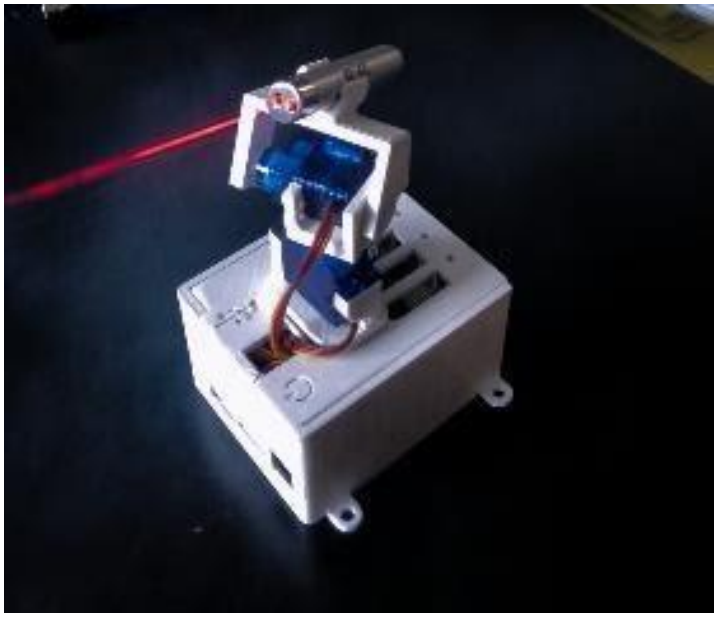


SMART LASER

Math word problem solver robot that points to the solution with a laser.

PROJECT SPRINT #5 +Extra.
DATE: 14 June 2020

Marcos Muñoz González
Roger Piera
Daniel Rojas Pérez (Made the expanded module)
Ange Xu

Table of Contents

Sprint #5 +Extra
Date: 14 June 2020

SMART LASER

Math word problem solver robot that points to the solution with a laser.

Project description

The project will be divided in 3 parts: robotics, computer vision or optical character recognition and natural language processing.

The computer vision or optical character recognition part will be done by making use of the camera embedded in the robot that will detect the question and possible solutions that someone is showing with a poster. The camera will detect the poster with Object Recognition and then transcribe it into text with Optical Character Recognition technique

The Natural Language Processing part will deal with understanding what the question means so it can compute the correct answer. The questions that this NLP part will handle are mathematical based, i.e. “hey smartlaser, give me the multiplication of 20 by 20 and then sum it to a hundred multiplicated by 2 , please.”, “sam had 4 apples but he lost one during his walk to the mountain. then he found two more in a tree and gave one of them to his”, “what is (120 multiplied by 3 and then minus 20) divided by (120 multiplied by three and then minus twenty) ?”, etc. We have taken two different approaches for this task: Deep Learning approach and Parsing and Naïve Bayes approach. Apart from this main task, the robot will also be able to perform 10 more different NLP tasks trained by the DecaNLP project by Salesforce.

The robotics part will allow the motion of the robot, which is based on a robotic arm with 2 degrees of freedom that will be grabbing a laser pointer and will point it to the correct solution out of the few proposed in the poster showed. Since we can not make the hardware part of this, we will be making a simulation of the laser point on a program.

Github Link to our project: https://github.com/danielrojasperez/smartlaser

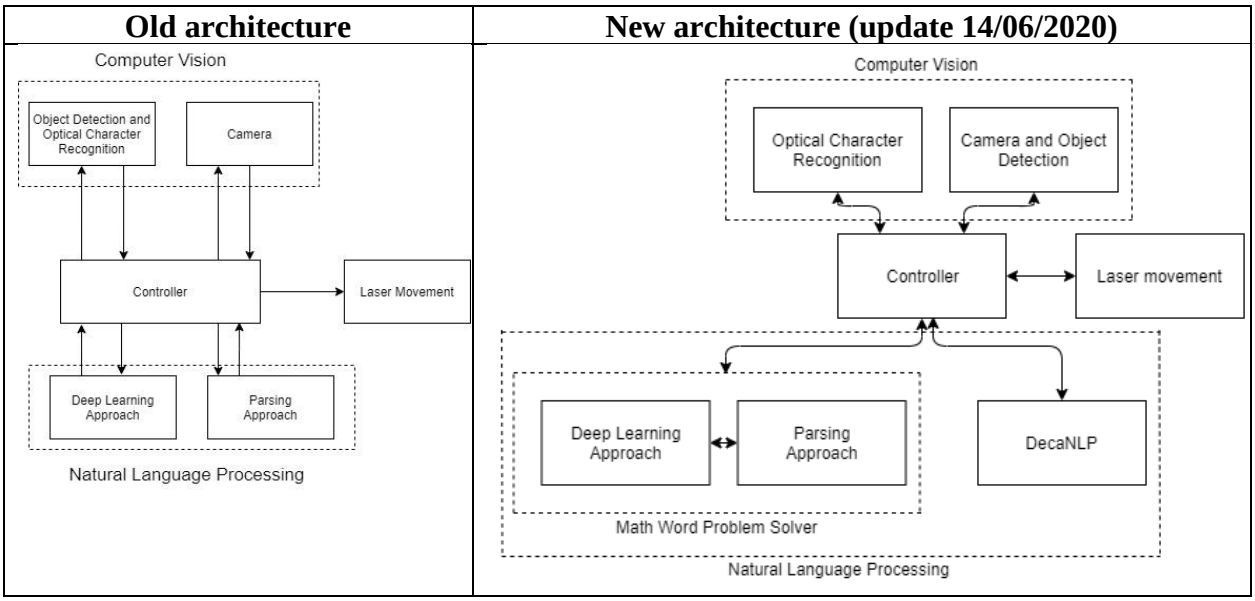
Electronic components

This is the list of the proposed components:

- 2 Micro Servomotors
- Raspberry Pi 4

- Power supply
- Camera
- Laser pointer
- Micro SD 32GB

Software Architecture



Natural Language Processing part will be divided by a minimum of two modules:

1. **Deep Learnning approach and Parsing and Naïve Bayes approach**

Input data: question in text format, possible answers, context (optional)

Output data: answer proposed

This module is a core part of the project, this Natural Language Processing task will be strictly developed by our team and is the main task to perform.

The objective of this task is to be able to answer mathematical problems asked in a natural way.

