A Cache-based Approach to Dynamic Switching between Different Dataflows in Crowdsourcing

Yusuke Suzuki[†], Masaki Matsubara[†], Keishi Tajima[‡],
Toshiyuki Amagasa[†], and Atsuyuki Morishima[†]

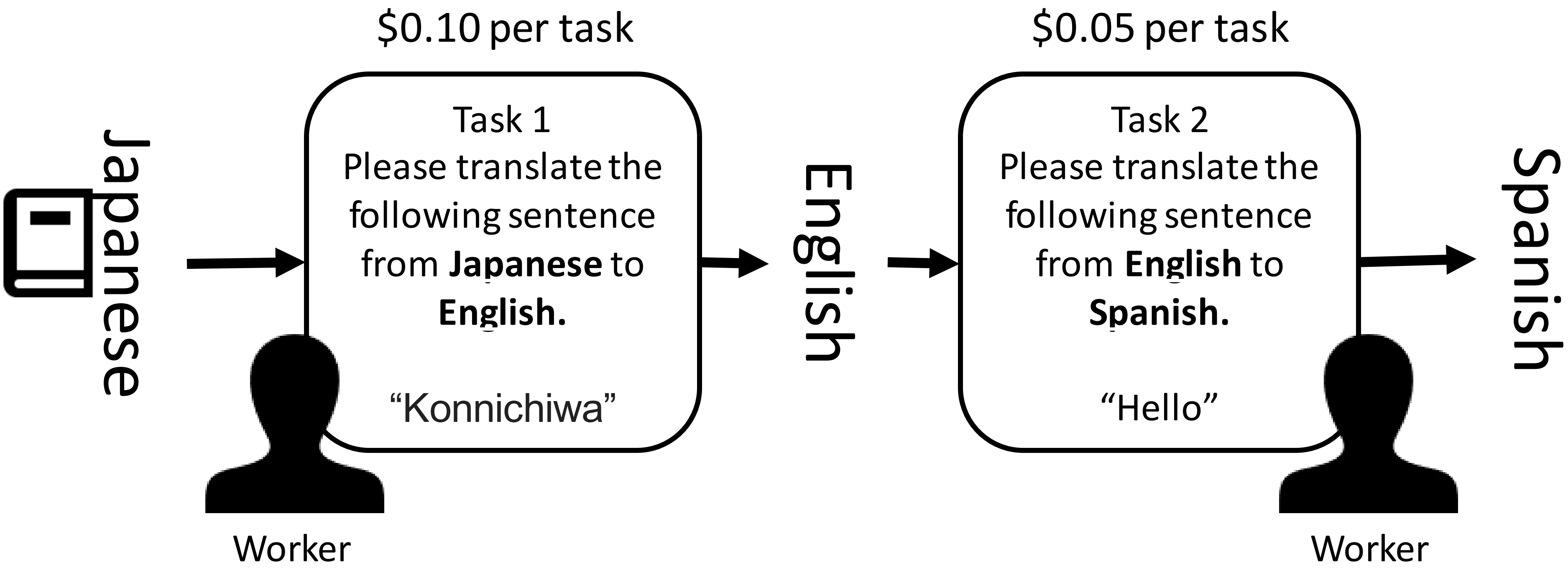
筑波大学
University of Tsukuba [†]

京都大学
KYOTO UNIVERSITY [‡]

Key points of the presentation

1. **【Background】 Crowdsourcing sometimes makes**
2. **dataflow change halfway, but it costs a lot of money.**
3. 【Related Work】 With the method using task result caches,
4. it cannot cope with dataflow change.
5. 【Our Approach】 We propose the method to use caches
6. coping with changing dataflows.
7. 【Simulation】 Our simulation results showed that it is
8. possible to identify the best point to minimize the total cost.

Background(1/4) : Crowdsourcing Takes Time and Monetary Cost



$0.10 per task

$0.05 per task

Japanese

Task 1
Please translate the following sentence from **Japanese** to **English.**

"Konnichiwa"

English

Task 2
Please translate the following sentence from **English** to **Spanish.**

"Hello"

