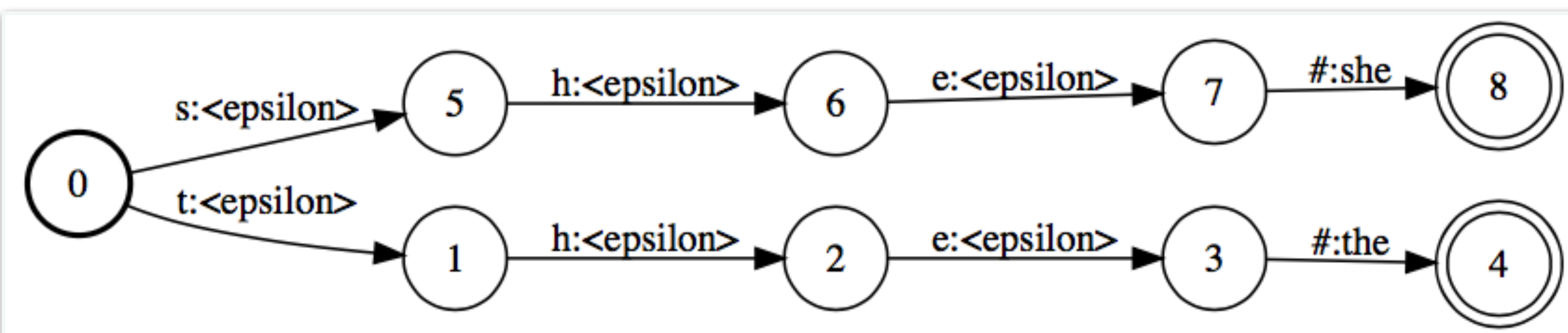
The OCLM Language Model

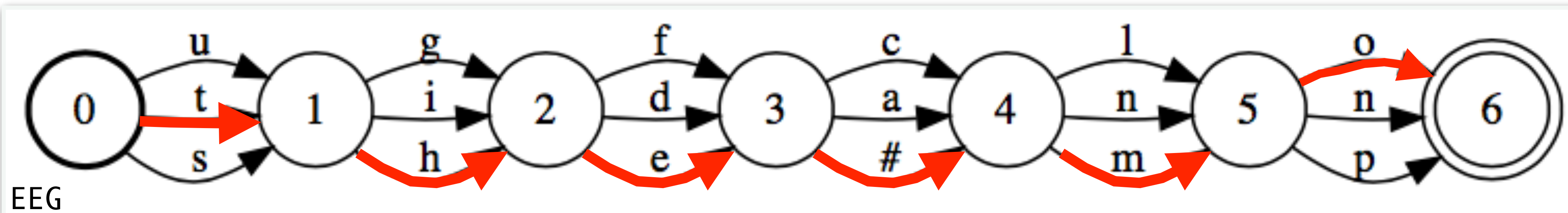

u
t
s
g
i
h
f
d
e
c
a
#
l
n
m
o
n
p
0
1
2
3
4
5
6
EEG

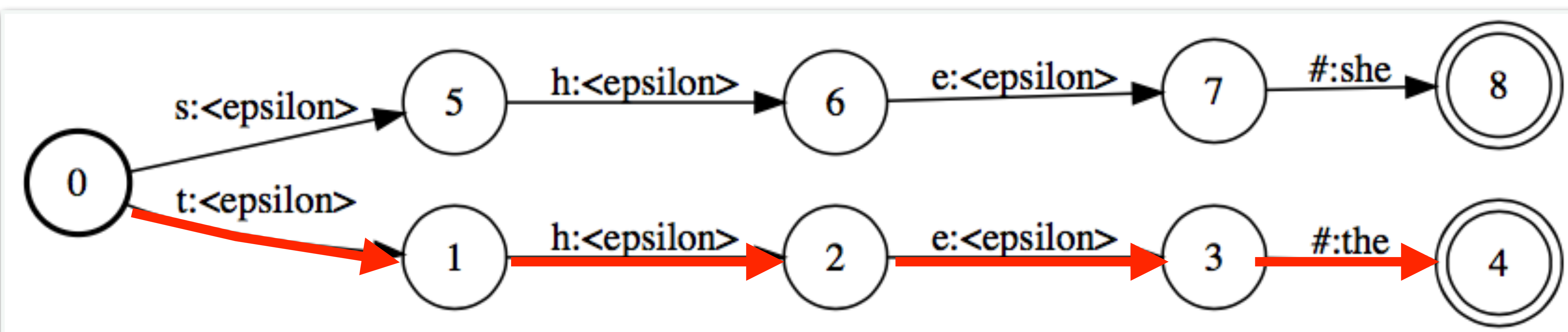


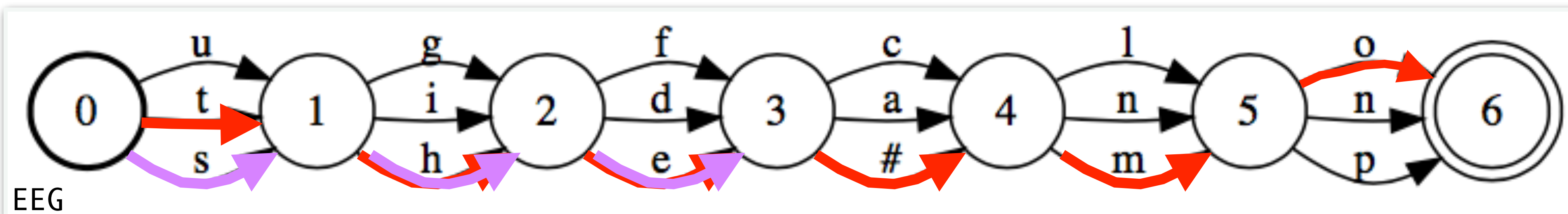
Arcs
contain normalized
EEG probabilities

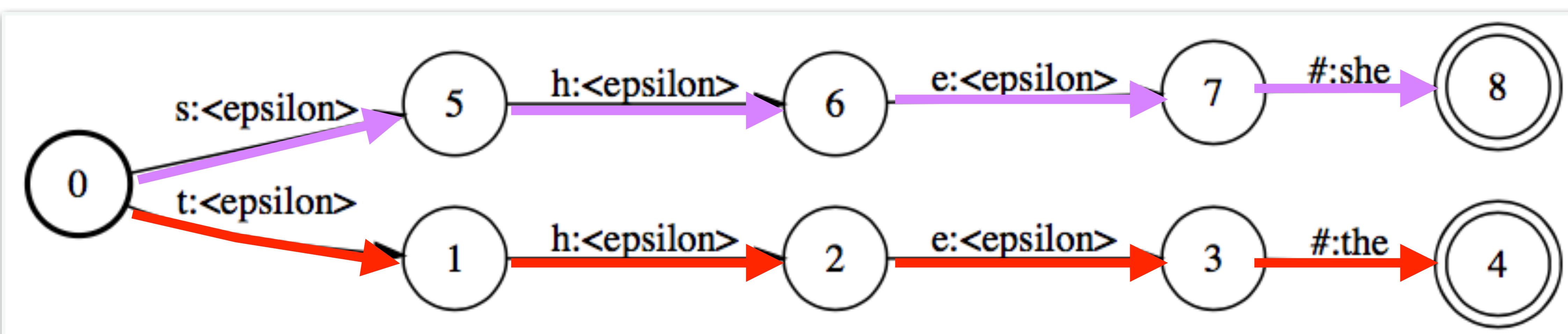

0
u
t
s
1
g
i
h
2
f
d
e
3
c
a
#
4
l
n
m
5
o
n
p
6
EEG


0
s:<epsilon>
5
h:<epsilon>
6
e:<epsilon>
7
#:she
8
t:<epsilon>
1
h:<epsilon>
2
e:<epsilon>
3
#:the
4

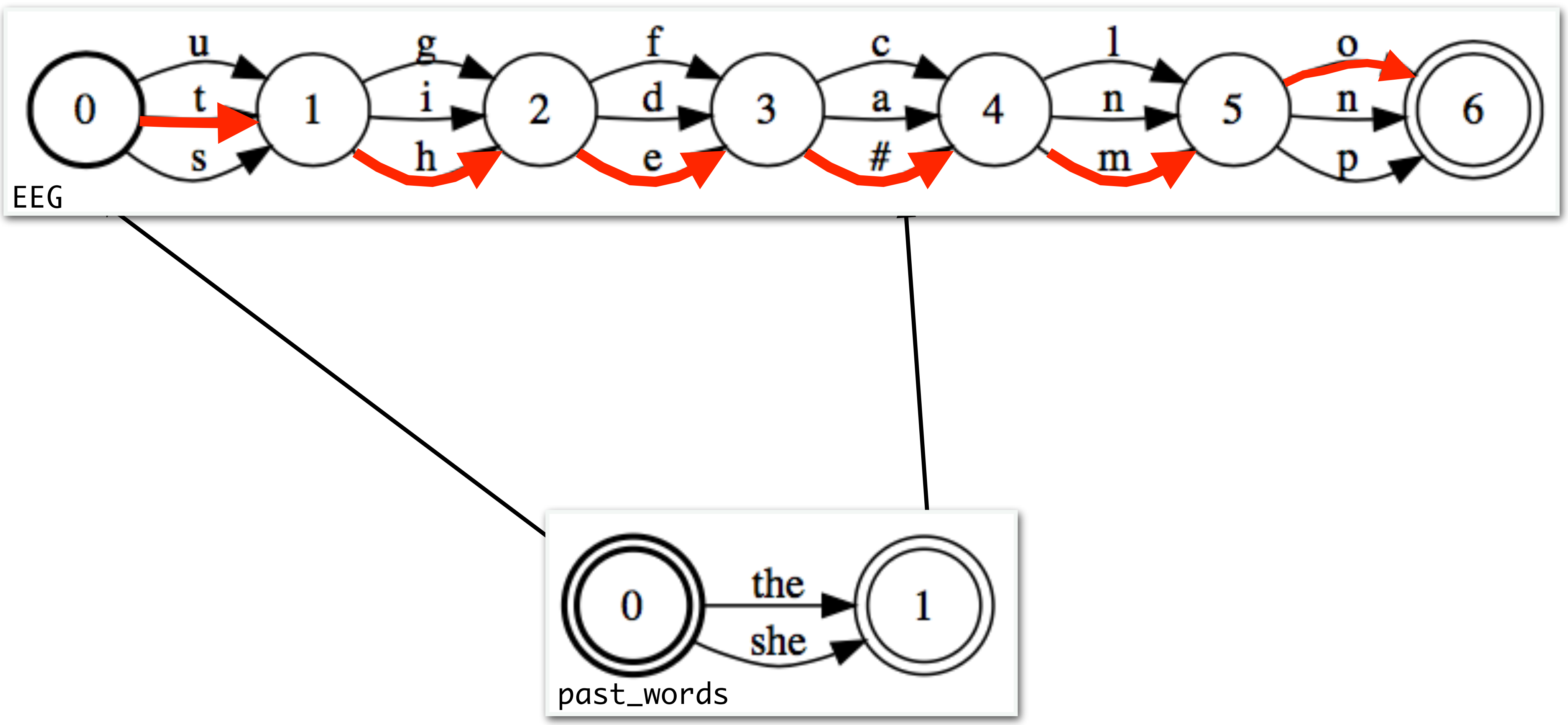

0
u
t
s
1
g
i
h
2
f
d
e
3
c
a
#
4
l
n
m
5
o
n
p
6
EEG


0
s:<epsilon>
5
h:<epsilon>
6
e:<epsilon>
7
#:she
8
t:<epsilon>
1
h:<epsilon>
2
e:<epsilon>
3
#:the
4


0
u
t
s
1
g
i
h
2
f
d
e
3
c
a
#
4
l
n
m
5
o
n
p
6
EEG

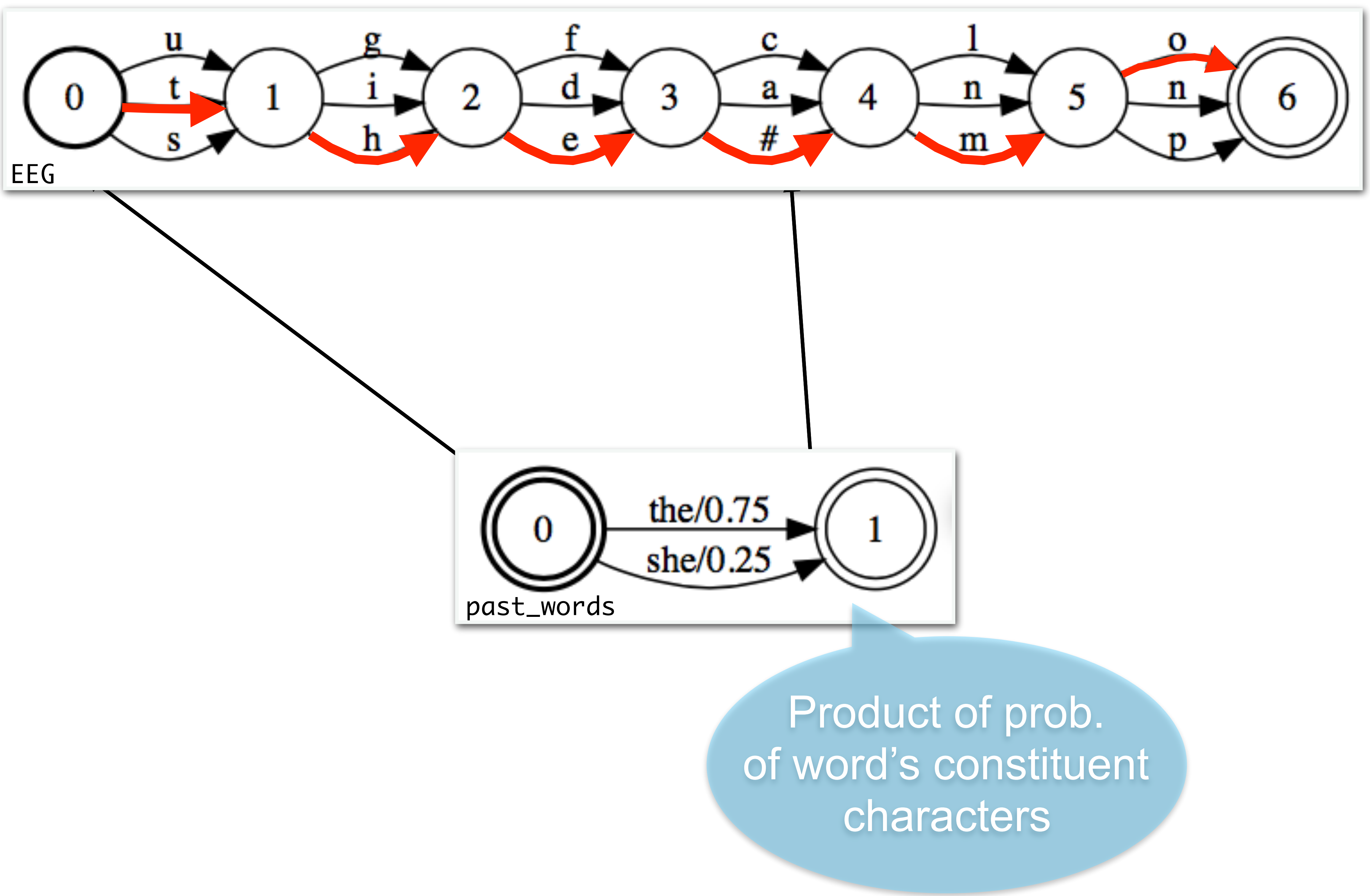

0
s:<epsilon>
5
h:<epsilon>
6
e:<epsilon>
7
#:she
8
t:<epsilon>
1
h:<epsilon>
2
e:<epsilon>
3
#:the
4