These are some examples of questions that we want to answer:

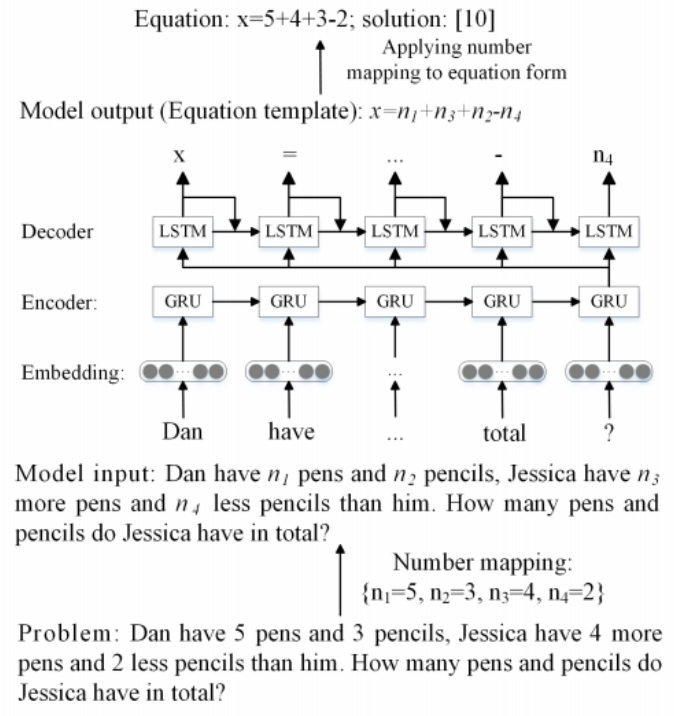
- Calcula 5 + 40.
- Calcula 5 más 40.
- ¿Cuánto es 5 por 40?
- Cuánto es 5 multiplicado por 40 y luego sumado por 20?
- Hey smartlaser, give me the multiplication of 20 by 20 and then sum it to a hundred multiplicated by 2 , please.Deriva $2x^2$
- what is 120 multiplied by 3 and then minus 20?If 5^21 x 4^11= 2 x 10^n what is the value of n?
- Dan have 2 pens, Jessica have 4 pens. How many pens do they have in total ?
- what is (120 multiplied by 3 and then minus 20) divided by (120 multiplied by three and then minus twenty) ?The sum of two numbers is -63. The first number minus the second is -41. Find the number
- What is 10 to the power of two to the power of 3 and then substract to it the multiplicatio n of a hundred by 2.
- Sam had 4 apples but he lost one during his walk to the mountain. then he found two mor e in a tree and gave one of them to his girlfriend.
- I am wondering how many coins I have and what happened to the 10 of them. I remembe r that John asked me to give him half of them. but then he gave me two.
- …

Now, making this mathematical type of questions as general and natural as possible is an extremely difficult thing to make and a key concern for us. Due to this uncertainty, the range of questions that our robot may be able to interpret may differ from our actual initial proposals.

We have decided that the language we will use for this task is English because we will use the Dolphin18k dataset in order to train our model.

The Dolphin18k dataset is a data structure created from ‘yahoo answers’ that specifies a question, a equation template and the solution to the unknowns variables. You can explore more kind of examples of math problems by browsing this dataset.

The algorithmic structure of the Deep Learning approach consists on a Recurrent Neural Network seq2seq (sequence to sequence) model with two Gated Recurrent Units for both encoder and decoder, we also include attention with Bahdanau Attention. Structure of the model:



We first apply number mapping in order to remove the numbers from the problems. This gets us a better training because the sentences have less variation. The inputs go through

an embedding layer for vector encoding and purposes and setting all of the inputs with the same length. The inputs go through the encoding layer and get encoded. The attention mechanism assign a weight to the input words and is then used by the decoder to predict the next word in the sentence. The decoder then return the predicctions and the decoder hidden state. We also apply the teacher forcing technique to decide the next input to the decoder, this technique passes the target word as the next input to the decoder.

Apart from the Deep Learning approach, we have turned our model into a hybrid model by adding another approach. This new approach is based on Parsing (syntactic analysis of the sentences). The program will analyze the sentence and detect the symbols and numbers needed in order to generate the equation. This approach also uses a Naïve Bayes Classifier model to classify whether a symbol is positive or negative.

So, for this task we will run the two approaches and choose the best for providing the solution.

2. DecaNLP module

The Natural Language Decathlon is a multitask challenge that spans ten tasks:

Question Answering: it receives a question and a context that contains information necessary to output the desired answer. decaNLP uses the Stanford Question Answering Dataset (SQuAD 1.1) as the dataset for this task.

Machine Translation: it receives an input document in a source language that must be translated into a target language. decaNLP uses the 2016 English to German training data prepared for the International Workshop on Spoken Language Translation (IWSLT), and we use the 2013 and 2014 test sets for the validation and test sets, respectively.

Summarization: it takes in a document and output a summary of that document. decaNLP uses the non-anonymized version of CNN/DailyMail (CNN/DM) corpus. We include the non-anonymized version of this dataset in decaNLP.

Natural Language Inference: it receives two input sentences: a premise and a hypothesis. Models must then output whether the premise entails contradicts or is neutral with respect to the hypothesis. decaNLP uses the Multi-Genre Natural Language Inference Corpus (MNLI).

Sentiment Analysis: it classifies the sentiment expressed by input text. decaNLP includes the unparsed, binary version of the Stanford Sentiment Treebank (SST), which consists of movie reviews with corresponding sentiment (positive, neutral, negative).

Semantic Role Labeling: given a sentence and predicate (typically a verb) and must determine ‘who did what to whom,’ ‘when,’ and ‘where’. decaNLP includes the Wikipedia domain of QA-SRL 1.0.

Relation Extraction: it takes in a text document and the kind of relation that is to be extracted from that text. decaNLP includes QA-ZRE, a dataset that maps relations to a set of questions so that relation extraction can be treated as question answering. At validation and test time, the relations are different than those seen in training, so that this is a zero-shot relation extraction task.

Goal-Oriented Dialogue: Dialogue state tracking is a key component of goal-oriented dialogue systems. Based on user utterances and system actions, dialogue state trackers keep track of which predefined goals the user has for the dialogue system and which kinds of requests the user makes as the system and user interact turn-by-turn. decaNLP includes the English Wizard of Oz (WOZ) restaurant reservation task, which comes with a predefined ontology of foods, dates, times, addresses, and other information that would help an agent make a reservation for a customer.

Semantic Parsing: it translates unstructured information into structured formats so that users can interact with structured information (e.g. a database) in natural language . decaNLP includes the WikiSQL dataset, which maps natural language questions into structured SQL queries.

