Corpus-based instruction interpretation in **virtual environments**

Luciana Benotti, Martín Villalba,
Tessa Lau & Julián Cerruti

IBM Argentina
Silvergate Group

Nat. University of Córdoba
Famaf

Universität Potsdam
Linguistics Department

Instruction interpretation?

Aren't all (non-trivial, computational) systems supposed to do that already?

Indeed! When we say

Instruction interpretation

we'll use it as a placeholder for

Interpretation of **task-oriented** instructions written in **natural language** by users and without any kind of special syntax

What is a **virtual environment**?

Environment

The surroundings which determine,
specify or clarify the meaning of an event

(according to Wiktionary)

Formally, an environment is defined by:

Its **actions** (no actions = no events)

Actions implies **reactions** to them.
This makes our environment interactive

Actions and reactions occur within a **circumstance,**
a combination of where and at which time an action took place

Virtual Environment

The *virtual* surroundings which determine, specify or clarify the meaning of a *virtual* event

Formally, a *virtual* environment is defined by:

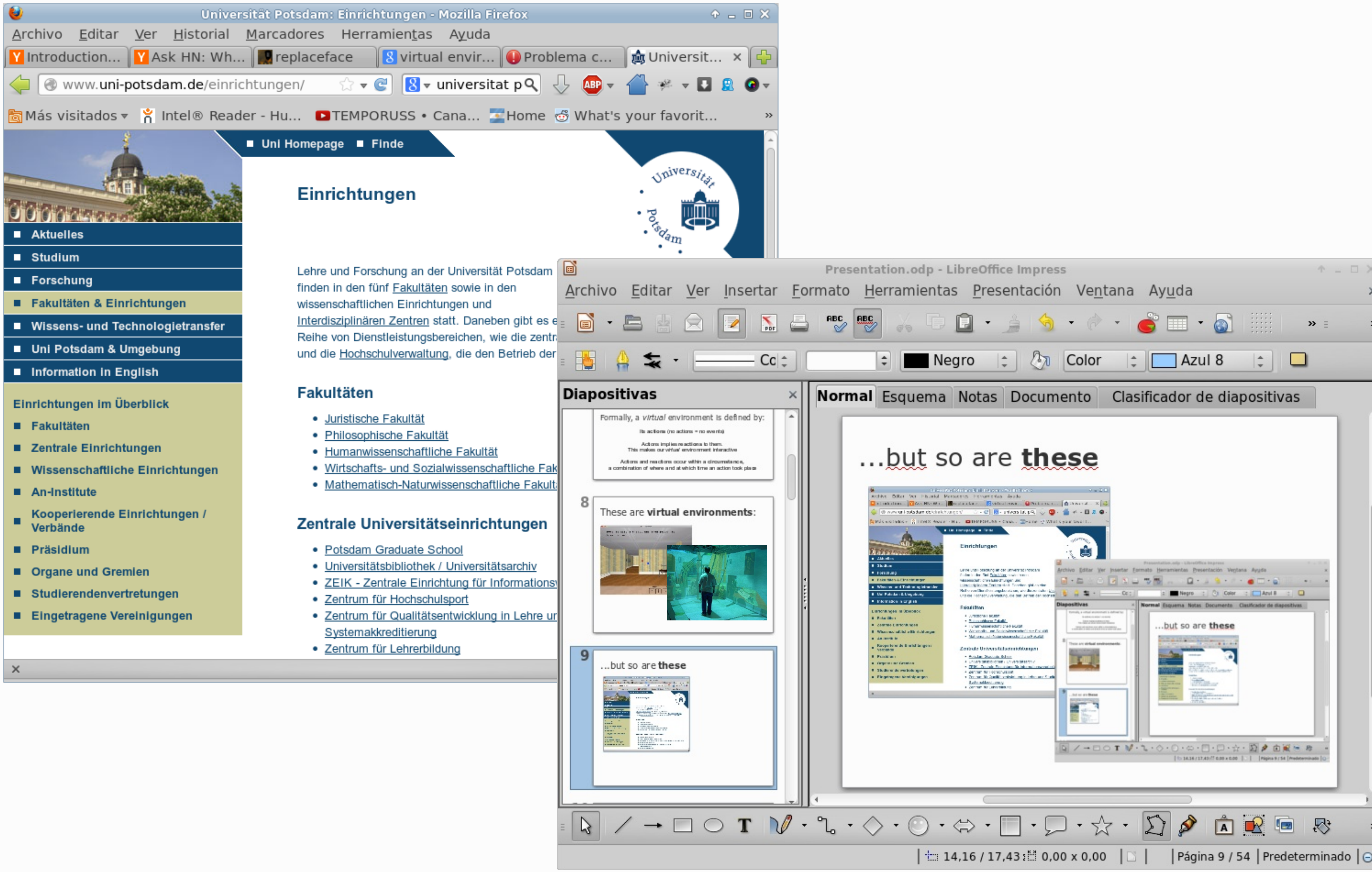
Its **actions** (no actions = no events)

Actions implies **reactions** to them.
This makes our *virtual* environment interactive

Actions and reactions occur within a **circumstance,**
a combination of where and at which time an action took place

These are **virtual environments**:

...but so are **these**



Word of the day:

Affordability

An action that can take place in the current environment given the current circumstances

So you'll be interpreting instructions* inside a virtual environment.

But why is it **corpus-based**?

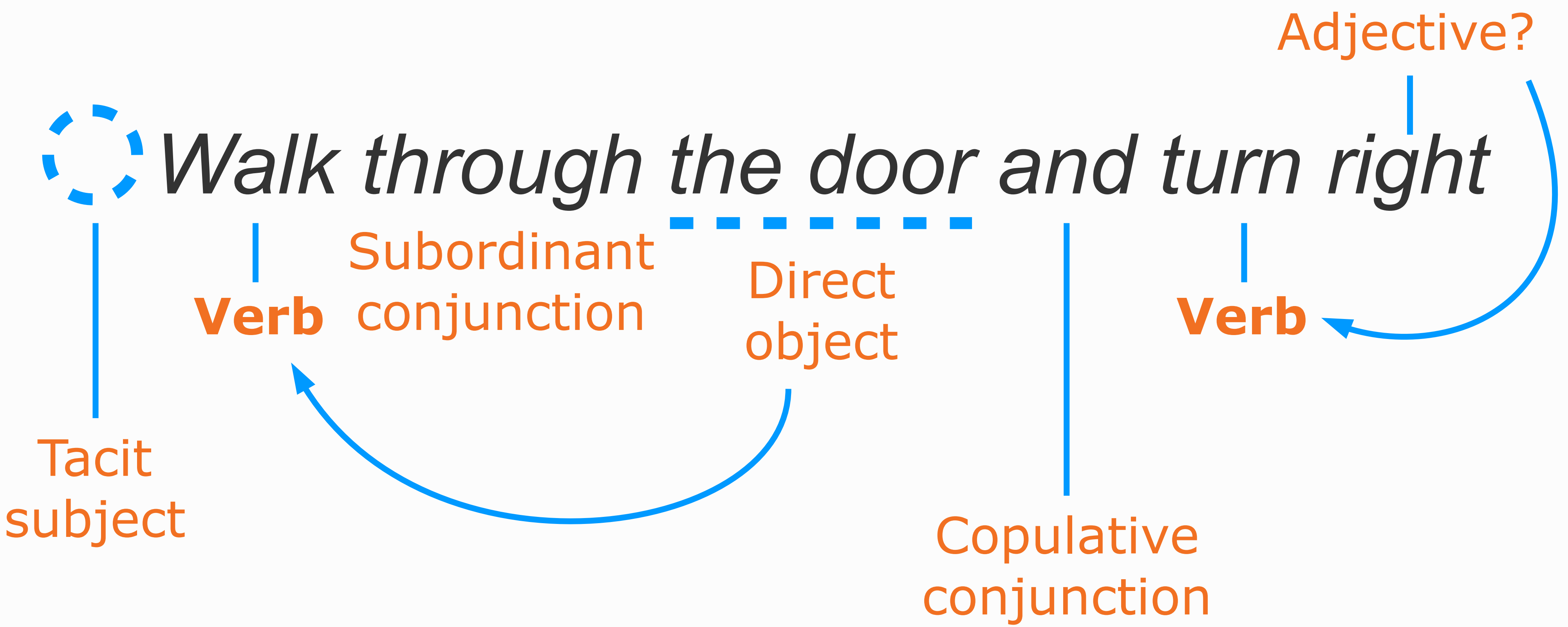
* interpreting task-oriented instructions written in natural language by users and without any kind of special syntax