Find possible word history sequence

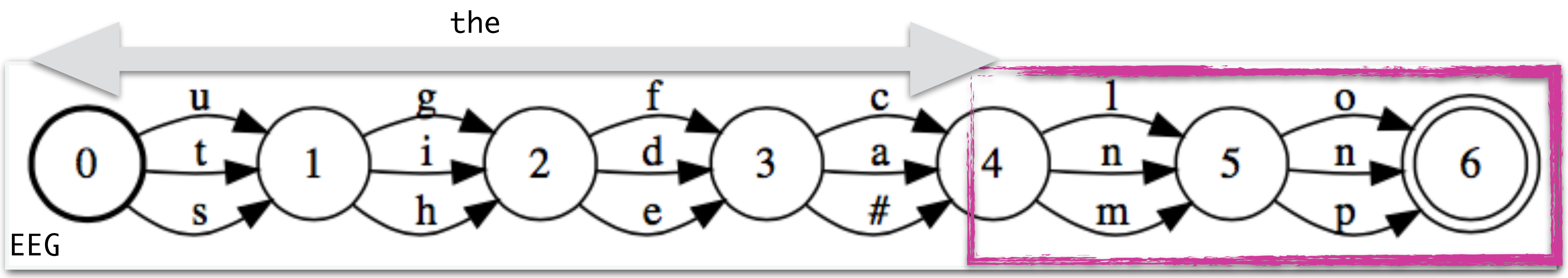


**state 0 is a final state as the entire sequence can be considered a prefix as well

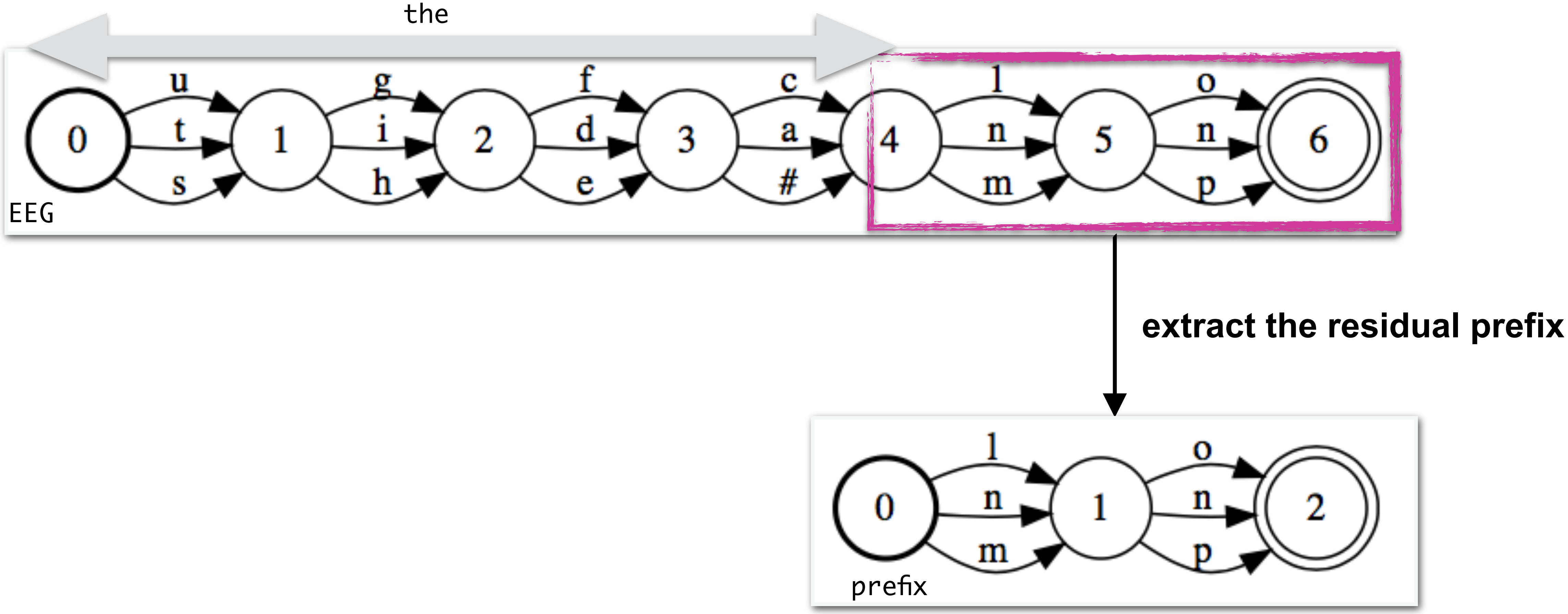
Find possible word history sequence



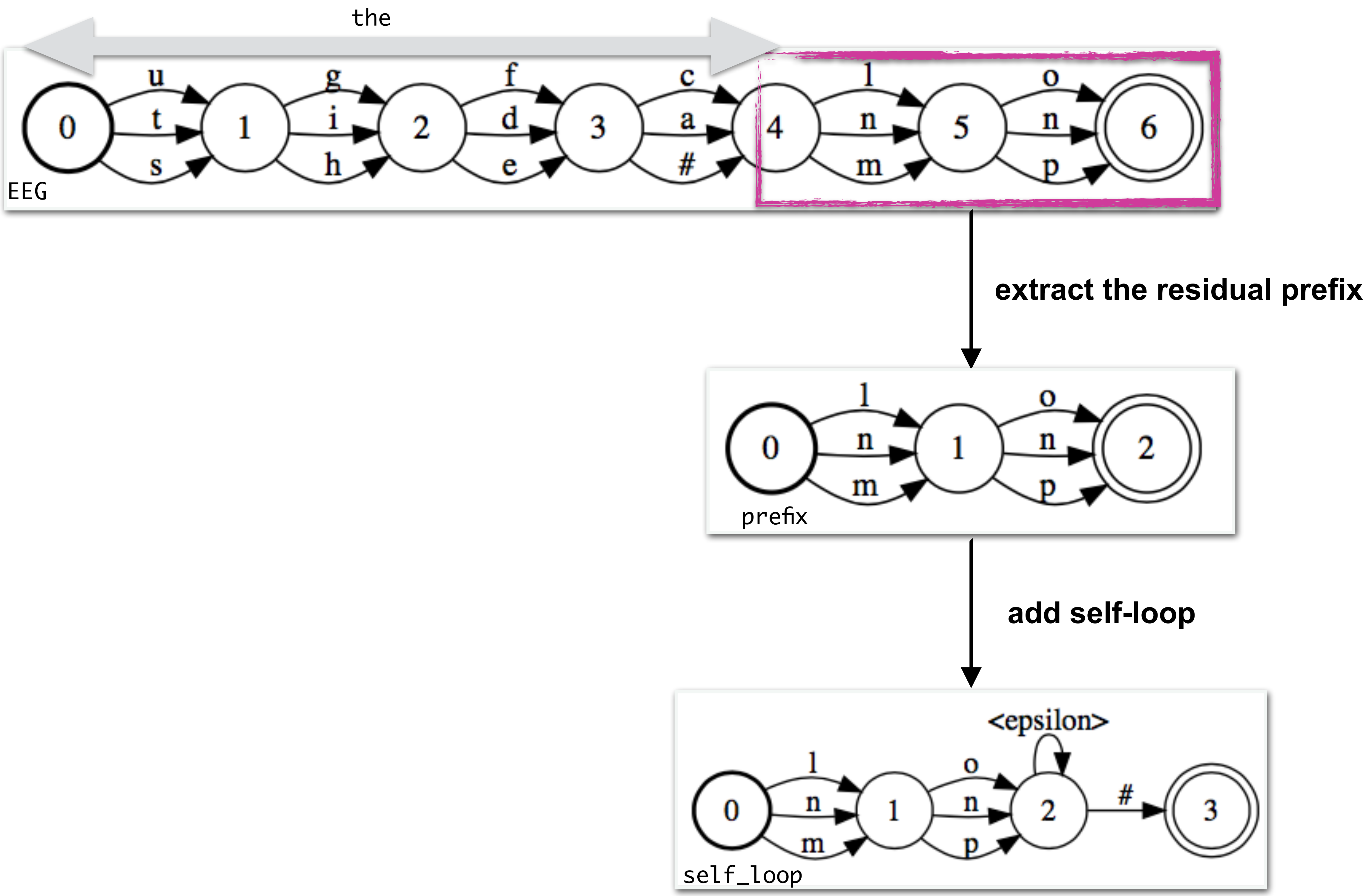
Find possible word completions w/ no context.



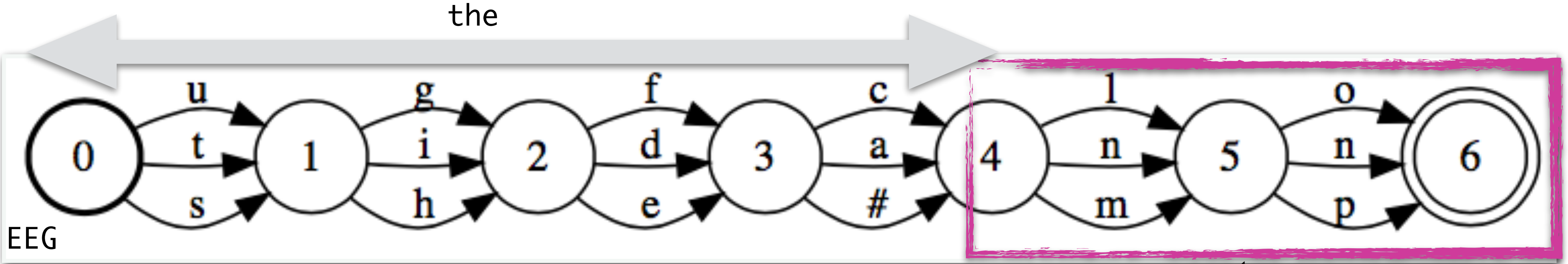
Find possible word completions w/ no context.



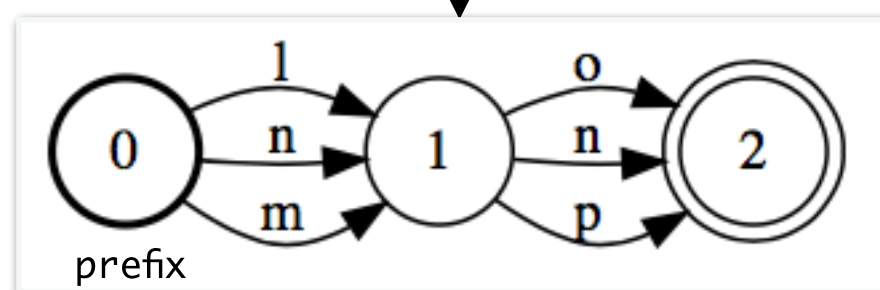
Find possible word completions w/ no context.



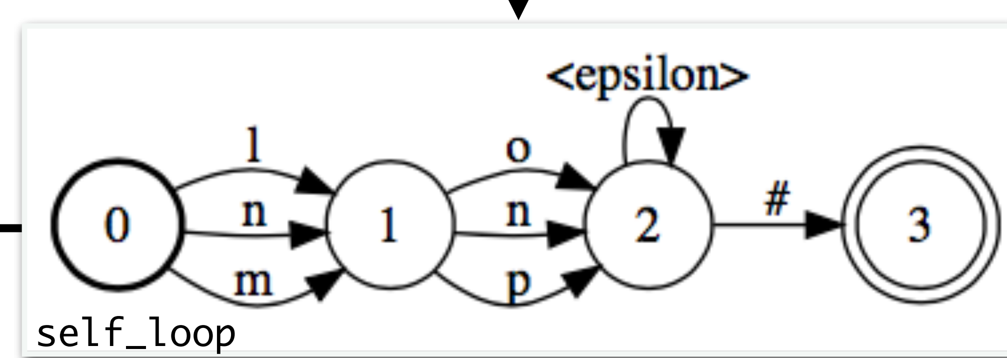
Find possible word completions w/ no context.



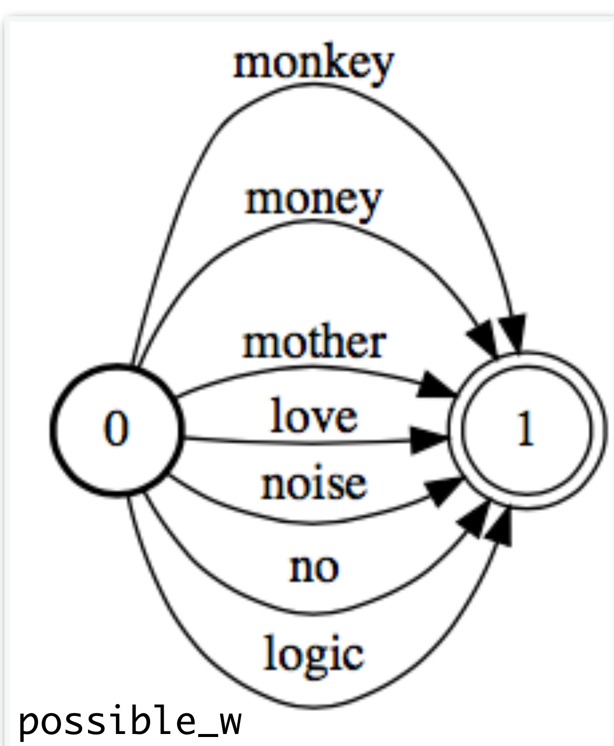
extract the residual prefix



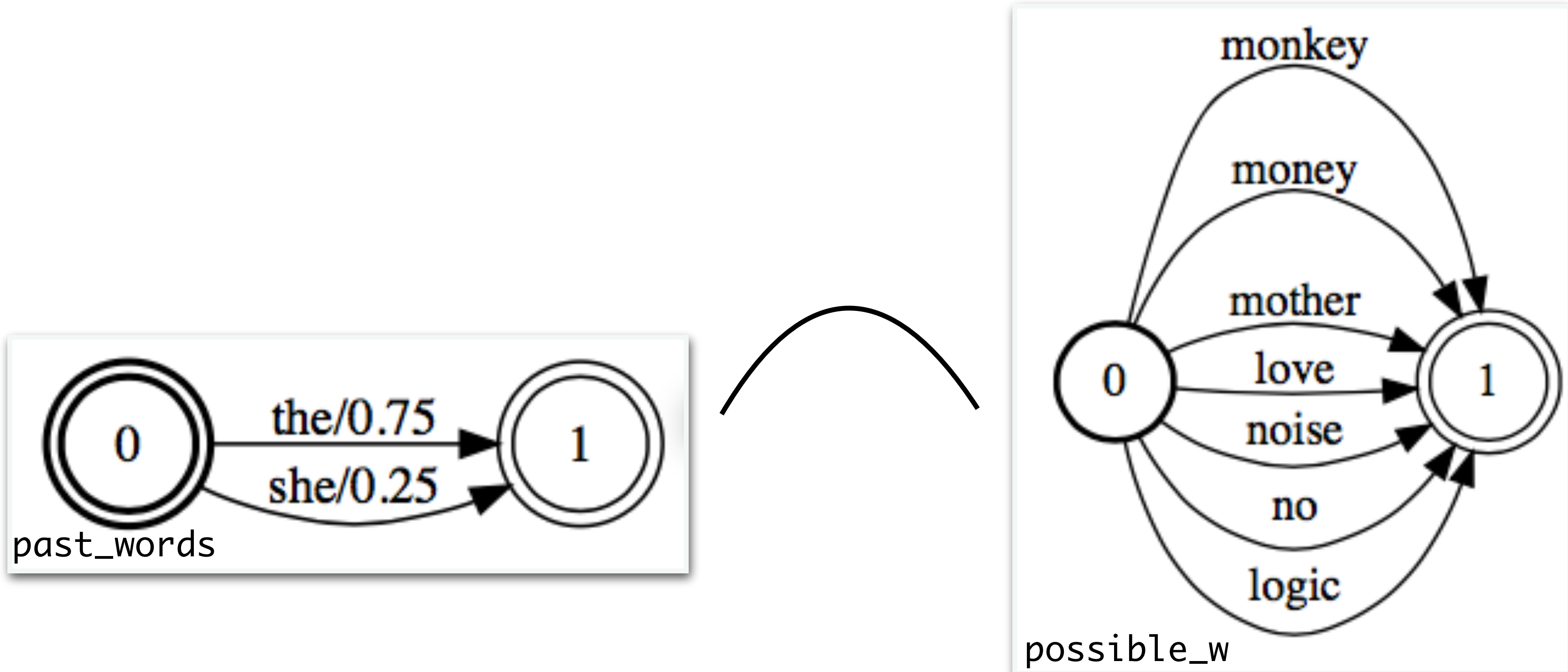
add self-loop



extract possible words (no context)

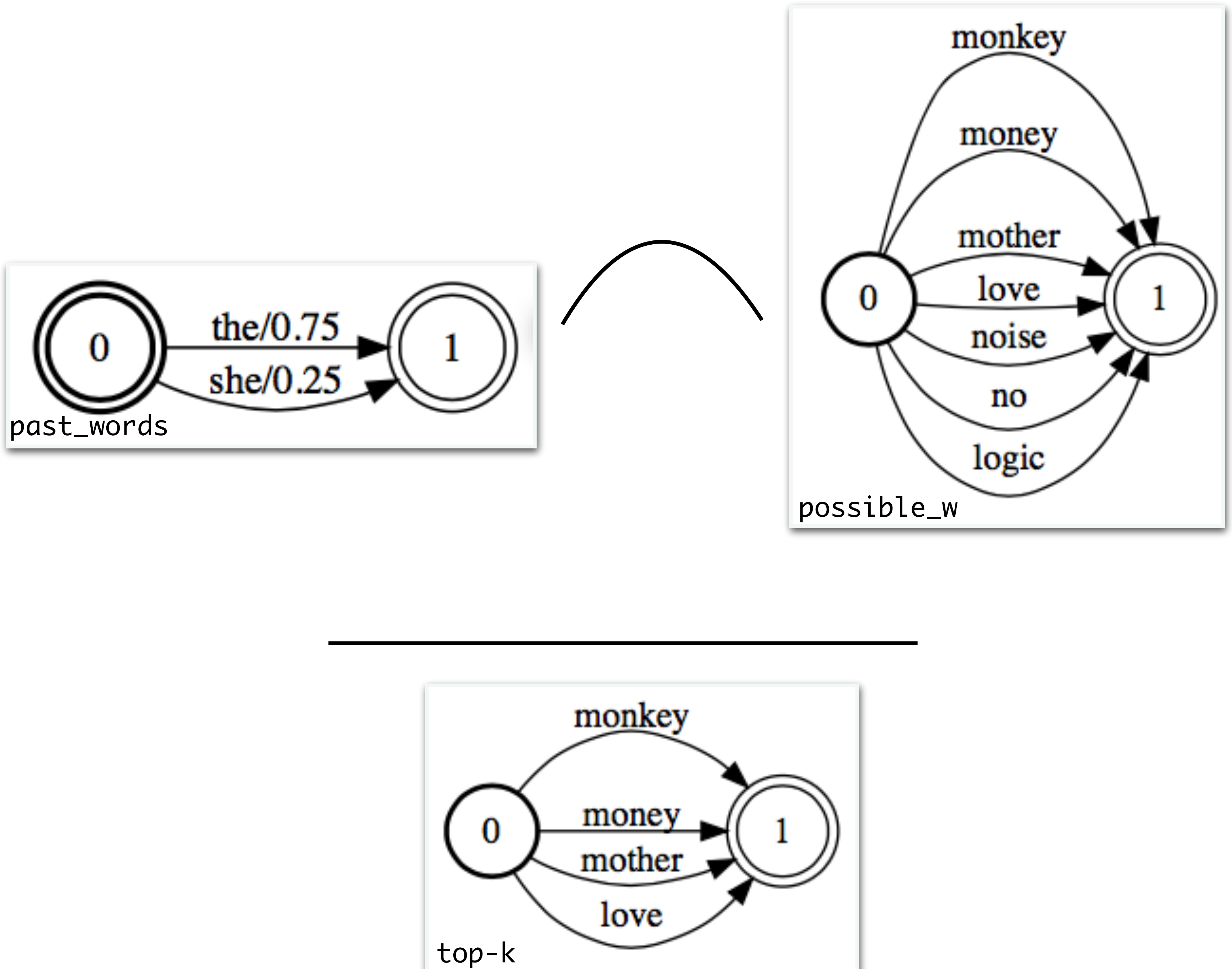


Find top-*k* words the user might be in the middle of typing



concatenate past and current words

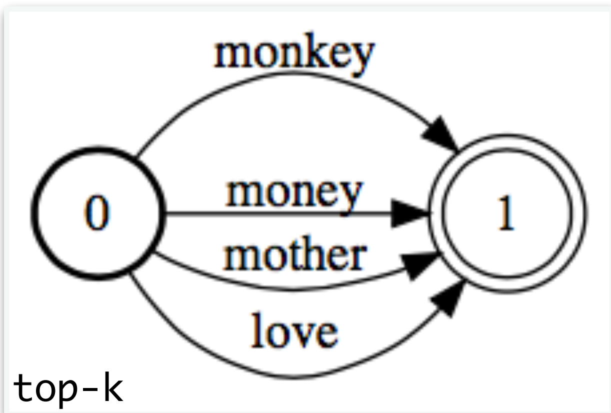
Find top-*k* words the user might be in the middle of typing