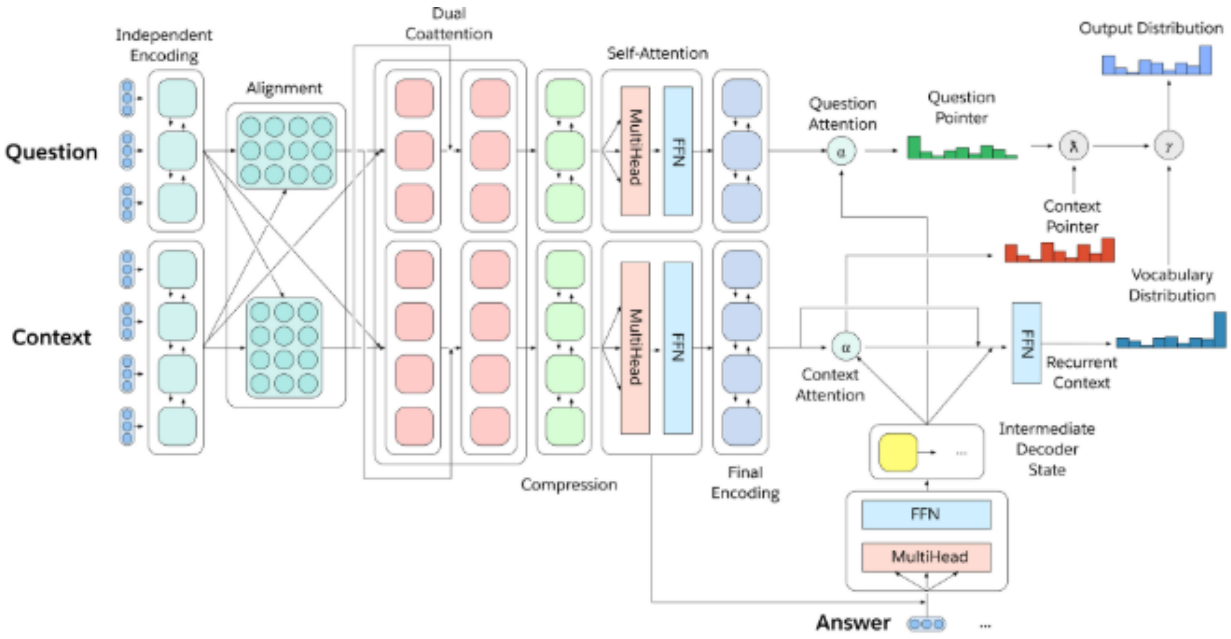
Commonsense Reasoning: models are required to answer questions like “Joan made sure to thank Susan for the help she had [given/received]. Who had [given/received] help? Susan or Joan?”.

This module will make the robot perform a larger range of Natural Language Processing Tasks. Before, the robot could solve math word problems with the module we built. Now, with the help of the DecaNLP project, the robot is able to perform 10 tasks more, including some state-of-the-art tasks.

These tasks are very diverse and are based on question answering, which suits perfectly with our robot model. Although not all of the tasks are optimal for it because our robot works better with shorter text so the camera can analyze the text better.

Examples of this implementation can be found on the ‘14/06/20 Extra module addition’ section (page 25).

All of these complex tasks are performed by a single system model. The following figure is the model structure of the Multi Question Answering Model:


Question
Context
Independent Encoding
Alignment
Dual Coattention
Compression
Self-Attention
MultiHead
FFN
Final Encoding
Question Attention
Question Pointer
Context Pointer
Output Distribution
Vocabulary Distribution
Context Attention
Recurrent Context
Intermediate Decoder State
FFN
MultiHead
Answer
...

Computer vision part will be divided into trees modules:

1. **Object detection**

   Input data: digital image

   Output data: boolean variable that tells us whether to continue to the next step

   This module is responsible for determining if the input image contains the information we want to process, that is, a poster with one handwritten question and next to it a few answers of the question. The module will calculate what is the probability that the image contains what we are looking for and according to the value of this probability, it will return true if the value exceeds the threshold that we set at the beginning, otherwise it will return false.

   If the return value is true, the controller will proceed to deal with the question, otherwise it does nothing.

2. **Handwritten Text Recognition**

   Input data: digital image

   Output data: recognized text

   In this part we will use an offline Handwritten Text Recognition (HTR) system to extract the question from the image and transform it to text. For this step we will need to use an open source HTR project on github (https://github.com/githubharald/SimpleHTR) that provides us the pre-trained model with Tensorflow. The advantage of using this project is that it's simple and does not require a large amount of computational power and extra memories.

   In order to facilitate the recognition of the text, it is necessary to segment the image into word-images, that is, images parts that contain a word. As smaller these word-images are, the better the recognition accuracy will be.

3. **Localization of the correct answer**

   Input data: digital image, answer to the question

   Output data: pixel coordinates

   Once we have obtained the correct answer to the question using the NLP technique, we then proceed to find the image area where the answer is located and return the

coordinates of the image. In case of not being able to find the correct pixel location, the module will return (-1,-1).

The Laser Movement module will deal with the Robotics part of the movement, making use of 2 Servomotors, because we need two degrees of freedom.

Once the coordinates to the answer are provided using Computer Vision we will calculate the movement that the Servos need to do so the laser can point to the correct answer.

The biggest responsibility of this module will be to interpret the coordinates provided by the Computer Vision module so that the laser points to the correct coordinate.

The first of the Servomotors will handle movement in the "x" axis, rotating left and right.

The second Servomotor will handle the movement in the "y" axis, rotating up and down.

To initially calibrate the laser we will need to move it until the light from the laser appears on camera, then we will move the laser until it points to the desired "neutral" position.

There are multiple ways to point the laser to the correct coordinate.

- We can coordinate with the Computer Vision module to find the limit of the paper with the answers and point the laser there as a "0,0" position in the axis. If the paper moves, the laser will follow the neutral position until it has to point to an answer.

- We can use the general image obtained from the camera and situate the pointer in the top left corner neutrally. Then we just need to receive the coordinates of the answer from the Computer Vision module and point to the correct spot. Even if the paper moves, the camera itself won't move, so we won't have to re-calibrate the laser.