Let's talk about the most **common approaches** to instruction interpretation

Symbolic approaches

Walk through the door and turn right

Symbolic approaches



Symbolic approaches

Stop before reaching the button next to the lamp, and push it

Give it to him

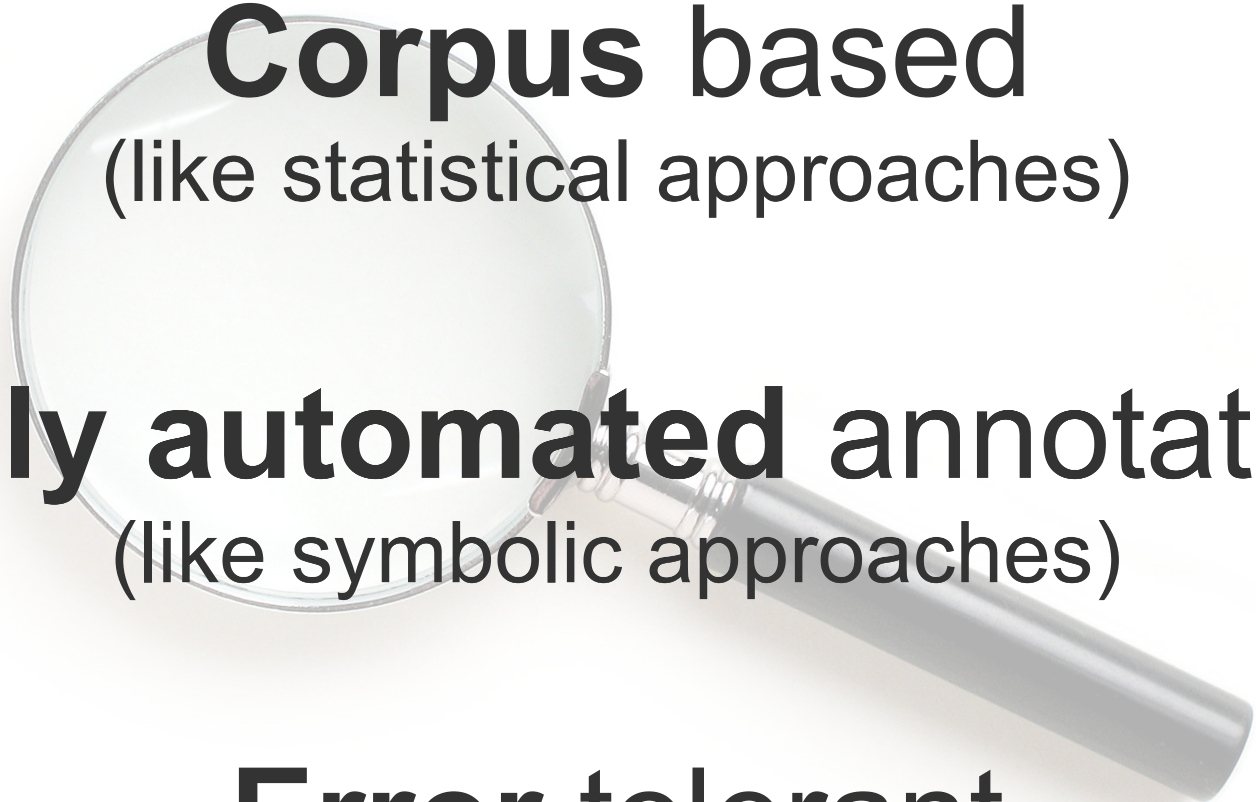
Shut up and answer me

Grab the mouse

Statistical approaches

They require **extensive manual annotation** of a **corpus** during the training phase

Our approach



Corpus based

(like statistical approaches)

Fully automated annotation

(like symbolic approaches)

Error tolerant

(orthography, grammar, etc)

Schedule for this talk:

Three parts

1) Foundations
What is our work based on?

2) Approach
What we did, and how we did it?

3) Results
Did it work?


Part 1
Foundations

What is the give challenge ?


GIVE 3D Client
Move forward 2 steps, and then press the blue button.
give
challenge
Walk
Turn
Esc
quit
H
What?

Corpus statistics

6 virtual worlds
100 games
5580 instructions
14:26hs. of interaction

Divided in two major branches

In the **Cs Corpus**, instructions were always given by the same person

Pros: consistent instructions and language
Cons: familiarity with the task may skew results, favoring efficiency over naturalness

---

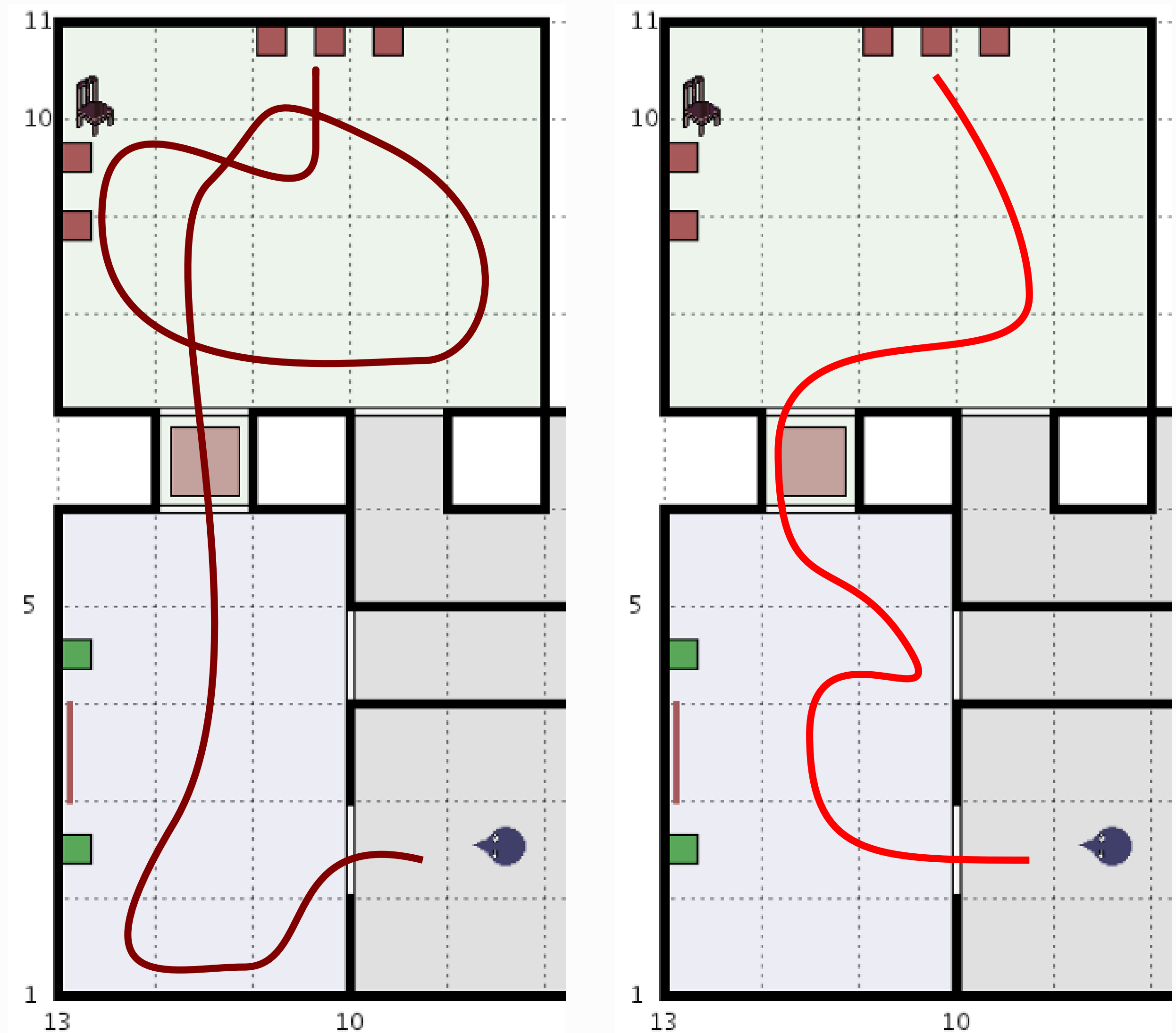
In the **Cm Corpus**, both roles were selected randomly

Pros: instructions are truly spontaneous, easier to collect
Cons: too much variability in the training set, instructions are inconsistent, swearing

```
perss the red - r - no into shade room the paise
stop spinning - door on your left - sorry
Through the opening to the right of the lamp
```

Question:

How do we deal with **continuous*** data?