Spanish

Worker

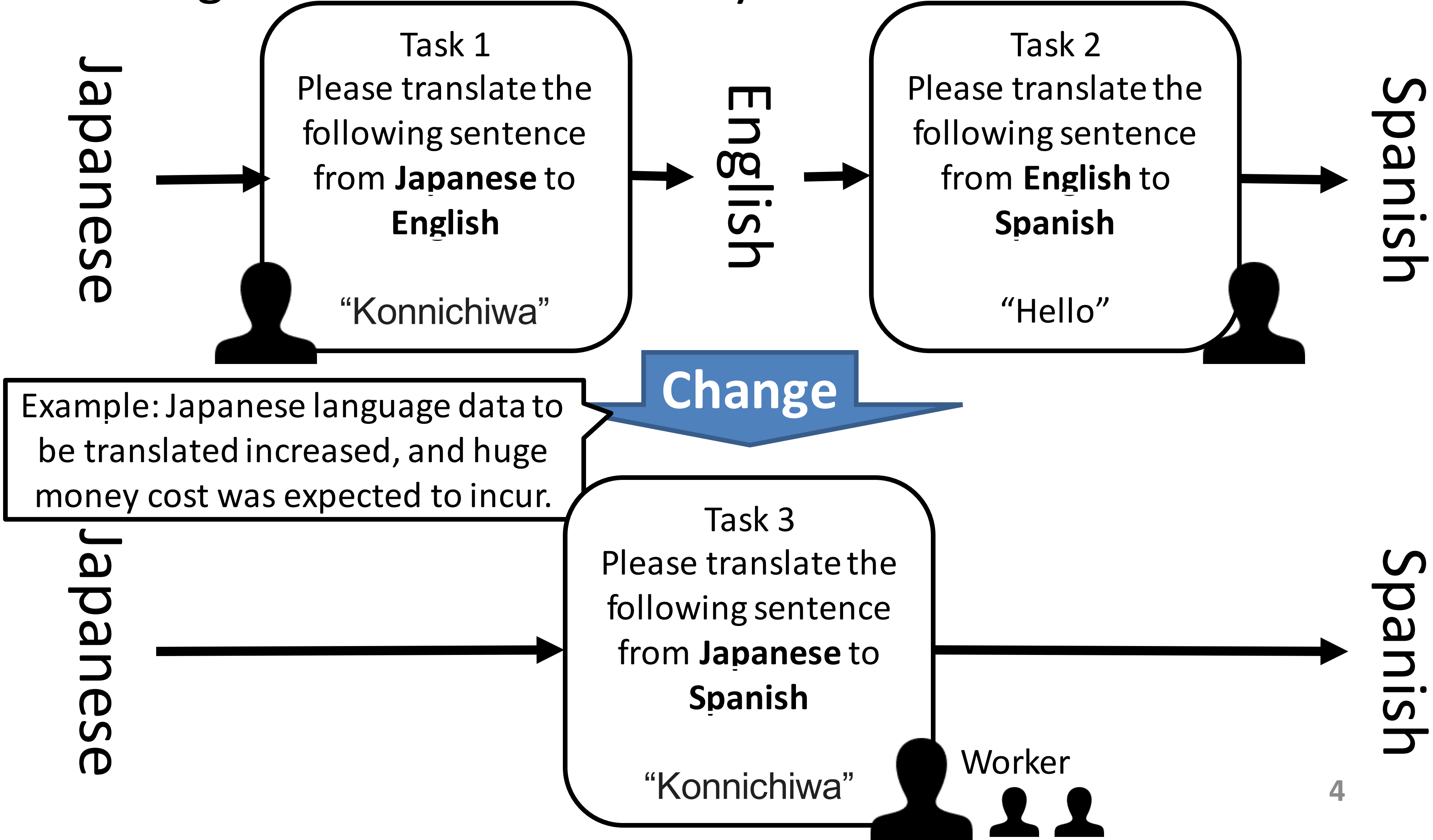
Worker

Crowdsourcing Characteristics

Because of slow processing by human, it requires time and monetary costs.

Background(2/4) : Changing Dataflows

Because completing all tasks takes long time, we may want to change the dataflow halfway due to various factors.



Japanese

Task 1
Please translate the following sentence from **Japanese** to **English**

"Konnichiwa"

English

Task 2
Please translate the following sentence from **English** to **Spanish**

"Hello"

Spanish

Change

Example: Japanese language data to be translated increased, and huge money cost was expected to incur.

Japanese

Task 3
Please translate the following sentence from **Japanese** to **Spanish**

"Konnichiwa"

Worker

Spanish

Background(3/4) : Changing dataflows in crowdsourcing are difficult

Switching query plans at appropriate timing.

(Eddies[1] et al.)

→ Since it is necessary to develop a mechanism for each type of operator, such a mechanism cannot be generalized for crowdsourcing settings where we have variety of tasks

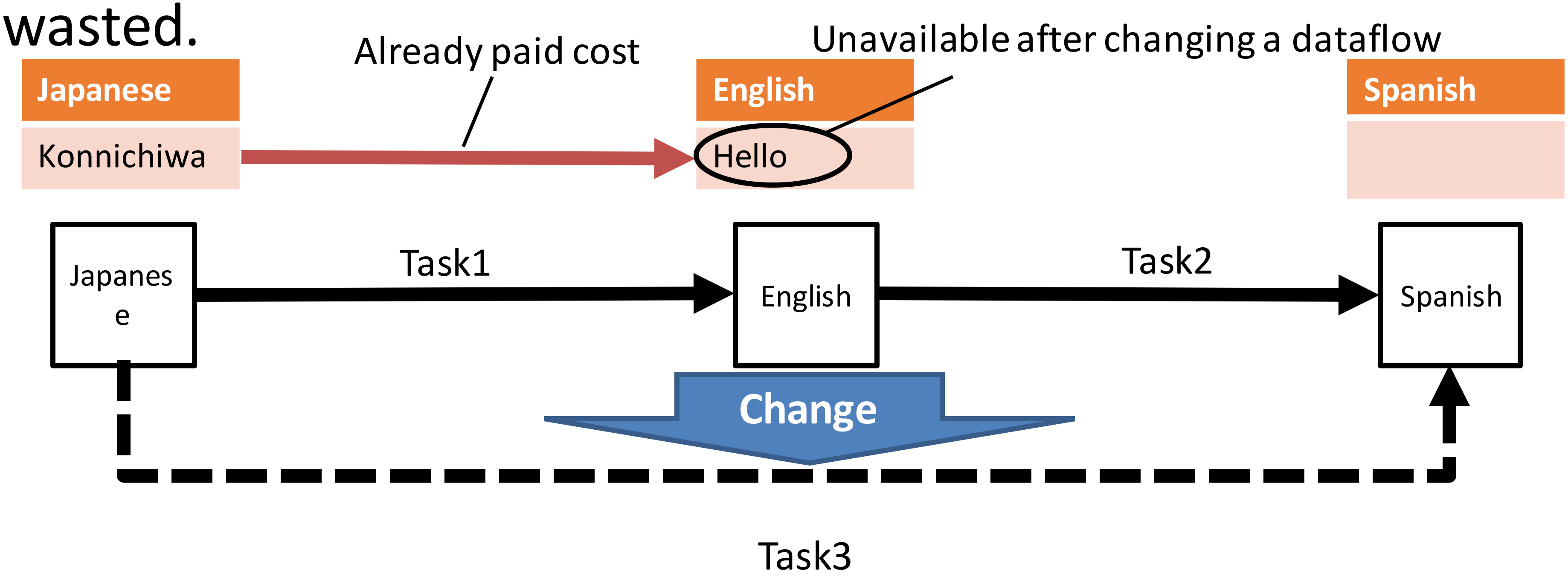
Propose a rerunning method with the task result cache.