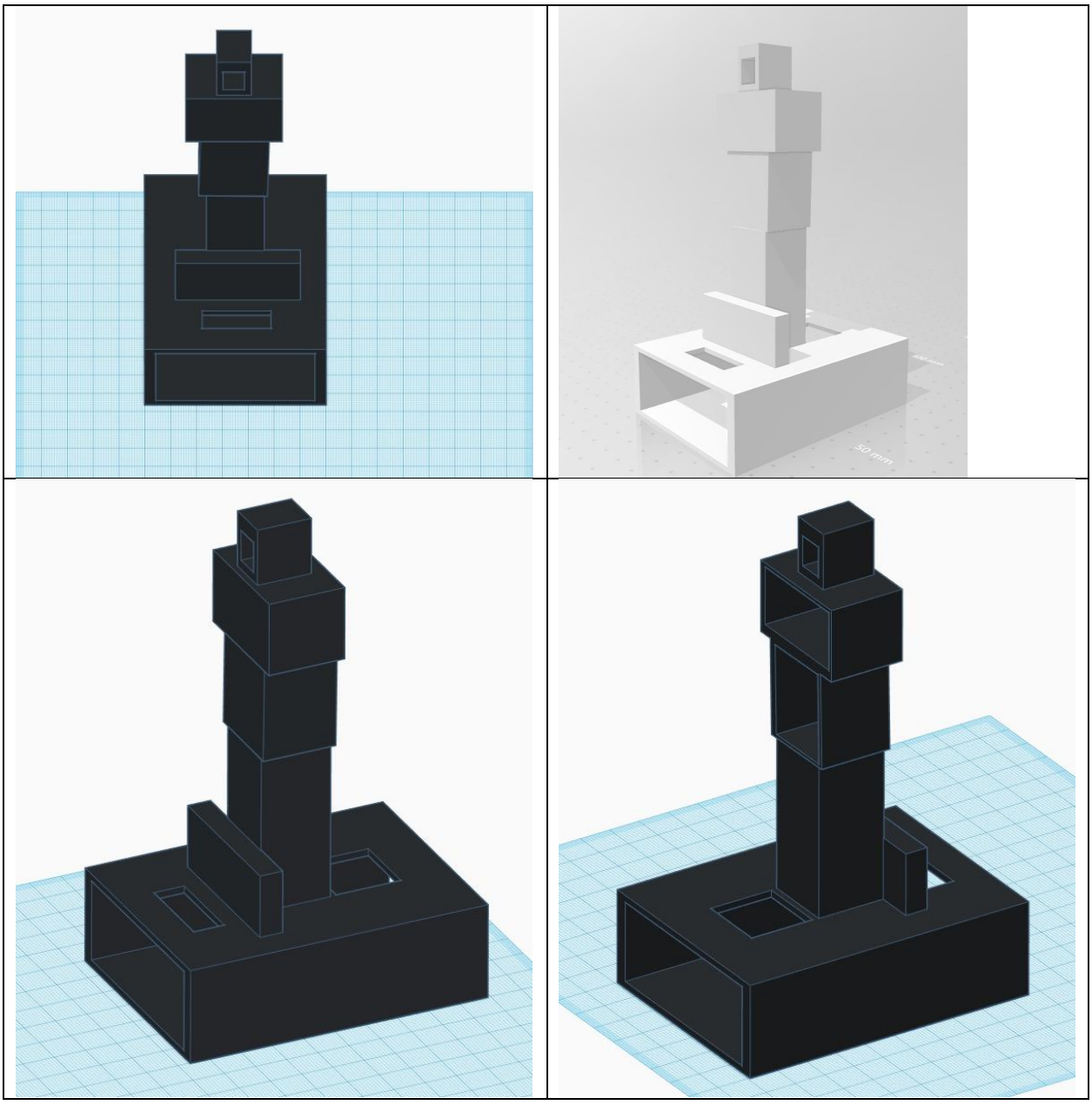
It is important to keep in mind that depending on the proximity of the paper towards the camera the same value for 1 pixel moved in the coordinates may not be the same as in another position of the paper. So, if the paper moves we will need to recalibrate how much distance is the laser moving every pixel so that we may move to the desired coordinates.

Once we get the coordinates of the answer we will transform them so that the laser points to it.

First, we will have to know the movement that the light of the laser needs to do so it can point to the correct answer, this could be implemented in different ways:

- Using the relative value of the coordinates. We can calculate the difference between the current coordinates and the new solution coordinates and move the pointer there.
- Using the absolute value of the coordinates. We can move the pointer back to neutral position and then move back to the new provided coordinates.

Then, we will need to transform the movement of the light of the laser to the movement of the laser itself, rotating the two servo motors to get the light of the laser to point correctly.



Amazing contributions

The Smart Laser project is a unique robot idea that gathers a few modern techniques that are of a great degree of relevance on these days, such as Natural Language Processing and Optical Character Recognition. Being this project of such a broad artificial intelligence based techniques, it makes this robot a very complete and up-to-date project.

The interaction with the human is our key concern, we want the robot to be as “casual” as possible. That’s why we want the range of questions or tasks that the human can ask to the robot to be as encompassing as possible, as well as enabling hand-writting posters and the position of the user holding the poster to not be strict.

Since the robot will realize state-of-the-art tasks, we consider that the project deserves a state-of-the-art mark.

Extra components and 3D pieces (COMPULSORY only a number of them, not all of them are compulsory in this sprint)

- Posters with questions and their possible solutions.
- 3D pieces for the box structure to hold raspberry pi and camera.
- 3D pieces to make the robotic arm.

Strategy for validation, testing and simulation (COMPULSORY, only the strategy, you do not need the tests to be done yet at this stage)

We will make a batery of tests to make sure the Robot can move the laser to any coordinate provided irregardless of its current location. To test this we will try 2 different models:

- The first one, we will reset the coordinates of the servos to a default location before moving to the new coordinates.

- The second one, we will calculate how to move to the new coordiantes from the current ones.

Using an accuracy system, we will calculate the best model and use that one moving forward.

To validate the accuracy of the OCR and HTR models that we have used for character recognition in images, we will prepare a dataset with images of different types (distinguishing between handwritten or not) and apply the cross validation strategy.

In regards to the math word problem solving validation, since we are using a dataset, we can validate and get metrics of the performance of our model by computing k-fold cross-validation.

Foreseen risks and contingency plan

Risk #	**Description**	**Probability** (High/Medium/Low)	**Impact** (High/Medium/Low**)**	**Contingency plan**
1	Mathematical NLP part developed by us is too complex to implement	Medium	Medium	Robot will still be functional thanks to the usage of the external tool.
2	Optical character recognition part is too complex to implement	Medium	Medium	We can use external tools to implement this part.
3	Laser pointer difficulties with pointing correctly	Low	High	We can implement different ways to make this part. Such as giving absolute coordinates, giving relative coordinates based on the actual position of the pointer, giving relative coordinates based on the position of the laser in the poster, etc.
4	Not enough processing power in the Raspberry Pi	Low	High	If there isn't enough processing power, we can migrate all the resource intensive tasks into the computer and

				communicate with the Robot using WIFI.

Actual State of the project (ordered by module and sprint)

Computer Vision Module

OCR recognition

On the part of OCR recognition for now we have achieved the following points:

1. Implementation of basic OCR that recognizes characters in an image.

2. HRT implementation that recognizes handwritten characters in an image(with no high accuracy).

- The pre-trained model (https://github.com/githubharald/SimpleHTR) we are using is completely offline and in each execution of the program it only accepts as input data, a reduced size image that contains only one word. For this reason, to use this model we need to pre-treat the input image.

- Using computer vision techniques the program is now able to locate in an image, the areas that contain a word. We extract these areas and save them as a mini image that only contains one word(word-image).

- By passing these word-image to a pre-trained model that only accepts reduced-size images, we achieve our goal in this part. The model returns the recognized words and we reconstruct the question that our robot must solve.

Problems that still have to be solved:

-The model used for character recognition works with an accuracy of 70% in the case of treating characters (depends on its thickness) and fails in the case of numbers.

-Performance is poor.

Proposed solutions:

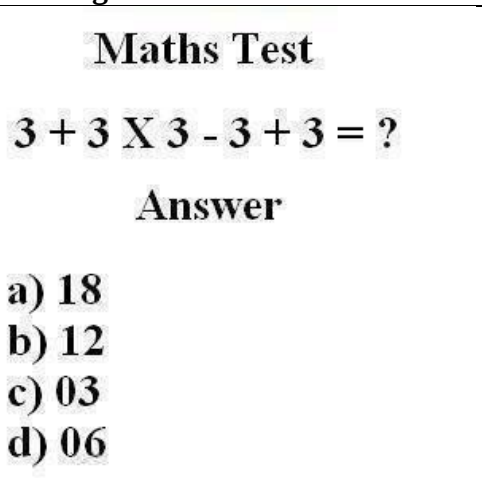
-Use the String Matching algorithm to increase the accuracy with the characters, or use some libraries that autocorrect words.

