Natural Language Processing CS Challenge Brain Dump

"Develop a tool to **define human understandable patterns** in sentences which can be used to **extract structured information**. For example, a sentence might have adjectives and **job titles** describing a person. **Examine existing work** to explore how these human understandable rules could be automatically inferred using AI techniques applied to **labelled sentence examples**. For example, using **decision trees**."

PLANNING SECTION

At the meeting with John on 20/02/2020, I explained my initial approach to locating papers which was keyword searching. John explained that while this may yield some useful results, it's possible that there are papers which talk about my area of interest but don't describe it using the same language. Time for a change of plan. My objective changes from "finding the best keywords to use to locate information" to "finding the most prominent/respected experts in the field of Linguistics and NLP, then look at abstracts of their papers to find papers which describe something similar to the end goal of my project: formalising text using rules".

First, I'm going to visit the Stanford website and note the names of any lecturers whose profile talks about NLP. I will repeat this for Washington.

Next, I'll find out what conferences include talks on this topic (right now, nothing I've been to addresses NLP nor formalising text)

Once I've found these, I'll update my learning log with new terms / insight into NLP. For the next submission, the specification requires some kind of new code. In my case, this plan results in nothing tangible so I'm better off improving my how-to guide by adding a new section which is also interactive and concise. Another objective is to act upon the feedback provided at the end of submission 1.

REVIEW OF PLAN FOR MEETING ON 27/02/2020

- This week, I gained lots of insight into the work of Professor Chris Manning https://nlp.stanford.edu/manning/ by watching some of his recorded postgraduate lectures and reading a paper that suggests NLP techniques could be used to aid Computer Vision and by extension, formalising text into graphics. I found the mathematical notation used in his presentations to be extremely challenging to understand. Maybe there's some use in making this easier for new learners to interpret?
- I learned about some cases of syntactic ambiguity by reading about Winograd schema and watching a video by Tom Scott.
- I've also learnt about the work of Santiago Gonzales in Cognizant; using Creative Evolutionary AI Modelling to generate totally new data. https://www.cognizant.com/whitepapers/machine-learning-the-first-salvo-of-the-ai-business-revolution-codex3697.pdf and https://www.youtube.com/watch?v=GtkZ-_zgoEg
- Time management is still quite difficult because the first half of the week after the meeting was dedicated to finalising my SDP project and revising for Databases so I'm effectively submitting 2 days of dedicated CSC work; slightly less than is ideal. Next week I have no deadlines for the other modules so I'll get more time to read papers fully and depending on the feedback about my idea for a formalising text into graphics tool, perhaps lay the foundations of that.
- In the week of research commencing 28/02/2020, I've noted a few other Professors and companies that I've come across in my research so far, namely; Timothy Dozat and OpenAI. I also noted two new Python libraries in the learning log that I am keen to investigate: Theano (optimise mathematical expressions with multidimensional arrays) and Keras (hl neural-network set to provide quick prototyping of models).

PLAN FOR WEEK COMMENCING 28/02/2020

- My first objective is to message Dr. Niloofer Shanavas, PwC's leading AI specialist in Belfast. In the email, I'll explain my goal and ask for advice regarding other people to contact / papers to read which might get me closer to my goal.

Planning for Week 11

- During the meeting in week 10, John encouraged me to dedicate some time to analysing the data structures behind popular common-sense knowledge bases. My first objective is to explore the inner workings of Cyc's Knowedge base in greater detail because I feel like the output of this research could be included on the timeline of key contributions to the field.
- John also suggested to appeal to a broader audience in a similar way to how Computerphile and Two Minute Papers reach technically minded people of all demographics, not just teenagers. Given that my revised target audience is likely the audience of these channels, I will watch some of their videos to understand their presentation style.
- Once I conduct this research, I'll stick to my process of transferring my understanding into documents which can be then accessed via the timeline app.
- I'll revise some Python file handling code; I feel like building an app will maintain and possibly improve my technical skills because I've already worked with HTML and Flask for submission 1. John advised me to consider how my final submission will benefit others; I feel that the target audience will acquire a sufficiently detailed overview of the field

LEARNING LOG

What have I learnt so far? (Up to 19/04/2020)

NLP Theory

- Tokenizing
- Stopwords and when they should be removed
- Stemming and Lemmatizing
- Chunking and Chinking
- Part of Speech Tagging
- Training datasets vs Testing datasets
- Named Entity Recognition with pre-trained SpaCy
- Training a model for NER
- WordNet lists, definitions and similarity scoring
- Sentiment Analysis
- Word Vector Similarity scoring