→ It is reasonable because the cost of the computer involved in using caches is low compared to the cost of processing tasks in crowdsourcing.

[1] Ron Avnur and Joseph M Hellerstein. Eddies: Continuously adaptive query processing. In ACM sigmod record, Vol. 29, pp. 261–272. ACM, 2000.

Background(4/4) :
It takes extra money cost to redo the task

Changing the dataflow makes it impossible to use the result of the task performed halfway and the cost for that is wasted.



Already paid cost

Unavailable after changing a dataflow

Japanese	English	Spanish
Konnichiwa	Hello	

The purpose of our research

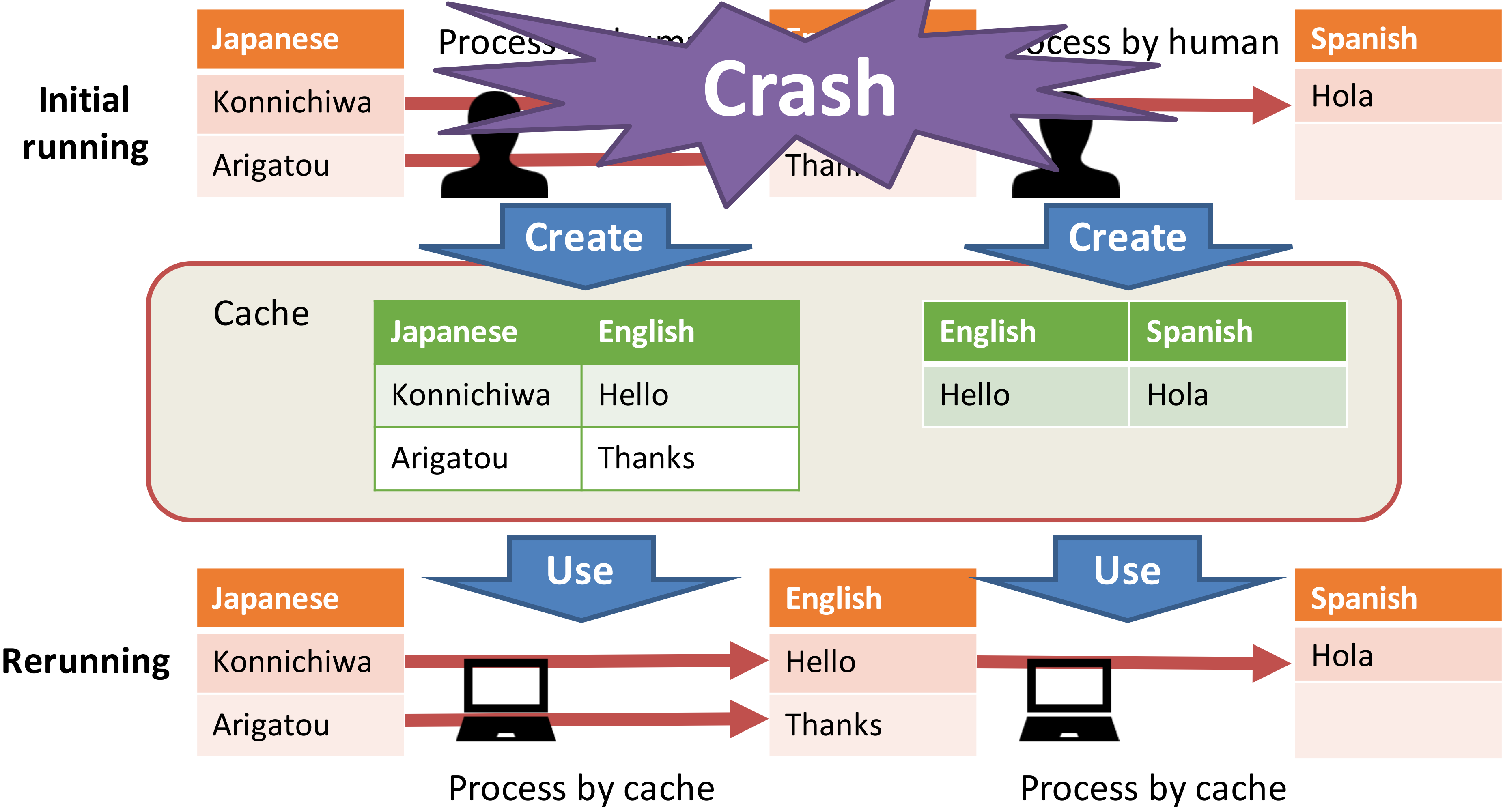
To reduce the total monetary cost in the dataflow change process

Key points of the presentation

1. 【Background】 Crowdsourcing sometimes makes
2. dataflow change halfway, but it costs a lot of money.
3. **【Related Work】 With the method using task result caches,**
4. **it cannot cope with dataflow change.**
5. 【Our approach】 We propose the method to use caches
6. coping with changing dataflows.
7. 【Simulation】 Our simulation results showed that it is
8. possible to identify the best point to minimize the total cost.