-Use another model for the recognition of handwritten digit characters.

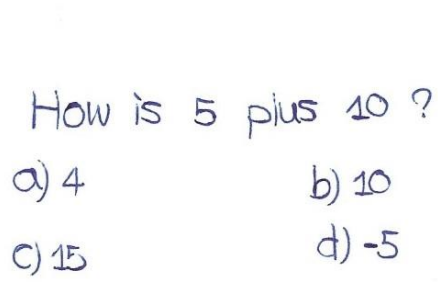
-Combine these two models.

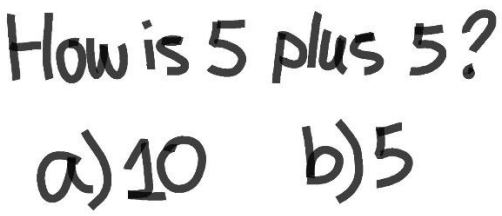
-Use Google Vision Cloud as alternative solution.

TEST 1(Basic OCR recognition)

Input image	Output
	Maths Test 3+3X3-34+3=? Answer a) 18 b) 12 ©) 03 d) 06

TEST 2(Handwritten characters recognition)

Input image	Output
	How is ' pls s0 ?) A b s0 , 15 o oj

	How is 's plus 's ? 30 b5

Object detection

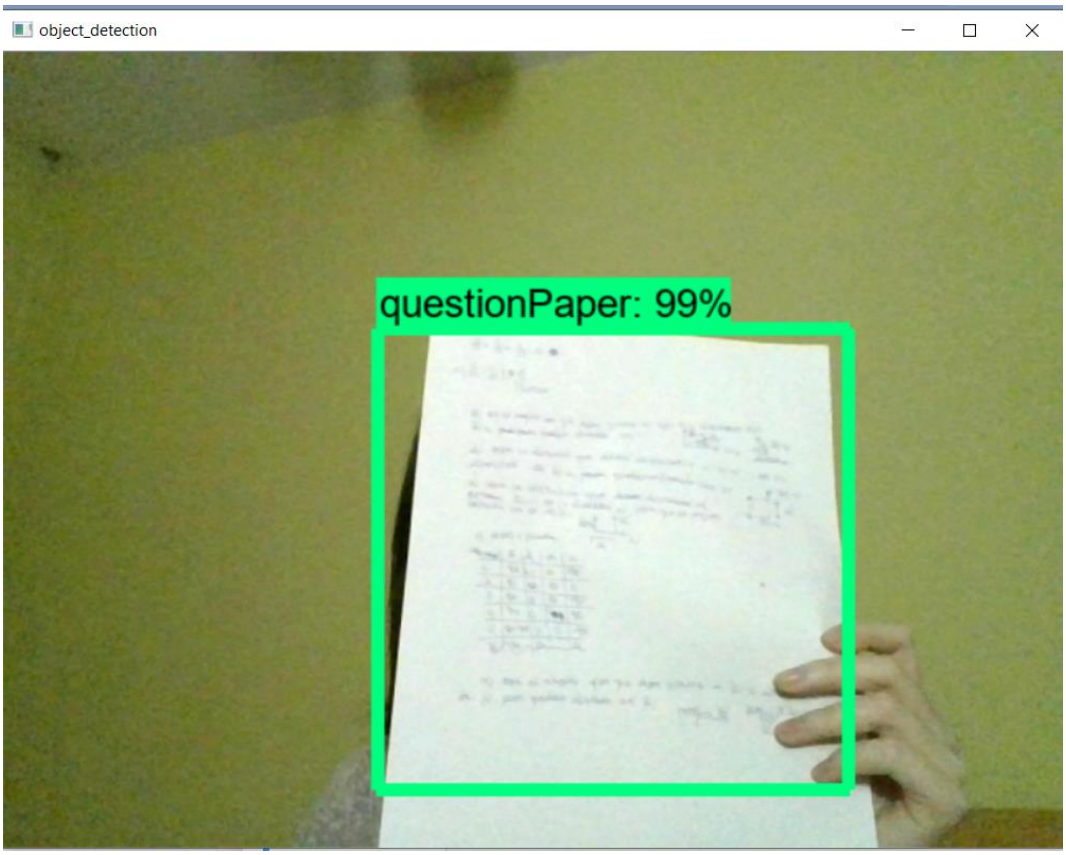
The objective of this part is to create an object detector that is able to locate where the question paper is located, which is held by a person in a scene. We have used the **Tensorflow object detection API (version 1.14)** as the main tool, and we have also created our own image dataset to train and validate the model.

The steps we have followed to achieve our objective are the following:

1. Gathering data
2. Labeling data
3. Generating TFRecords for training
4. Configuring training
5. Training model
6. Exporting inference graph
7. Testing object detector

The results obtained are shown below:

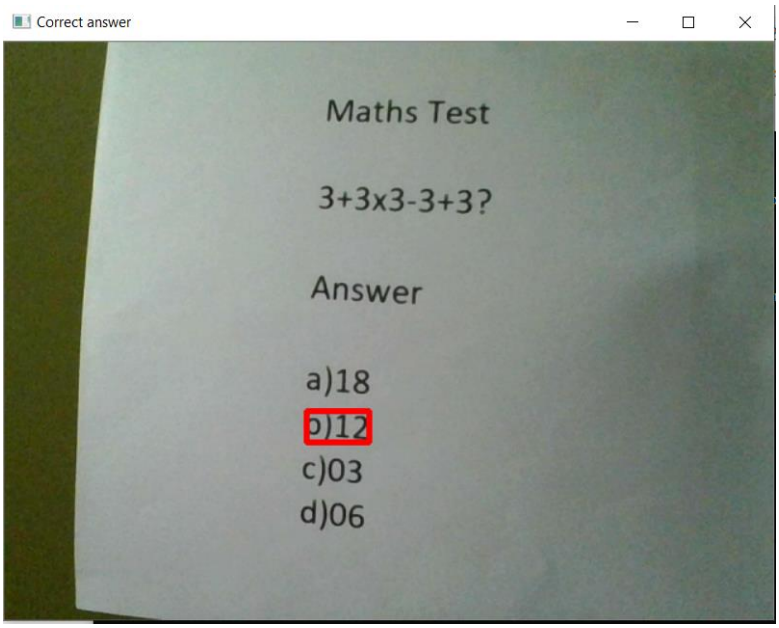
The input data are images captured by the webcam and the detector marks the area where the paper is located. The next step is to extract this area and apply in this case, Handwritten characters recognition(HRT).