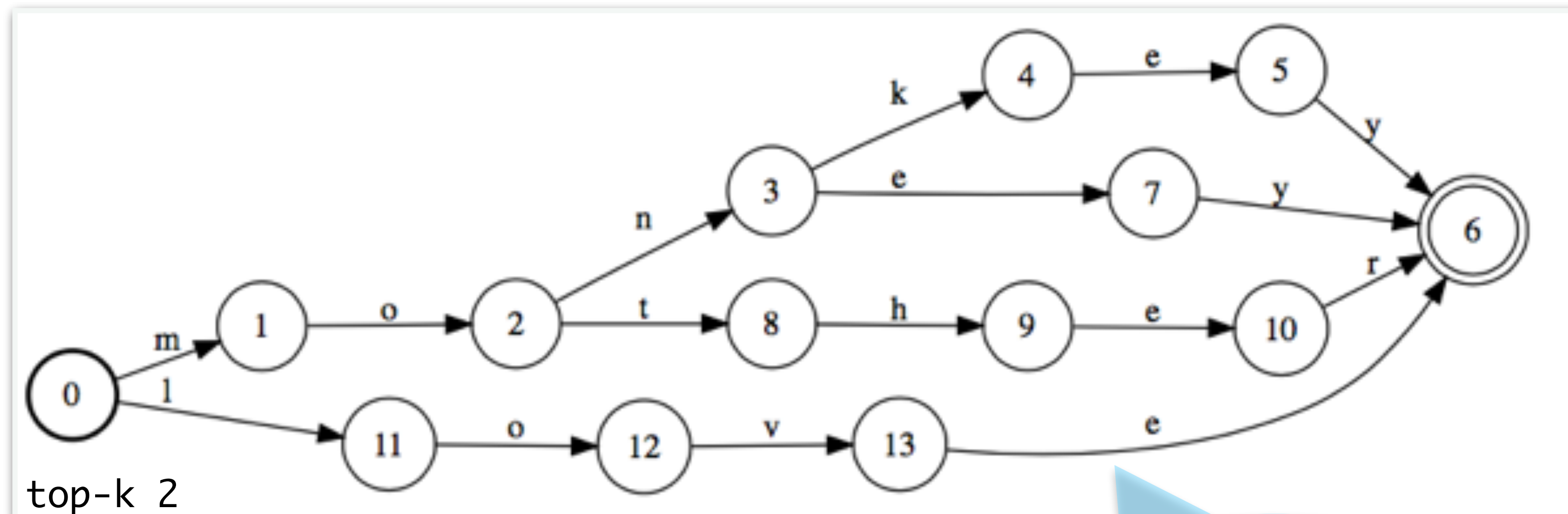
concatenate past and current words

find top-*k* words given context

Make the symbol language model

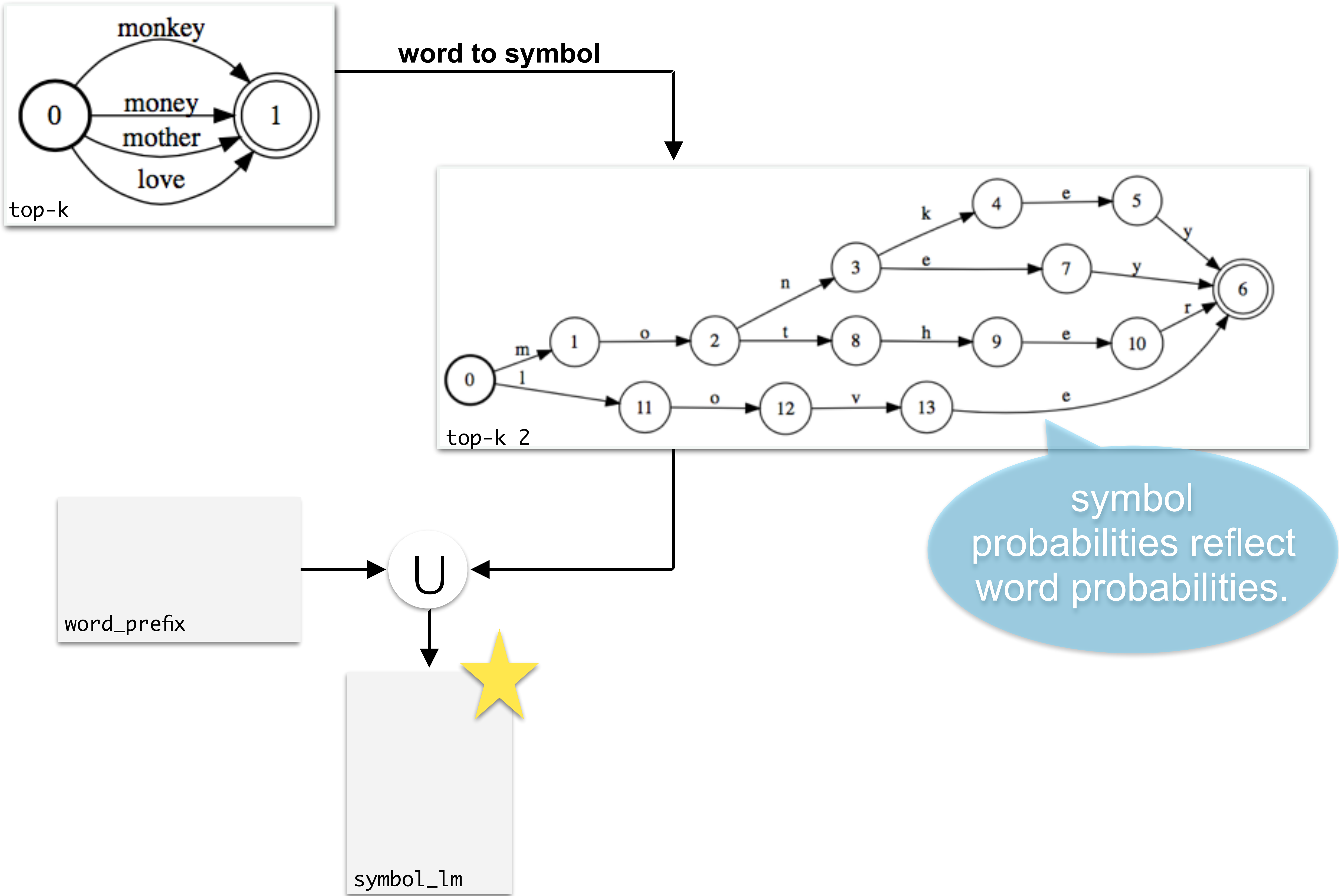


word to symbol

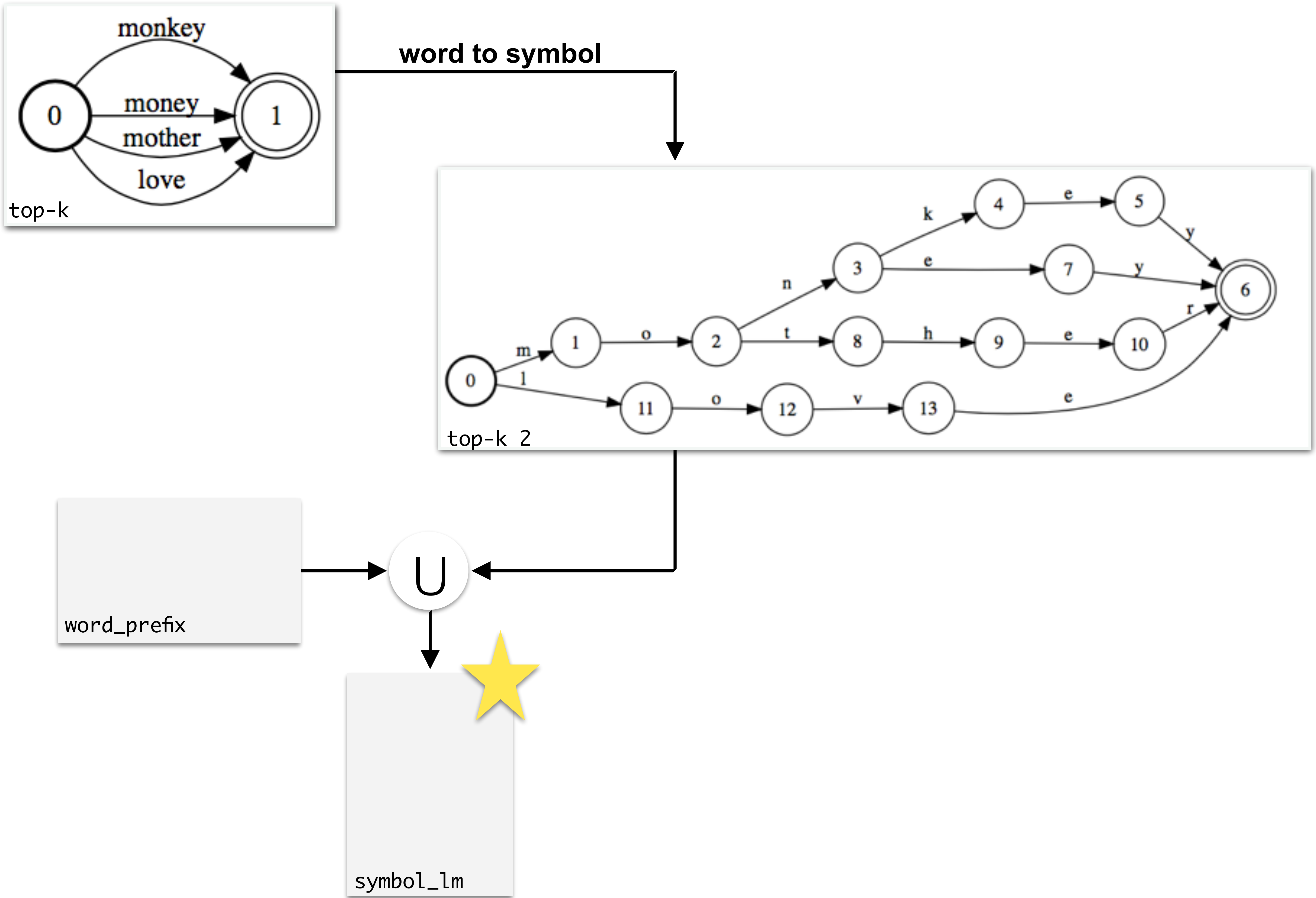


symbol probabilities reflect word probabilities.

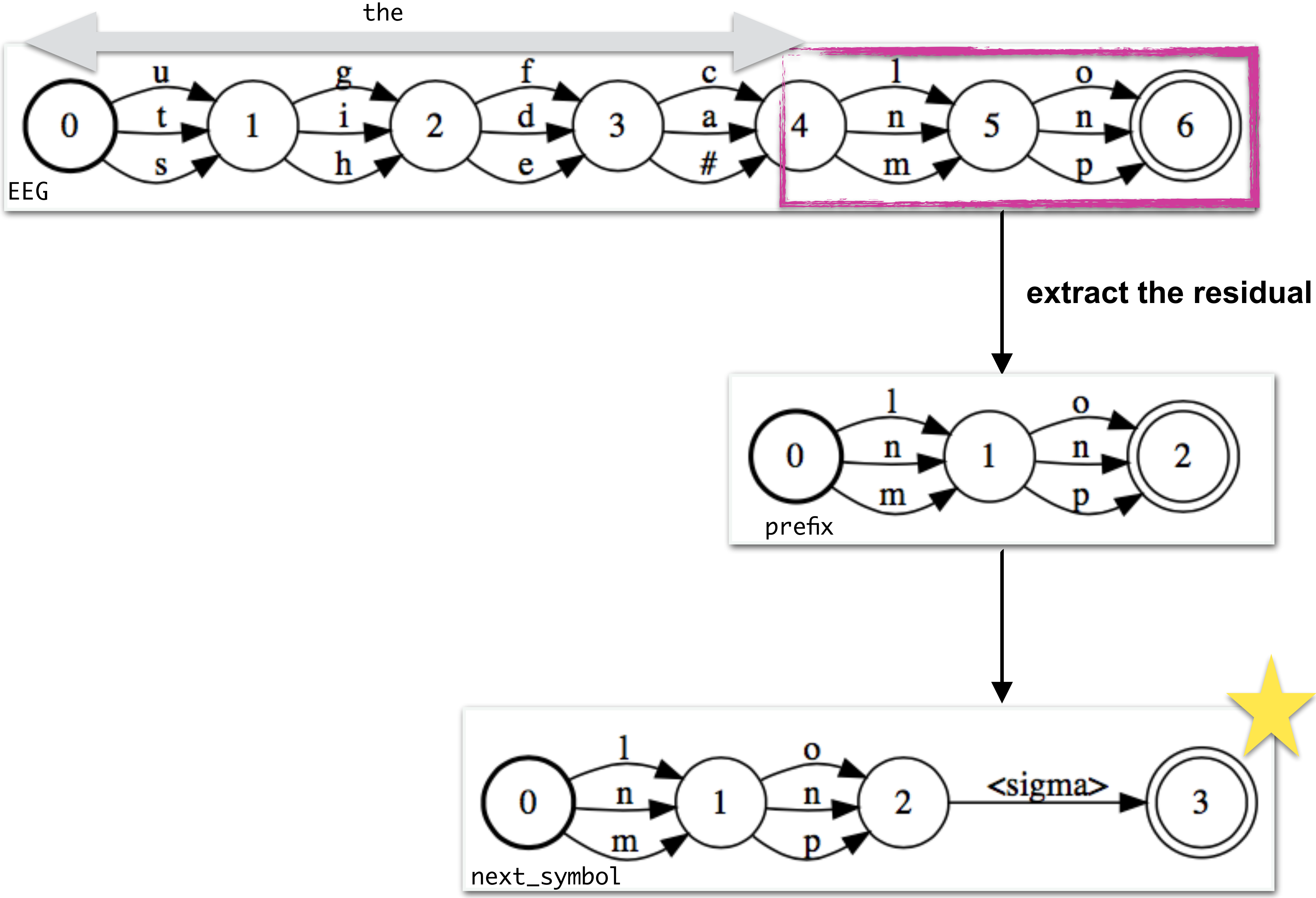
Make the symbol language model



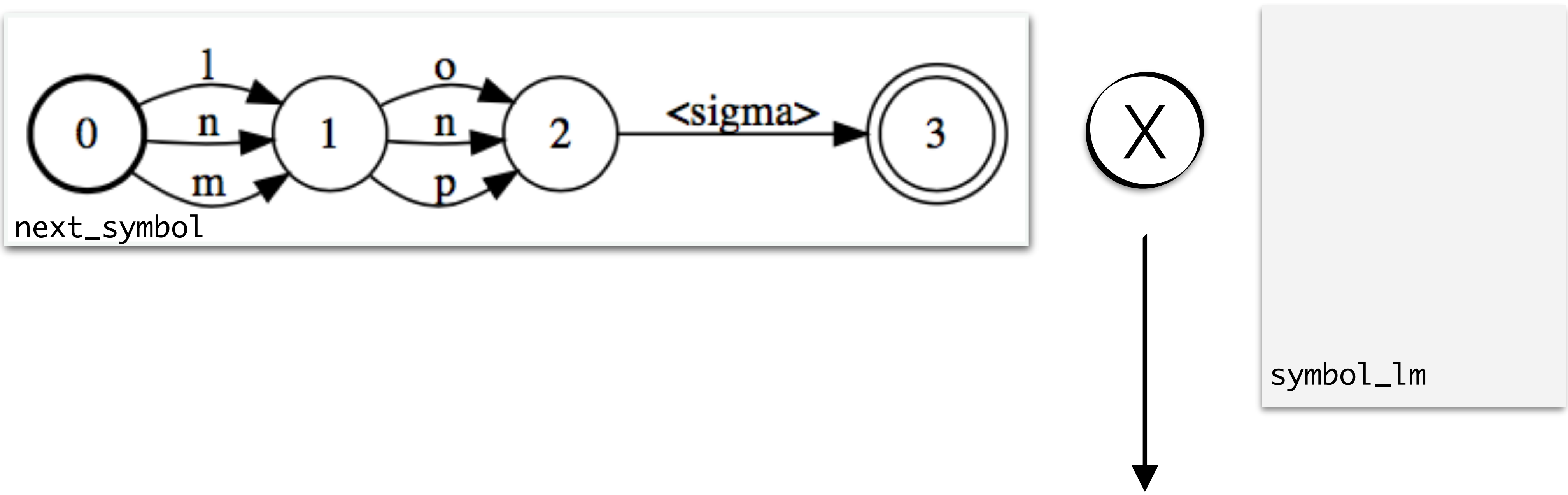
Make the symbol language model



Prepare the current trailing prefix to find its next symbol distribution



Intersect symbol_lm (the machine in #3) with next_symbol (the machine in #4) and get the distribution over final symbols to return to front-end.