Existing Method : Rerunning method with the task result caches [2]

It can resume halfway when crashing



Japanese
Konnichiwa
Arigatou

Spanish
Hola

Japanese	English
Konnichiwa	Hello
Arigatou	Thanks

English	Spanish
Hello	Hola

Japanese
Konnichiwa
Arigatou

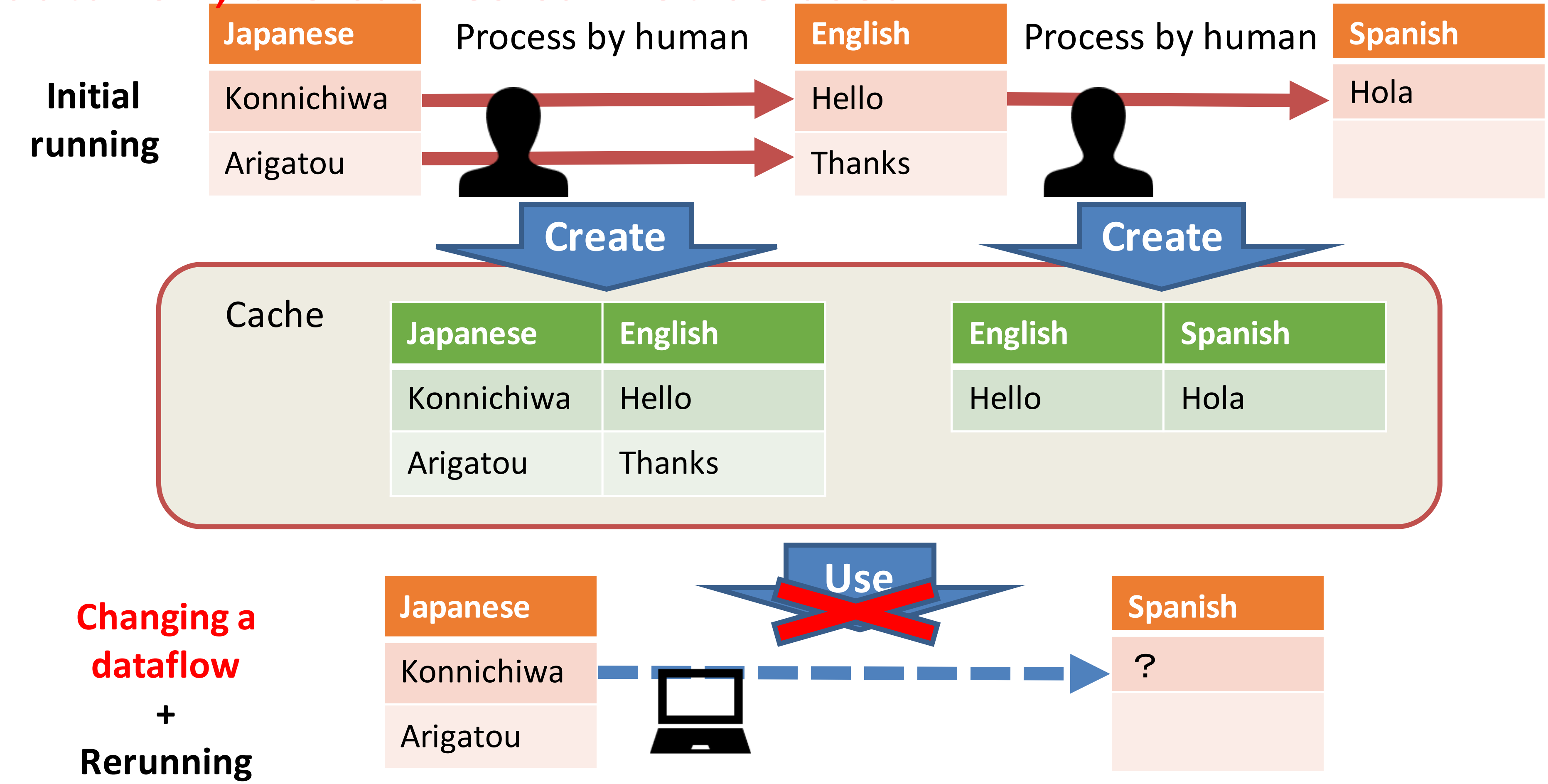
English
Hello
Thanks

Spanish
Hola

[2] G. Little et al. “Turkit: human computation algorithms on mechanical turk”. In: UIST. 2010.

Existing Method : Cannot use caches after changing the dataflow[2]

Because caches does not correspond to the task in new dataflow, the caches cannot be used.