Miscellaneous Programming Theory

- OS library basic features
- Python-Docx basic features
- Regex

Flask Programming Theory

- Routes
- url_for(): HTML templates
- url_for(): Images
- GET and POST requests
- Using SpaCy NER in a live Flask website

Mathematics

- Naïve Bayes Equation
- Entropy
- Binary Decision Trees

Real-life Implications and Examples of NLP:

- Viv.ai
- SocialMind
- Overcoming the processing of sarcastic/hyperbolic data
- GLUE Benchmark
- Cycorp approaches to formalising knowledge about the world
- Meena Chatbot

Acronyms:

KDD: Knowledge Discovery in Databases

DEFT: Deep Exploration and Filtering of Text

NLTK: Natural Language ToolKit

Definitions (Updated as of 29/01/2020):

The idea of this section is that while I'm reading through documentation, I'll write the word here and afterwards, go looking for an explanation of the word.

Text Mining: The mining process applied on free text to discover actionable insights, useful information and patterns within it.

DEFT: Access implicitly expressed, actionable information within a *corpora*.

Tokenizing: Grouping tokens together. Word Tokenizers separate a *corpora* by word. Sentence Tokenizers separate a *corpora* by sentence.

Lexicon: A grid of words and their various meanings depending on the context in which they are used.

Lemma	Topic	Rating
waiter	service	
waitress	service	
wait		bad
quick		good
.*schnitzel	food	
music	ambience	
loud		bad

Figure 1: Sample Lexicon. Source: Introduction to Sentiment Analysis by Thomas Aglassinger

Stopwords: Words which have little to no effect on the overall meaning of the sentence. These are removed before processing large datasets however, leaving them in while looking at smaller datasets won't hinder your model.

Stemming: Used to normalise similar sentences. For example, the tenses within two similar sentences may not affect their overall meaning. A disadvantage to this process is that sometimes, the results can non-existent words.

Lemmatizing: Grouping together the inflected forms of a word so they can be analysed as a single item, identified by the word's *lemma*.

Lemma: How a word appears in the dictionary.

Training Data: Used to teach the machine to perform a process.

Sample Data: Used to test the machine.

Polysemous: The coexistence of multiple meanings for the same word or phrase.

Disambiguation: Determine the correct category from a set of possible categories; determine the pronunciation of a letter based on what letters surround it, determine the POS of a word based on the contextual words surrounding it, determine where to attach a prepositional phrase based on the set of other phrases, determine the meaning of a polysemous word in a given context.

Segmentation: Determine the correct boundary of a segment from a set of possible boundaries.

Part of Speech Tagging: Training a machine to determine what part of the target language a single word in a *corpora* belongs to; Proper Noun, Possessive, Preposition etc.

Sentiment Analysis:

Chunking: Identify the named entity within a *corpora*, then identifying other words within the corpora that affect that named entity.

Pipeline: Turns text into a Doc object. By default, the processes carried out are tokenization, dependency parsing and entity recognition.

Tuple: an ordered sequence of values.

WEEK 6

Computer Vision: Making a computer understand a digital image or video with the intention of automating what the human visual system can do. An application of this would be supporting a driver or pilot in hazard detection. It could also be used to look for forest fires.

The Mars Rover (NASA) is an example of Computer Vision being used for Space exploration – recognising and identifying pre-learned objects.

Neural Network: A concept of learning from past experiences to update understanding. In AI, a neural network is trained with a range of inputs and given their expected outputs. They then compute the input, compare their output with the expected output and learn to improve the results if required.

Morphology: The study of words, how they are formed and how they relate to other words in the same language.

Word-Vectors: A word is described using a set of decimal numbers and a key of characteristics relate to all words in that word group.

Feed-forward Neural Network: A unidirectional neural network; nodes do not form a cycle. It has three node types: Input, Hidden and Output.

Syntactic Ambiguity: A sentence with a referent ('it') which also has two nouns in it where the referent could be either noun: "The ball went through the table because it was made of *metal/glass*". An AI struggles immensely with this kind of sentence because to classify the referent as either ball or table, it needs to understand material properties.

The Winograd Schema: If a machine can determine an intended referent based on clues from the context of a sentence (especially a syntactically ambiguous one), then it is using reasoning to parse the sentence. Terry Winograd proposed this concept in 1972. A team of researchers at the University of New York have devised a collection of 150 sentences which follow the Winograd Schema; more or less the Turing Test of NLP. For example, *Lily spoke to Donna, breaking her [concentration/silence]. Whose [concentration/silence]?*

https://cs.nyu.edu/faculty/davise/papers/WinogradSchemas/WS.html

GPT2: A text-generating model built by OpenAI. It can accept a prompt and then finish the passage off for you. It has 1.5 billion parameters and outperforms models trained on domain specific datasets such as Wikipedia. You can generate a whole article based on small input sentences!