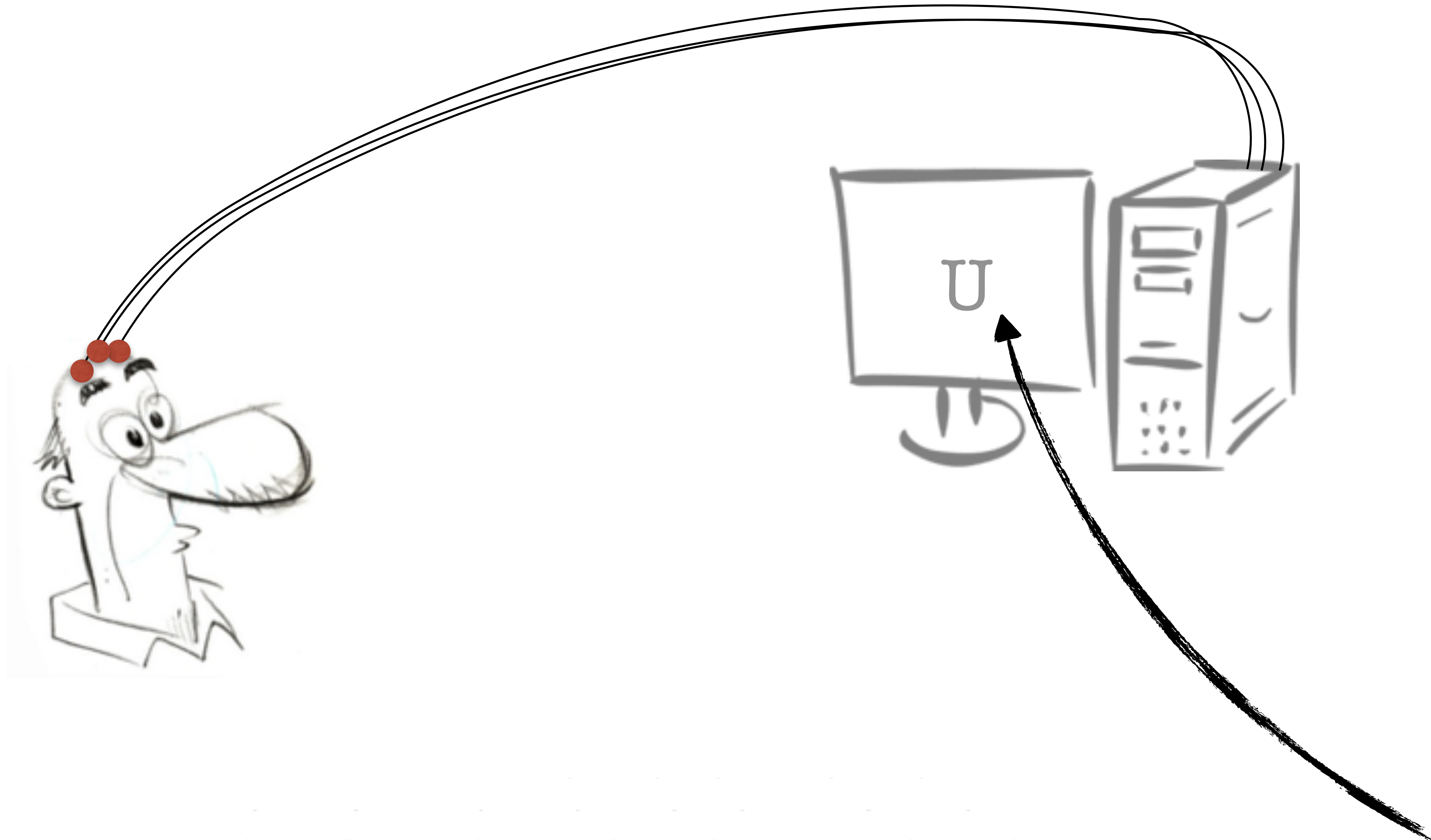


symbol	probability
n	0.103
t	0.102
g	0.09
o	0.09
a	0.06
r	0.04
s	0.04
v	0.01

OCLM in short

1. Extract the potential previous words and possible current ones to figure out the current word and create a targeted LM
2. Extract the potential prefix
3. Extract the next symbol in the potential prefix given the LM

symbol/letter prediction with word knowledge to improve letter prediction



$$P(sym|EEG) \propto P(sym)P(EEG|sym)$$

$$P(sym) = P(sym|X_m)$$

$$X_0 = \{sym_{t-i}\}, i > 0$$

$$X_1 = \{word_{t-i}\}, i > 0$$

$$X_2 = \{EEG_{t-i}\}, i \geq 0$$

Candidate symbols

symbol: confidence

u: 0.95
m: 0.05

Evaluation of OCLM

Method:

Train different LM types:

1) 5 gram LM applied on basic LM algorithm

2) Prefix LM applied on basic algorithm*

3) OCLM algorithm

80% train 20% test, on Brown corpus

Evaluation Metric #1

Mean Reciprocal Rank (MRR):

Or how close were we in our guess for target symbol?

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank_i}$$

Evaluation Metric #2

Perplexity:

Or how likely are we to reproduce the data with our model

$$PPL(t, x) = 2^{-\frac{1}{T}\sum_{t=1}^{T} \log q_{model}(x=target_t)}$$

Evaluation Metric #3

ACC@10:

Or how often the target was in top 10 guesses

metric	NGRAM	PreLM	OCLM
MRR	0.4	0.7	0.75
PPX	4.4	1.8	1.9
ACC@10	0.69	0.96	0.96

Table 1: Evaluation Results (n=1)

nbest	metric	PreLM	OCLM
	MRR	0.29	0.51
n=2	PPX	3.5	3.0
	ACC@10	0.69	0.87
	MRR	0.26	0.44
n=3	PPX	4	3.9
	ACC@10	0.63	0.83

Table 2: Evaluation Results (n=2, n=3)

Conclusions

Both OCLM and PreLM outperform NgramLM in terms of PPX and MRR on a single EEG evidence

On multiple hypotheses both algorithms degraded but OCLM was performing much better than PreLM

One shortcoming: OCLM’s and runtime is longer

Additional material

The research paper: https://aclweb.org/anthology/W18-1210

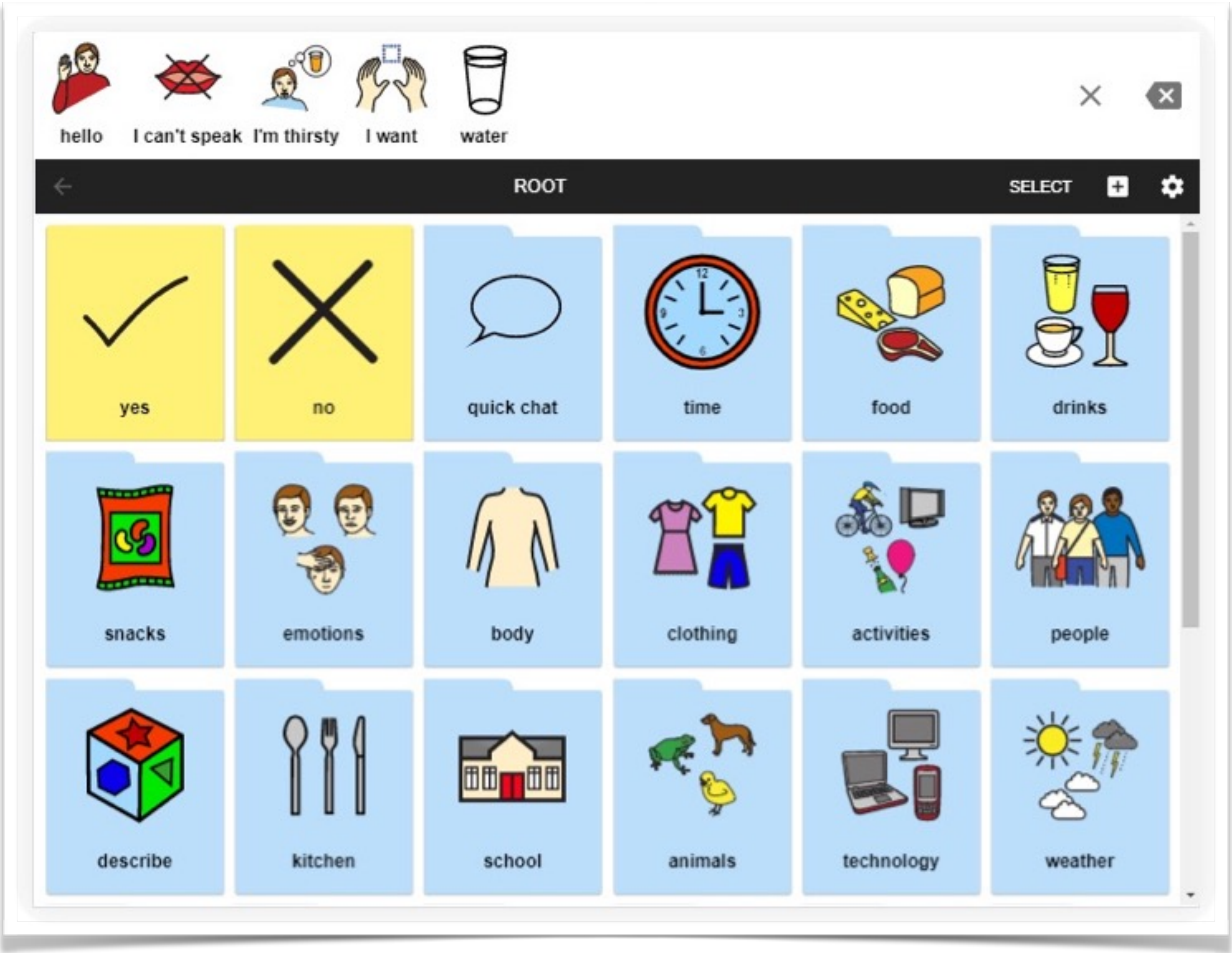
The git repo: https://github.com/shiranD/oclm

Compositional Language Modeling for Icon-Based Augmentative and Alternative Communication

Shiran Dudy

April 23, 2019

What is AAC? Augmentative and Alternative Communication
https://www.youtube.com/watch?v=r3m8_YmTDDM



Open-source communication board
https://www.cboard.io/cboard/open-source/2017/12/05/open-source-communication-board/

electronic communication board
https://www.rachelmadel.com

Scaling?

Selection modality

Our goal:

Creating an icon language-model based AAC system

Symbolstix Icon set

Used by communities who are in need of icon-based communication

Symbolstix Icon set

Used by communities who are in need of icon-based communication

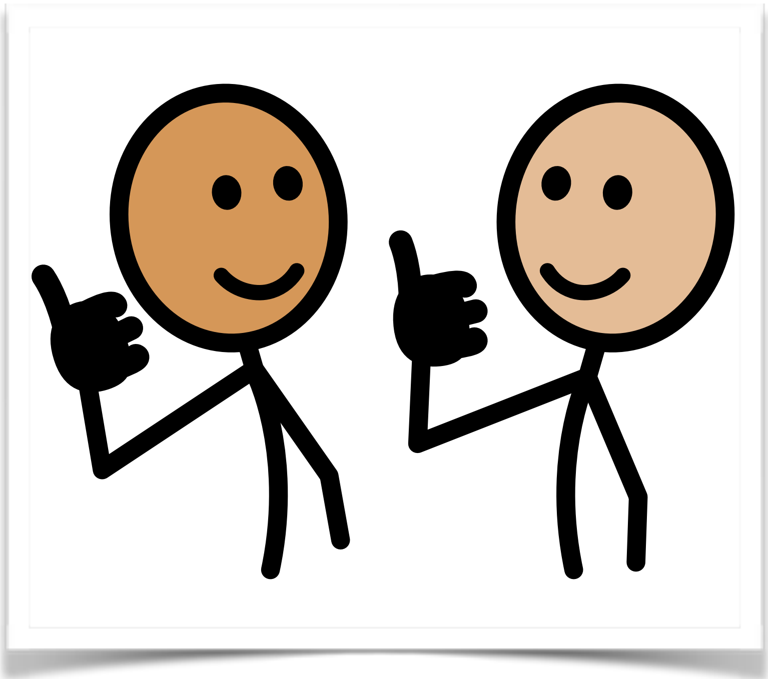
Human Curated icons

Symbolistix Icon set

Used by communities who are in need of icon-based communication

Human Curated icons

34,837 icons: 13,951 single words, 12,434 unique single words

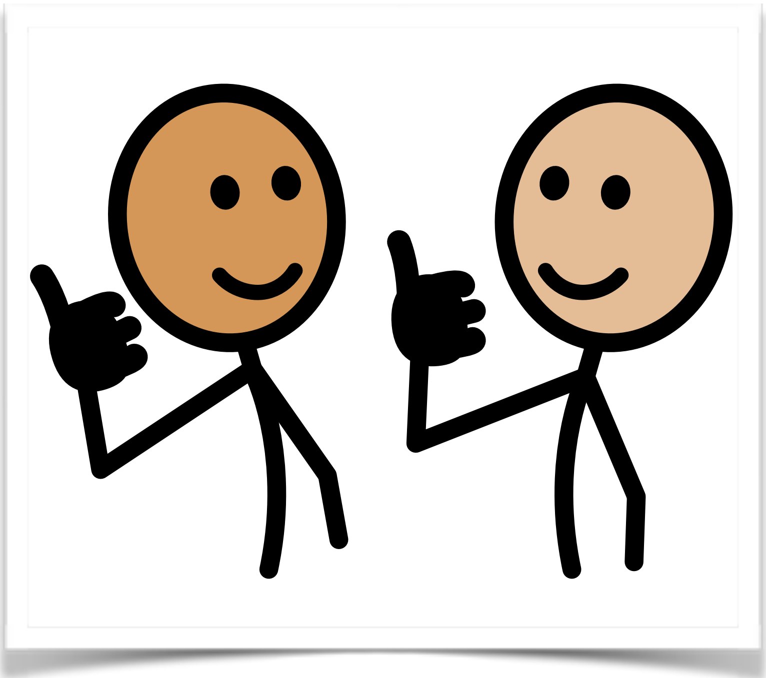


name: agree

word type: verb

synonyms:

agreement
agreed
agreeing
flexibility
on the same page
see eye to eye
agrees
approve
concur



name: agree

word type: verb

synonyms:

