"A cloud solution for analysing patterns in NGO projects"

Team Number: Team 36

Names: Rachel Mattoo, Mark Anson, Yansong Liu

Date: 28/02/2020

Our progress so far

1. Chunking (Rachel)

I have been using the NLTK software available online to construct an abstract syntax tree for a passage with corresponding labels. This was an alternative idea to replace our machine learning solution and provide an alternative brute-force method, as proposed by our NLP expert at our previous meeting. The main purpose of the chunking method was to extract numeric information because they had separate labels (NP) to the surrounding information. The aim was to extract the financial information. Each word is tokenized and labelled corresponding to a dictionary of labels relating to the word type. This is why this method is convenient for numerical extraction because words of a numeric type will have a different label to words of a non-numeric type.

When a passage is passed into the NLTK software for chunking, a subtree is created. The subtree includes labels for each word in the passage. This subtree can now be traversed to search for data, specifically numeric data.

We decided not to pursue this algorithm because we found that the machine learning solution, BERT, was much more powerful. This is because of its ability to train on datasets.

2. Form + NGO name extraction (Mark)

In order to get the name of the NGO from the documents, we can't necessarily rely on our question and answer system. I have built an algorithm to extract the names from each pdf.

It achieves this by comparing all words in the first three pages of the document with both the file name and any URLs within the document to create the name. At the moment the algorithm is not 100% accurate but is still accurate enough to be useful.

The aim here was to attempt to extract data from tables within the files. This algorithm is not yet completed, but currently I am first extracting the tables, and running the rows through a filter to try and find values we can work with.

3. Training + Fine-tuning data

We created a SQUAD maker based on ALBERT, which is essentially a question and answer system. We created a function which would allow for a question to be asked to a page of the pdf report, and an answer would be returned to that question with a high level of accuracy.

4. ANCSSC Database (Mark)

Following from guidelines sent to us by Matt, the client for Team 37 (a team we are working in collaboration with), I have created the first iteration of the ANCSSC database structure, to be used primarily by the other second year team and the master's team. Below is the ER diagram of the ANCSSC database.
One we get the ok from Matt, we can undergo any revisions required by the master's team.
We are assuming that multiple revisions will be required to get the database to a state which fully corresponds with the front-end web app.

Timeline of development:

Before 09/02/2020:
We are facing the problem that most algorithms available online cannot extract information properly. Due to the high flexibility of natural language text information, it is difficult for machines which are deterministic, to understand indeterministic things such as language. We tried using frequency statistics to locate where the possible desired information may locate, as part of the early pdf extraction tool mentioned in earlier reports. However, this method can be only used to locate information that has general form. For example, since financial information in a report usually contain several keywords, such as total income or total expanses, we can use this algorithm successfully locate where this type of information is. However, for some information which doesn't have keywords like project information, our algorithm cannot handle it. The corpus for natural language is so varied that brute-force solutions will not work, so we must turn to machine learning solutions.

09/02/2020:
As we were facing the technical challenge mentioned above, we decided to consult with experts for a possible solution. Discussing with Dr. Pontus Stenetorp, we were told that there are several natural language processing toolkits can be used for addressing our technical difficulties. One was Chunking, which was a brute-force solution for extracting data containing different data types. The other, a toolkit we adopted, is BERT, Bidirectional Encoder Representations from Transformers. BERT is capable of comprehending text and answering any question based on the text. This machine learning solution essentially allows us to make our machine understand text, so we considered this is the ultimate solution for the difficulty we faced.