Language Modelling: An NLP task to probabilistically predict the next word or character of a document.

'Advice Taker': This is the first ever common-sense based program which was proposed to take formalised statements, understand their implications and then draw conclusions based on its understanding of these. Having an awareness of this paper is valuable because it was written when the field of common-sense knowledge was entirely novel and unexplored. Hence, it has an ambitious metric for success which over half a century later is still being pursued.

GLUE and SuperGLUE:

https://gluebenchmark.com/diagnostics#knowledgeandcommonsense

I investigated GLUE because I feel that it's useful to include a current way of evaluating the overall common sense of a system. From a black box perspective, GLUE is an iQ test for intelligent systems. This benchmark was created by researchers at Google (DeepMind), NYU and the Paul Allen Institute.

"Strictly speaking, world knowledge and common sense are required on every level of language understanding, for disambiguating word senses, syntactic structures, anaphora, and more. Our entire suite (and any test of entailment) does test these features to some degree. However, in these categories, we gather examples where the entailment rests not only on correct disambiguation of the sentences, but also application of extra knowledge, whether it is

concrete knowledge about world affairs or more common-sense knowledge about word meanings or social or physical dynamics."

By evaluating an AI's performance at a range of tasks, you are showing that it doesn't fall far into the 'one trick pony' category that most 1st Generation AI are categorised in. By writing about this, the reader will appreciate that the best AI systems will grasp natural language similarly to how humans do; in a general, flexible and robust manner. This relates closely to McCarthy's vision in 1958 (system drawing conclusions based on what it has understood).

It was valuable to research GLUE because not only did I gather insight into an accepted benchmark, I also got access to the GLUE Leaderboard for overall intelligence which is valuable because it shows me what intelligent systems are currently 'high performance' so that I can review their papers this week coming.

BRAIN DUMP

Introduction to Pre-Processing 18/01/2020

Pre-Processing describes how the data is manipulated before it is passed into a Neural Network.

I will use sent_tokenize and word_tokenize from NLTK to group tokens together. Word Tokenizers separate a corpora by word. Sentence Tokenizers separate a corpora by sentence. In nltk, tokenizing will break a corpora up into strings.

Eliminating 'stopwords' improves efficiency of the algorithms that handle the data later on. These words have little to no effect on the overall meaning of the sentence.

Chunking vs Chinking

Chunking uses regular expressions to identify parts of a corpora which satisfy the regex. This is mainly used to group words into 'Noun Phrases'; the idea that you group a noun with words that are related to/affecting it.

Chinking affects a Chunk. You remove a small part of your Chunk that you do not want, then you keep the larger Chunk.

It is likely that I will use both of these processes during the pre-processing of data. I had to consult a regex cheatsheet because I am unfamiliar with the process of defining a regex.

19/01/2020: Researching Ideas

Seeking Training Data

I am still unsure as to the exact dataset that I want to process and extract insight from. I have considered using the Tweepy library to web-scrape recent Tweets which include a keyword; perhaps the name of a politician or location. I feel that this would be most insightful as analysis of a current affair could be carried out to gauge how the public feel about a decision. However, my doubt regarding this style of project is that other developers have done this before. I would need to identify a way of making my project unique. Perhaps I could add in a factor to the sentiment analysis such as the number of interactions with the tweet since it had been posted. Ideally, I would like the machine to interpret these tweets and decide whether it is for or against the motion in question and to explain to me why it came to that decision.

An advantage to using Twitter is the brevity of each tweet which makes them quicker to process as opposed to news articles that would be scraped from news sources. However, using social media to gauge the political climate raises the uncertainty that comes from rumours and speculation being posted, which could influence the machine, especially if words with a high sentiment score are using such as expletives. Exaggerations and sarcasm also pose a challenge for the process of sentiment analysis. Having realised this potential setback, I investigated the approaches that PwC are taking to overcoming it.

PwC Case Study 1: Sarcastic Tweets affecting Elections

Phil Mennie, Social Media Governance Leader from PwC explains in the article linked below, Twitter "hashtag hijacking" is a risk that organisations face on social media. UKIP published "#WhyImVotingUKIP" in an attempt to encourage voters to explain to others why they are voting for this party. However, users who disagreed with those values used the hashtag to poke fun at the party. For a machine, it is highly challenging to distinguish between serious remarks and hyperbolic or sarcastic remarks, given that he struggled to interpret some of them. Phil suggests that analysing a user's previous tweets may indicate whether the tweet in question is sarcasm or not.