agreement

agrees

agreed

approve

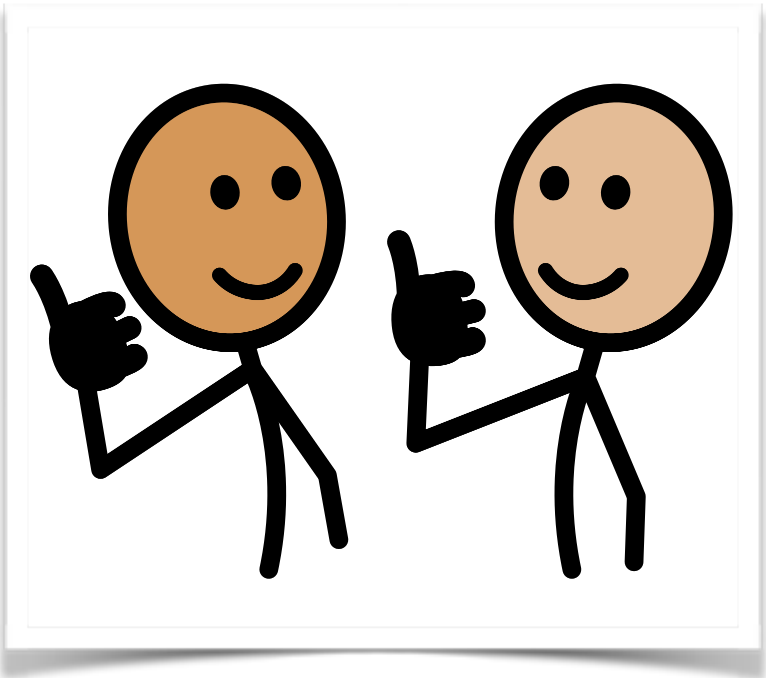
agreeing

concur

flexibility

<u>on the same page</u>

<u>see eye to eye</u>

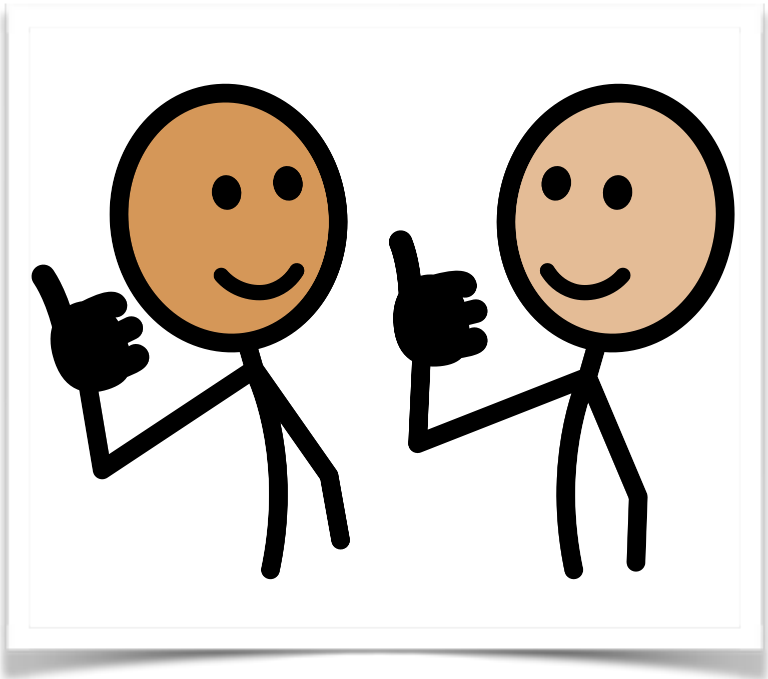


name: agree

word type: verb

synonyms:

agreement
agreed
agreeing
flexibility
on the same page
see eye to eye
agrees
approve
concur



name: agree

word type: verb

synonyms:

agreement	agrees
agreed	approve
agreeing	concur

flexibility

on the same page

see eye to eye

Symbolistix Icon set

Used by communities who are in need of icon-based communication

Human Curated icons

34,837 icons: 13,951 single words, 12,434 unique single words

Symbolistix Icon set

Used by communities who are in need of icon-based communication

Human Curated icons

34,837 icons: 13,951 single words, 12,434 unique single words

<u>No Corpus available!</u>

Our Question:

How to create language models for corpus-less symbol-set

We don’t have:

We don’t have:

icon corpus

We don’t have:

~~icon corpus~~

We don’t have:

~~icon corpus~~

we have:

We don’t have:

~~icon corpus~~

we have:

- Icon meta-data

We don’t have:

~~icon corpus~~

we have:

- Icon meta-data
- pre existing **textual** corpora such as Gigaword

We don't have:

~~icon corpus~~

we have:

- Icon meta-data
- pre existing textual corpora such as Gigaword
- Word embedding representation

.

.

agree
agreeing
bring
clinic
concur
connected
flexibility
hospital
telephone
patient

.

telephone

.
.
agree
agreeing
bring
clinic
concur
connected
flexibility
hospital
telephone
patient
.

concur

agree

agreeing

patient

clinic

hospital

telephone

concur

agree

agreeing

.
.
agree
agreeing
bring
clinic
concur
connected
flexibility
hospital
telephone
patient
.

patient

clinic

hospital

We don't have:

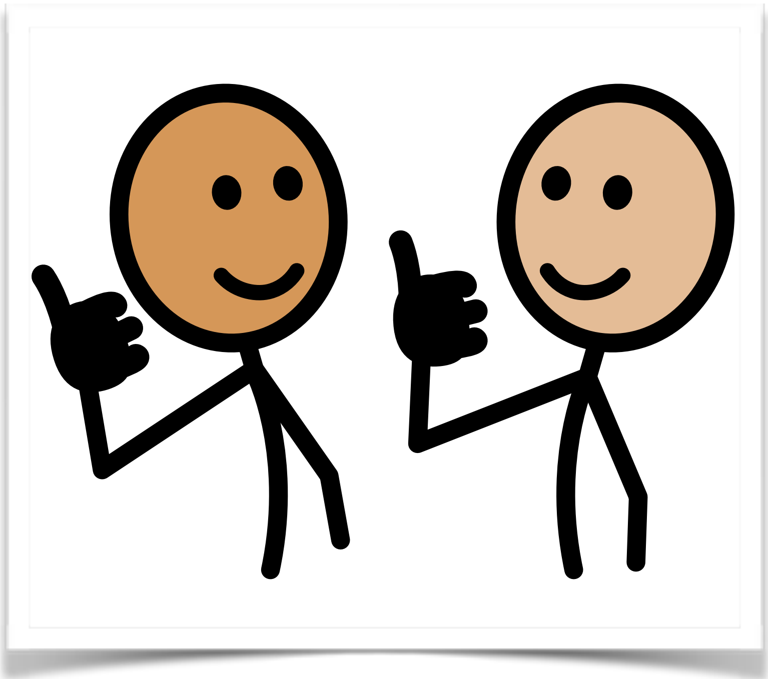
~~icon corpus~~

we have:

- Icon meta-data
- pre existing textual corpora such as Gigaword
- Word embedding representation

Perhaps we can utilize the synonyms, apply to textual corpus?