13/05/2020 Update

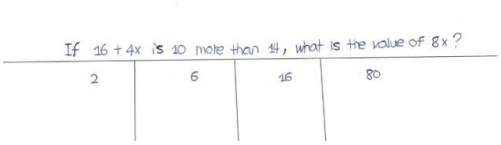
Localization of the correct answer

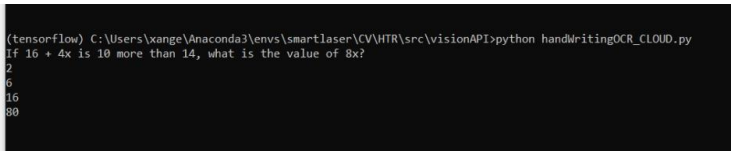
After detecting the question sheet, we capture it with the camera and save it as an image. This image will be sent to the OCR recognition module and this will be in charge of generating a file with the recognized text (the question that our robot will solve) . This file will be passed to the NLP module that will return the correct answer to the question. Now we can point out the correct answer in the image.



26/05/2020 Update

Some errors have been resolved and we included the ability to transcribe hand-written text image to text using the Google Vision API. Although we are not using it for our tests.





Natural Language Processing Module

Currently we are working on designing and developing the math word problems solver by ourselves. The artificial intelligence task that we are developing has not been developed by anyone exactly as the way that we are doing it (in terms of using a different dataset) so there is no step-to-step guide that facilitates our work. In order to learn and develop how to do the math word problem solving, we are relying on academic papers that we will list in the references and the official Tensorflow documentation.

After all the research done, we are currently developing a Recurrent Neural Network seq2seq model for transforming problem text to a math equation. This model is great for our task because the inputs and outputs of our neural network are sequences (text). This model is composed by an embedding layer, an encoder (we are currently using a Gated Recurrent Units) and a decoder (Long Short-Term Memory).

Why are we using an embedding layer? Given the categorical nature of the words, the model will first look up the source (problem question) and target (equation) embeddings to retrieve the corresponding word representations

Why are we using GRU and LSTM? Because they are good at keeping in memory states for a long time, so they work well for text sequence handling. We might change this design because the inspiration of having this model system is by a paper that was training with a dataset of around 28K samples. We are using a smaller, but cleaner, dataset of around 1800 samples.

We currently have analyzed, preprocessed and cleaned the dataset, created a vocabulary of the words that appear in text problems and equations necessary for the neural network, designed and developed the RNN-based model (Encoder, Decoder, loss function) and we already have the training algorithm.

In the next days, we will focus on debugging the program because it is still not able to train the data due to some input errors. We will also develop the evaluation function and plot results.

Overall, the task is being complex to learn and develop, but we are currently in a good path to achieve our goals. Our main concerns right now are:

- The model may not achieve the accuracy that we want.
- The model has trained on math word problems that we are not really interested on resolving due to them being kind of complex and the user using SmartLaser may not be able to create a problem of that kind. For this problem, we hope that learning the complex problems that the dataset has, will also make it able to solve the easier problems that we want SmartLaser to solve.

If any of these cases happens, we have thought on redoing our NLP math word problem solver by changing the dataset and creating one ourselves, which could be even much more complex. Our idea of creating our own dataset would be realized by stablishing a large amount of equation templates and generating N semi-random problem texts to each of those equation templates, creating a very large dataset of the kind of problems that we want SmartLaser to solve. We are really motivated by this idea for being unique and would be very learning-satisfying.

In the next weeks, we will focus on optimizing the math word problem solver model by tuning different parameters and applying some processing or technique (such as number mapping) and using the external tool DecaNLP for solving NLP tasks other than math word problems.

13/05/2020 Update

We have finished the math word problem AI and we are currently trying to tune it. We have got some good results but it still needs some tuning. We are still concerned about the two points mentioned before.

These are some of the predictions that our AI did in order to get an equation out of a word math problem (the main task).

```
Sentence:  ['<start>', 'when', 'the', 'smaller', 'of', 'two', 'consecutive', 'integers', 'is', 'added', 'to', 'three', 'times', 'the', 'larger,', 'the', 'result', 'is', '43.', 'find', 'the', 'two', 'integers.', '<end>']
Equation:  ['<start>', 'equ', ':', 'x', '-', '3', '(', 'x', '-', '1', ')', '=', '43', '<end>']
Prediction:  ['equ', ':', 'x', '-', '3', '(', 'x', '-', '1', ')', '=', '67', '<end>']
```

```
Sentence:  ['<start>', 'the', 'sum', 'of', 'two', 'consecutive', 'even', 'numbers', 'is', '30.', 'what', 'are', 'the', 'numbers?', '<end>']
Equation:  ['<start>', 'equ', ':', '2', '*', 'k', '+', '2', '*', 'k', '+', '2', '=', '30', '<end>']
Prediction:  ['equ', ':', '2', '*', 'k', '+', '(', '2', '*', 'k', '+', '2', ')', '=', '288', '<end>']
```

Once we are confident with our equation maker, we will develop the equation resolver, which should be easy.

The DecaNLP module should also not be as complex as our main task to implement.

26/05/2020 Update

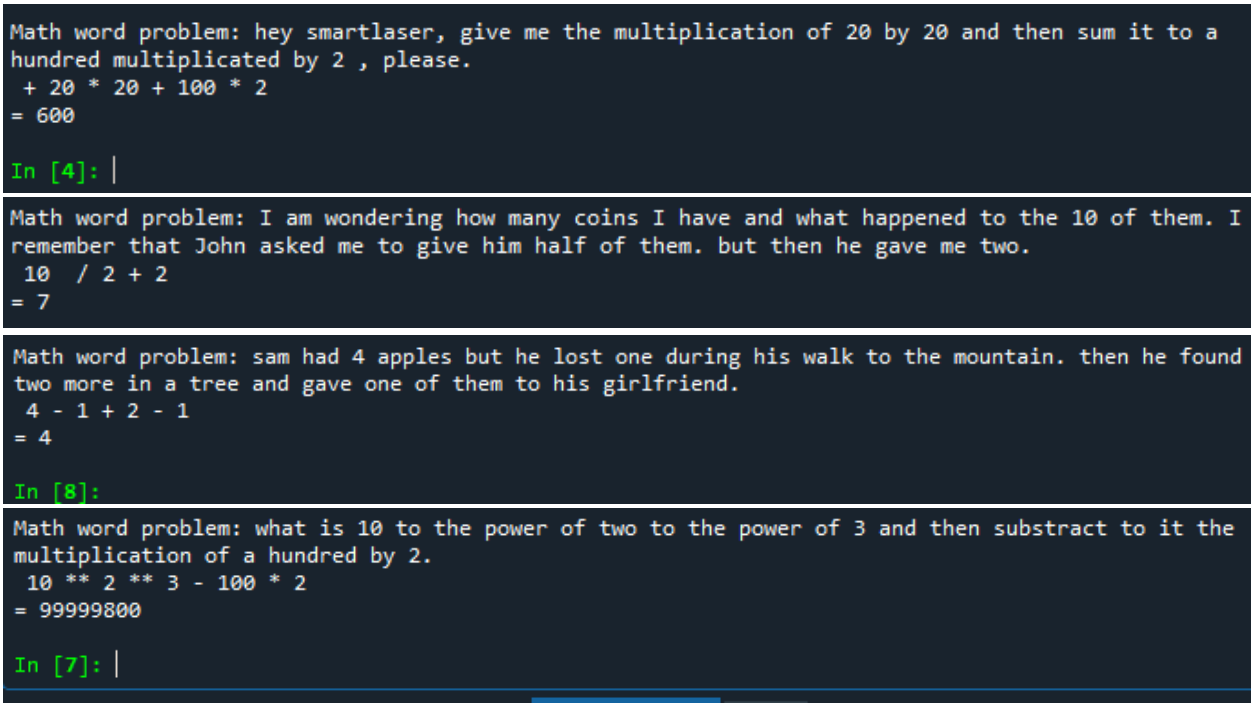
(we have updated the NLP software architecture report)

