

ACKN: A context-aware keyword programming system

Shu Aochi, Masuhara Hidehiko (Tokyo Institute of Technology)

Introduction

A criteria to choose a **programming environment** is the **efficiency** in developing code.

Code recommendation is a code assist technique to save a programmer's time by providing a list of possible code.

Input of a code recommendation system can be:

- An abbreviation
- Previous context
- A keyword query
- A partial code fragment

Goal:

Build a code recommendation system that:

- Suitable for **all** programmer
- Concern the **context** of the editing file
- Can provide a **complicated** expression.

Keyword Programming [2009, Little]

Keyword programming generates **all possible expressions** and organizes them by their **scores**.

```
public List<String> getLines (BufferedReader src) throws Exception{
    List<String> array = new ArrayList<String> ( );
    while(src.ready()){
        add line
    }
    return array;
}
```

Keywords: add line
Expression: array.add(src.readLine())

Score: + 1.772

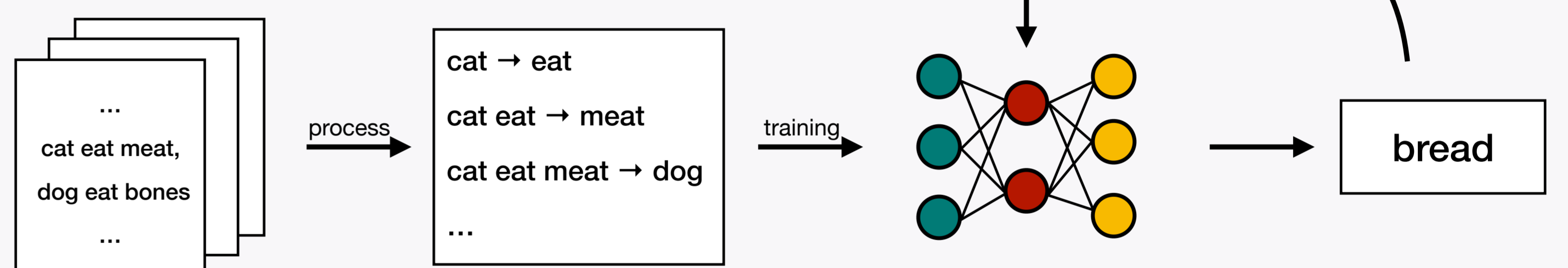
array.	add(src.	read	Line()
-0.01	+1.0	-0.01	-0.01	+1.0
-0.05	-0.05	-0.05	-0.05	-0.05
+0.001		+0.001		

Ranking algorithm:

- +1.0, $token \in keywords$
- -0.01, $token \notin keywords$
- -0.05, for each *depth*
- +0.001, if *token* is a local variable, a member variable, or a member method

Neural Text Generation

Predict **next token** according to corpus



Problems

Hard to generate complicated expressions

```
...
public void read(){
    read standard in
}
...
new InputStreamReader(System.in).read()
...
new BufferedReader(new InputStreamReader(System.in))
```