* not really continuous but close enough

Answer:

We don't.
We discretize them
using a **planner**

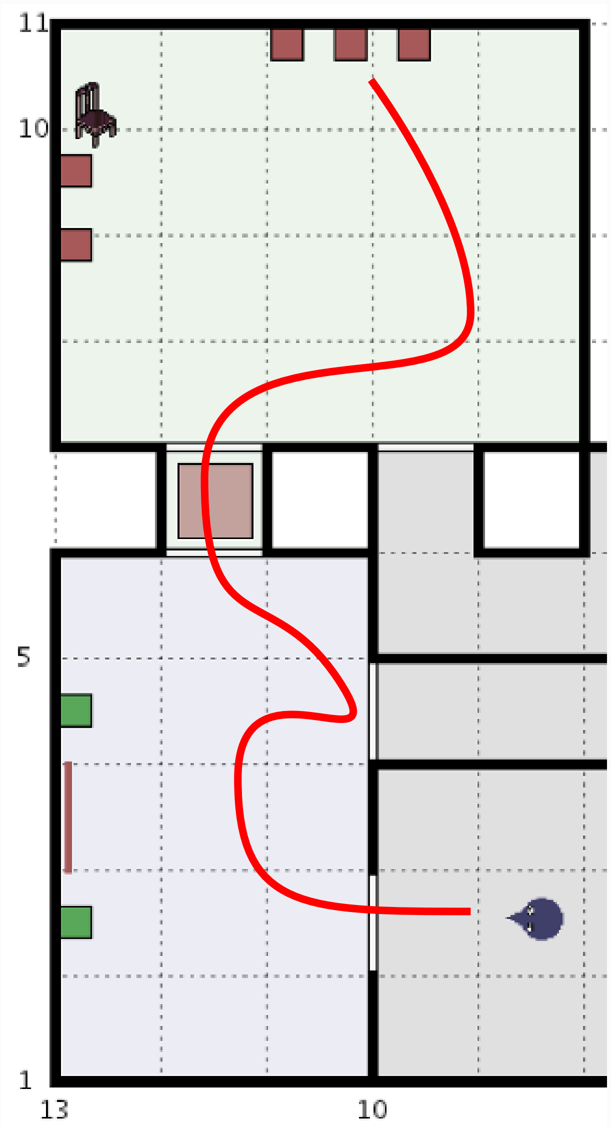
An **automated planner** is a system that, given

an initial state, and
a set of tuples
`(state x action->new state)`

returns a sequence of actions that achieve a certain target state.

This sequence of actions is called the **solution** to a planning problem

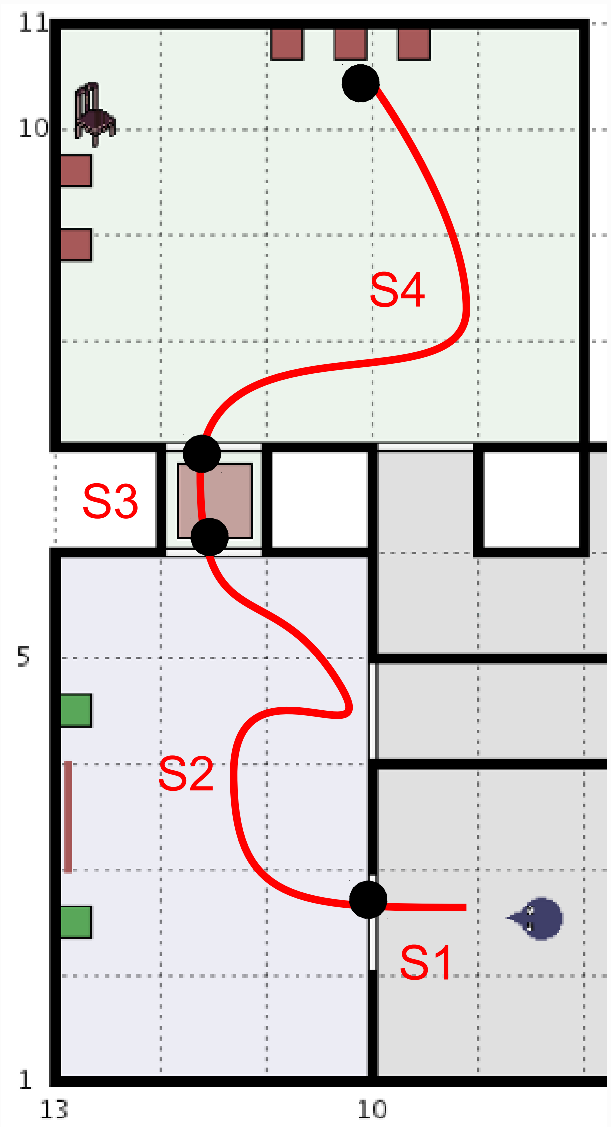
Discretization



Step 1: Define a set of conditions, operators and states common for both the environment and the planner

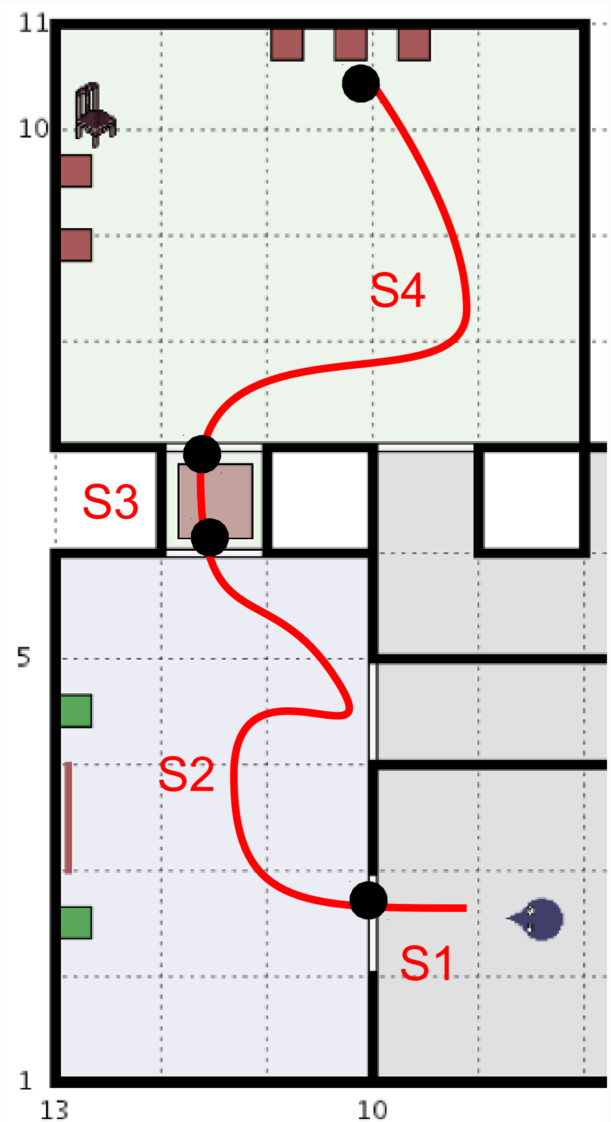
```
[ button b1...b7,
  door d1..d5 ]
[ walk_through(door d)
  push_button(button b) ]
[ initial_pos = p0 ]
```