The article encouraged me to look beyond the tweet when searching for text to analyse. Considering factors such as the demographics of the user and who they follow may indicate their political persuasion.

https://www.pwc.co.uk/services/risk-assurance/insights/can-sentiment-analysis-spot-sarcasm.html

PwC Case Study 2: SocialMind NLP Tool

According to the Consumer Financial Protection Bureau Supervision and Examination Manual, penalties for unfair or deceptive practices using social media can reach $1,000,000/day. PwC's Analytics services state that a social media presence is not enough to ensure marketing success. How a brand is perceived on social media can either positively or negatively affect the brand's reputation considerably. Regulators have started using social media to detect unlawful practices. Dissatisfied customers on social media can be disastrous for business performance. Their negative experience will deter others from using services from the brand in question.

SocialMind was developed to provide insight to the client about what customers are saying about their services, as well as their competitors'. It listens to all relevant social media data and client-specified websites. Dashboards are compiled after the data has been processed to visualise its results.

I feel motivated to implement elements of this tool in my project such as using additional data to just the text from the Tweet to extract information. Presenting the information in the form of an interactive dashboard makes the processed information more readable.

https://www.pwc.com/gx/en/issues/innovation-technology/analytics/assets/social-mind.pdf

Named Entity Recognition

This process identifies proper nouns such as "John" and "Smith".

WordNet is a lexical database designed by Princeton and implemented in NLTK. It can be used to find the meanings of words, synonyms and antonyms.

If you input a word, WordNet can return its definition. If there are multiple definitions of that word, it will return the first one unless a second argument is entered.

You can print a list of synonyms or antonyms for that word.

WordNet: Similarity

The NLTK library allows you to compare two words and return a decimal of how similar they are to each other. A score towards 1 suggests a synonym. The lower the score, the more likely it is that the other word is an antonym.

Text Mining: Docx

Python has a python-docx library which facilitates the generation and manipulation of .docx documents. It is likely that I will use this library to output a written report after having tested for sentiment analysis, identifying which words appeared most times in positive product reviews and vice versa.

Developer Diary Entry 22/01/2020: Mathematics 1

Following research carried out on *https://pythonprogramming.net*, the first algorithm to investigate is the Naïve Baye classifier. It is often used in text classification. It assumes that the presence of a feature in a class is unrelated to the presence of any other feature. This algorithm doesn't require much processing.

Posterior = Prior Occurences * Likelihood / Evidence

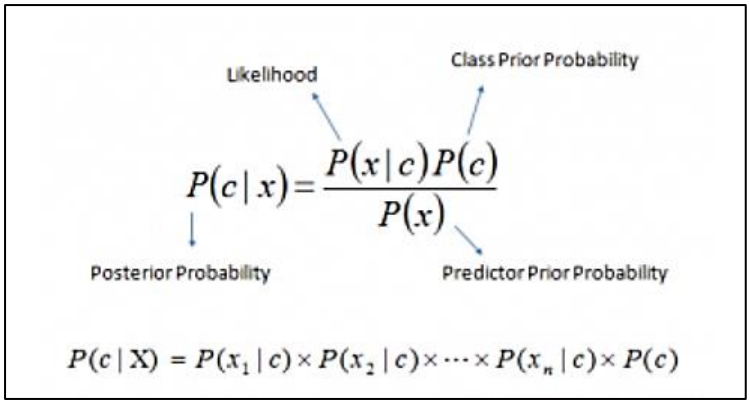


Figure 2: Naives Baye Formula

- $P(c|x)$ is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Training vs Testing

It is good practice to use two datasets (A and B) where A is used as the training set and B is used as the testing set. This eliminates bias.

26/01/2020 Fundamentals of Spacy

Training a model is ideal if I want my tool to generalise based on examples provided. It can prove effective if you are tracking a named entity.

A rule-based approach is more useful if you are can express a pattern with a regex, such as tracking an IP address. Regexes operate on single tokens rather than corporas.

Sentiment Analysis

https://ep2018.europython.eu/conference/talks/introduction-to-sentiment-analysis-with-spacy

Thomas Aglassinger notes that it can be useful to make an enum of whatever factors you are assessing sentiment on eg; for product reviews, factors such as durability, value for money, cost effectiveness, dispatch time etc may be ideal headings.

There are 3 common ways of expressing sentiment; Positive/Negative, Star Rating, Float. The most accurate is float, so I will be using it.

