

Towards Understanding Situated Text: Concept Labeling & Extensions

Antoine Bordes

antoine.bordes@lip6.fr

Nicolas Usunier

nicolas.usunier@lip6.fr

&

Jason Weston

jaseweston@gmail.com

Ronan Collobert

ronan@collobert.com

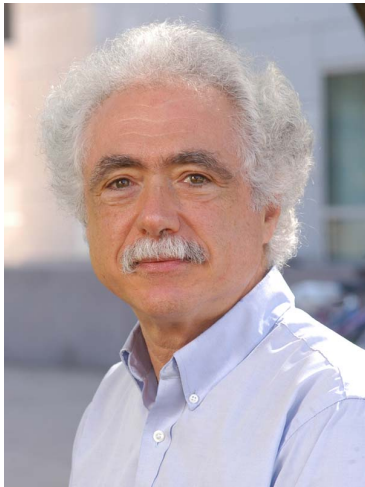
LIP6 - Université Paris 6
Paris, France

NEC Laboratories America
Princeton, USA

World Knowledge

“When a human reader sees a sentence, he uses knowledge to understand it. This includes **not only grammar, but also his knowledge about words, the context of the sentence, and most important, his knowledge about the subject matter.**

A computer program supplied with **only grammar** for manipulating the syntax of language **could not produce a translation of reasonable quality.**”



Terry Winograd – 1971

in *Procedures as a Representation for Data in a Computer program for Understand Natural Language*

Connect Natural Language to the World

- **Our goal:** learning from scratch to use both **syntax** and **the surrounding environment** to “understand” natural language.
- We understand language because it has a deep connection to the world it is used in/for → *strong prior knowledge*.

“John saw Bill in the park with his telescope.”

“He passed the exam.”

“John went to the bank.”

World knowledge we might already have:

Bill owns a telescope.

Fred took an exam last week.

John is in the countryside (not the city).

The Concept Labeling Task

- **Definition:**

Map **any natural language sentence** $x \in \mathcal{X}$ to **its labeling in terms of concepts** $y \in \mathcal{Y}$, where y is a sequence of concepts.

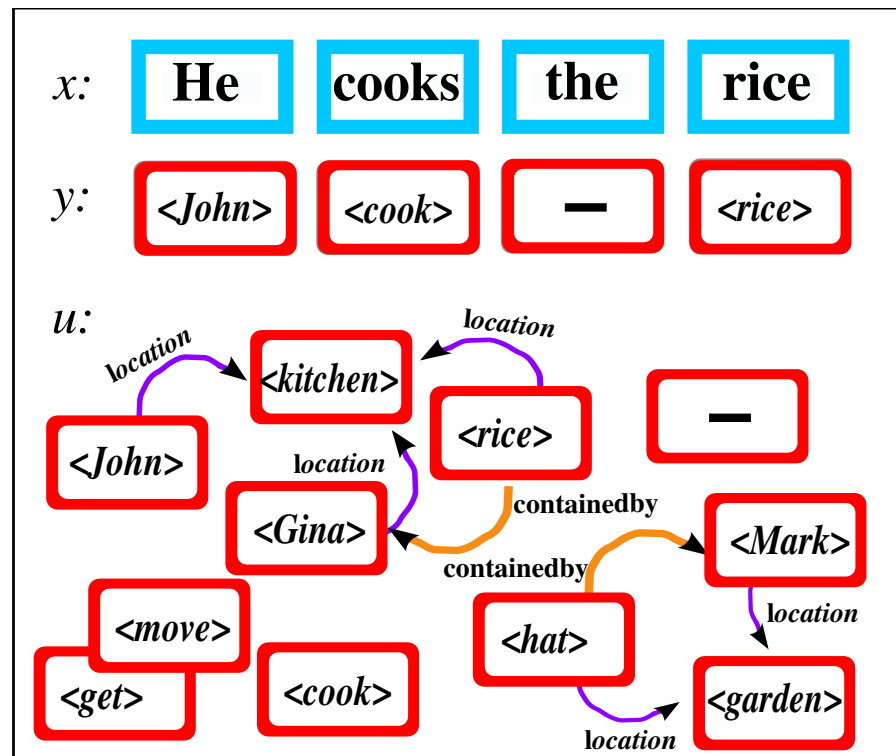
- One is given **training data triples** $\{\mathbf{x}_i, \mathbf{y}_i, \mathbf{u}_i\}_{i=1, \dots, m} \in \mathcal{X} \times \mathcal{Y} \times \mathcal{U}$ where \mathbf{u}_i is the current state the world.
- **Universe** = **set of concepts and their relations** to other concepts, $\mathcal{U} = (\mathcal{C}, \mathcal{R}_1, \dots, \mathcal{R}_n)$, where n is the number of types of relation and $\mathcal{R}_i \subset \mathcal{C}^2, \forall i = 1, \dots, n$.

Example of Concept Labeling

Define two relation tables:

- $location(c) = c'$ with $c, c' \in \mathcal{C}$,
- $containedby(c) = c'$ with $c, c' \in \mathcal{C}$.

A training triple $(\mathbf{x}, \mathbf{y}, \mathbf{u}) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{U}$:



Semantics

- Concept labeling is not sufficient for semantic interpretation.
- Just add **Semantic Role Labeling**:

He cooks the rice
<John> <cook> - <rice>
ARG1 REL - ARG2 → <cook>(<John>, <rice>)

- The system can **update its own world representation** and carry on **story understanding**.
- For example:
“John went to the kitchen and Mark stayed in the living room.”
“He cooked the rice and served dinner.”

Ambiguities We Will Handle

The **main difficulty** of concept labeling → **ambiguous words**.

He picked up the hat **there**.

The **milk** on the table.

The **one** on the table.

She left the kitchen.

The adult left the kitchen.

Mark drinks the **orange**.

...

(e.g. for sentence (2) there may be several milk cartons that exist...)

Mix of word sense disambiguation, reference resolution and entity recognition.

Concept Labeling Is Challenging

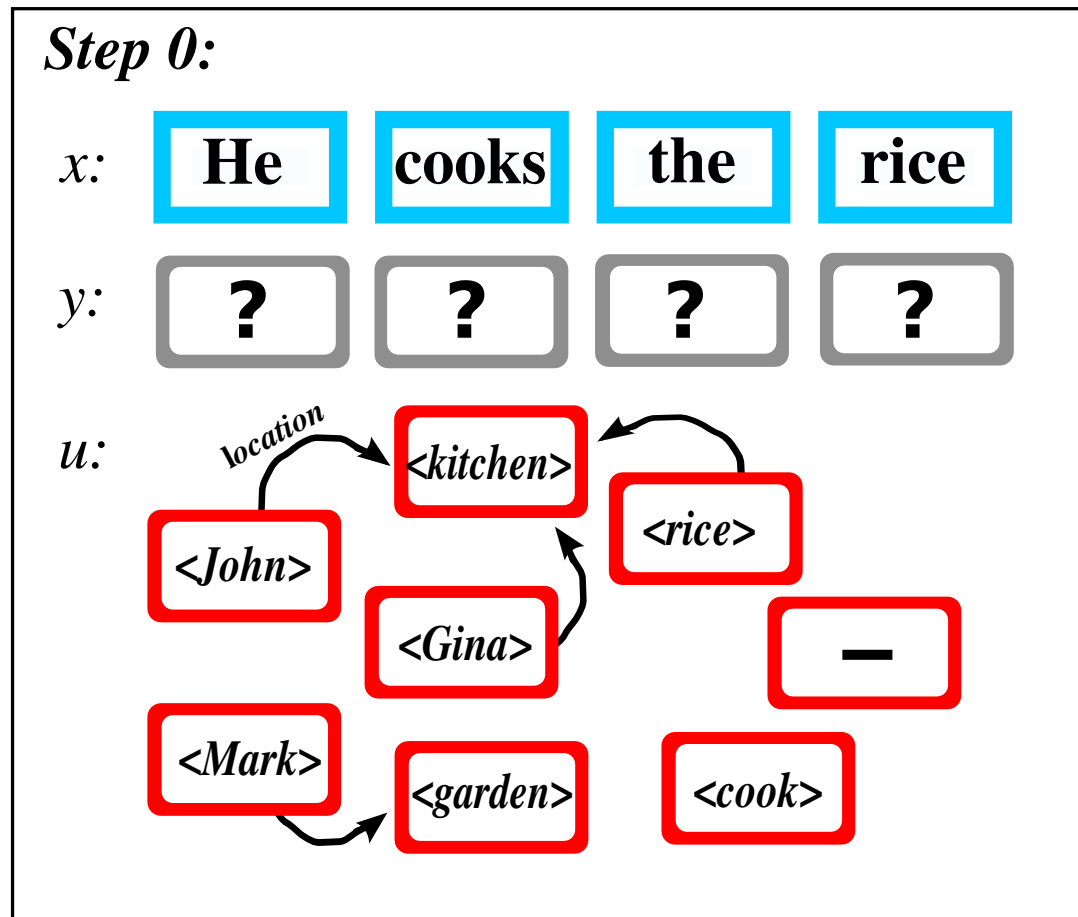
- Solving ambiguities requires to use **rules** based on **linguistic information** and **available universe knowledge**.
- **But**, these rules are **never made explicit** in training.