Discretization



Step 2: Segment the user data, using planner events as boundaries

Discretization

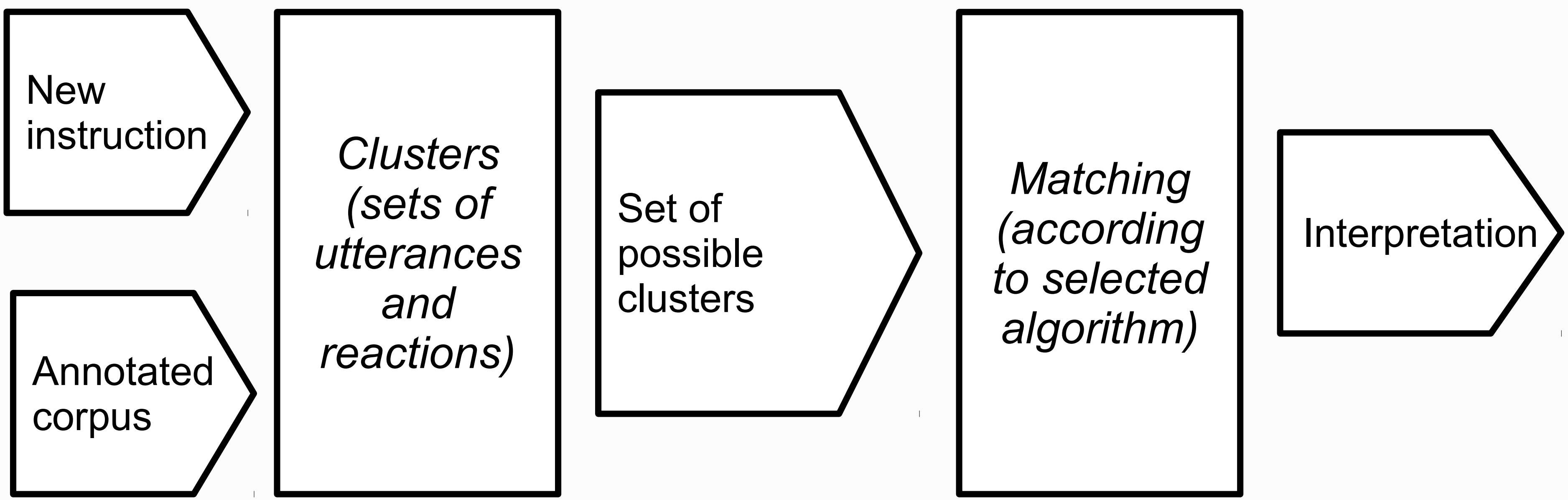


Step 3: create a plan for each segment. Together, they form the discrete version of this interaction

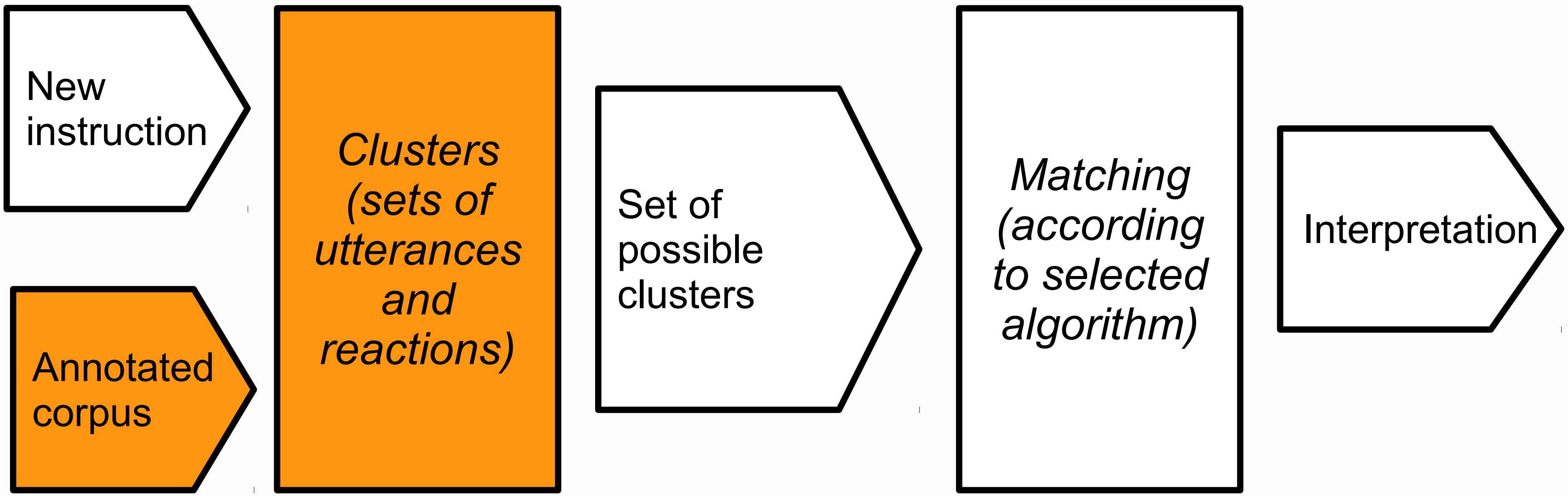
```
Plan S1:
  walk_through(door d1)
Plan S2:
  walk_through(door d3)
Plan S3:
  walk_through(door d4)
Plan S4:
  push_button(button b2)
```


Part 2
Searching for solutions

Our **architecture**



Our **architecture**



We'll pick the correct interpretation from our set of clusters

What is a **correct interpretation**?

Automated annotation

Press the red button in the furthest room
[enters the room]
[presses the red button]
[turns right]
Nowgo right, green next to lamp in next room
[enters to the room to the right]
[presses the green button]

Behavior-based annotation

Press the red button in the furthest room

[enters the room]
[presses the red button]
[turns right]

Nowgo right, green next to lamp in next room

[enters to the room to the right]
[presses the green button]

Visibility-based annotation

Press the red button in the furthest room
[enters the room]
[presses the red button]
[turns right]

Nowgo right, green next to lamp in next room
[enters to the room to the right]
[presses the green button]

Clearly, the method is not **perfect**

Press the green button
[steps next to the wrong button]

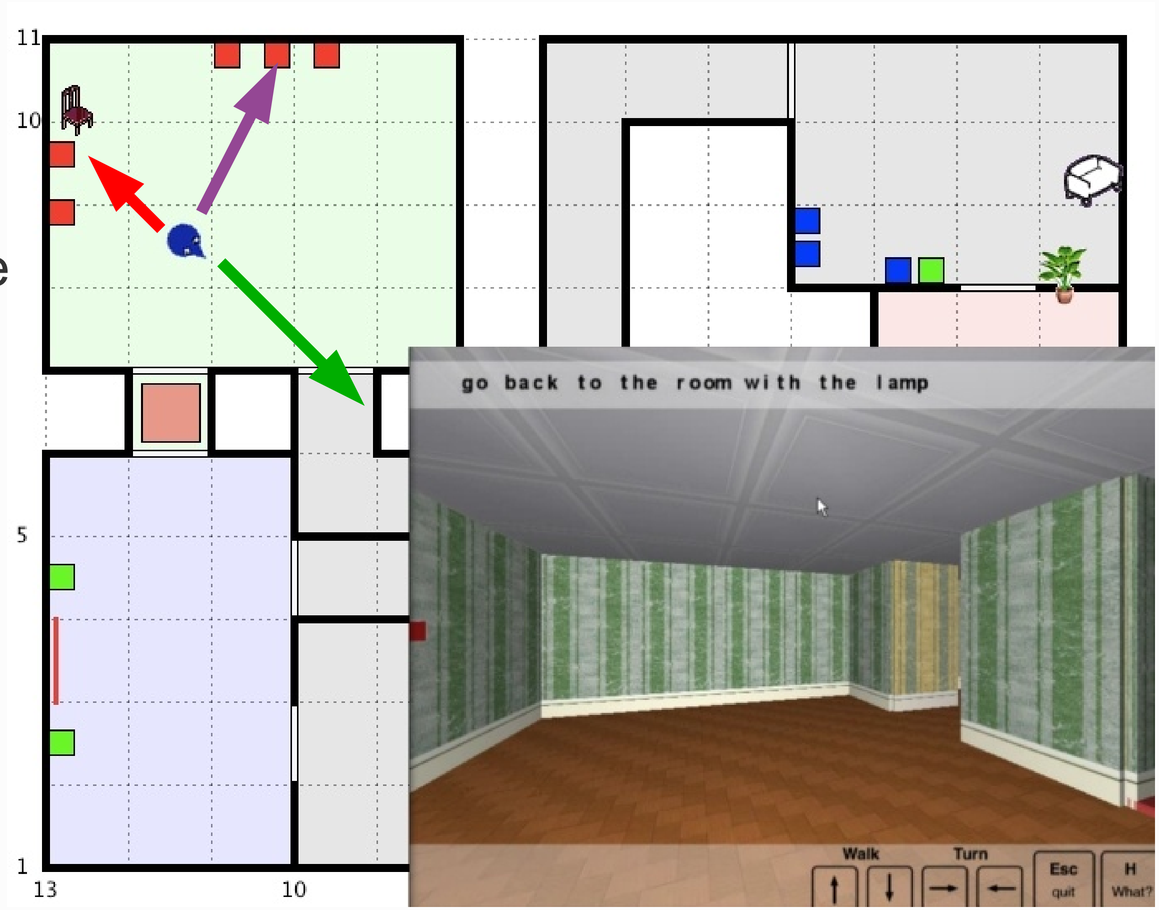
No
[moves towards the correct one]
[presses the right one]