Expected but shown in the 17th

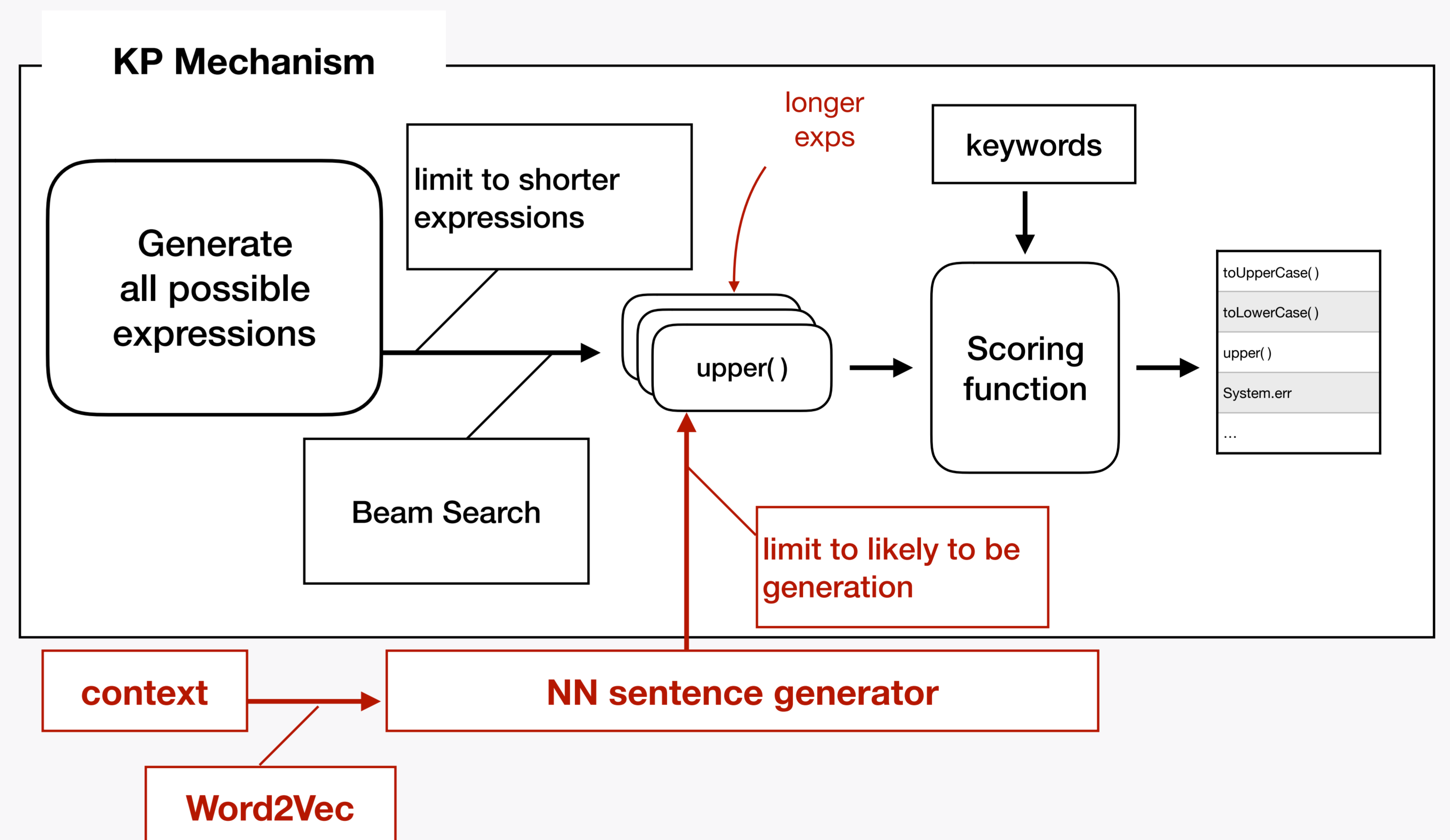
Keyword should be the substrings

```
...
Frame localFrame = newFrame(this.mOutputFormat);
// expected expression: filename.endsWith(ext.toUpperCase()) |
...
```

Keyword: path ends upcase ext Keyword: filename ends upper ext

Result: not on the list Result: 3rd

Proposal: KP + NTG



Evaluation

① Prepare 15 expressions

```
...
new BufferedReader(new
InputStreamReader(System.in))
...
```

② Collect 21 programs containing the exp

GitHub → [Programs]

③ Train LSTM with 20x15 programs

LSTM

④ Create task

```
---
---
---
Keyword
read standard in
```

⑤ Run

ACKN KP

Result

KP: 10 of 15 shown on the list, 6 of 15 are in the top-5

ACKN: 12 of 15 shown on the list, 7 of 15 are in the top-5

Future Work

- Using a larger training dataset to avoid overfit
- Set weight for the keyword query