From 0.0 to 1.0, I will use these standards:

RATING	MEANING
0.0	VERY POOR
0.25	POOR
0.5	AVERAGE
0.75	GOOD
1.0	VERY GOOD

26/01/2020 Rule Based Matching Existing Software Case Study 1: Explosion.ai's Matcher

This application lets the user create their own rules and run them against their text. You can specify values, POS tags or Boolean flags. Essentially, it is a high-level abstraction of rule based matching which lets the user construct rules without having to manually code them.

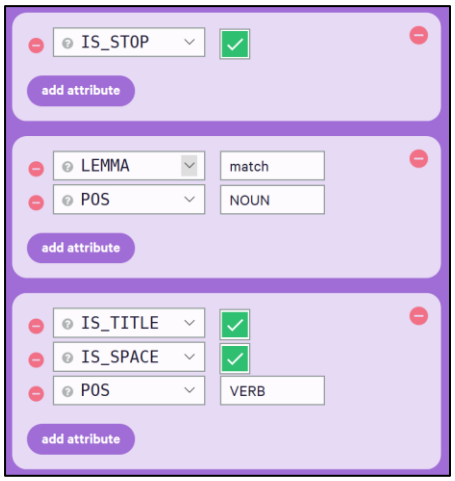


Figure 3: UI for the Rule Creation Tool

```
pattern = [{'IS_STOP': True},
           {'LEMMA': 'match', 'POS': 'NOUN'},
           {'IS_TITLE': True, 'IS_SPACE': True,
'POS': 'VERB'}]
```

Figure 4: Result of the criteria from Figure 3

27/01/2020 University Case Study 1: NLP at Washington University "Decision Trees and NLP: A Case Study in POS Tagging"

Part of the Project Specification requires me to analyse existing work. A case study from the University of Washington identifies a linguistic knowledge acquisition bottleneck which suggests that each slightly new NLP variation requires linguistic knowledge bases to be built from scratch. Hence, "automatically acquired language models" would be very useful since training for all grammatical exceptions and sub-regularities is heavily time consuming if someone was to manually encode it. A Corpus-based approach such as NLTK is an abstraction/black-box system simulated by massive statistical tables. This abstract nature means that some linguistic trends and phenomena could be overlooked. In 1995, Magerman states that most NLP problems are classification problems, hence, a machine learning approach is suitable. There are two types of linguistic problems; **Disambiguation** and Segmentation.

This case study has two parts; POS Disambiguation tasks, as well as Unknown Word Guessing. The target natural language is modern Greek, which hasn't been investigated widely from a computational perspective. Washington recorded a 93-95% performance rate in POS

Disambiguation and 82-88% performance rate of guessing the POS of unknown words. Three tree induction algorithms are used to produce decision trees. The first two produce generalised decision tree while the third produces binary decision trees (see Figure 5 below).

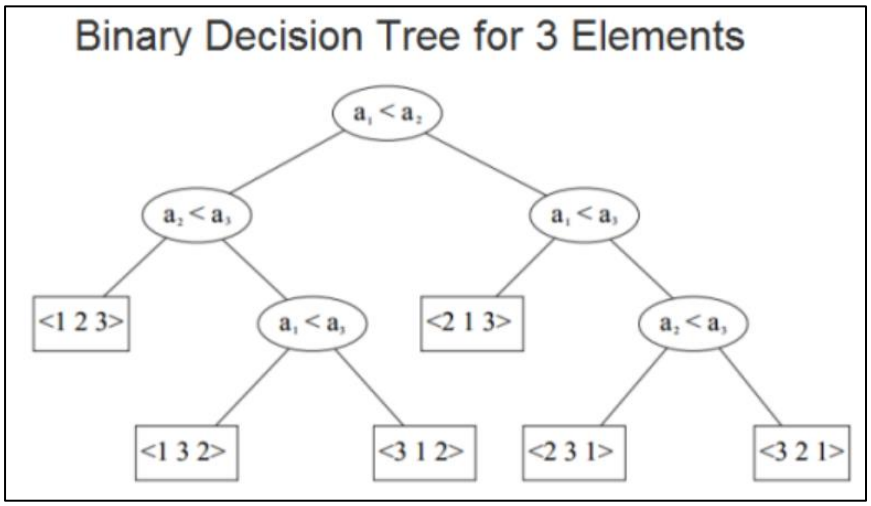


Figure 5: A model Binary Decision Tree

Figure 5 depicts the decisions that are made to identify the order of 3 numbers.

Is Element 1 less than Element 2?

YES: Is Element 2 less than Element 3?