Now that we have **canonical reactions**, we can classify instructions:

If two instructions have **the same reaction** in **the same location**, they belong to the same **group** or **cluster**

Graphically, it looks like **this**

- Red left of chair
  Press the one behind u
- Press middle button
  In group of 3, press middle one
  Counting right, 2nd one
- To the room with lamp
  Go back to the big room
  Go back to the hallway
  Go back out the room
  Down the passage
  Go back out
  Yes
  Out the way you came in
  Exit the way you entered
  Go through the opening on the left with the yellow wall paper




What if I get a **shorter** response? Should that count as a **valid** reaction?
Alice
(c) Briberry @ FLickr

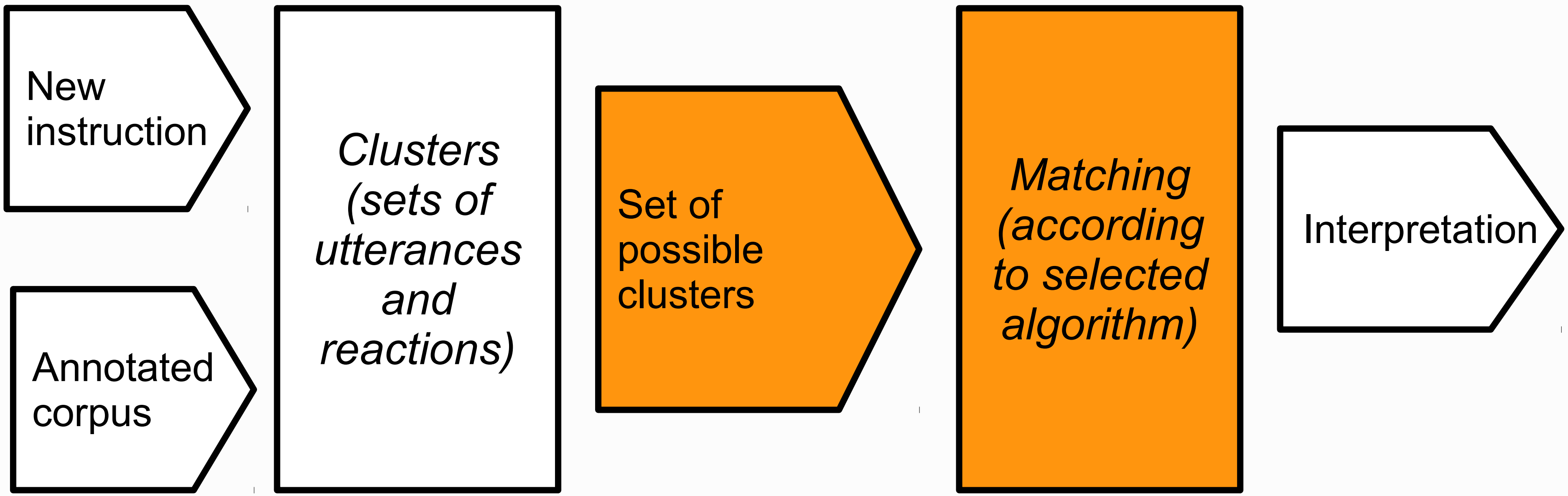
If we are **goal oriented**, then a shorter reaction shouldn't count as right

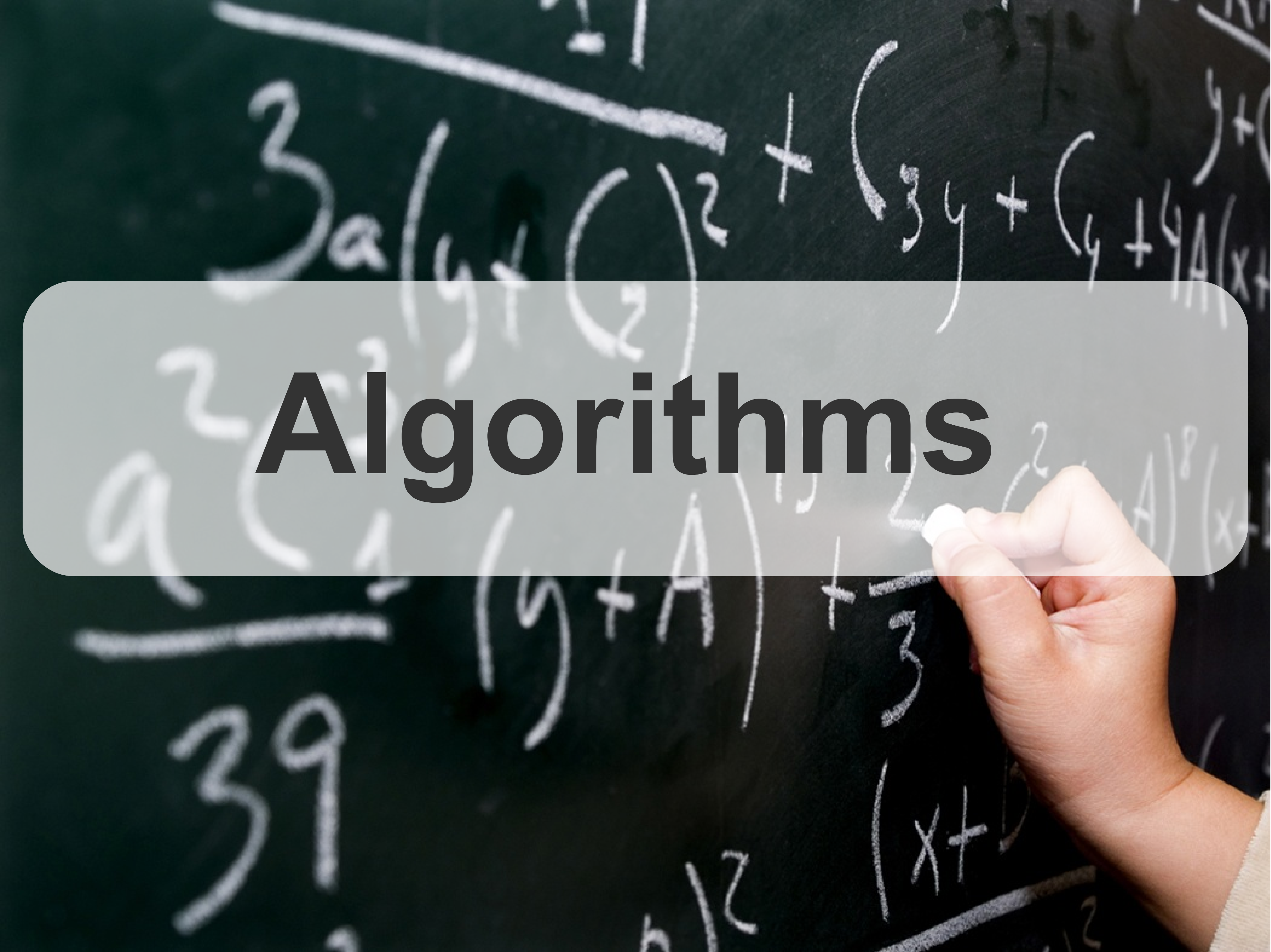
Instruction:
1. Go to your room
2. Clean your desk
3. Make your homework

Reaction:
Goes to room

However, if we are **process oriented**, then the player is now closer to the goal, and we accept it

Our **architecture** (part 2)




Algorithms

Similarity indexes

Jaccard

$$\frac{|A \cap B|}{|A \cup B|}$$

Overlap

$$\frac{|A \cap B|}{min(|A|,|B|)}$$

Distance indexes

Levenshtein (over letters and words)

Example: Country → City: 4

(Country → Ciuntry → Cintry → Citry → City)

Other strategies

BLEU

Evaluation algorithm for machine-translated text quality

Idea: we'll assume every cluster is a set of possible translations of the utterance, and choose the one with the highest average score

Other strategies

Support Vector Machines

Linear binary classification

Features: unigram count and player location.
A new classifier is trained for each evaluation
(keeps the feature number low).