→ A concept labeling algorithm has to **learn them**.

- **No engineered features** for describing words/concepts are given.

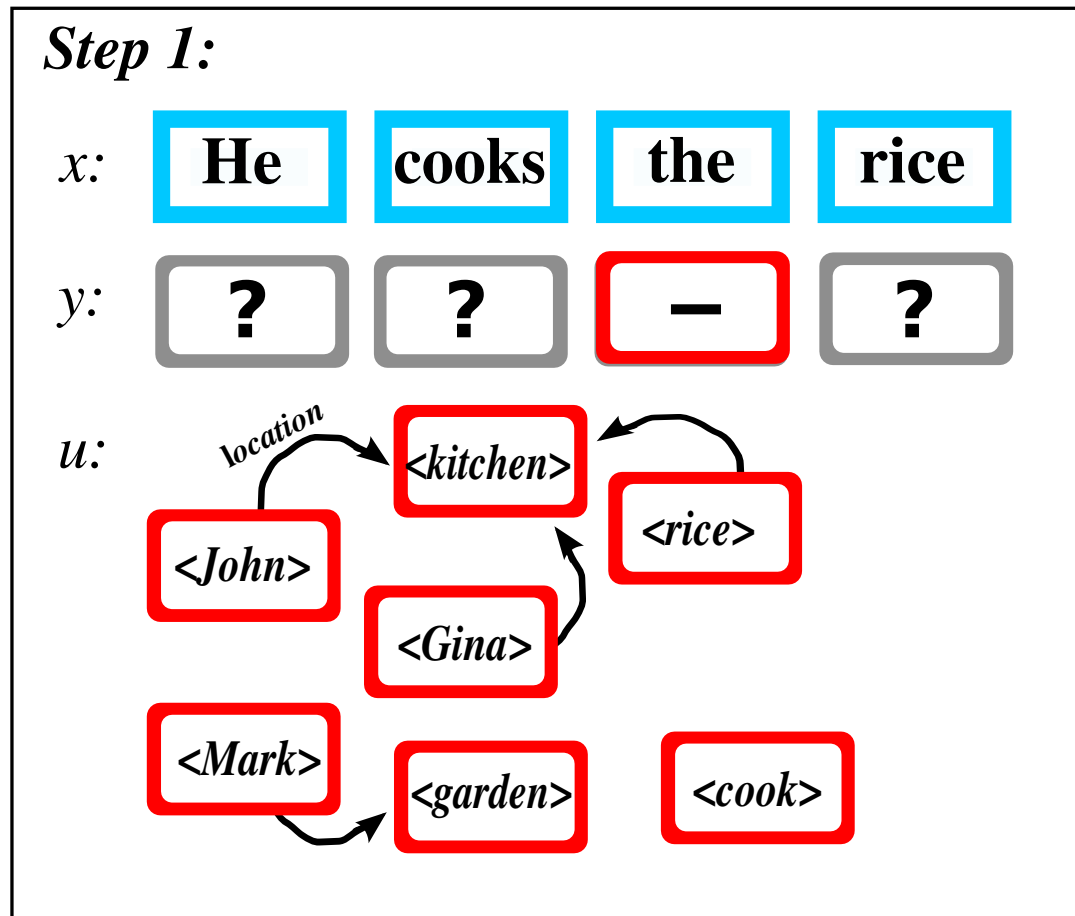
→ A concept labeling algorithm has to **discover them from raw data**.

Disambiguation Example



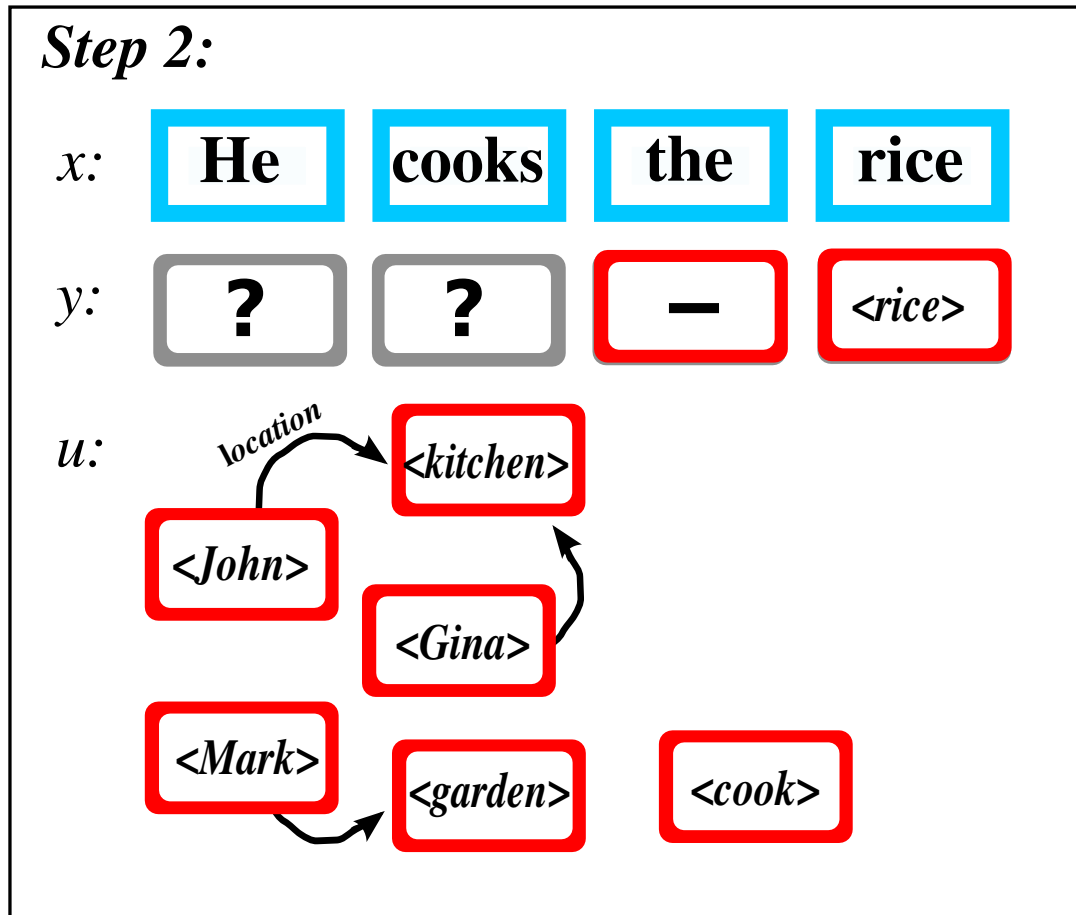
Label the above sentence.