Icon representation

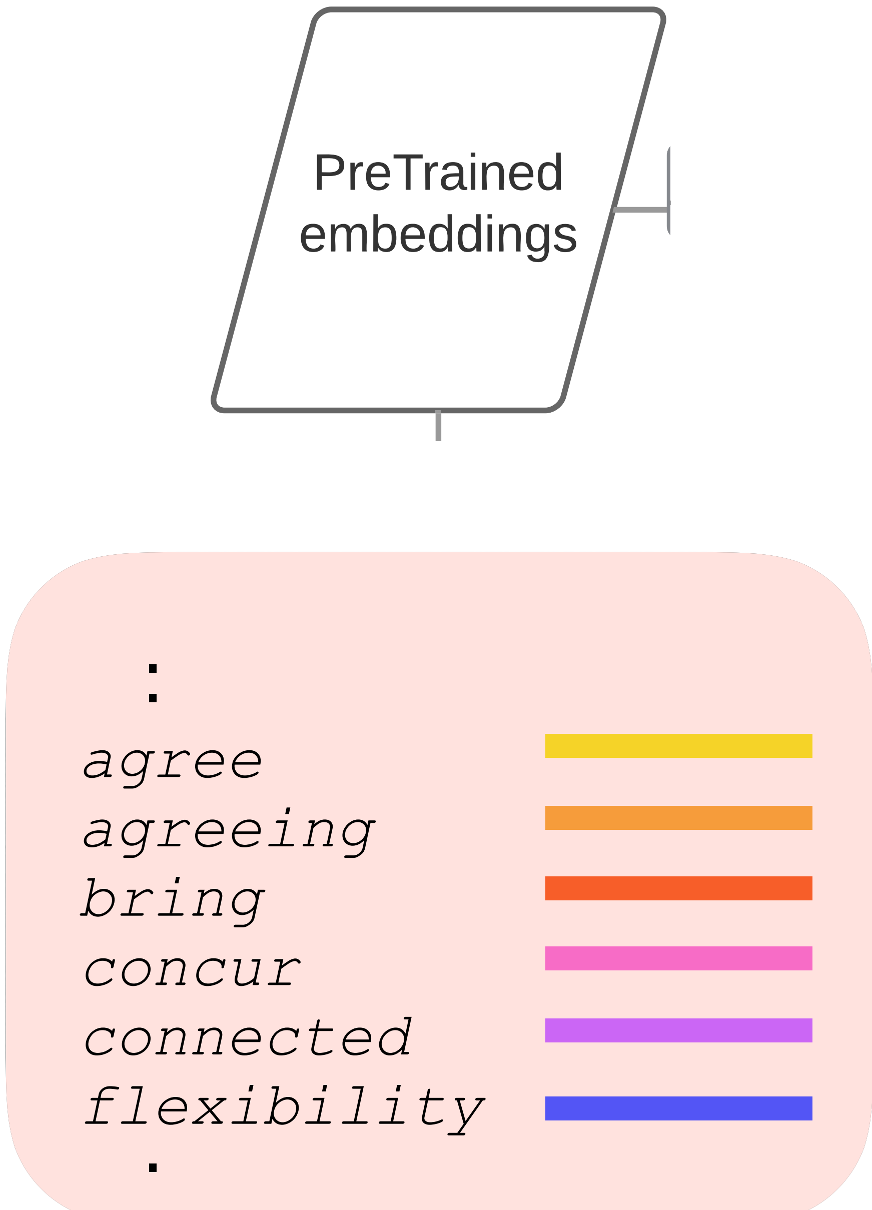


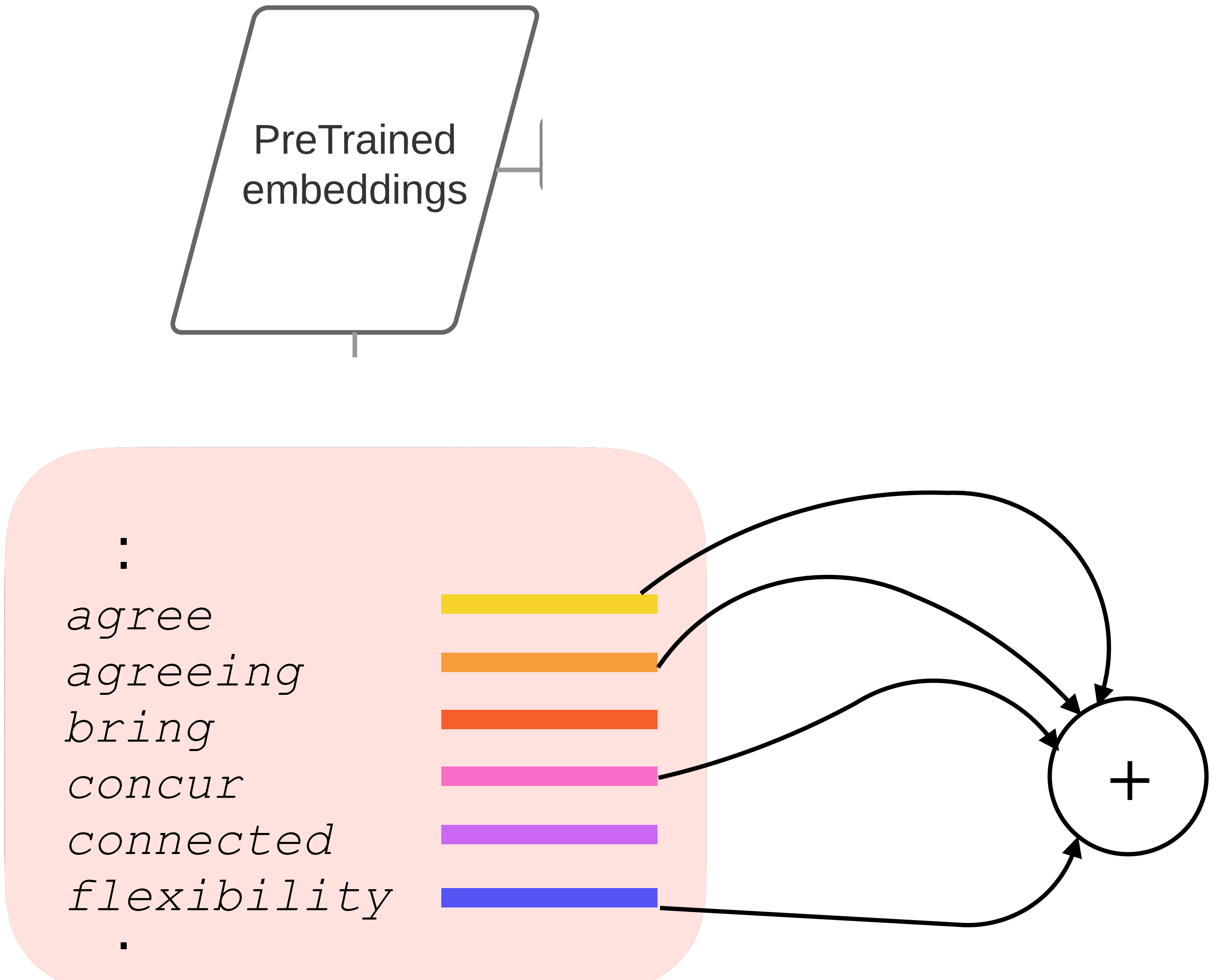
name: agree

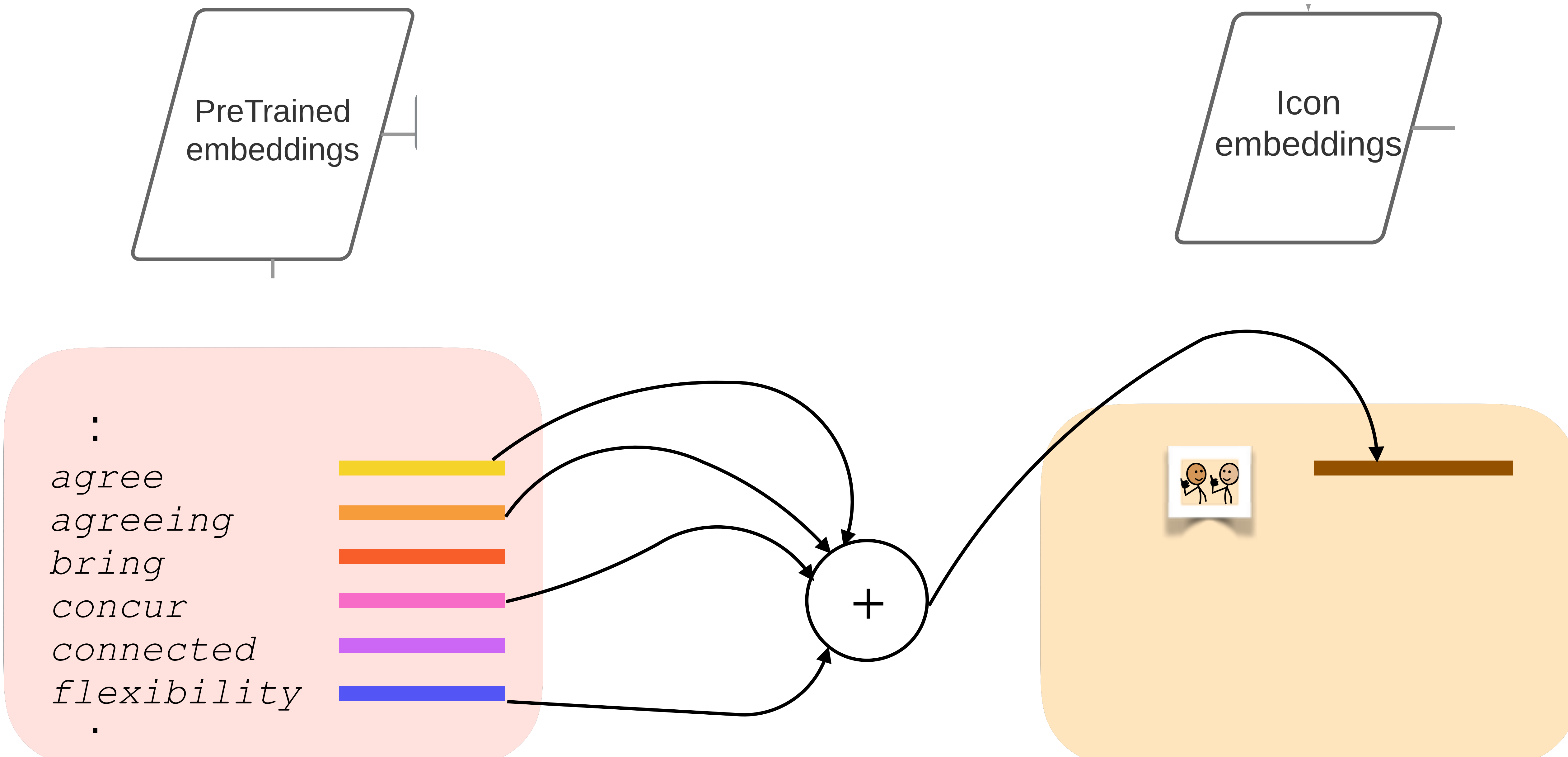
word type: verb

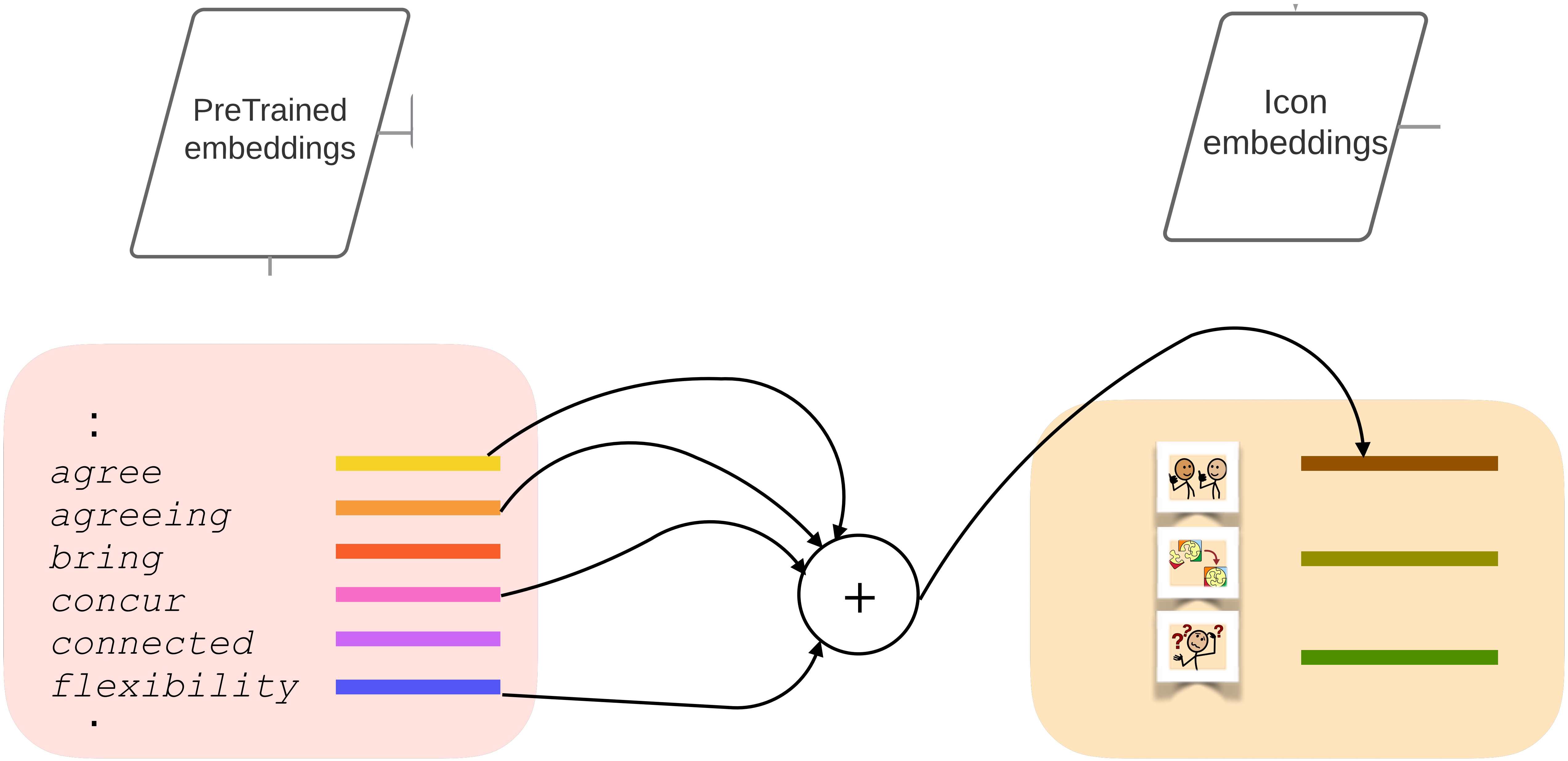
synonyms:

agreement
agreed
agreeing
flexibility
on the same page
see eye to eye
agrees
approve
concur

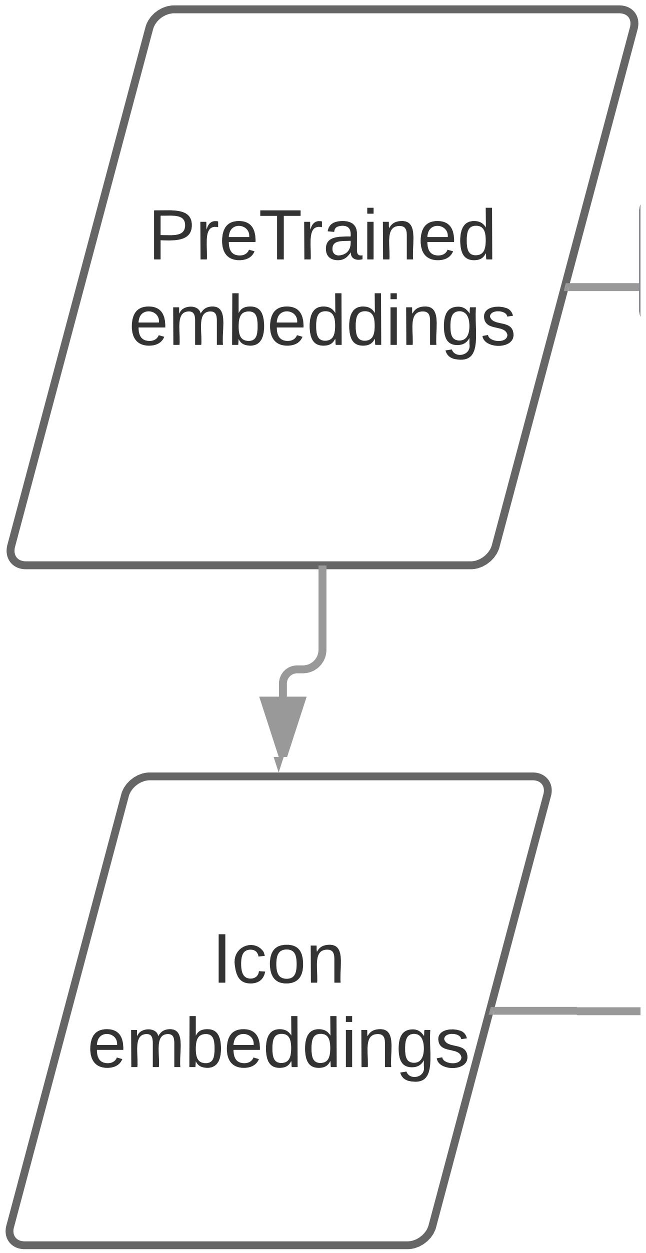

PreTrained
embeddings
:
agree
agreeing
bring
concur
connected
flexibility
.

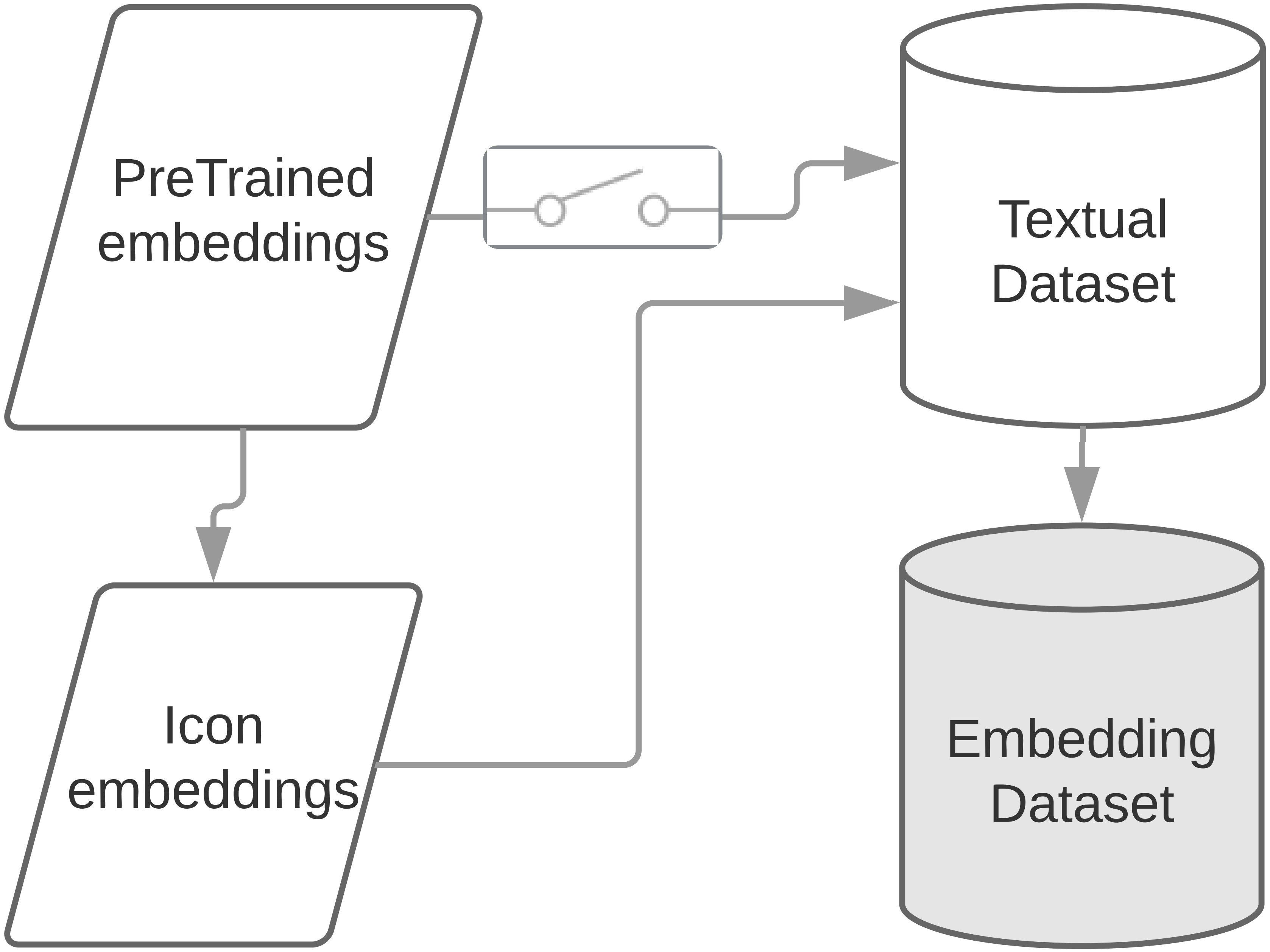

PreTrained embeddings
agree
agreeing
bring
concur
connected
flexibility
+

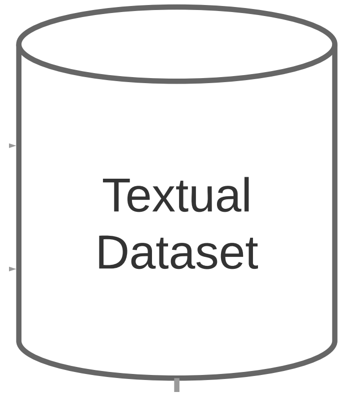

PreTrained embeddings
Icon embeddings
agree
agreeing
bring
concur
connected
flexibility
+



PreTrained embeddings
Icon embeddings
:
agree
agreeing
bring
concur
connected
flexibility
.
+