Initial running

Japanese
Konnichiwa
Arigatou

Process by human

English
Hello
Thanks

Process by human

Spanish
Hola

Japanese	English
Konnichiwa	Hello
Arigatou	Thanks

English	Spanish
Hello	Hola

Changing a dataflow + Rerunning

Japanese
Konnichiwa
Arigatou

Spanish
?

[2] G. Little et al. “Turkit: human computation algorithms on mechanical turk”. In: UIST. 2010.

Key points of the presentation

✓ 1. 【Background】 Crowdsourcing sometimes makes
2. dataflow change halfway, but it costs a lot of money.
✓ 3. 【Related Work】 With the method using task result caches,
4. it cannot cope with dataflow change.
➡ 5. **【Our Approach】 We propose the method to use caches**
6. **coping with changing dataflows.**
7. 【Simulation】 Our simulation results showed that it is
8. possible to identify the best point to minimize the total cost.

Combining caches

Combine cache from task input and task output

Initial running

Cache

Japanese	English
Konnichiwa	Hello
Arigatou	Thanks

English	Spanish
Hello	Hola

Combine

Cache

Japanese	Spanish
Konnichiwa	Hola

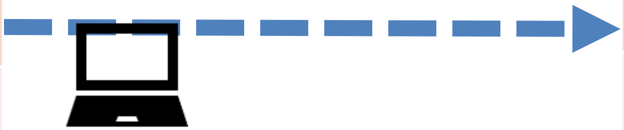
Use

Changing a dataflow

+

Rerunning

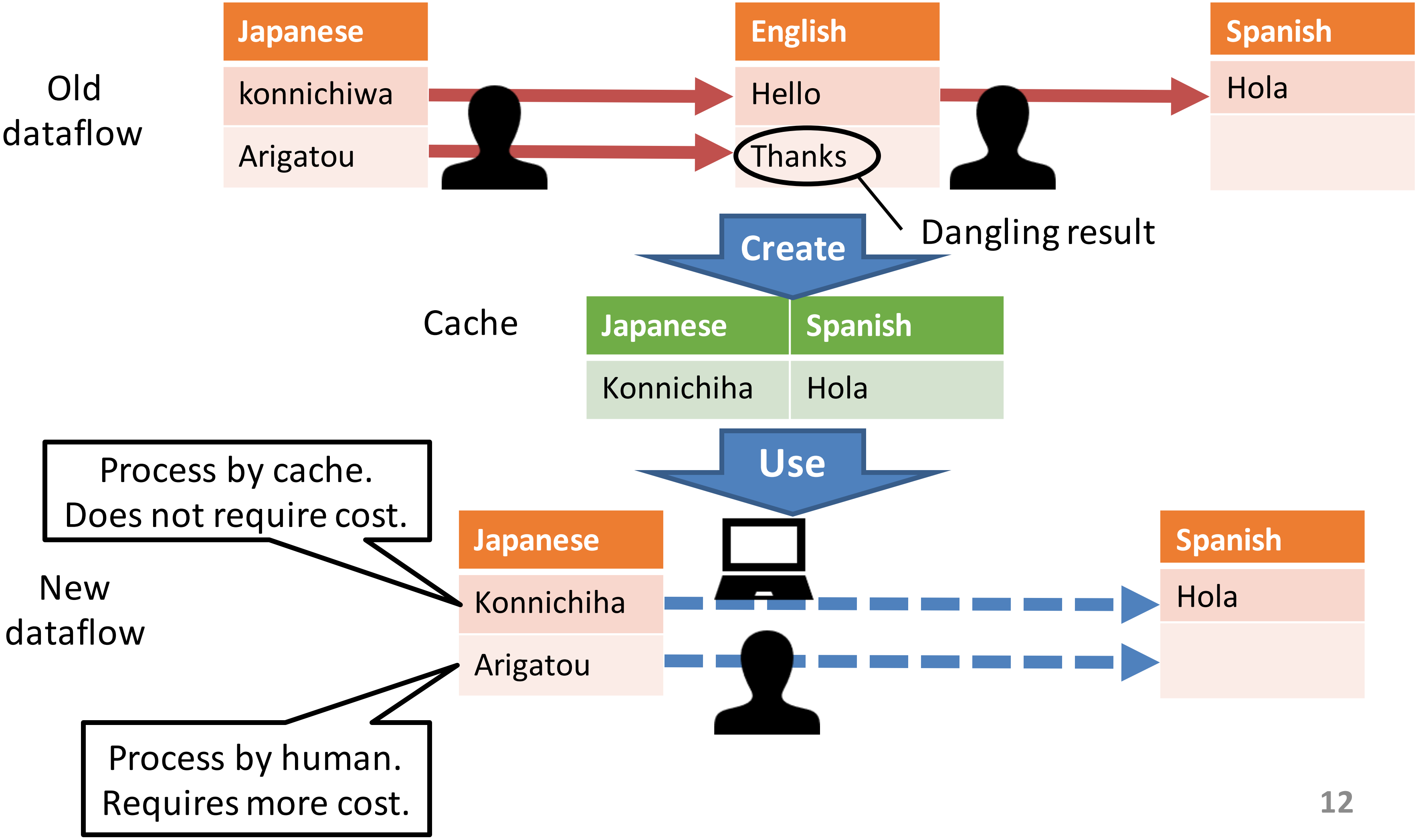
Japanese
Konnichiwa
Arigatou



Spanish
Hola

Processing data without corresponding cache

In the old dataflow,
only data with no corresponding cache will be processed.