Disambiguation Example



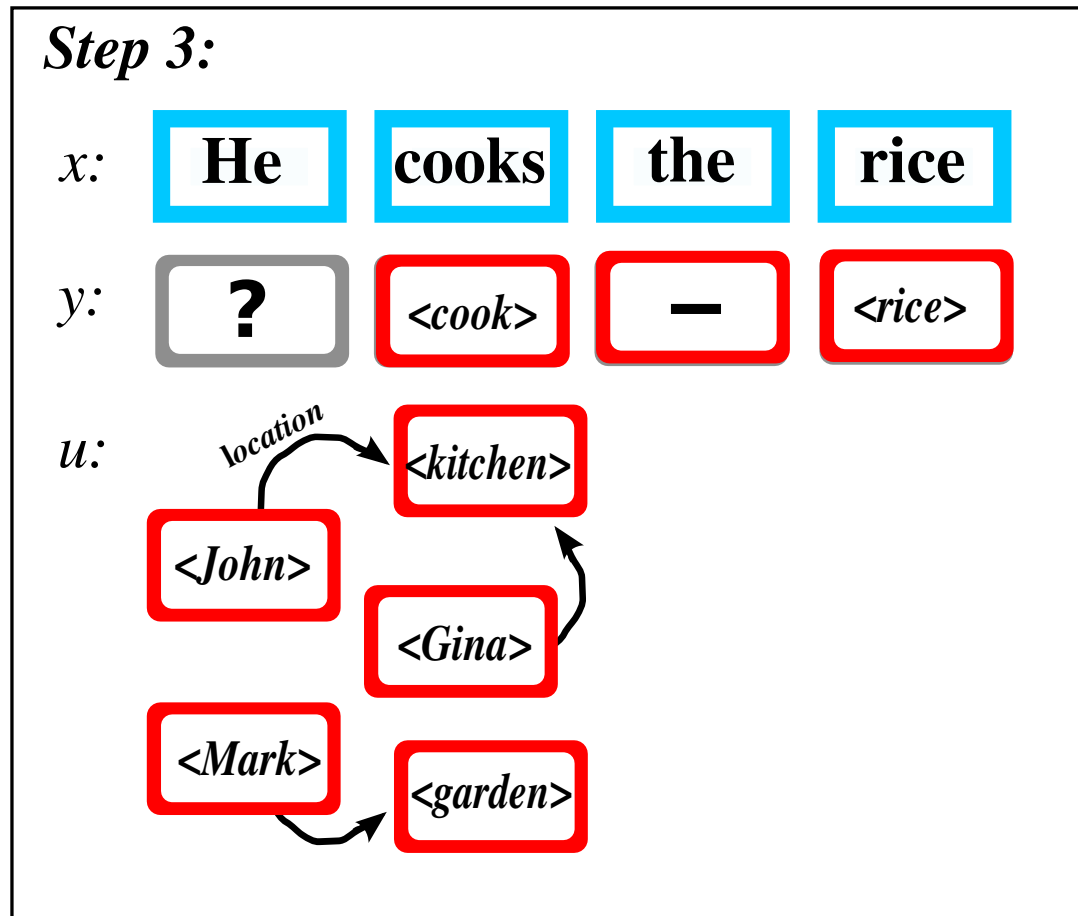
You have to start by labeling non-ambiguous words.

Disambiguation Example



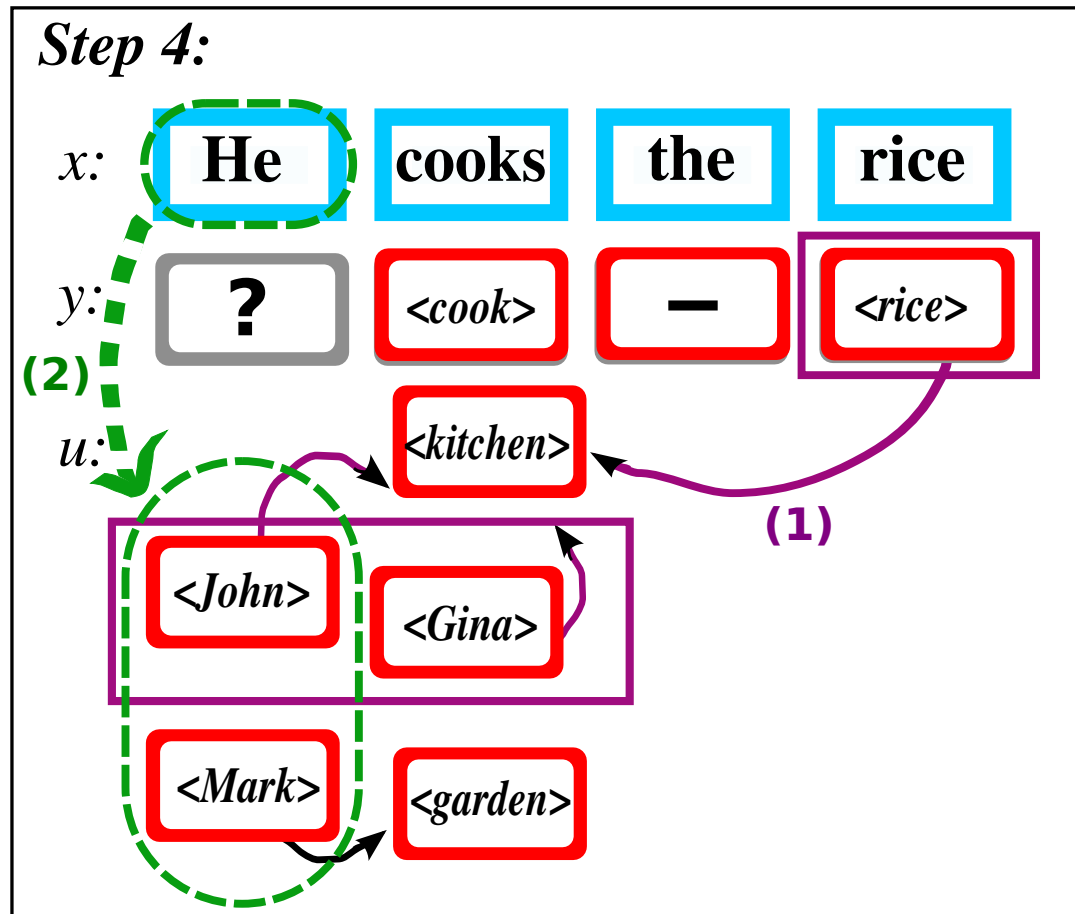
Again...

Disambiguation Example



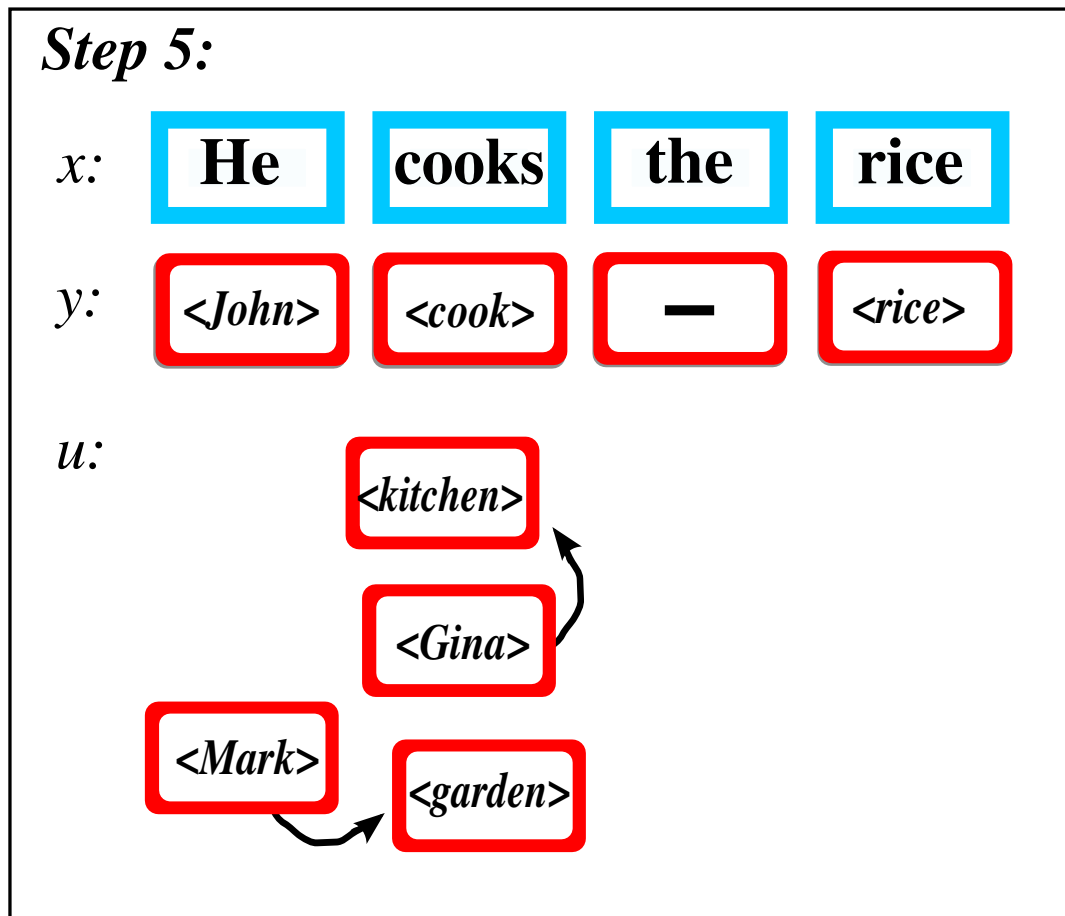
Still obvious...

Disambiguation Example



Label "He" requires two rules which are never explicitly given.

Disambiguation Example



“John” is the only male in the kitchen!

