



SD213

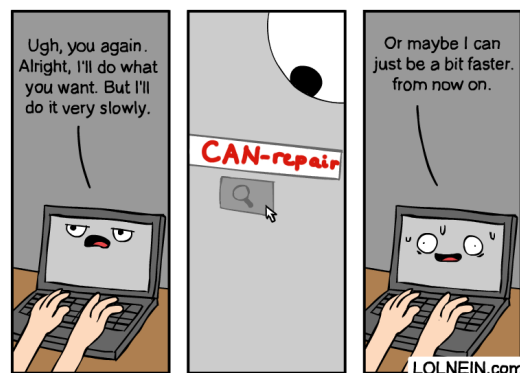
## Cognitive Approach to Natural Language Processing

Micro-study

<https://simplicity.r2.enst.fr/CANLP>

Name: Kanvaly FADIGA

## CAN-Repair : Speed up my slow laptop



## Introduction

Computers get slower as we use them, because of virus and software we install on them. So user often see an incremental decrease in performance over time. Here we will present a discussion between a computer owner and a repair service. The owner has a certain budget and preferences to solve this problem. Given these element, we will generate a dialogue using the CAN method. This tool can be used by online diagnostic websites to give solutions automatically according to users needs. Also, the solution will be give on fly.

## Existing solutions

There are different ways of finding solutions when we have a problem on a computer. The first is to find an exhaustive list of all the things that have already worked and try them all. This is likely to be very time consuming. Another way is not to try them all and to look for cases that resemble our constraints. However, finding the right one can be difficult for a non-expert and also time consuming. Or finally ask an expert to solve the problem for us. We try to do thing differently, creating a discussion with a user and a virtual expert that have internal knowledge using CAN Method.

## Proposition

Our virtual expert has a certain amount of knowledge that we will represent by a causal graph in `prolog`. This graph will allow us to navigate between the solutions to find the one that corresponds to our user's needs. Then our user will give us his preferences on various variables. For example, we can ask him if he has a lot of time because he could be in a hurry. Finally, the user can be asked what is have already done (initial fact). With all these elements we have everything we need to apply the CAN method.

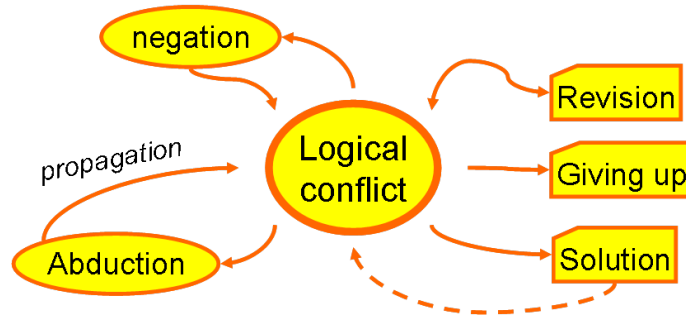


Figure 1: The C-A-N procedure [2]

Let's give an example of CAN procedure for the problem of speeding up a computer :

- **Conflict** : The first step is about detecting a conflict, here the procedure might detect that we have a slow computer, and we wish to have it faster.
- **Solution** : Then the procedure will try to find a solution to that. We take decision at this step. For instance, 'upgrade RAM' can be identified as a cause of speeding up computer, then taking this action will create a new conflict : 'upgrade RAM' wished but not realized.
- **Abduction** : consist in finding cause of an event. This step is useful for the solution step because it give us choices of actions to solve our problem. Here for instance, abduction will return 'upgrade RAM' is a possible cause of speeding up computer.
- **Negation** : Here we do counterfactual reasoning process. It consist of considering **What if an action was done differently**. For instance, if 'upgrade RAM' have preferences of  $v=10$ , and we did it, we will modify the intensity of not doing it to  $v=10$ . We did this so that the next time we look for a conflict we will prioritise not doing it first.
- **Give-up**: if choosing an action fail we set it's intensity to low values to memorises the fact that it doesn't succeed. For instance, setting the intensity of 'upgrade RAM'  $v=-10$  if it fail to solve the issue
- **Revision**: We will come to this if all of our solution fail. We will ask the user to change his preferences otherwise we have no solution to propose. For instance, the user can say that we can consider that he have more money or more time.

## Knowledge

We will build our knowledge given what we know about component slowing down computer. Here are some of the things that might slow down a computer [1]:

- Running out of RAM (Random Access Memory)

- Running out of disk drive space (HDD or SSD)
- Old or fragmented hard drive
- Too many background programs
- Too many startup programs
- Falling behind on Windows or driver updates
- Malware or viruses
- Special effects and visual features

In our work we will consider some of them to build our Knowledge base. We also consider new variables that is important for the user (spend money, do tough\_work,...).

```
% initial facts
initial_situation(-fast_computer).
initial_situation(-get_free_ram).
initial_situation(-get_free_storage).

% default predicates: these predicates are true unless proven false
default(-have_external_disk, _).

%%% causal clauses %%%
% ram
get_free_ram <=== buy_ram.
get_free_ram <=== clear_temp + -have_background_software + -startup_software.
get_free_ram <=== reset_computer.

% storage
get_free_storage <=== remove_some_file.
get_free_storage <=== reset_computer.
get_free_storage <=== buy_external_disk + -move_file_to_ext.

fast_computer <=== reboot + get_free_storage + get_free_ram.

%%% consequences %%%
spend_small_money <=== buy_ram.
spend_lot_money <=== buy_external_disk.
lost_small_file <=== remove_some_file.
lost_all_file <=== reset_computer + -have_external_disk..
```

The remaining parameter are the preference that user have to set.