Japanese
konnichiwa
Arigatou

English
Hello
Thanks

Spanish
Hola

Japanese	Spanish
Konnichiha	Hola

Japanese
Konnichiha
Arigatou

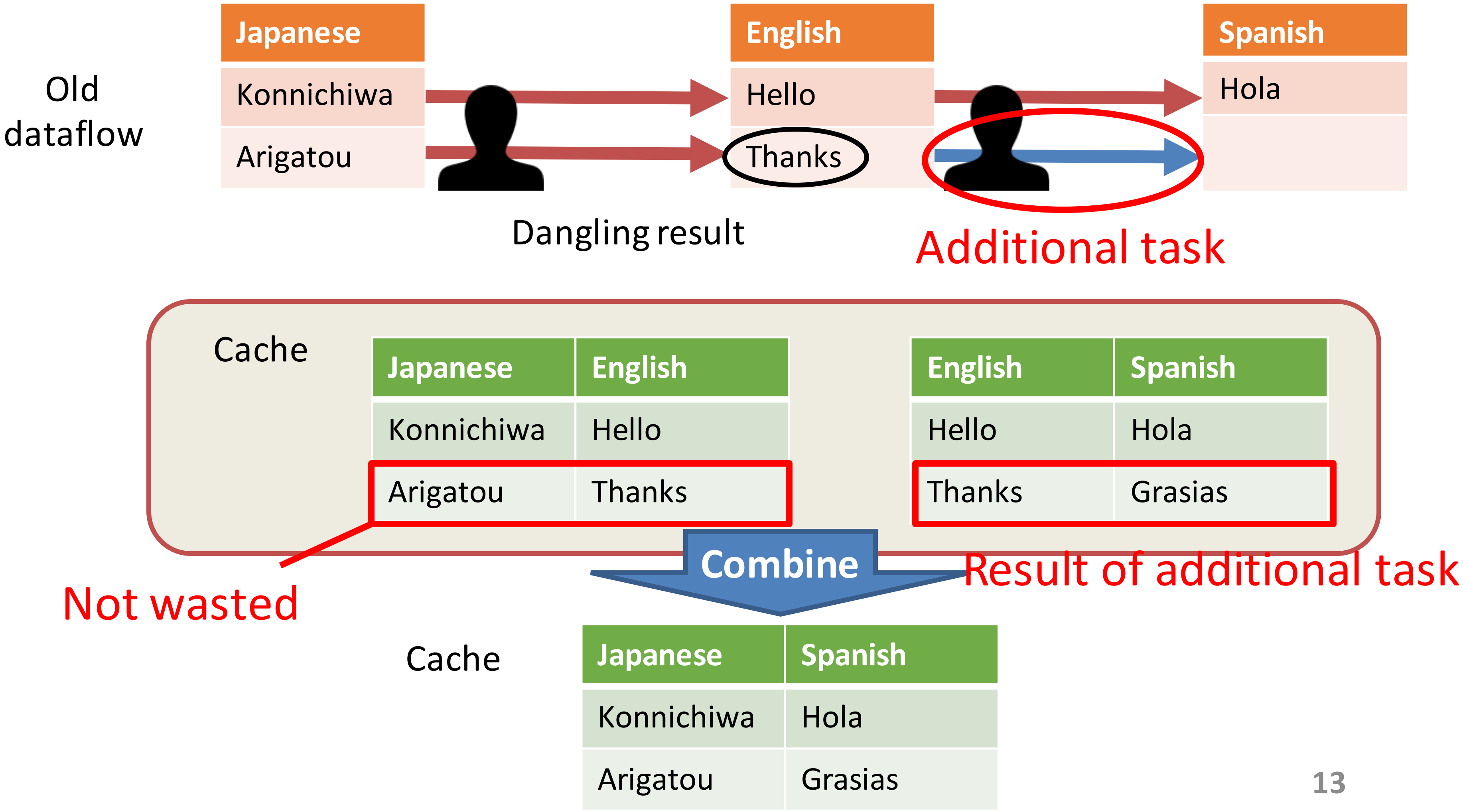
Spanish
Hola

Process by cache.
Does not require cost.

Process by human.
Requires more cost.

Process the dangling result in the old dataflow

Reduce wasted task results by executing additional tasks
on dangling results.