PreTrained embeddings
Icon embeddings


PreTrained embeddings
Icon embeddings
Textual Dataset
Embedding Dataset

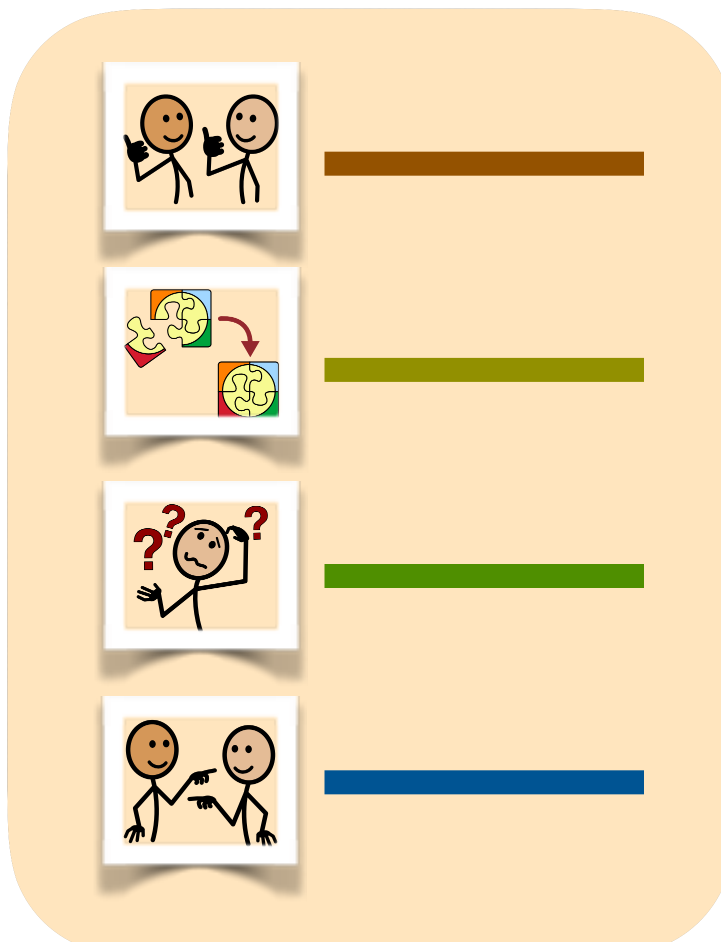


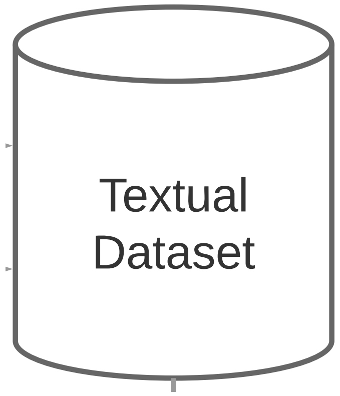
`… you agree to solve problems …`

Textual Dataset

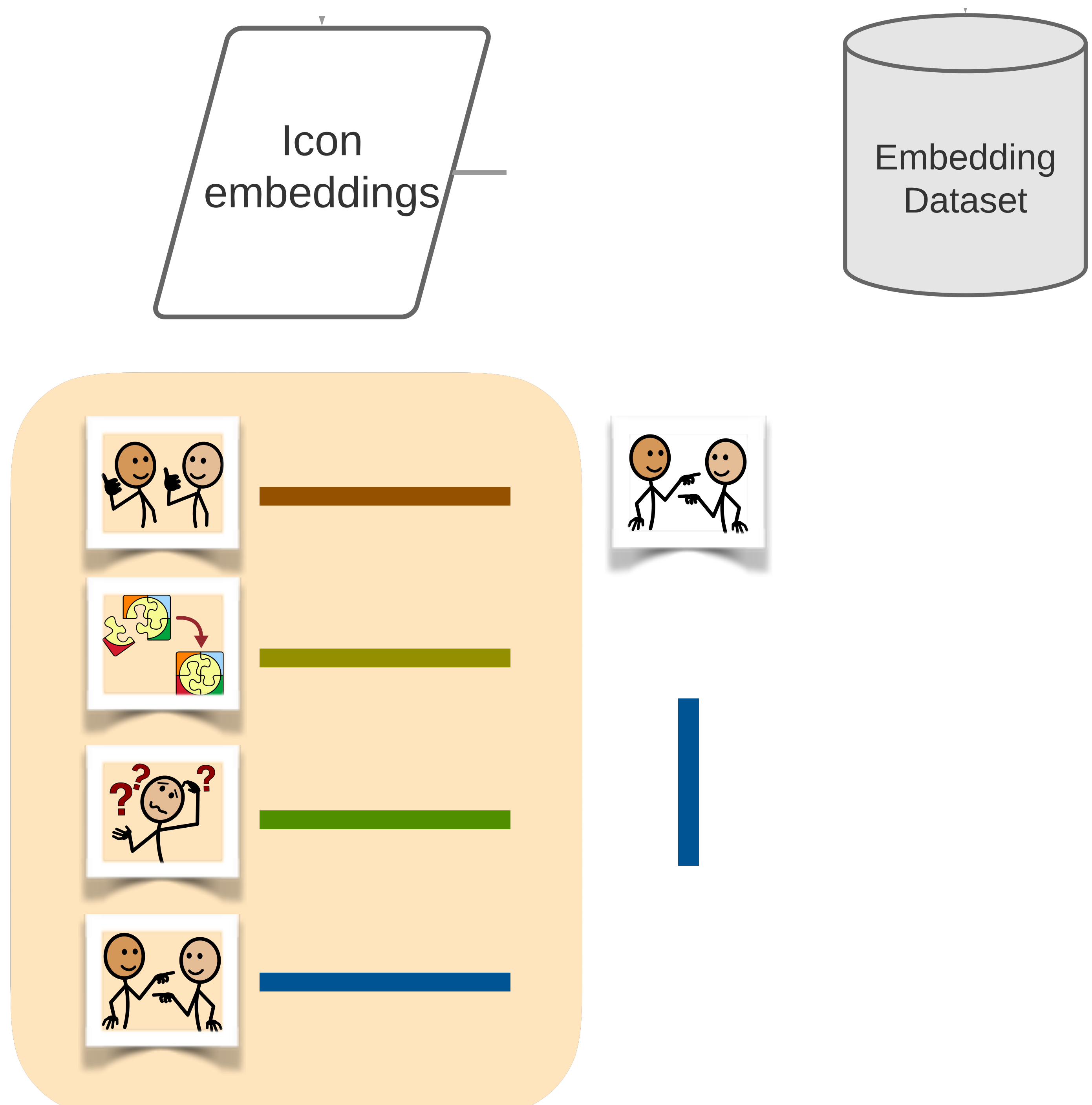
Icon embeddings

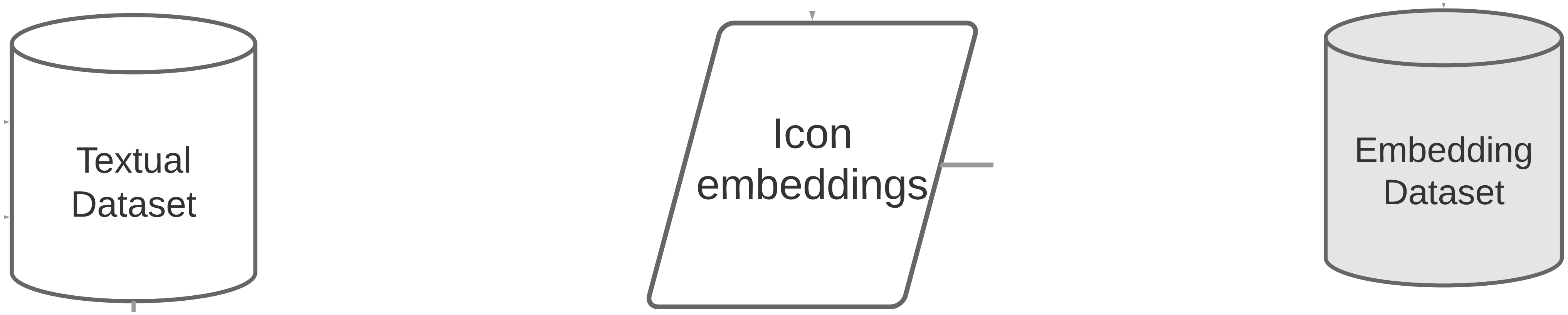
… you agree to solve problems …



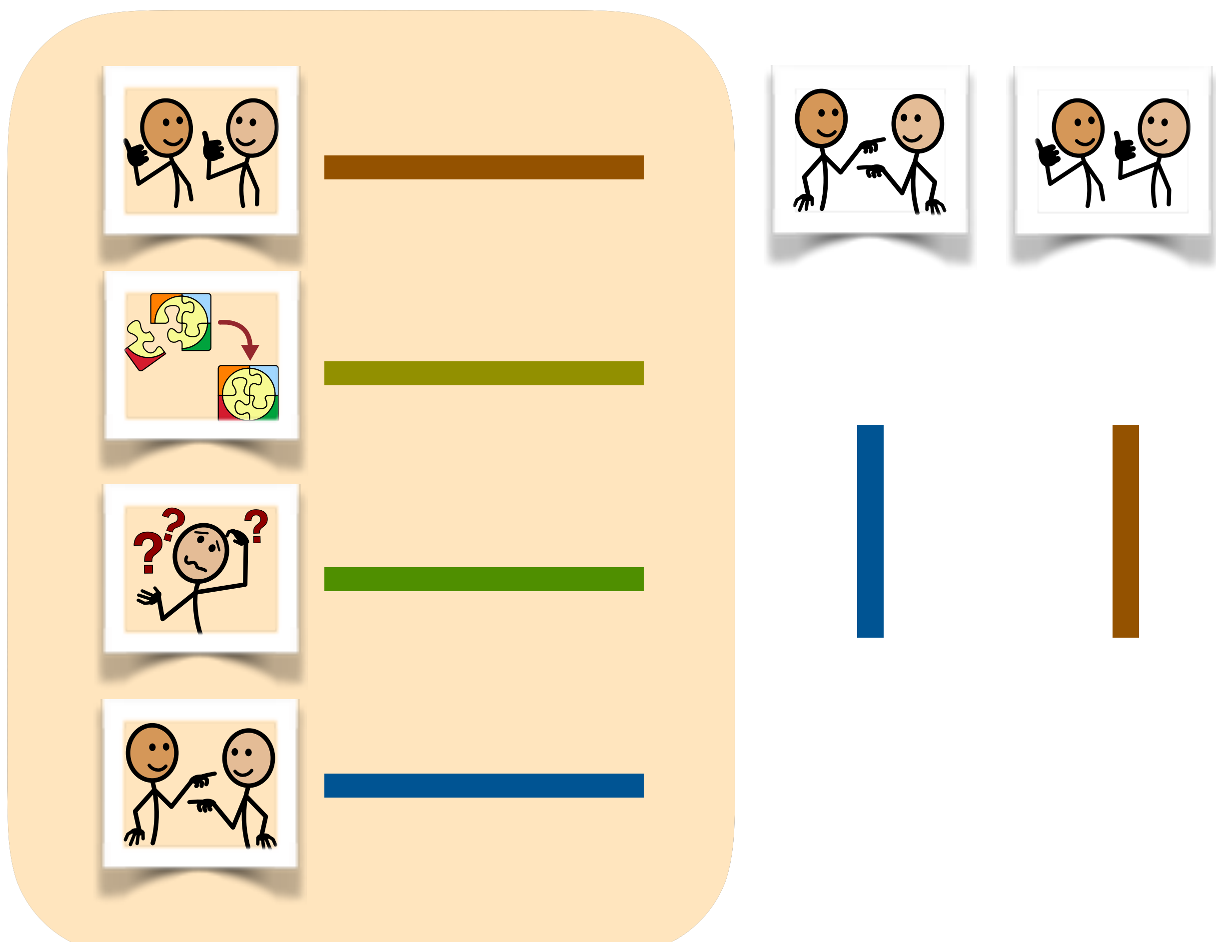


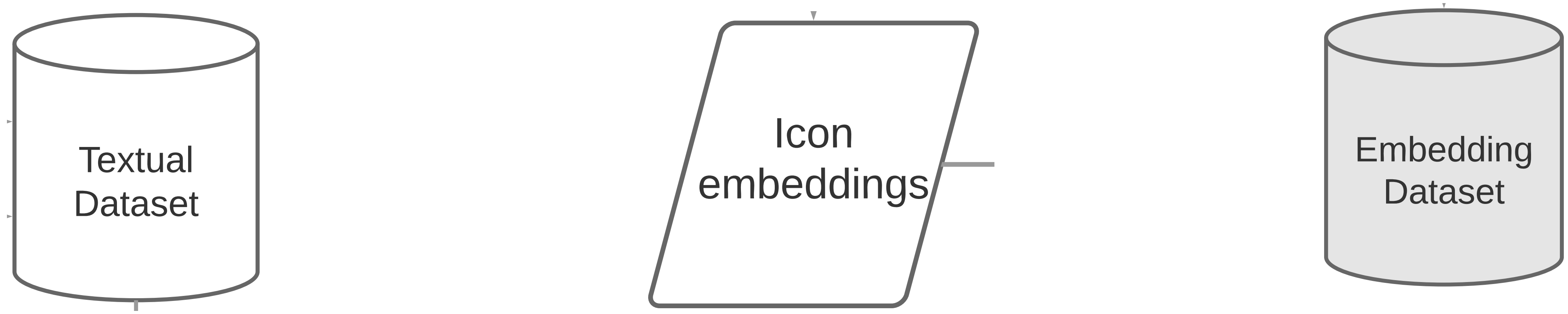
… **you** agree to solve problems …



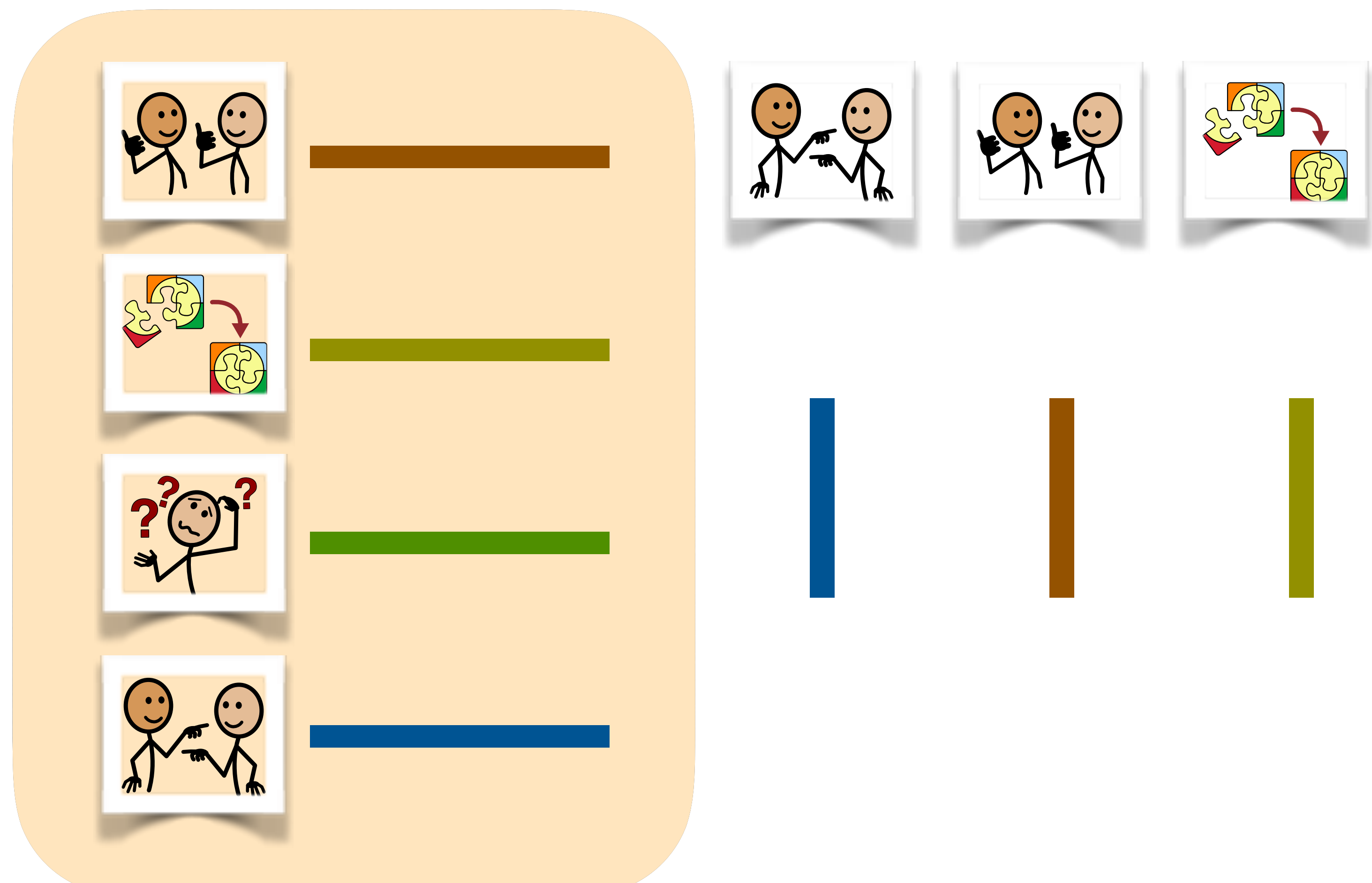


… you **agree** to solve problems …





… you agree to **solve** problems …

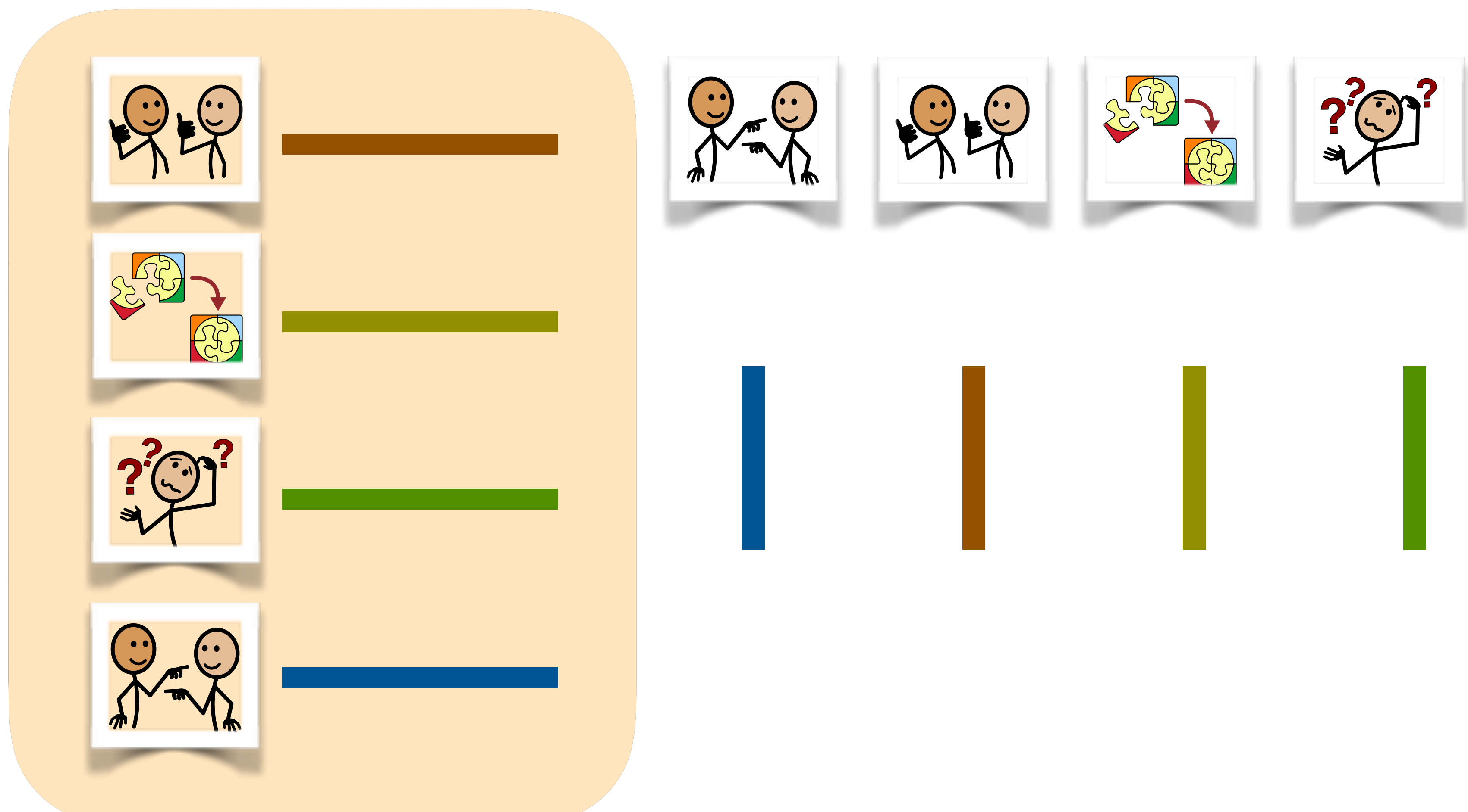


Textual Dataset

Icon embeddings

Embedding Dataset

… `you agree to solve` **`problems`** …



We have embedding in sequences

We have embedding in sequences

Now we are ready for Icon LM training

We have embedding in sequences

Now we are ready for Icon LM training

Simple (any architecture we want)

We have embedding in sequences

Now we are ready for Icon LM training

Simple (any architecture we want)

5 fold cross validation

We have embedding in sequences

Now we are ready for Icon LM training

Simple (any architecture we want)