## Experiments

We have created a web interface that allows the user to define his preferences. Then we launch the program to propose solutions to our problem. **Here is a demo video link** : <https://www.loom.com/share/901fb4b74ff2417d98ba9d29db3b3f31>

## CAN-Repair : Speed up my slow laptop

### Initial fact

- ☐ do you have a fast\_computer
- ☐ do you have a free\_ram when all your app are running
- ☐ do you have a free\_storage on your disk

### Preferences

fast_computer	spend_small_money	spend_lot_money	lost_small_file	lost_all_file
20	-10	-10	-10	-30

Figure 2: User interface created with streamlit [3]

Below you will find a clear transcription of the conversation:

```
preference(spend_small_money, -10).
preference(spend_lot_money, -10).
preference(fast_computer, 20).
preference(lost_small_file, -10).
preference(lost_all_file, -30).
```

KANVALY 😞: I want my computer to be faster.

CAN-REPAIR 😊: ok, **reboot** your laptop.

KANVALY 😞: ok it made it. still slow.

CAN-REPAIR 😊: then, **reset\_computer**.

KANVALY 😞: I can't do that I don't **have\_external\_disk**. So will **lost\_all\_file**.

CAN-REPAIR 😊: ok fine, so you can **buy\_external\_disk** and **move\_file\_to\_ext**.

KANVALY 😞: I really don't want to **spend\_lot\_money**.

CAN-REPAIR 😊: Got it. So I have a simple solution: **remove\_some\_file** it will free space a bit.

KANVALY 😞: I was ok with **lost\_small\_file** but, I can live with it. [lost\_small\_file to 30]

CAN-REPAIR 😊: Great we solve space issue. For the RAM, can you **buy\_ram** ?

KANVALY 😞: I can't because I don't want **spend\_small\_money**.

CAN-REPAIR 😊: I understand, so just **clear\_temp** and avoid **have\_background\_software** and **startup\_software**.

KANVALY 🥳: Nice it works 🚀! Thanks.

Here is the raw output of the program:

```
?- go.
Conflict of intensity 20 with fast_computer
```

```

---> Decision : reboot
Conflict of intensity 20 with fast_computer
---> Decision : reset_computer
Conflict of intensity -30 with lost_all_file
---> Decision : -reset_computer
Conflict of intensity 20 with fast_computer
---> Decision : buy_external_disk
Conflict of intensity -10 with spend_lot_money
Conflict of intensity -10 with spend_lot_money
We are about to live with a problem concerning spend_lot_money ( -10 )!
If you want to change preference for spend_lot_money ( -10 ),
enter number followed by '.' (or else: 'n.')
|: -30.
Conflict of intensity 30 with spend_lot_money
---> Decision : -buy_external_disk
Conflict of intensity 20 with fast_computer
---> Decision : move_file_to_ext
Conflict of intensity 20 with fast_computer
---> Decision : remove_some_file
Conflict of intensity 20 with fast_computer
Conflict of intensity 20 with get_free_storage
Conflict of intensity 20 with fast_computer
---> Decision : buy_ram
Conflict of intensity -10 with spend_small_money
Conflict of intensity -10 with spend_small_money
We are about to live with a problem concerning spend_small_money ( -10 )!
If you want to change preference for spend_small_money ( -10 ),
enter number followed by '.' (or else: 'n.')
|: 30.
Conflict of intensity -10 with lost_small_file
Conflict of intensity -10 with lost_small_file
We are about to live with a problem concerning lost_small_file ( -10 )!
If you want to change preference for lost_small_file ( -10 ),
enter number followed by '.' (or else: 'n.')
|: 30.

true.

```

## References

- [1] CCleaner. *Why your PC is so slow... and how you can speed it up*. URL: <https://www.ccleaner.com/knowledge/why-your-pc-is-so-slow-and-how-you-can-speed-it-up>.
- [2] Jean-louis Dessalles. "A Cognitive Approach to Relevant Argument Generation." In: *M. Baldoni, C. Baroglio, and F. Bex, editors, Principles and Practice of Multi-Agent Systems* (). URL: [https://perso.telecom-paristech.fr/jld/papers/Dessalles\\_16061002.pdf](https://perso.telecom-paristech.fr/jld/papers/Dessalles_16061002.pdf).
- [3] Streamlit. *The fastest way to build and share data apps*. URL: <https://streamlit.io/>.