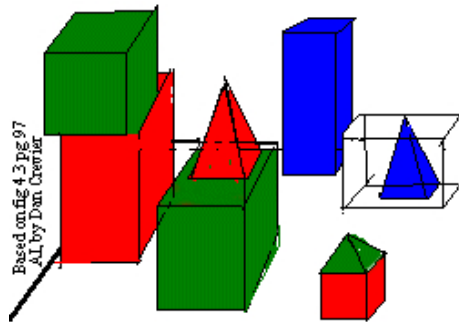
(Some of the) Previous Work

No use of world knowledge as input (only natural language):

- Mapping language with visual reference: [Winston '76], [Thibadeau '86], [Siskind '96], [Yu & Ballard '04], [Barnard & Johnson '05], [Fleischman & Roy '07].
- Mapping from sentences to meaning in formal language: [Zettlemoyer & Collins, '05], [Wong & Mooney, '07], [Chen & Mooney '08]
- Example applications:
 - (i) word-sense disambiguation (from images),
 - (ii) generate Robocup commentaries from actions,
 - (iii) convert questions to database queries.

SHRDLU & Block Worlds

- **SHRDLU**: early natural language understanding computer program. [Winograd, '72],[Bobrow & Winograd, '76]
- Use both language and world knowledge as input.



- Great success of AI → **great hopes**.
- **No later success on more realistic situations.**
- **Problem:** SHRDLU involves hand-coding in 2 ways,
(1) World model (block world).
(2) **Mapping natural language to world.**

Learning Algorithm : Basic ArgMax

We use a matching score :

$$\hat{y} = f(x, u) = \operatorname{argmax}_{y'} g(x, y', u),$$

$g(\cdot)$ is a scoring function which should be large if concepts y' are consistent with both the sentence x and the current state of the universe u .

Due to the complexity of the tagging problem, a complete argmax computation could be very slow...

Greedy “Order-free” Inference

Adapted from LaSO (Learning As Search Optimization) [Daumé & al., '05].

Inference algorithm:

1. For all the positions **not yet labeled**, “guess” what the corresponding concept would be.
2. **Select** the pair (**position, concept**) you are the most confident in. (*hopefully the least ambiguous*)
3. **Remove** this position from the set of available ones.
4. Collect all **universe-based features** of this **concept** to help label remaining ones.
5. Loop.

Scoring Function

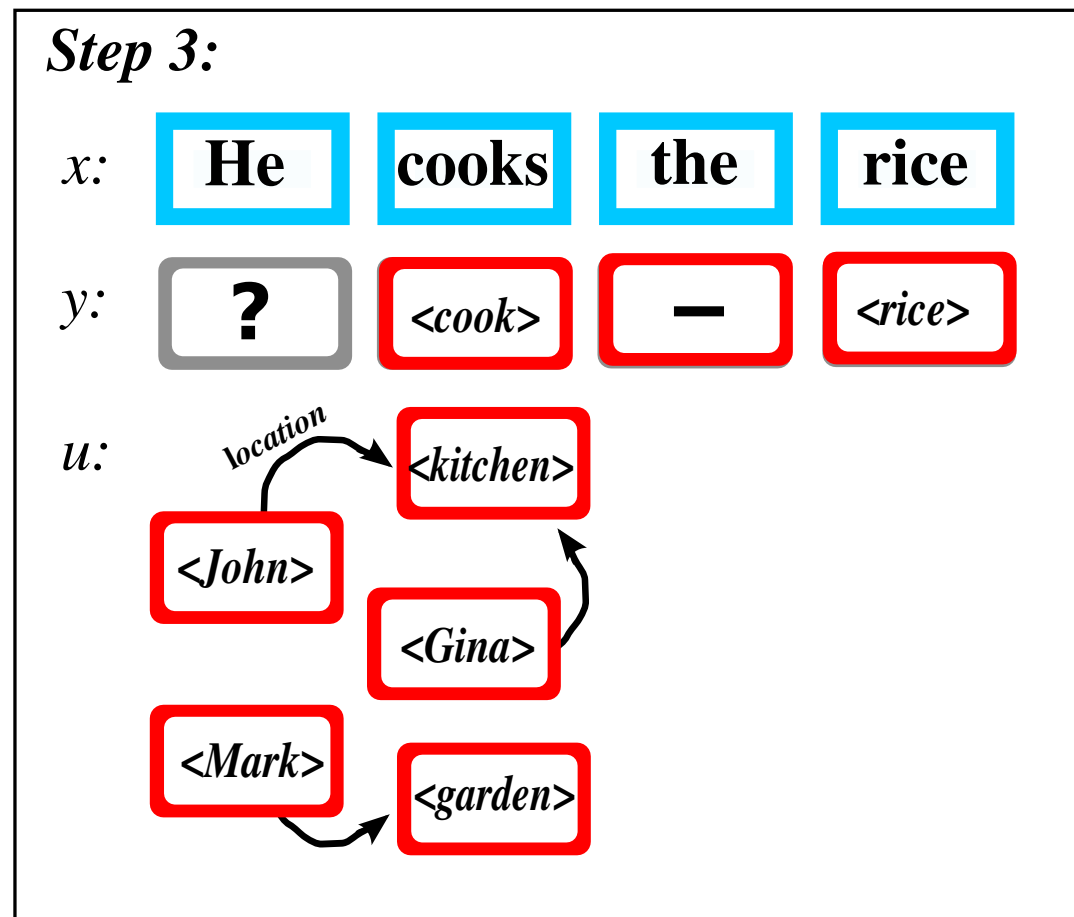
Our score combines two functions $g_i(\cdot)$ and $h(\cdot) \in \mathbb{R}^N$ which are neural networks.

$$g(x, y, u) = \sum_{i=1}^{|x|} g_i(x, y_{-i}, u)^\top h(y_i, u)$$

- $g_i(x, y_{-i}, u)$ is a **sliding-window** on the text *and* neighboring concepts centered around i^{th} word \rightarrow embeds to N dim-space.
- $h(y_i, u)$ embeds the i^{th} concept to N dim-space.
- **Dot-product**: confidence that i^{th} word labeled with concept y_i .

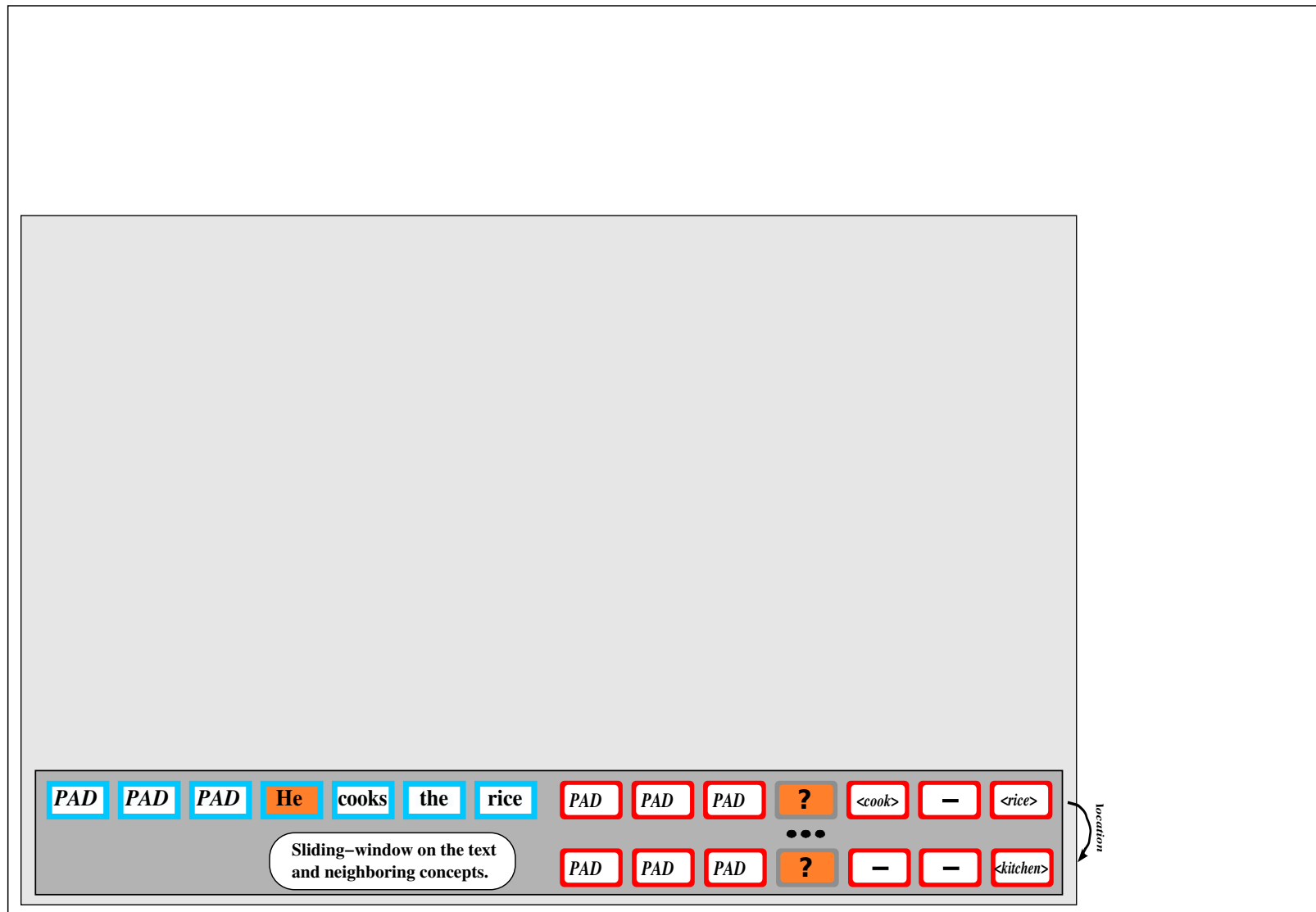
Scoring Illustration

Let's get back to our previous example:



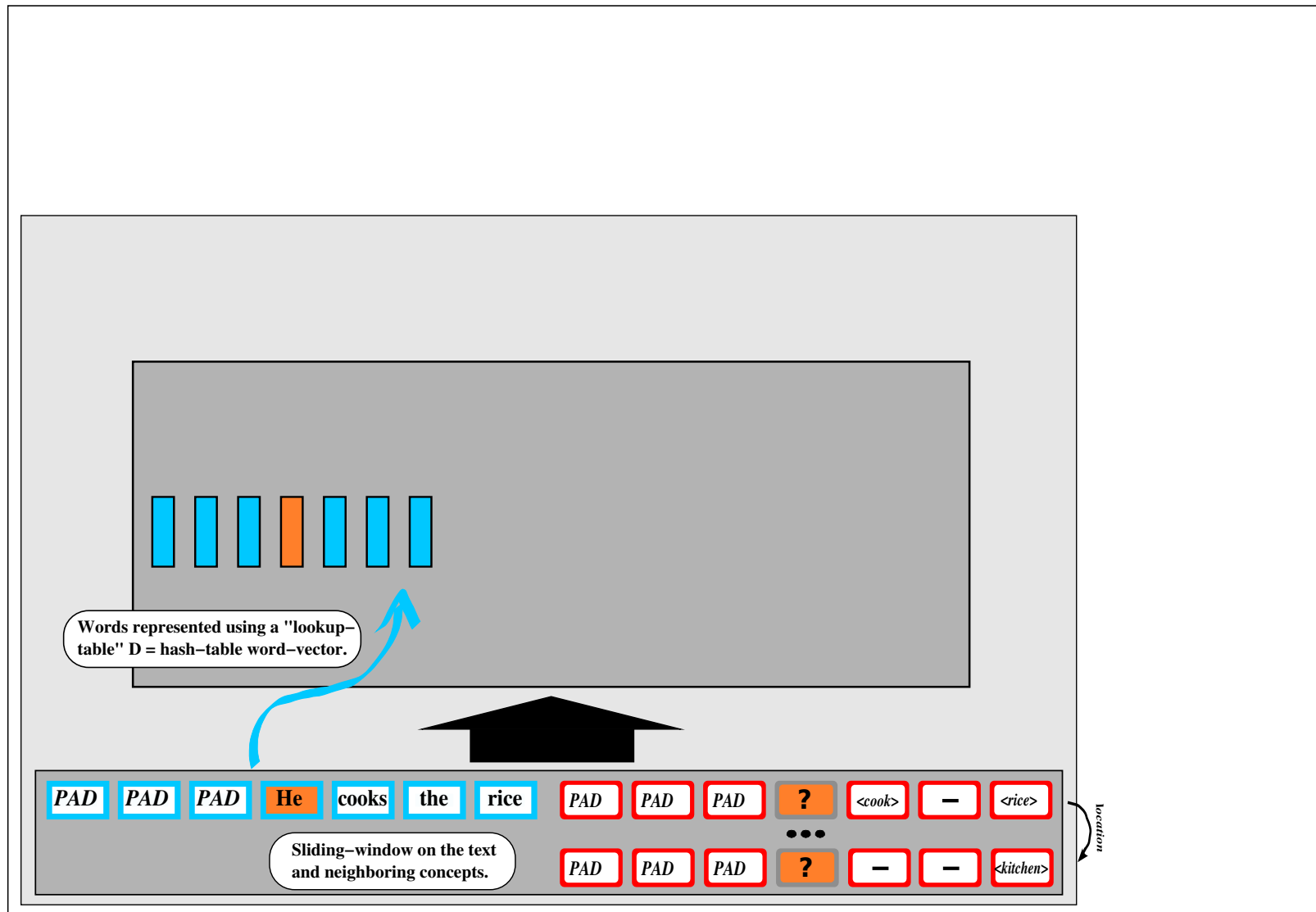
Scoring Illustration

Step 0: Set the sliding-window around the 1st word.



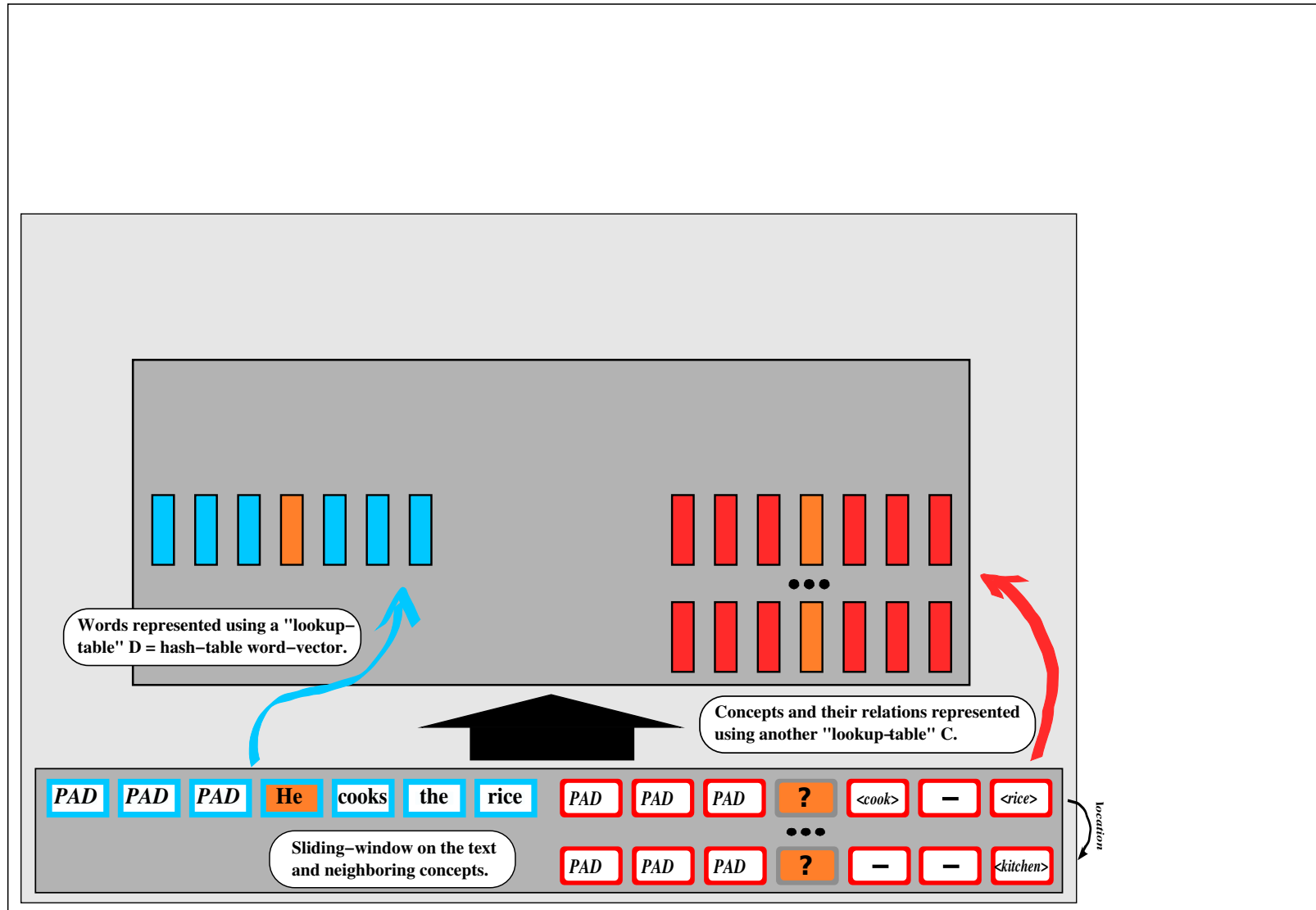
Scoring Illustration

Step 1: Retrieve words representations from the "lookup table".