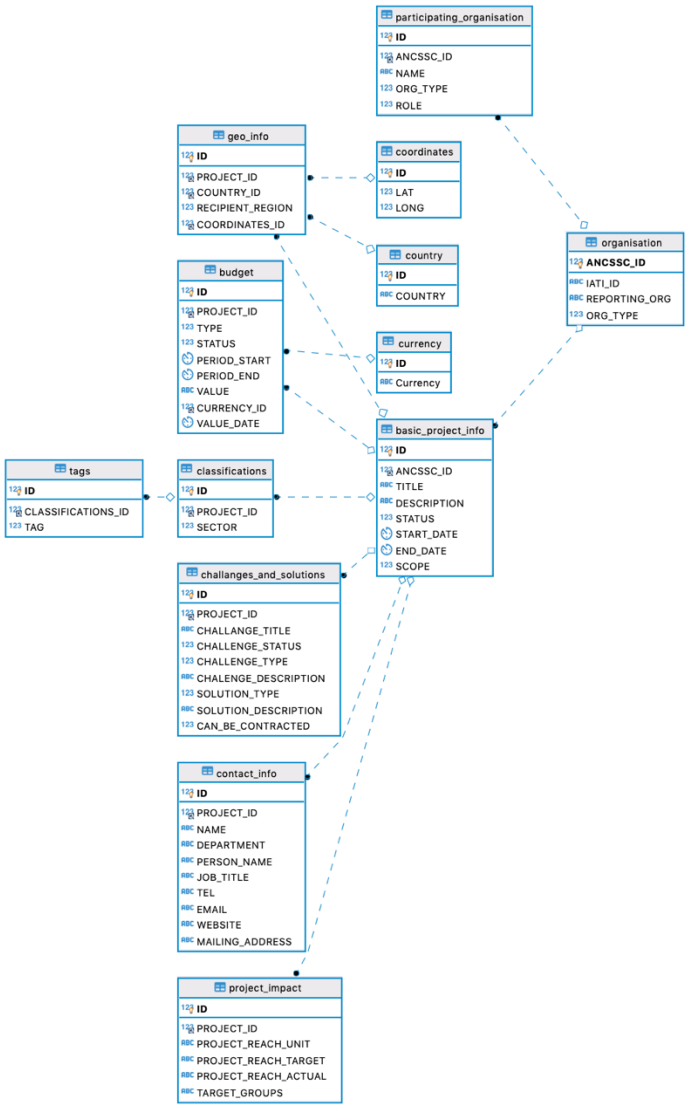
10/02/2020:
After digging several online sources, we found ALBERT (A Lite BERT for Self-Supervised Learning of Language Representations). ALBERT is published by google research team in December last year, so it is very recent. According to SQuAD (The Stanford Question Answering Dataset) ranking, ALBERT is the most powerful BERT algorithm in the world (16/02/2020). Then we decided to implement our BERT function based on ALBERT. However, after spending almost three days understanding how to use the source code they provided, we figured out the code they published is only for evaluating and testing. The code itself doesn't contain any information about how the algorithm works and is poorly documented. The poor documentation is due to this algorithm being published just a month ago. Combining all those conditions, we decided not moving forward with ALBERT and instead focus on other models based on BERT, which were robust and well-documented.

11/02/2020:
The second approach we tried is RoBERTa (A Robustly Optimized BERT Pretraining Approach). RoBERTa is published by a Facebook research team. This algorithm is ranking lower than ALBERT. However, this algorithm still performs 2 percent better than a normal person on reading comprehension task. On another hand, RoBERTa was published earlier than ALBERT, which means the documentation of this algorithm is more detailed comparing to ALBERT. Soon after installing RoBERTa, the code is capable of extracting the embedding layer of the passage. However, this code does not work with a high-level implementation. Due to our tight timeline, we went visit Dr. Pontus (NLP expert) for help again. It is then that he and his PhD students suggested to use the Hugging Face library, a python wrapper for BERT models. This means that we could now use ALBERT.

17/02/2020:
After installing Hugging Face library, we are now able to create our own network for question and answering, so essentially, we can now query any length of text.

24/02/2020:
After negotiating with computer science department technology support, we are allowed to use a dedicated machine to train and test our model.

ER Diagram for ANCSSC Database (Database 1) – as per Matt's guidelines



List of tasks completed and whether project is running on time

We are on track to completing our project.
We have completed:

- ANCSSC database created
    - Based on Matt's guidelines
- Consulted with Dr. Pontus Stenetorp, our NLP expert at High Holborn
    - Meeting 1: Introduced to different methods

        - Chunking
        - BERT models
    - Meeting 2: Introduced to Hugging Face
        - Suggested by his PhD students

Plan for coming 2 weeks

- Implement feedback from master's team
    - On ANCSSC database
    - Will need to keep updating as they update their prototype
- Train different BERT models
    - Using new dedicated machine
    - Testing accuracy of different models
    - Training multiple times
    - Time-consuming task