Other strategies

Majority

Pick the largest available cluster

Random

Pick one at random

Note on random: this is not the *true* random – the baseline is *very* tricky to calculate


What if we could
correct mistakes?

Retries

If the prediction is **incorrect**, we can also allow the system to try again

(no more than 3 times)


Part 3
What were the results?

General results

70% accuracy for the best performing algorithm
(and no error corrections)

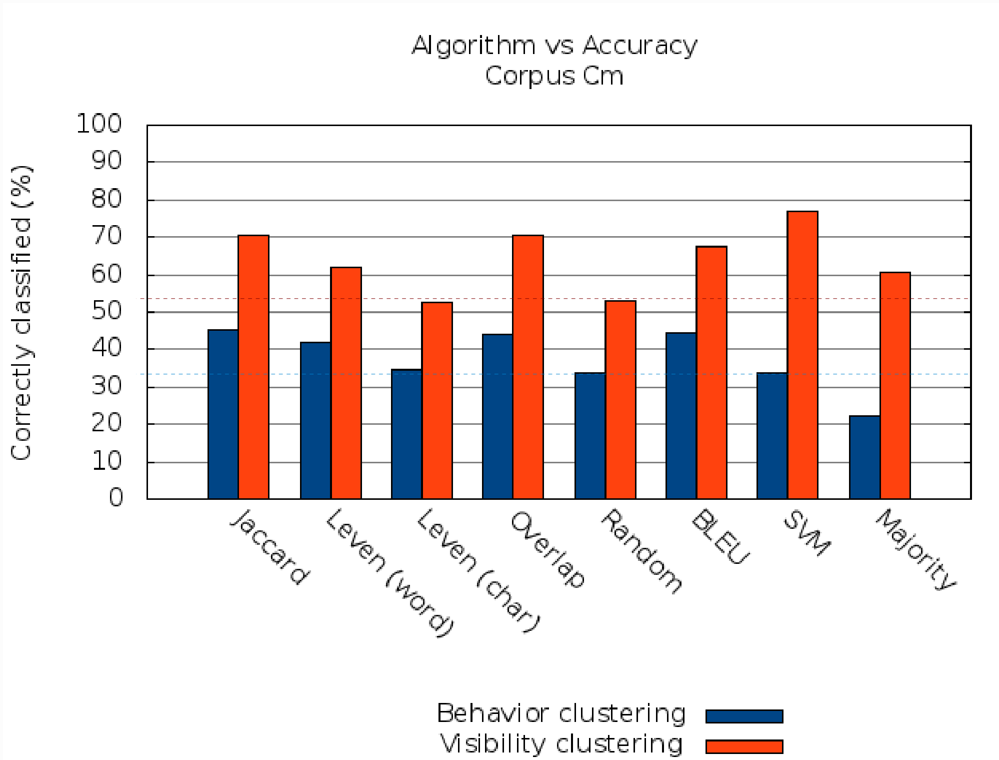
Chen & Mooney (2011) achieved 56%
Gorniak (2007) achieved 72%
using manual annotation

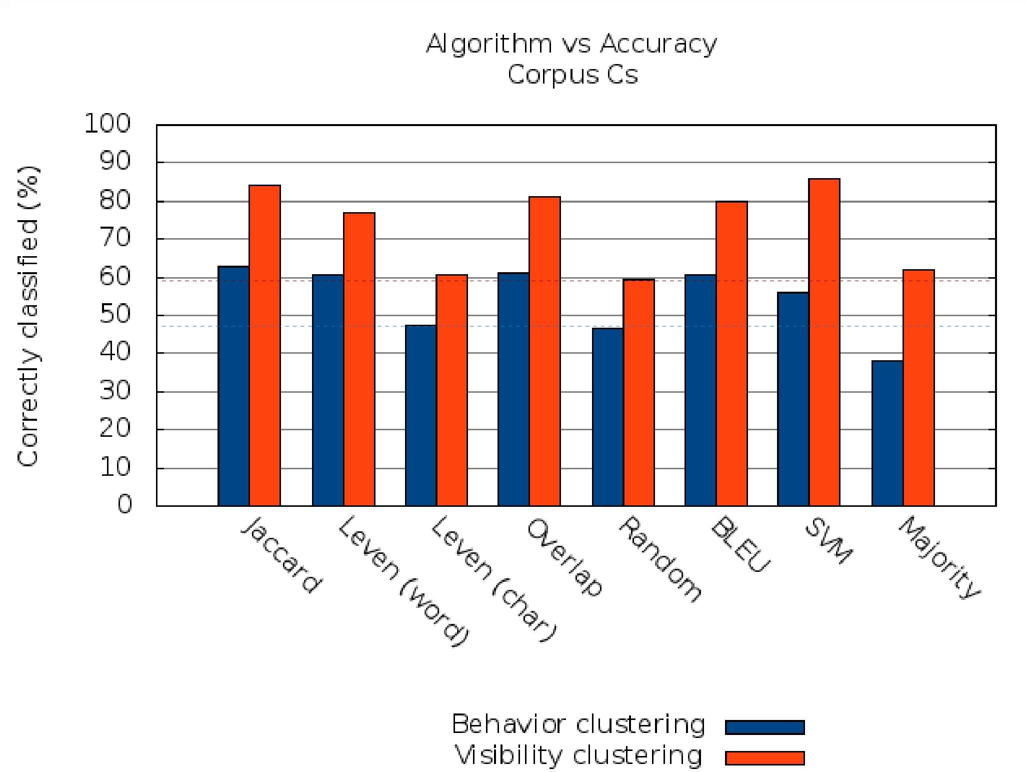
Human accuracy is closer to 69%
(according to Chen & Mooney)

Question:

¿Which one was the **best** algorithm?

General results





Similarity indexes (**Jaccard** and **Overlap**) achieved the best results at first

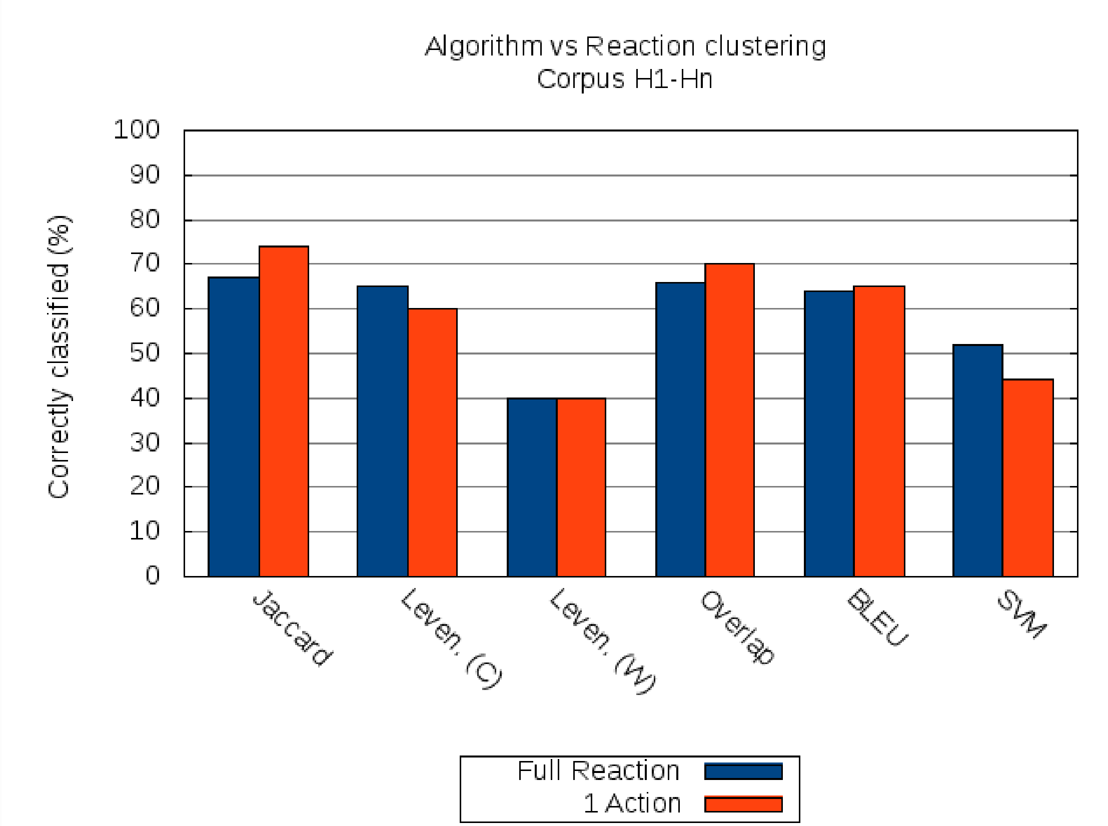
BLEU and **Levenshtein** *(by words)* achieved a decent performance