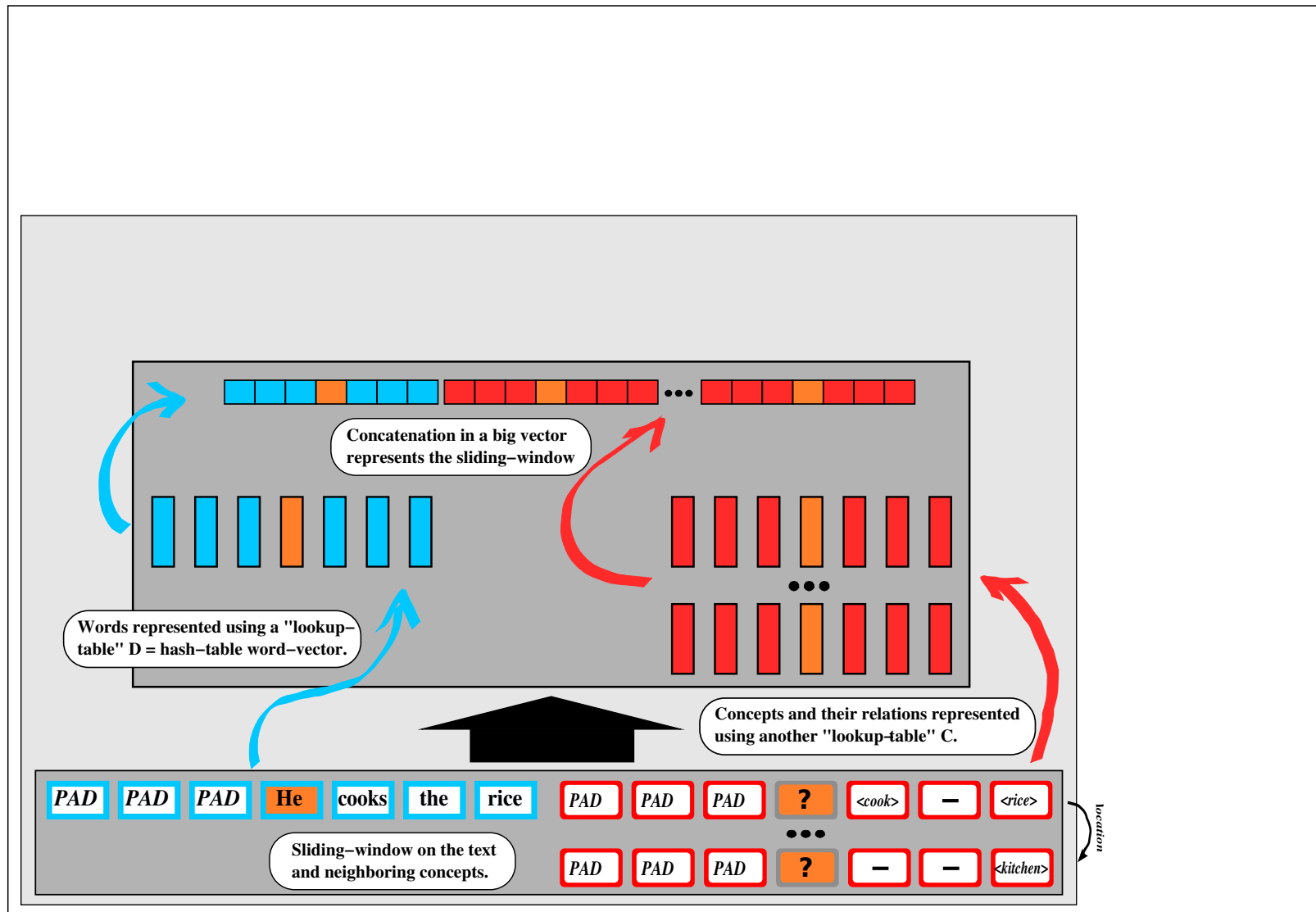
Scoring Illustration

Step 2: Similarly retrieve concepts representations.



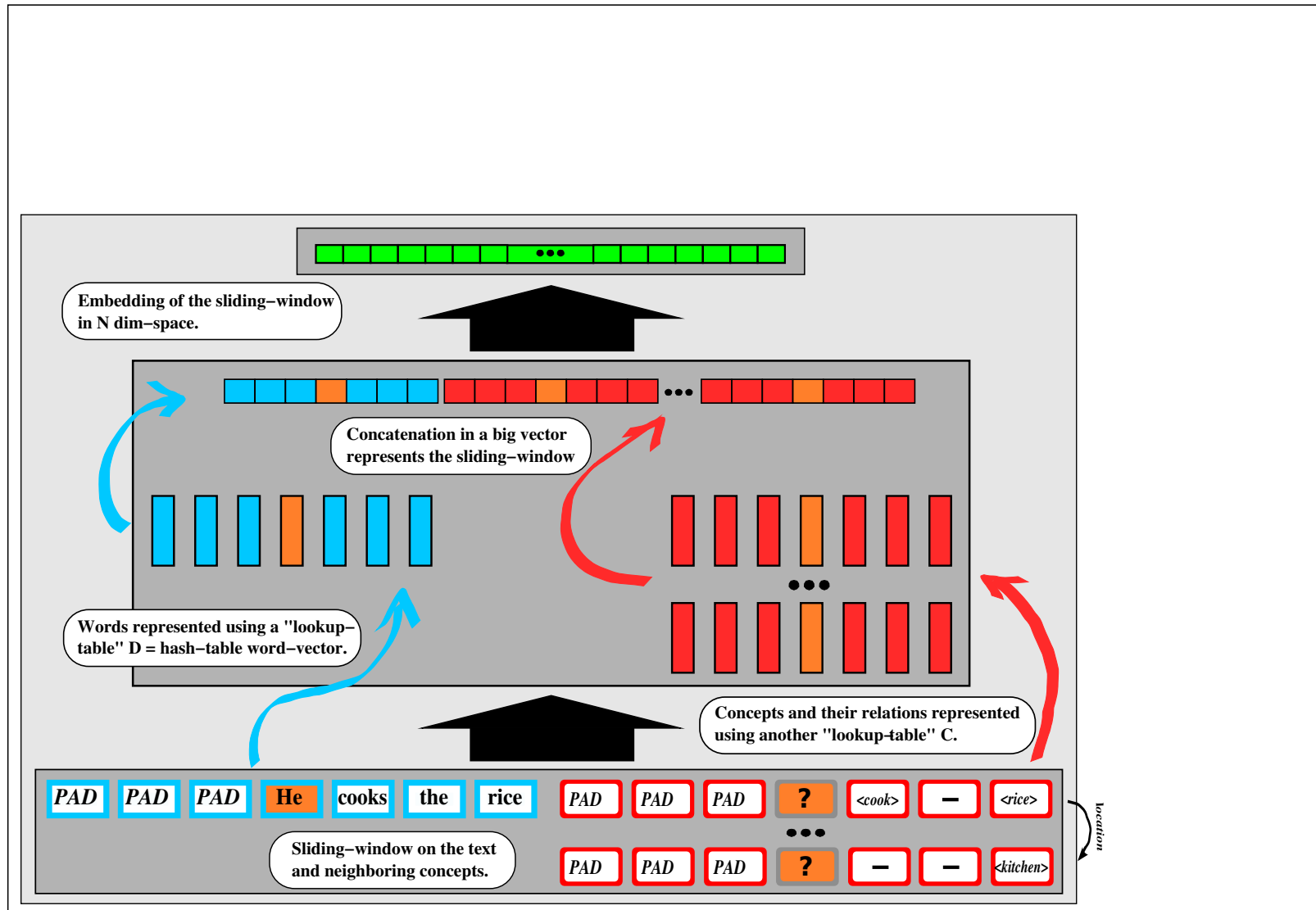
Scoring Illustration

Step 3: Concatenate vectors to obtain window representation.