YES: ORDER IS 1, 2, 3

NO: Is Element 1 less than Element 3?

YES: ORDER IS 1, 3, 2

NO: ORDER IS 3, 1, 2

NO: Is Element 1 less than Element 3?

YES: ORDER IS 2, 1, 3

NO: Is Element 2 less than Element 3?

YES: ORDER IS 2, 3, 1

NO: ORDER IS 3, 2, 1

---Analysis of this research paper will continue on 31/01/2020---

28/01/2020 4. Using Flask to make a simple application that runs via a browser (Part 1).

I found a series of videos by Corey Schafer which demonstrate how flask can be applied in the creation of a web application and have closely followed it, given that I have never used this library previously.

https://www.youtube.com/channel/UCCezIgC97PvUuR4_gbFUs5g

Installation:

In Anaconda, type the following:

activate <name_of_virtual_env>

pip install flask

pip install flask-wtf

Figure 6 is the source code for the Hello World flask application. It is worth explaining what each important line of code does.

```
from flask import Flask
app = Flask(__name__)


@app.route('/')
def hello_world():
    return 'Hello, World!'

if __name__ == '__main__':
    app.run()
```

Import command

Instantiate Flask app

Specify behaviour which happens when the '/' route is the URL.

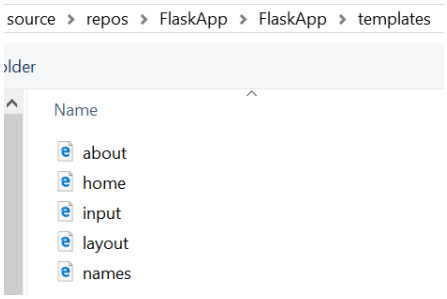
In this case, it just prints "Hello, World" onto the screen.

Run the Flask app

Figure 6: Hello World Flask App

@app.route("/dir_name") makes a new webpage which can display html using the render_template('template_name',args) function.

My editor is VS Code. There was no templates folder which was created automatically as described in several tutorials. Hence, I created one and populated it with templates which were used by the web pages. One of these templates, layout.html was a base template which was extended upon by the others.



Once you have coded or downloaded some templates, you can create a new webpage which calls them in the function for that page's route.

Making a secret key protects against forgery attacks. It is recommended that you use 16 random characters for this. An effective way to generate a security tag is to go to Anaconda and type:

python

import secrets

secrets.token_hex(16)

29/01/2020 University Case Study 2: Decision Trees at Trinity College, Dublin

The resource I am referencing was published by Dr. Kevin Koidl, School of Computer Science and Statistic Trinity College Dublin and can be viewed at https://www.scss.tcd.ie/~koidlk/cs4062/Lecture11DecisionTrees.pdf

This is a summary of the knowledge I have gained from reading Dr. Koidl's notes:

Decision Tree learning is used for inductive inference, the process of reasoning a statement from specific to a generalisation. For example, in a sample, if all the sample cars are black, then there is a hypothesis; "Are all cars black?". You can model any algorithmic decision-making process as a tree. Dr. Koidl reiterates what was stated in case study 1; that there is a linguistic knowledge acquisition bottleneck. Figure 7 describes the parts of any decision tree:

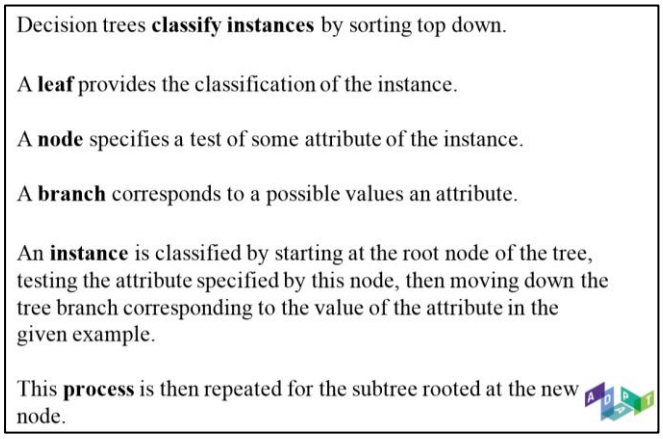


Figure 7: Parts of a Decision Tree

The ID3 Algorithm is works by taking each attribute and evaluating it to see how well it can classify training examples on its own. The attribute which scores the highest is used as the root node. A descendant of the root node is created for each possible value eg; {"Yes", "No"}, {"A","B","C"}. Training examples are placed at their most suitable node. You repeat the process using the training examples at each point in the tree to determine what attribute should be tested at that point. The algorithm never backtracks/reconsiders so it is called 'Greedy search'. "Information gain measures how well a given attribute separates the training examples according to their target classification." This is the best way to quantify the worth of an attribute.