SVM had to be carefully tuned, and it eventually outperformed the rest (in Vis clustering)

Levenshtein *(by letters)* is not better than random

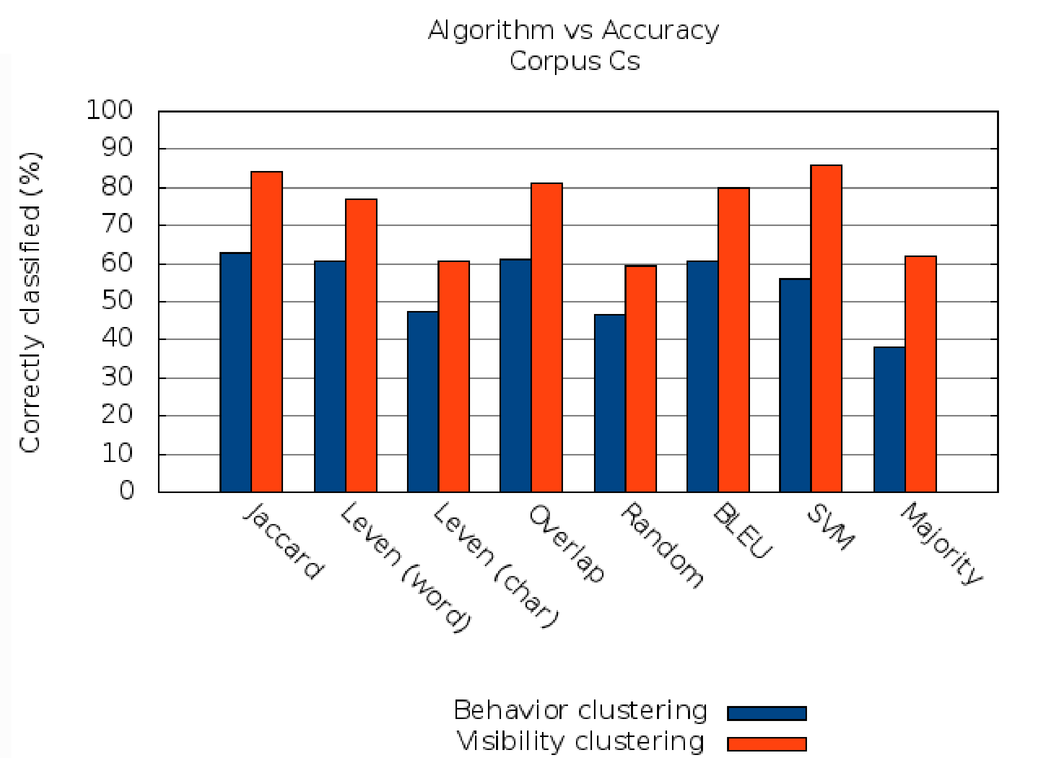
Question:

Which classification strategy is better, **visibility** or **behavior**?



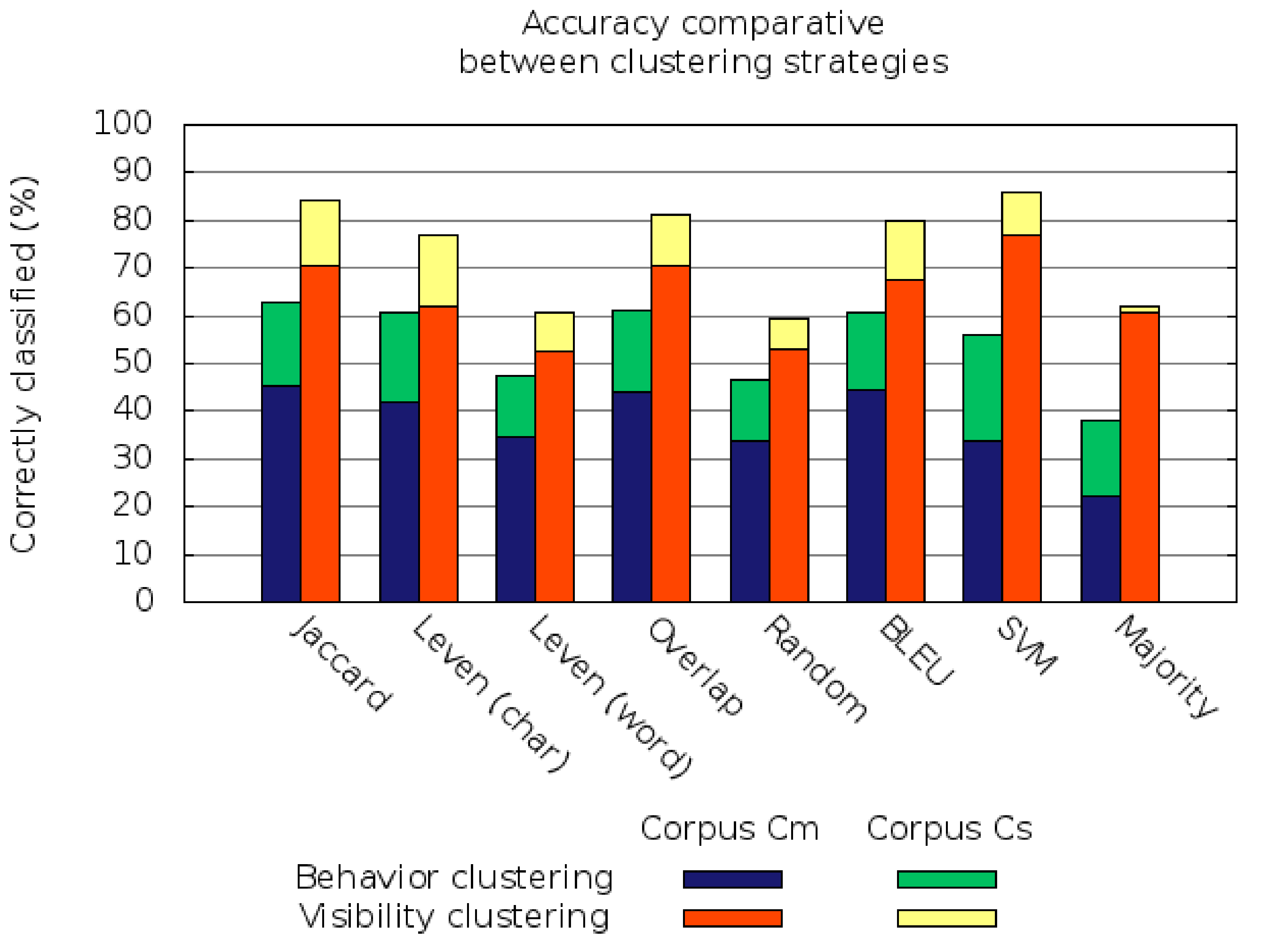
Answer: if we accept shorter results as correct, there is no (statistically significant) difference

However, if we only accept a perfect matching, then **Visibility** is better



Question:

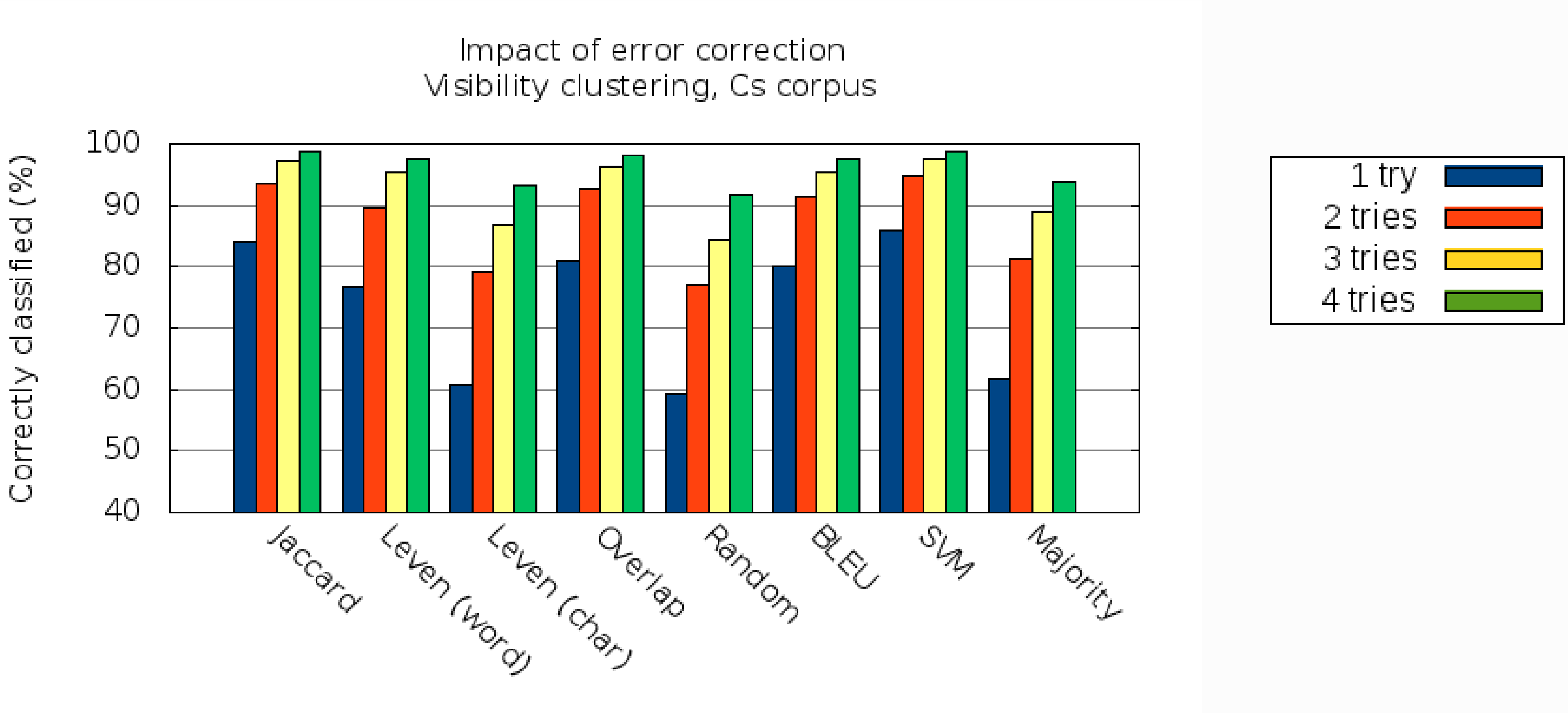
Is it relevant whether the instructions came from the **same person** or not?



Answer: the **Cs** data got better results in **all** tests

Question:

How much better are the results when we allow **mistake corrections**?

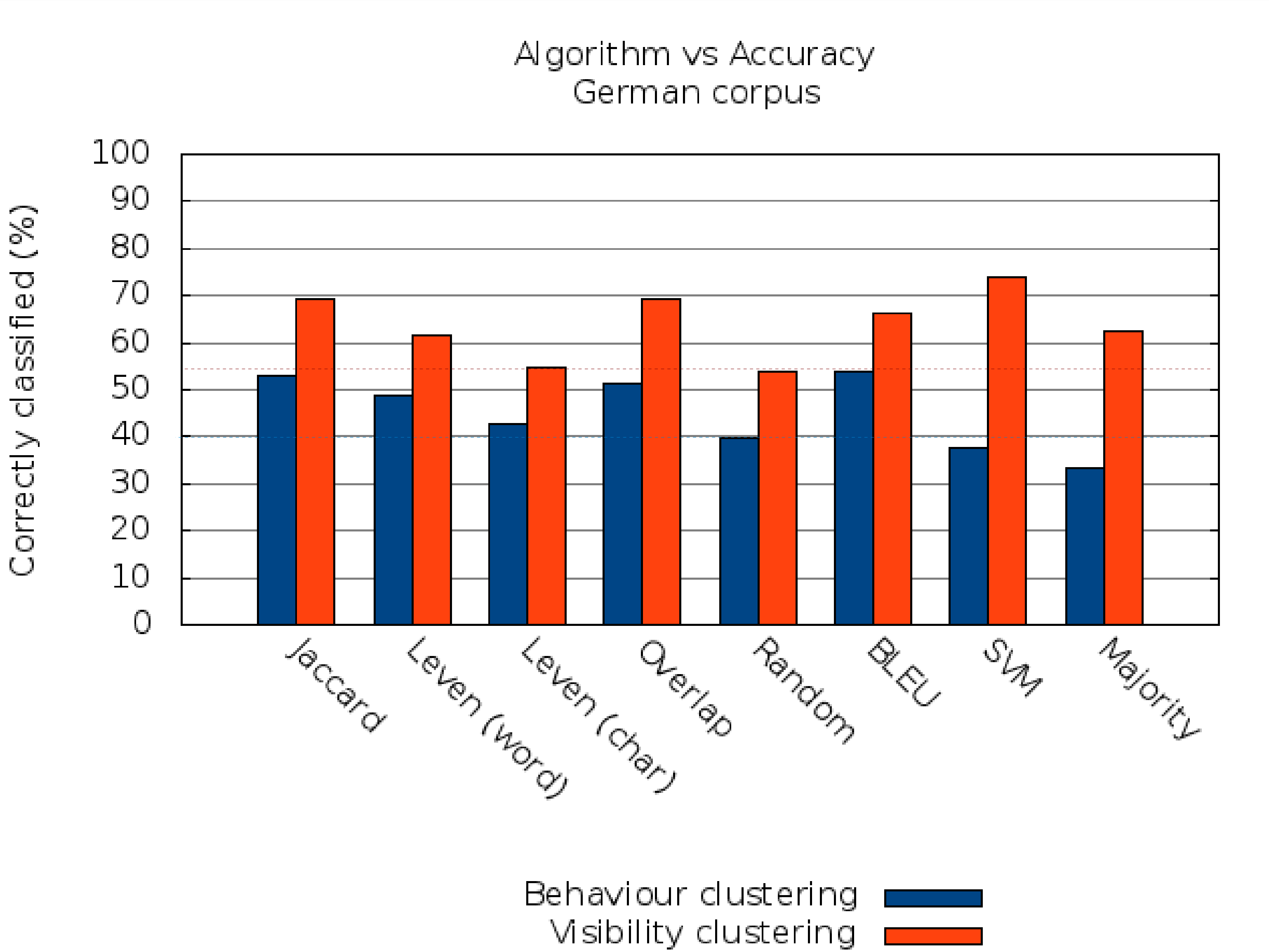


Answer: A **good** algorithm rarely needs more than two retries.

A **bad** algorithm improves linearly.

Question:

Are this results the same in a different **language**?



Answer: predictably, results appear to be consistent

In short:

Three ideas

Foundations

- Corpus data from the Give Challenge
- Continuous actions are discretized using a planner

Approach

- Utterances are clustered according to their (discrete) reaction and Bhv/Vis approach
- For a new utterance, we chose the correct affordability through cluster similarity
- We also test the impact of corrections on accuracy

Results

- 70% accuracy without retrying, 92% with one retry
- Best strategy: Single-speaker corpus, Visibility clustering, SVM prediction
- Language independent


Now is up to you:
Questions?

Thank you for your attention

Tessa **Lau**
tessa.lau@gmail.com

Luciana **Benotti**
luciana.benotti@gmail.com

Julián **Cerruti**
jcerruti@gmail.com

Martín **Villalba**
mvillalb@uni-potsdam.de

Further reading:

- **Benotti, Villalba, Lau and Cerruti** - Corpus-based interpretation of instructions in virtual environments. *50th annual meeting of the ACL (2012)*

- **Benotti and Denis** - CL system: Giving instructions by corpus based selection. *13th European Workshop on NLG (ACL, 2011)*