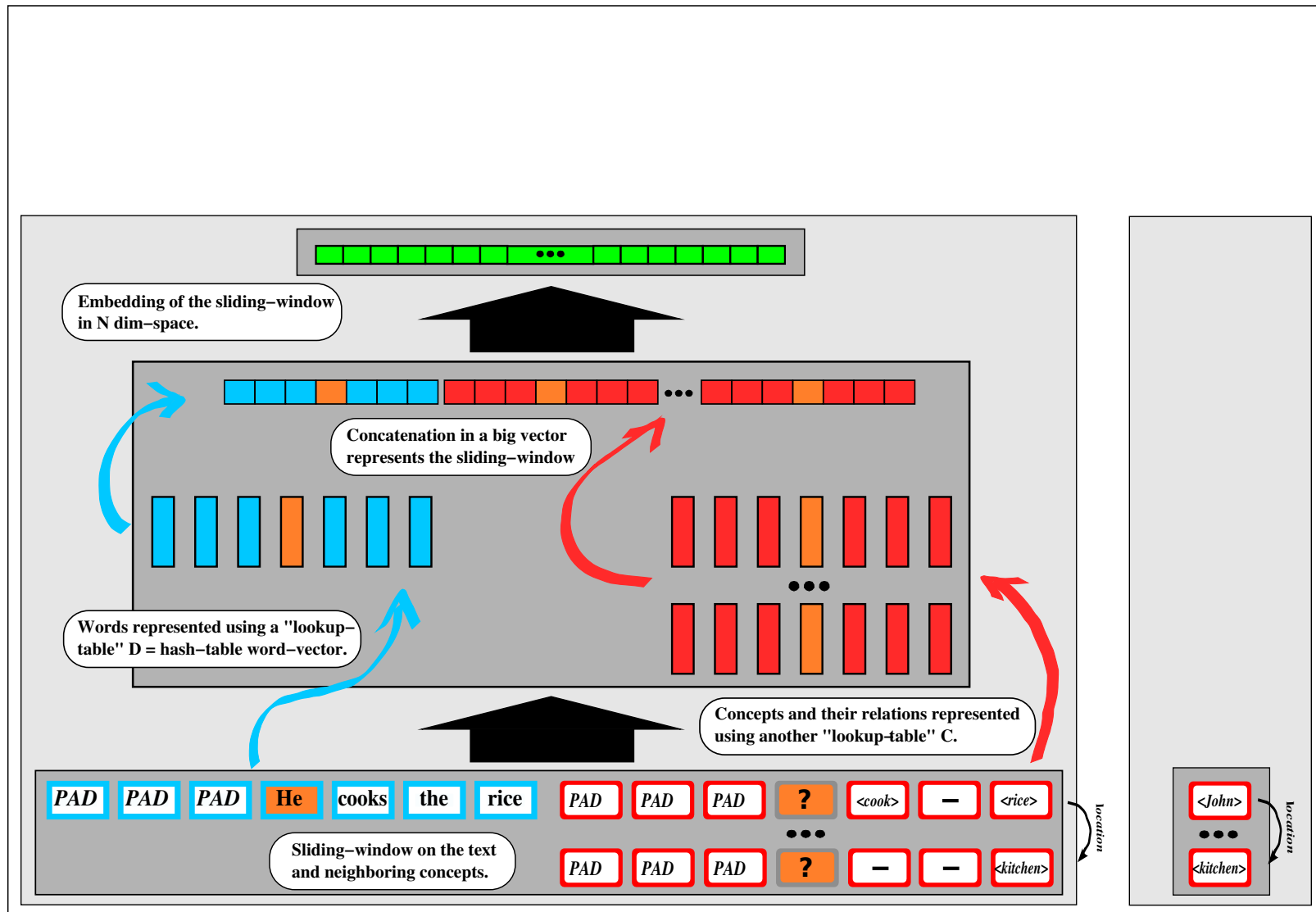
Scoring Illustration

Step 4: Compute $g_1(x, y_{-1}, u)$.



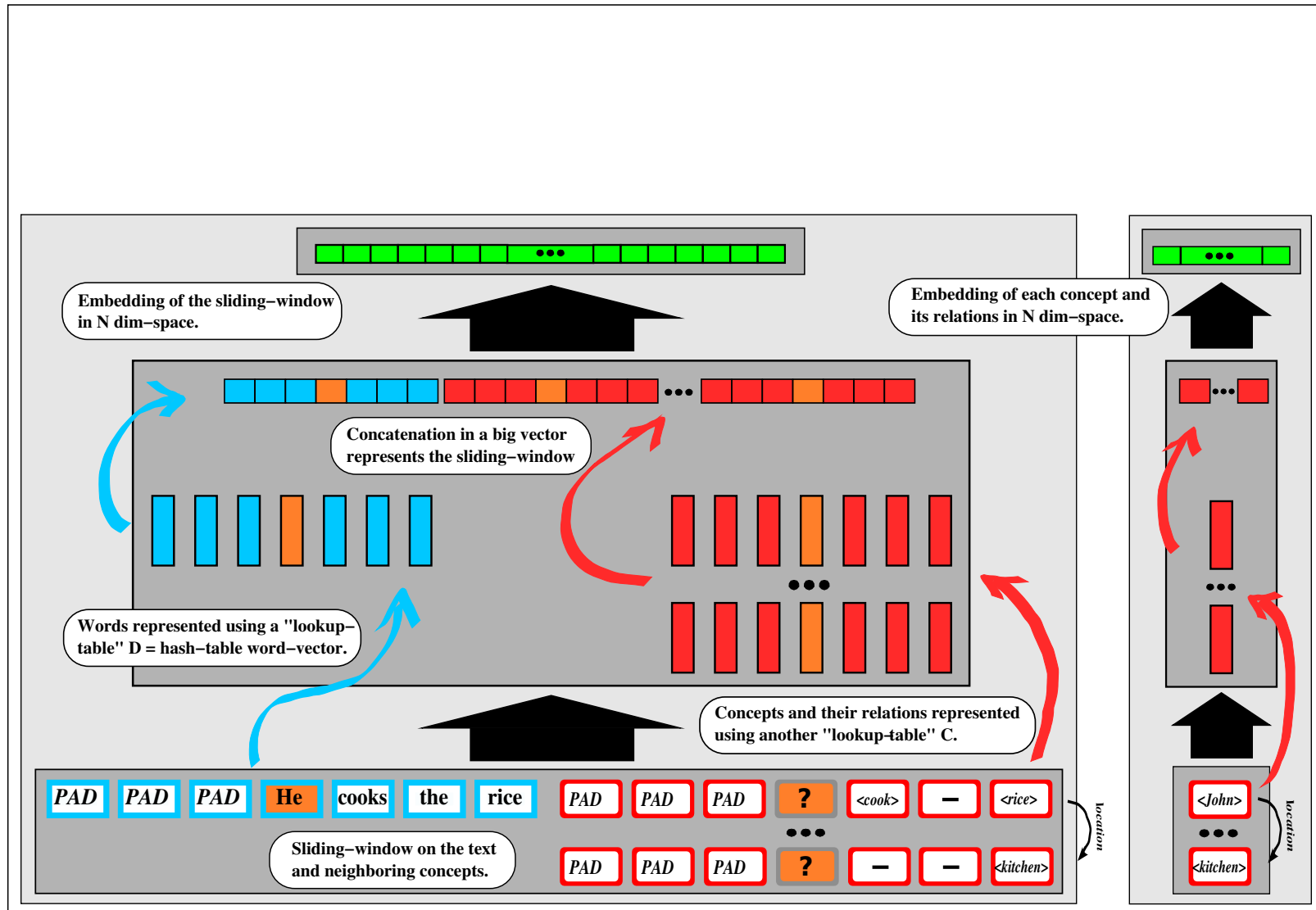
Scoring Illustration

Step 5: Get the concept $\langle John \rangle$ and its relations.