Old dataflow

Japanese
Konnichiwa
Arigatou

English
Hello
Thanks

Spanish
Hola

Dangling result

Additional task

Cache

Japanese	English
Konnichiwa	Hello
Arigatou	Thanks

English	Spanish
Hello	Hola
Thanks	Grasias

Not wasted

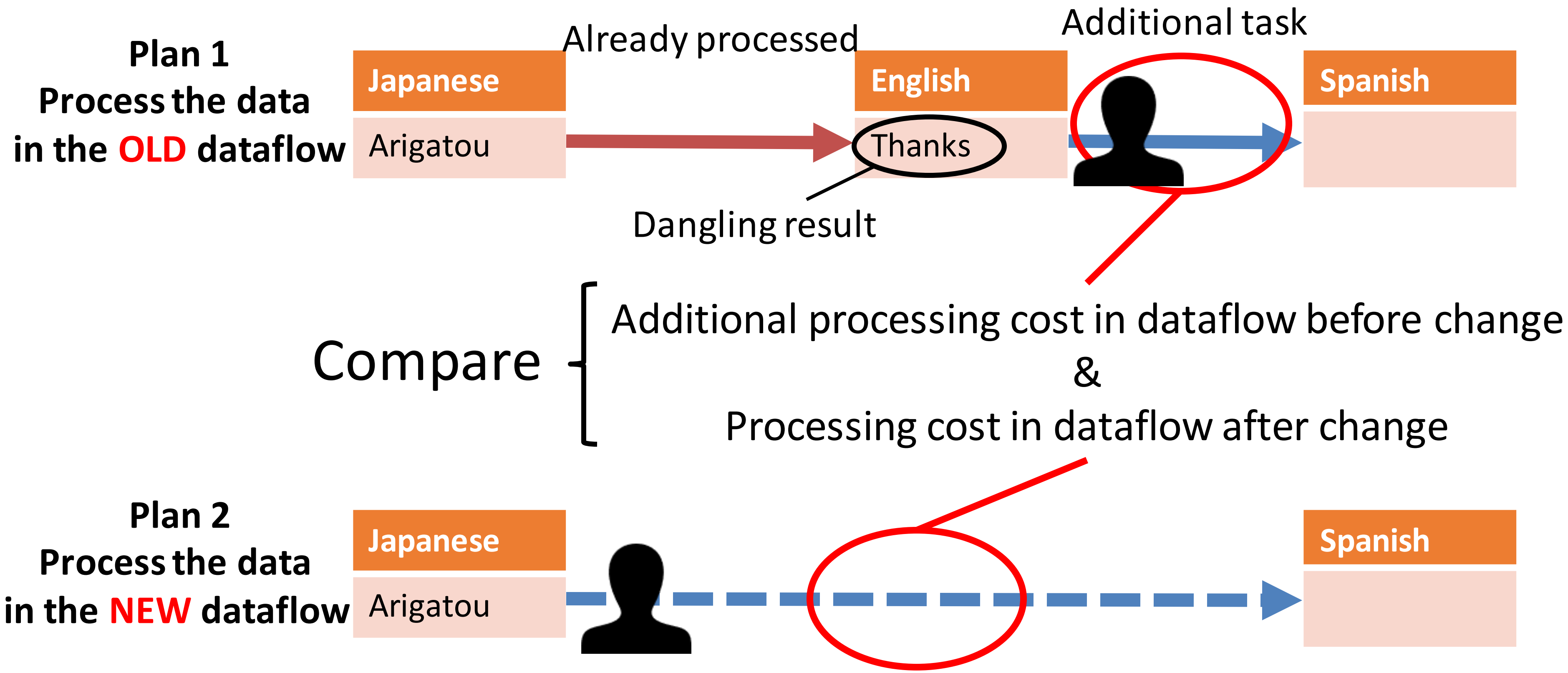
Result of additional task

Cache

Japanese	Spanish
Konnichiwa	Hola
Arigatou	Grasias

Relationship between dangling results and costs

It is not enough to simply process dangling results to the end



Plan 1
Process the data
in the OLD dataflow

Japanese
Arigatou

Already processed

English
Thanks

Dangling result

Additional task

Spanish

Compare

Additional processing cost in dataflow before change
&
Processing cost in dataflow after change