$$Entropy(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

BRAIN DUMP/DAILY DIARY

Current Goals for the week 26/01/2020 – 02/02/2020:

1. ~~Identify the target data I will process with my tool~~
2. ~~Fix teething issues with spaCy and Power BI* (Get the English language toolkit working and acquire a licence for Power BI)~~
3. **~~Begin learning about applying sentiment analysis~~**
4. Investigate how to use Flask and SqlLite for making a Webserver.
5. Learn the fundamentals of Tweepy or Scrapy
6. *Carry out research using various academic sources to learn more about decision trees.*

27/01/2020 Overcome a Challenge 1: Spacy's en_core_web_sm platform not linking

Source of challenge: The platform was being downloaded to the Python 3.8 environment but the Spacy package is stored in the 'spacy' environment. In the admin console, these are the steps to resolve the issue.

(base) C:\Windows\system32>activate spacy

(spacy) C:\Windows\system32>C:

(spacy) C:\Windows\system32>cd C:

C:\Windows\System32

(spacy) C:\Windows\system32>cd ..

(spacy) C:\Windows>cd ..

(spacy) C:\>cd Users

(spacy) C:\Users>dir

(spacy) C:\Users\rjdpo>python -m spacy download en_core_web_sm

Result:

Requirement already satisfied: en_core_web_sm==2.0.0 from https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.0.0/en_core_web_sm-2.0.0.tar.gz#egg=en_core_web_sm==2.0.0 in c:\users\rjdpo\miniconda3\envs\spacy\lib\site-packages (2.0.0)

Linking successful

C:\Users\rjdpo\Miniconda3\envs\spacy\lib\site-packages\en_core_web_sm

-->

C:\Users\rjdpo\Miniconda3\envs\spacy\lib\site-packages\spacy\data\en_core_web_sm

You can now load the model via spacy.load('en_core_web_sm')

31/01/2020: The How-to Guide

There are several high-quality SpaCy guides available which cover the theory behind the library in detail. It would be unrealistic to improve the syntax explanations provided on the SpaCy website. To produce an alternative guide which is also useful to someone looking to learn about Rule-Based matching, I feel that it would be beneficial to **combine**, then **condense** the information I recorded about this process, then produce a concise series of steps with **troubleshooting**. Interactive Learning is another area that I am interested in. How-to guides which contain only text, screenshots, links and videos do not captivate a beginner as much as ones which let the user try out what they have learnt. SpaCy's official site achieves this through editable code snippets, Explosion.ai. I have under 10 days left to produce this how-to guide so while I would like to implement some form of quality user interaction, this may not be achieved within the time constraint especially since I will need to learn how to link displaCy or an NLTK drawing tool to Flask. This weekend, I will check StackOverflow to see if anyone has done this before. I'm unsure as to how full my troubleshooting page will be.

During my research, I came across interesting research articles from Washington University, PwC and other sources which could benefit someone who now understands the fundamentals of NLP and would like to see them applied in industry/research.

http://faculty.washington.edu/fxia/courses/LING572/decison_tree99.pdf

https://www.pwc.co.uk/services/risk-assurance/insights/can-sentiment-analysis-spot-sarcasm.html

In terms of the HTML page that my 'How-to' guide is being published on, I intend for it to be of reasonably good quality, given that I have just under one week to produce it. I am not overly concerned about struggling to finish this webpage at a high standard, having learnt all the necessary Flask and produced material to publish in the past week. As stated above, a key feature of this webpage will be troubleshooting, something that lots of the other

tutorials which focus more on teaching the theory omit. I am going to showcase lots of what SpaCy is capable of, but also some examples of the shortcomings of NLP such as dealing with homographs and informal text (eg; tweets).

I have researched the format for a good quality 'how-to' guide on University of Bath's Digital Marketing & Communications guidance portal linked below; https://www.bath.ac.uk/guides/creating-a-how-to-guide/

01/02/2020 Overcoming a Challenge 2: Images not loading onto my Flask website.

When designing the front-end for the webpage, locally stored images weren't loading properly. I tried various approaches to solving this problem; loading the images from a file hosting website, from Google Drive, and from inside the 'templates' folder. Eventually, I realised that images needed passed into the HTML document via 'url_for' from the Flask library. I created a 'static' folder and set the app's url_path to it. This resolved the problem. It was frustrating trying to work out if the images not loading was a directories problem, a HTML problem or a Flask problem. However, this means that in future, I know exactly how to approach this.