5 fold cross validation

LM prediction accuracy: MRR, ACC@1, ACC@10

We have embedding in sequences

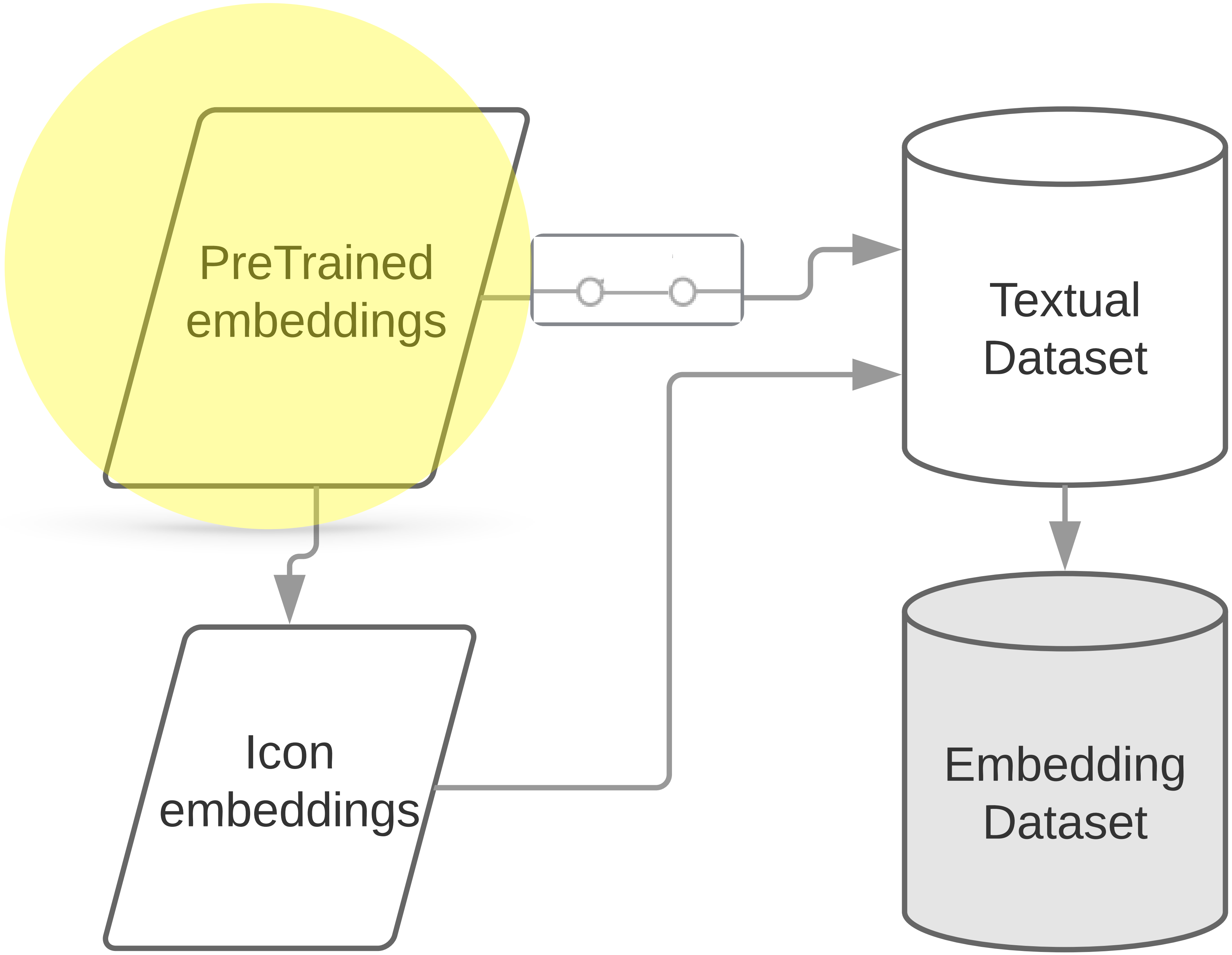
Now we are ready for Icon LM training

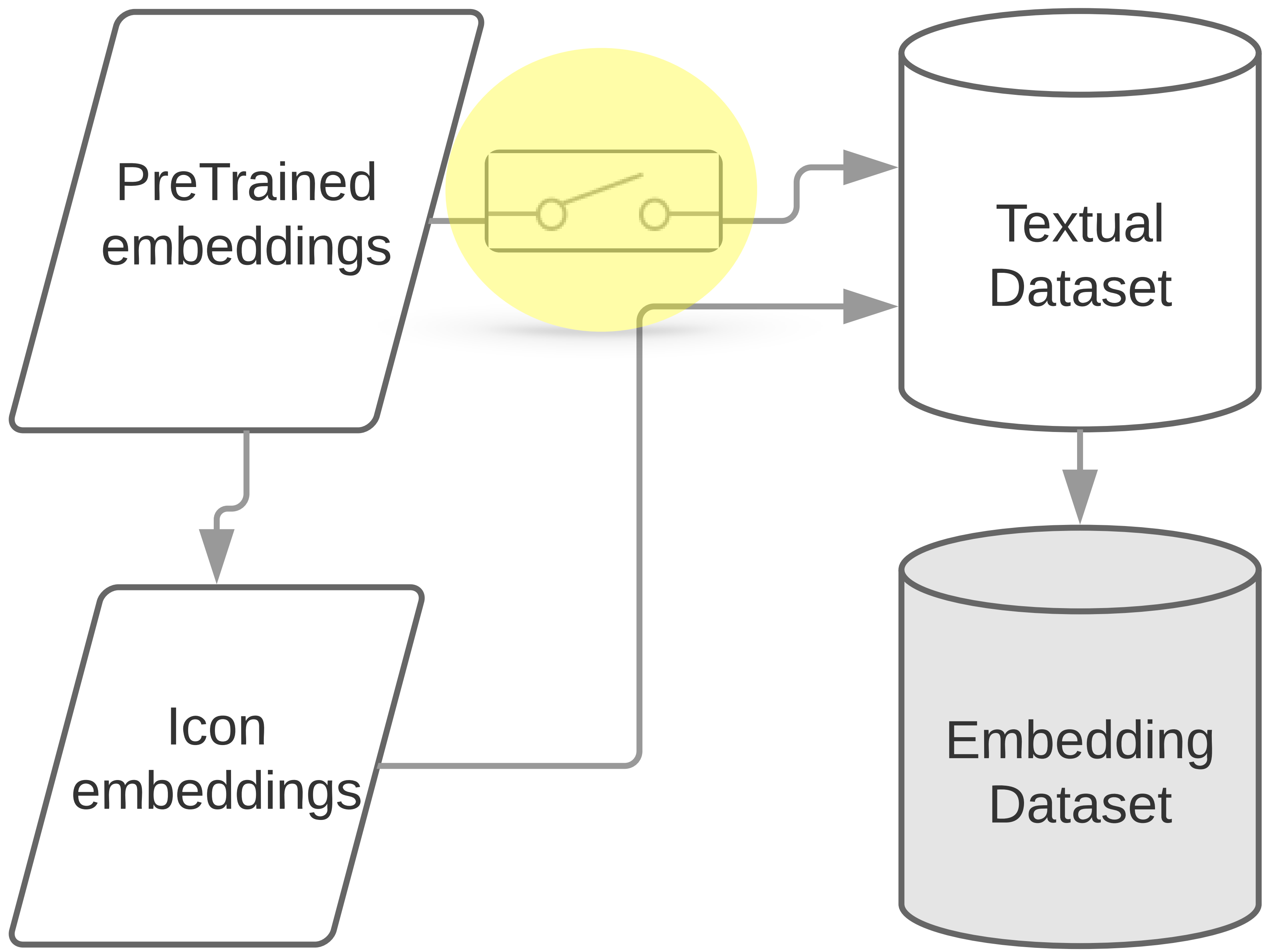
Simple (any architecture we want)

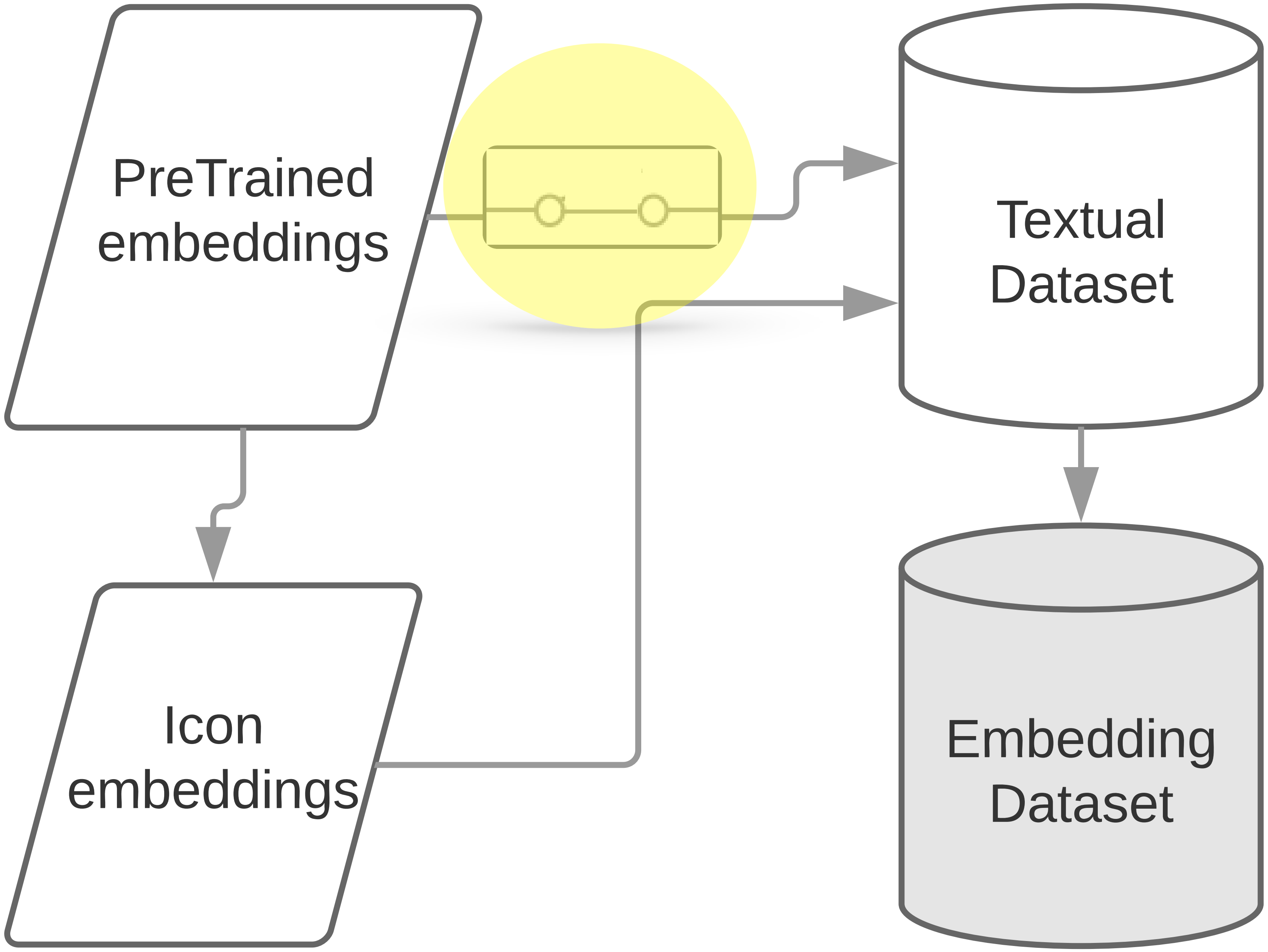
5 fold cross validation

LM prediction accuracy: MRR, ACC@1, ACC@10

How to evaluate our choices throughout the process


PreTrained
embeddings
Textual
Dataset
Icon
embeddings
Embedding
Dataset

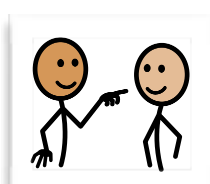
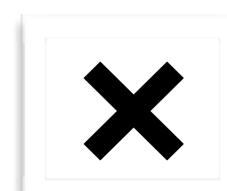
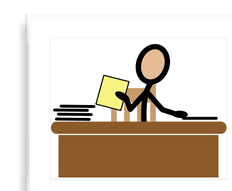

PreTrained embeddings
Textual Dataset
Icon embeddings
Embedding Dataset


PreTrained embeddings
Textual Dataset
Icon embeddings
Embedding Dataset

Experiment 2: Icon constraint

English: “your warning did not work”

non-pure:

<your> <warning> <did> <not> <work>

Experiment 2: Icon constraint

English: “your warning did not work”

non-pure:

<your> <warning> <did> <not> <work>

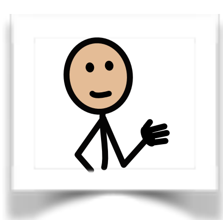
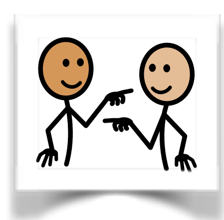
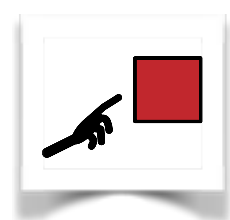
pure:

<your> <warning> <not> <work>

Experiment 2: Icon constraint

English: “so you did it”

non-pure:

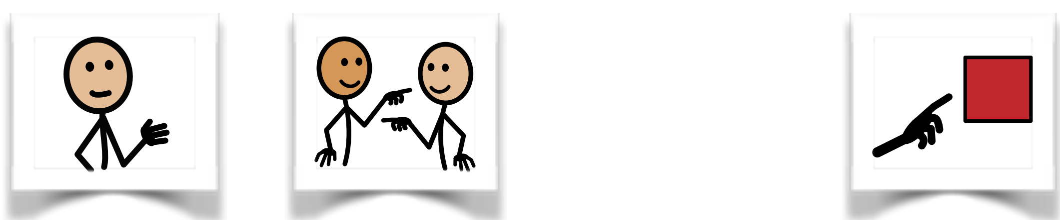
  

<so> <you> <did> <it>

Experiment 2: Icon constraint

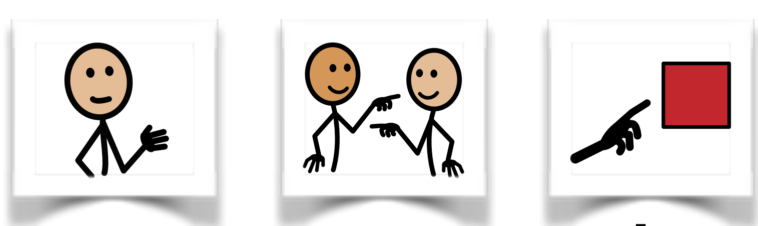
English: “so you did it”

non-pure:

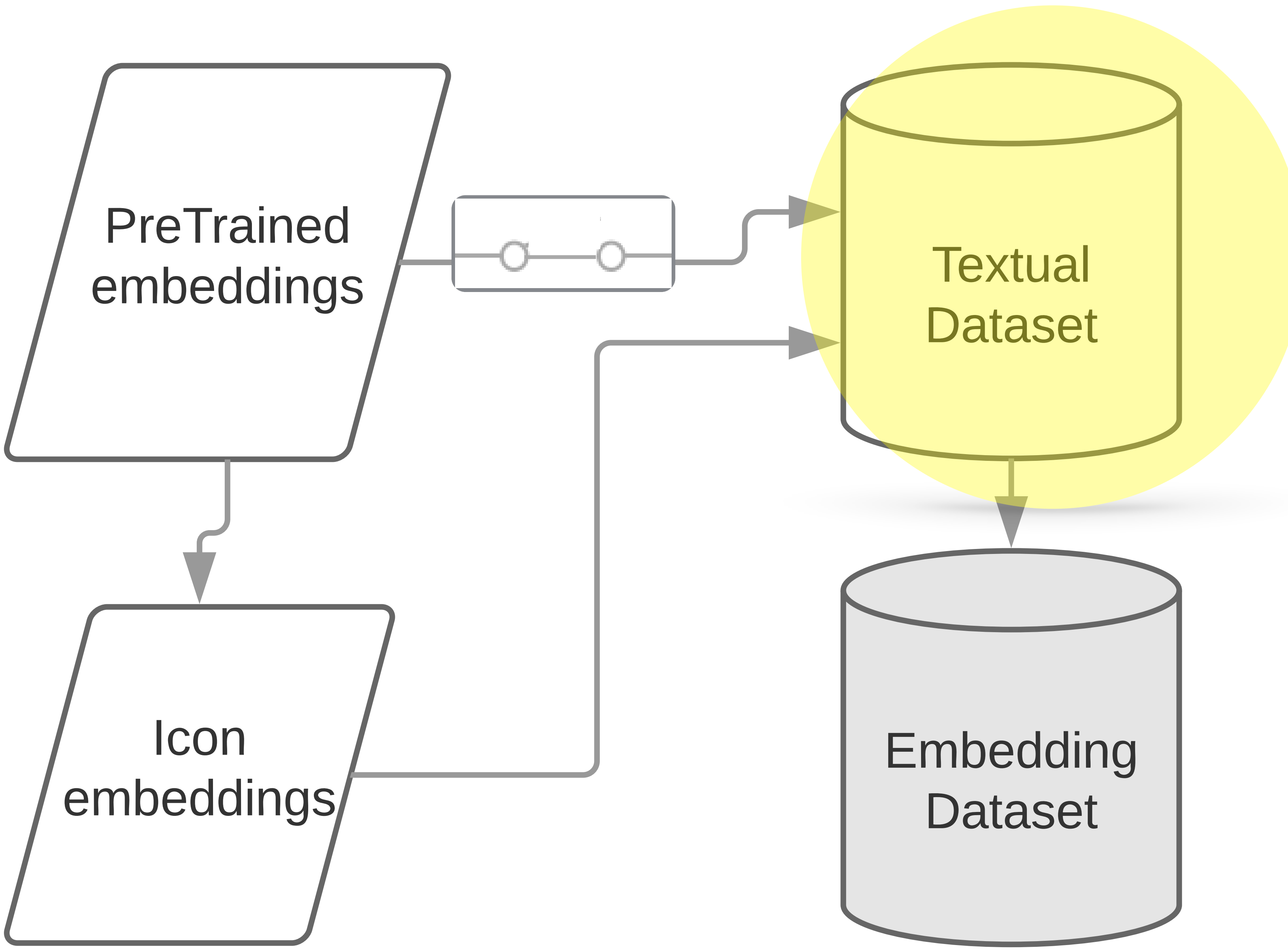


<so> <you> <did> <it>

pure:



<so> <you> <it>


PreTrained embeddings
Textual Dataset
Icon embeddings
Embedding Dataset

Experiment 1+2: Pretrained, Icon constraint

Glove	non-pure
MRR	0.85
ACC@1	49.29
ACC@10	92.29

c2v	non-pure
MRR	0.85
ACC@1	50.99
ACC@10	90.51

Experiment 1+2: Pretrained, Icon constraint

Glove	non-pure	pure
MRR	0.85	0.33
ACC@1	49.29	45.72
ACC@10	90.29	54.29

c2v	non-pure	pure
MRR	0.85	0.33
ACC@1	50.99	46.79
ACC@10	90.51	54.92

Main Limitation or advantage:

Icon language projection on textual language

Future work:

Subjective evaluation by end users

Multi phrase representation for the icons

Multi sense representation for the icons

Additional material

The research paper: https://www.aclweb.org/anthology/W18-3404

The git repo: https://github.com/shiranD/icon_lm

Acknowledgements:

OHSU team:

Melanie Fried-Oken
Betts Peters
Brandon Eddie

Northeastern University team:

David Smith
Shaobin Xu

Our Funders:

NIH, award
number 5R01DC009834-09

My advisor: Steven Bedrick