The results from the Deep Learning were not as accurate as we desired. In order to support the lack of accuracy, we built another approach for the task. This approach consists in the parsing technique. The math word problem will be syntactically analyzed and will detect symbols and numbers from the text and build an equation to solve with the help of a Naive Bayes model classifier that will classify whether a symbol is positive or negative.

We decided to build a hybrid module consisting of the two different approaches: Deep Learning approach and Parsing and Naïve Bayes approach.

As a result of this hybrid model, it will translate the math word problem to an equation and then solve it satisfactorily.

These are some results:



```
Math word problem: what is ( 120 multiplied by 3 and then minus 20 ) divided by ( 120 multiplied by
three and then minus twenty ) ?
 ( + 120 * 3 - 20 ) / ( + 120 * 3 - 20 )
= 1

In [6]:
```

```
Math word problem: what is 120 multiplied by 3 and then minus 20?
 120 * 3 - 20
= 340

In [5]:
```

Hardware

In this Sprint we made some progress in the Laser Robot simulation. We have managed to make a moving robot with a laser pointer. In order to get to this point we have faced many difficulties. First, we tried different robot simulators, among them:

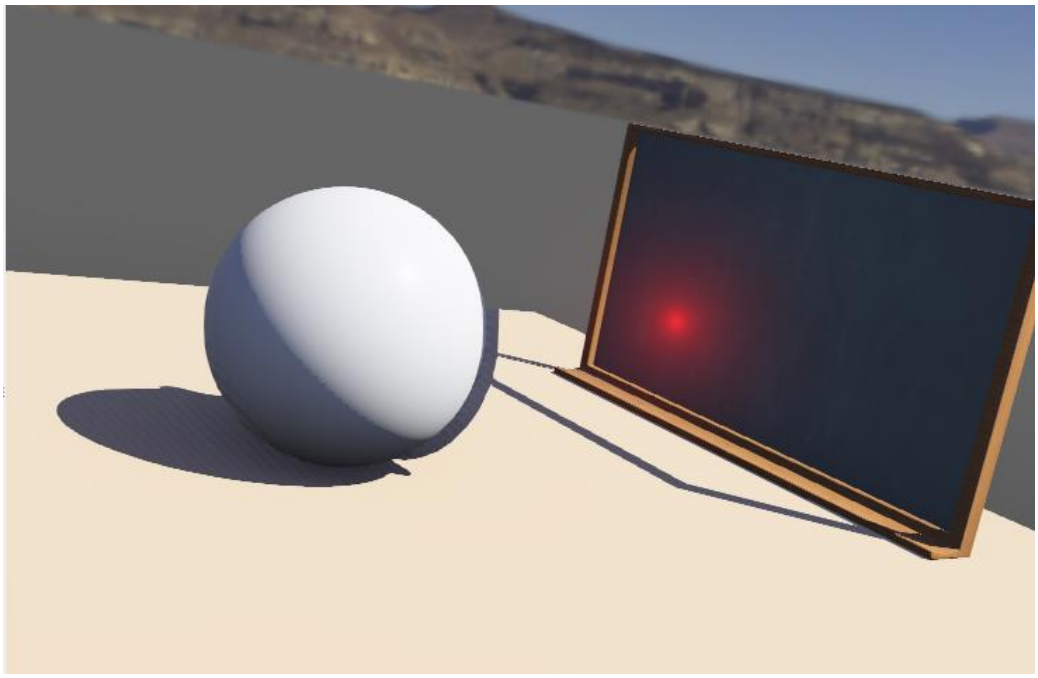
Tinkercad, but it was too limited and simple for our needs.

Rokisim, this one also felt to limited and we didn't find any way to make new 3d simulations.

Gazebo, this one had more options than the other ones, but it was very badly documented, and the software itself was very counterintuitive, we gave up trying to make it work. It also gave us a lot of problems trying to get it to work on Windows.

We attempted to use Fritzing but it was hidden behind a paywall.

We finally settled for Webots. This software is highly complex but at the same time it offers a wide array of options to simulate our robot. At the same time, it has extensive documentation. We believe that with it we can obtain very good results. We already have a proof of concept laser made, which can point in various directions.



Currently the robot can get an input from the keyboard and move the laser in a predefined location. Also, we are planning on remodeling the scenario to have a different look, something

like a real presentation, but in our current 3D environment and making the robot simulation model more like the 3D model that we already made.

```
INFO: laser: Starting controller: python.exe -u "laser.py"
[laser] Commands:
[laser]  I for displaying the commands
[laser]  A Move laser to the right
[laser]  D Move laser to the left
[laser] Moving laser to the right
[laser] Moving laser to the left
[laser] Moving laser to the left
```

13/05/2020 Update

In this sprint we have managed to advance with respect to the movement and precision of the coordinates:

- Now you can enter a coordinate input of type x, y, z using the keyboard

- You can now increase the x, y, z values gradually so that you see the movement of the laser

- A keyboard management system has been created so that the user has more control over it