Plan 2
Process the data
in the NEW dataflow

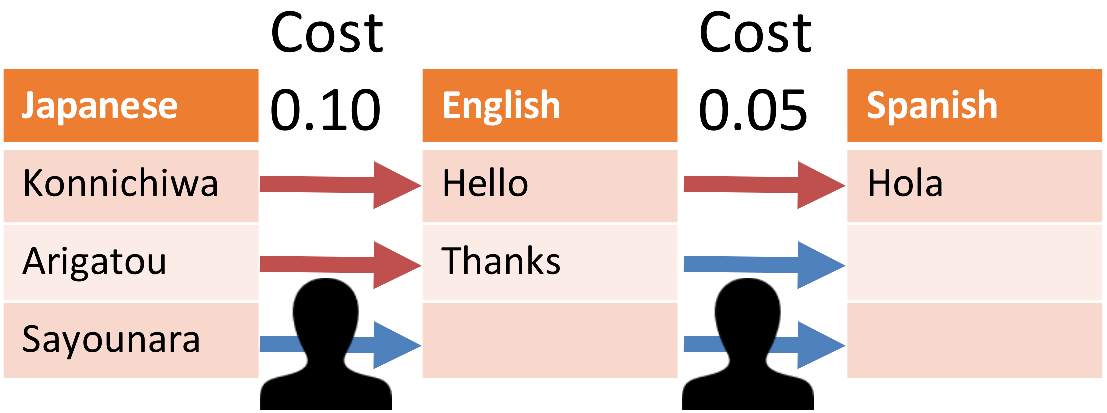
Japanese
Arigatou

Spanish

Compare the costs and process with the cheaper dataflow

Cost estimation for each dataflow

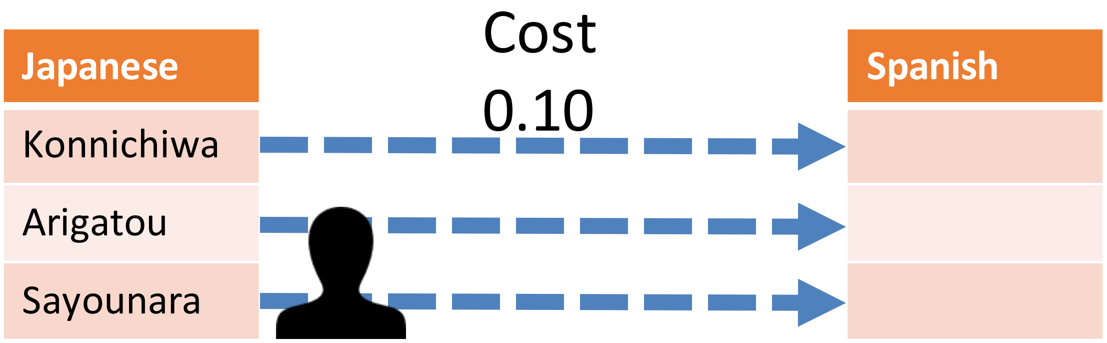
Monetary cost for the old dataflow



Japanese	English	Spanish
Konnichiwa	Hello	Hola
Arigatou	Thanks	
Sayounara		

Input	Monetary cost for the old dataflow
Konnichiwa	**0.00**
Arigatou	**0.05**
Sayounara	0.15

Monetary cost for the new dataflow



Japanese	Spanish
Konnichiwa	
Arigatou	
Sayounara	

Input	Monetary cost for the new dataflow
Konnichiwa	0.10
Arigatou	0.10
Sayounara	**0.10**

Key points of the presentation

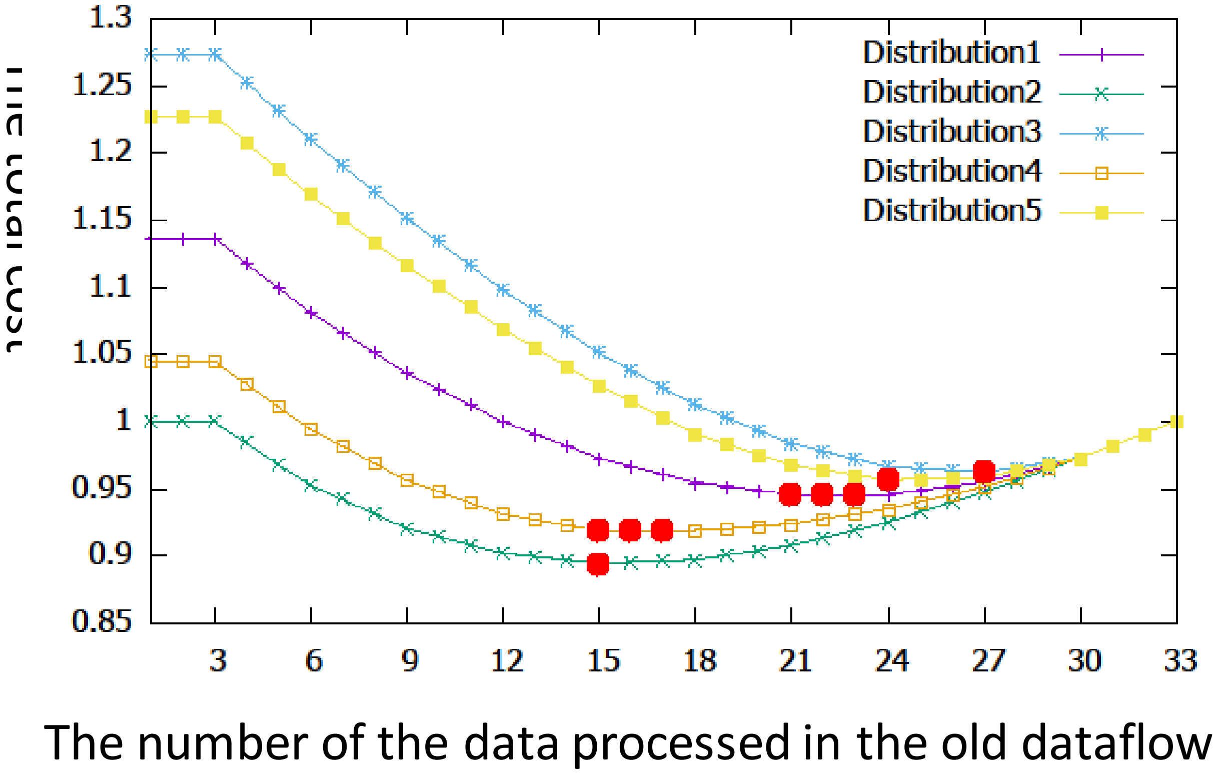
1. ✓ 【Background】 Crowdsourcing sometimes makes
2. dataflow change halfway, but it costs a lot of money.
3. ✓ 【Related Work】 With the method using task result caches,
4. it cannot cope with dataflow change.
5. ✓ 【Our Approach】 We propose the method to use caches
6. coping with changing dataflows.
7. **【Simulation】 Our simulation results showed that it is**
8. ➡ **possible to identify the best point to minimize the total cost.**

Simulation purpose and Results

Purpose: To show relationship between data processing cost and cost before change

Results:

- it is possible to identify the best point to minimize the total cost.
- There are no obvious solutions.



Summary

- We proposed the method to use caches corresponding to changing dataflows.
- Our simulation showed that it is possible to identify the best point to minimize the total cost.
- I explored the dataflow including join operation. (poster)

Future Work

- Since cost estimation is costly for calculation,
- we will consider a method to reduce calculation cost.