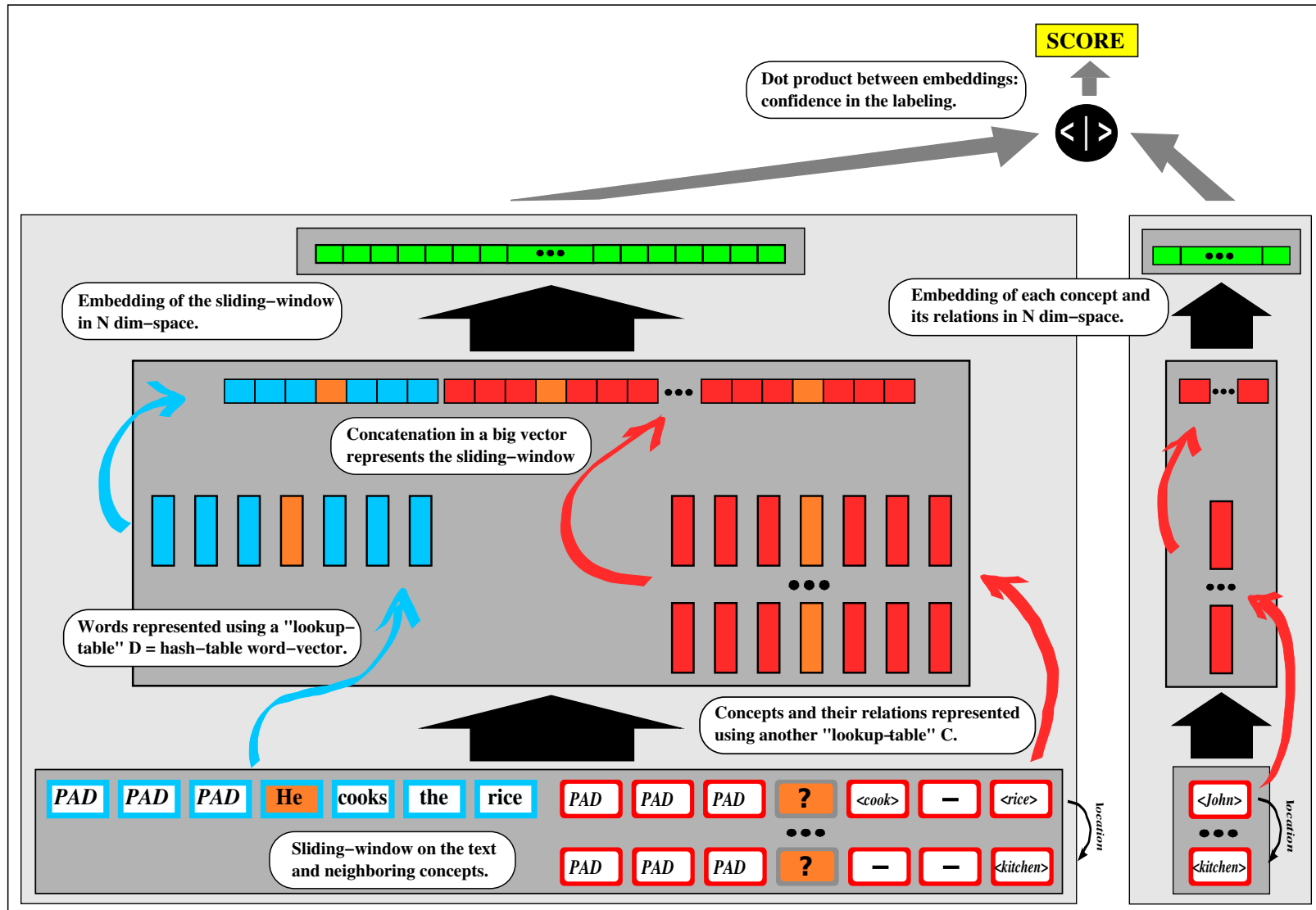
Scoring Illustration

Step 6: Compute $h(\langle John \rangle, u)$.



Scoring Illustration

Step 7: Finally compute the score: $g_1(x, y_{-1}, u)^T h(\langle \text{John} \rangle, u)$.



Train the System

- **Online training** i.e. prediction and update for each example.
- **At each greedy step**, if a prediction \hat{y}^t is **incorrect**, several updates are made to the model to satisfy:

For each **correct** labeling alternative $\hat{y}_{+y_i}^{t-1}$, $g(x, \hat{y}_{+y_i}^{t-1}, u) > g(x, \hat{y}^t, u)$.

- **Intuitively**, we want any **incorrect** partial prediction to be ranked **below all correct partial labeling**.
 - **“Order-free”** is not directly supervised.
- All updates performed with **SGD + Backpropagation**.

Generate Data by Simulation

- A.** An universe is **initialized** i.e. concepts and relations are created.
- B.** The simulation algorithm is run with:
1. **Generate a new event**, $(v, a) = event(u)$.
Generates verb + set of args \rightarrow a *coherent* action given the universe. *E.g. actors change location and pick up, exchange objects. . .*
 2. **Generate a training triple**, i.e. $(x,y)=generate(v, a)$.
Returns a sentence and concept labeling pair given a verb + args. *This sentence should describe the event.*
 3. **Update the universe**, i.e. $u = exec(v)(a, u)$.

Labeled Data

Simulation of a house with 82 concepts: 15 verbs, 10 actors, 15 small objects, 6 rooms and 12 pieces of furniture...

...

x: the father gets some yoghurt from the sideboard
y: - <John> <get> - <yoghurt> - - <sideboard>

x: he sits on the chair
y: <Mark> <sit> - - <chair>

x: she goes from the bedroom to the kitchen
y: <Gina> <move> - - <bedroom> - - <kitchen>

x: the brother gives the toy to her
y: - <Mark> <give> - <toy> - <sister>

...

→ Generation a dataset of 50,000 training triples and 20,000 testing triples (≈55% ambiguous), without any human annotation.

Experimental Results

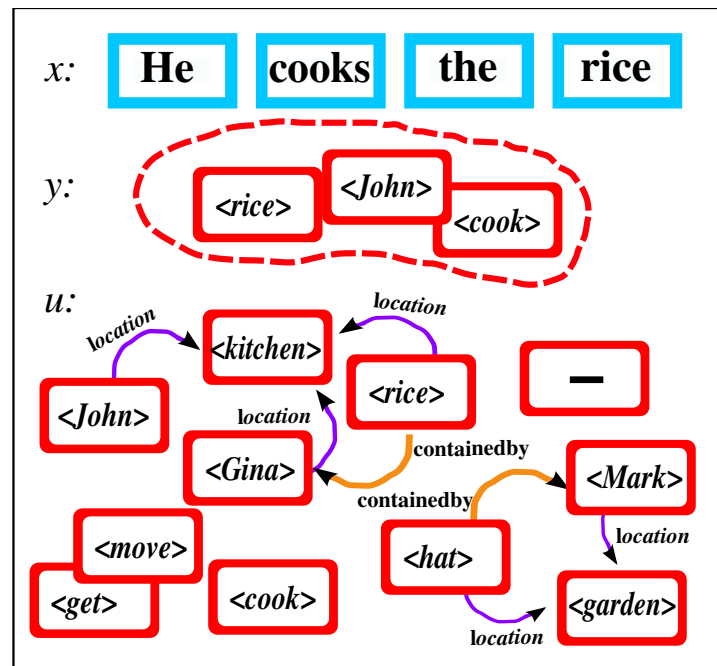
- Different tagging strategies.
- Different amounts of *universe* knowledge: no knowledge, knowledge about *containedby*, *location*, or *both*.

Method	Features	Train Err	Test Err
SVM_{struct}	x	42.26%	42.61%
SVM_{struct}	$x + u$ (<i>loc</i> , <i>contain</i>)	18.68%	23.57%
NN_{mult}	x	35.80%	36.97%
NN_{LR}	x	32.80%	35.80%
NN_{LR}	$x + u$ (<i>loc</i> , <i>contain</i>)	5.42%	5.75%
NN_{OF}	x	32.50%	35.87%
NN_{OF}	$x + u$ (<i>contain</i>)	15.15%	17.04%
NN_{OF}	$x + u$ (<i>loc</i>)	5.07%	5.22%
NN_{OF}	$x + u$ (<i>loc</i> , <i>contain</i>)	0.0%	0.11%

→ More world knowledge & OF leads to better generalization.

Extension: Weak Concept Labeling

More challenging setting: training data $\{x_i, y_i, u_i\}_{i=1, \dots, m}$ as before. However, y is a set (bag) of concepts - **no alignment to sentence**.



This is more realistic:

A child sees actions and hears sentences \rightarrow must learn correlation.

Results

Solution: modified LaSO updates – rank anything in the “bag” higher than something not in the bag.

Results on our dataset:

Method	Features	Train Err	Test Err
SVM_{struct}	$x + u$ (loc, contain)	18.68%	23.57%
NN_{OF}	$x + u$ (loc, contain)	0.0%	0.11%
NN_{WEAK}	$x + u$ (loc, contain)	0.64%	0.72%

Results on Robocup dataset [Chen & Mooney '08]:

Method	Matching F1-score
Random	0.465
Wasper	0.530
Krisper	0.645
Wasper-gen	0.650
NN_{WEAK}	0.669

Summary

- *Simple*, but *general framework* for language grounding based on the task of *concept labeling*.
- *Scalable, flexible learning algorithm* that can learn without hand-crafted rules or features.
- *Simulation validates our approach* and shows that learning to disambiguate with world knowledge is possible.

AI goal: train a character “living” in a “computer game world” to learn language from scratch i.e. *from interactions alone* (communication, actions, ...)

Thank You



Good AI never dies!