- The visual interface of the system has been improved

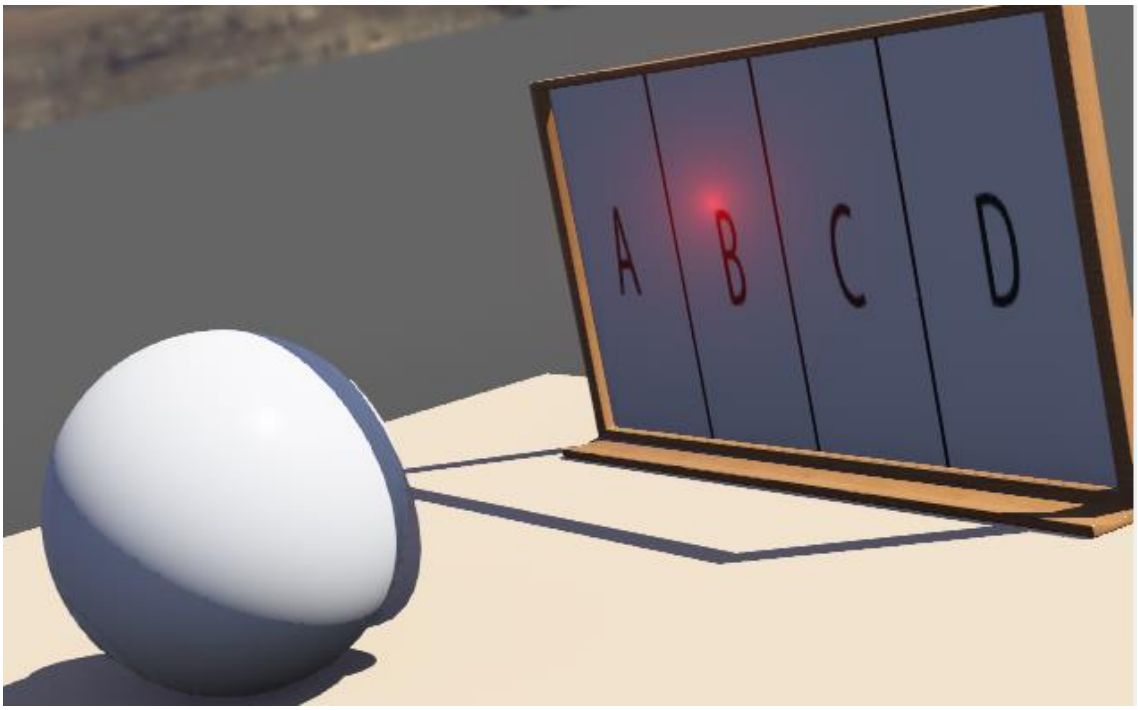
26/05/2020 Update

In this Sprint we made great achievements in the development of the Laser module. The laser now points to the specified response in the board. We can as we wish, change the image in the board dynamically and we have added several inputs to allow a better interaction.

This achievements are thanks to the Python controller and some modifications of the proto file in Webots, like adding textures.

We can be thankfully that Webots is a friendly, well documented and easy to use system for robot simulation.

The controller also opens a txt file which contains the response from the NLP and the vision computer module and can print and show the correct answer in the table and coordinates.



We have also faced severe difficulties like coordinate translation in the table as Webots is a 3D environment and coordinates are set up within the world.

So far so good, the results are very good and we are now very prideful about this module.

14/06/20 Extra module addition

We have added a whole new NLP module. The module will make the robot perform a larger range of Natural Language Processing Tasks. Before, the robot could solve math word problems with the module we built. Now, with the help of the DecaNLP project, the robot is able to perform 10 tasks more, including some state-of-the-art tasks. These new tasks are the nexts:

Question answering, machine translation, summarization, natural language inference, sentiment analysis, semantic role labeling, relation extraction, goal-oriented dialogue, semantic parsing and pronoun resolution.

These tasks are very diverse and are based on question answering, which suits perfectly with our robot model. Although not all of the tasks are optimal for it because our robot works better with shorter text so the camera can analyze the text better.

These are some examples of the performance of DecaNLP:

```
Question Answering task
-----------------------
Context: ...Southern is a major economic center for the state of
California and the US...
Question: What is a major importance of Southern California in
relation to California and the US?
Answer: major economic center

In [10]:
Machine Translation task
------------------------
Context: Most of the planet is ocean water
Question: What is the translation from English to German?
Answer: Der Großteil der Erde ist Meerwasser

In [11]:
Summarization task
------------------------
Context: Harry Potter star Daniel Radcliffe gains access to a reported
£320 millon fortune...
Question: What is the summary?
Answer: Harry Potter star Daniel Radcliffe gets £320M fortune...

In [12]:
Natural Language Inference task
------------------------
Context: Premise: Conceptually cream skimming has two basic dimensions
- product and geography.
Question: Hypothesis: Product and geography are what make cream
skimming work. Entailment, neutral, or contradiction?
Answer: Entailment

In [13]:
```

```
Sentiment Analysis task
-----------------------
Context: A stirring, funny and finally transporting re-imagining of
Beauty and the Beast and 1930s horror film.
Question: Is this sentence positive or negative?
Answer: positive

In [14]:
Semantic Role Labeling task
-----------------------
Context: Areas of the Baltic that have experienced eutrophication.
Question: What has something experienced?
Answer: eutrophication

In [15]:
Relation Extraction task
-----------------------
Context: Cycle of the Werewolf is a short novel by Stephen King,
featuring illustrations by comic book artist Bernie Wrightson.
Question: Who is the illustrator of Cycle of the Werewolf?
Answer: Bernie Wrightson

In [16]:
Goal-Oriented Dialogue task
-----------------------
Context: Are there any Eritrean restaurants in town?
Question: What is the change in dialogue state?
Answer: food: Eritrean

In [17]:
Semantic Parsing task
-----------------------
Context: The table has column names... Tell me what the notes are for
South Australia
Question: What is the translation from English to SQL?
Answer: SELECT notes from table WHERE "Current Slogan" = "South
Australa"

In [18]:
Pronoun Resolution task
-----------------------
Context: Joan made sure to thank Susan for all the help she had given.
Question: Who had given help? Susan or Joan?
Answer: Susan

In [19]:
```

References

This project has been inspired by the following Internet projects:

Structure of the robot similar to: https://www.jjrobots.com/remotely-controlled-laser-pointer/

Math word problem solving initially inspired by: https://www.wolframalpha.com/

DecaNLP: https://github.com/salesforce/decaNLP

Dolphin18k dataset: https://www.microsoft.com/en-us/research/project/sigmadolphin-2/?irgwc=1&OCID=AID2000142_aff_7806_1246483&tduid=%28ir__mtc1liz06wkftwzfkk0sohz3xm2xnkxw9lnbxykl00%29%287806%29%281246483%29%28%28b8d7daadaa518831fac5029563f7f01d%29%2881561%29%28686431%29%28at106140_a107739_m12_p12460_cES%29%28%29%29%28b8d7daadaa518831fac5029563f7f01d%29&irclickid=_mtc1liz06wkftwzfkk0sohz3xm2xnkxw9lnbxykl00

TinkerCAD: https://www.tinkercad.com/

RoKiSim: https://www.parallemic.org/RoKiSim.html

Gazebosim: http://gazebosim.org/

Fritzin: https://fritzing.org/services/

Cyberbotics: https://www.cyberbotics.com/

Deep Neural Solver for Math Word Problems (mainly this one)
https://ai.tencent.com/ailab/media/publications/Yan_Wang-EMNLP2017-Deep_Neural_Solver_for_Math_Word_Problems.pdf

Sequence to Sequence Learning with Neural Networks https://arxiv.org/pdf/1409.3215.pdf

Solving Math Word Problems
https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1184/reports/6866023.pdf

Learning to Automatically Solve Algebra Word Problems
https://www.aclweb.org/anthology/P14-1026.pdf

Object Detector: https://towardsdatascience.com/creating-your-own-object-detector-ad69dda69c85