01/02/2020 Designing Tutorial 1

As stated previously, the tutorials which cover the theory behind rule-based matching cannot be improved upon in the sense that the definitions are official. However, what I can do is provide example test cases of data which helps a novice user of SpaCy understand HOW and WHY input data causes matches for a given token. I have also included the pre-loaded URL links to the official SpaCy Matcher from Explosion.Ai, encouraging them to tweak the attributes inside the token(s) to see what the effect is.

The most useful tutorials that I learnt from all started out simple and got progressively more difficult with each page/example; a trend that I intend to replicate in the design of this website. There is no point starting off with an explanation of complex cases involving large texts compared against multiple multi-attribute tokens because that will overwhelm them, discouraging them from trying to learn. However, it is also important to make the final examples noticeably more technical than the starting ones.

I feel comfortable designing and explaining test cases which include all the attributes that a token can assume, and which apply tokens to texts which gradually increase in length and complexity.

One decision that I will make tomorrow is whether to include the shortcomings of Rule-based matching on this Tutorial Page, or on Troubleshooting which will be designed on 04/02/2020.

02/02/2020 Creating the code to be referenced in Tutorial 1

Today I created an application called WebsiteTutorial1.py which aims to teach users how Rule-Based Matching works by showing it working forwards (passing pre-written text into the matcher to see if any of its text satisfies a pre-written rule) and backwards (showing the user a rule and getting them to enter text which satisfies it). None of the tutorials I used to learn Rule-Based Matching used an approach to teaching like this, despite how useful and hands-on I feel it is. This way, the user is introduced to Rule-Based Matching with example cases which get gradually more complex and then can test their understanding.

Having read the official SpaCy tutorial, as well as several Rule-Based Matching tutorials on YouTube cited below, I decided that a code walkthrough would also be useful for the

reader, especially if the source code was linked for them to edit. I replicated this by hosting WebsiteTutorial1.py on GitHub. On the website, I will annotate the code screenshots to describe the SpaCy features such as the arguments of matcher.add().

In my solution, I also used displaCy as my visualisation tool. Given how useful this feature is, my tutorial will show how it can be implemented in the main "check_for_matches" method to show how each token affects the others. Not all of the tutorials mentioned that you can edit the appearance of the output by passing in an options argument so I will address this and show the effects of tweaking these options. For example, you can enhance the appearance of the visualisation by changing the font.

It is also useful to have displaCy create a separate diagram for each sentence. This is done by using the .sents property of the target text to create a list of sentences. In tutorial 2, I will show the effect of using a different style of diagram.

03/02/2020 Creating the code to be referenced in tutorial 2

Named Entity Recognition is the other area of NLP which I feel that I can provide a useful tutorial about. Once again, it requires the SpaCy library as well as the 'en_core_web_sm' vocabulary. I intend to make a program to teach the reader about the various types of Proper Noun that NER can be used for; Locations, Money, People, Artwork etc. because I am currently working on improving my own understanding about this. Since I have shown a limitation of Rule-Based Matching, it is also useful to show the user a test case which makes it difficult for NER to categorise a word. According to towardsdatascience.com, SpaCy's NER engine was trained on OntoNote5.

My test case for identifying a shortcoming of NER is 'Turkey' (Country) vs 'Turkey' (Food).

"Turkey tastes amazing at Christmas!" returns GPE for Turkey which is incorrect as in this context, it isn't referring to a country. This test case justifies a need for Rule-Based Entity recognition.

Another obscure test case is people with places for names; India is a girl's name as well as a country. Austin is a city in Texas, as well as a boy's name. Other examples are Brooklyn and Adelaide. None of the tutorials described how to go about distinguishing between semantically identical tokens in entity recognition so it would be beneficial to readers if my website addressed this challenge.

However, I am concerned that I will run out of time to implement an Entity Ruler example which solves this, as well as being able to contextualise my approach in the form of a tutorial given that the deadline is in less than 100 hours. Once I finish writing tutorial 1, my focus will shift to making tutorial 2 as engaging as possible. Fortunately, having read about Flask from the start of the project, I am more comfortable in producing a webpage given that the process for tutorial 2 is a replication of tutorial 1 and uses the same layout, but will have a different kind of interactive feature. Therefore, it is a matter of balancing programming the short tutorial application and outlining the steps for others to reproduce it. I intend to overcome this lack of time by dedicating time that would have been spent on improving the aesthetic of my frontend to reading up on possible approaches to creating a function that returns the correct answer for these obscure test cases. I reckon there are two potential approaches to training this tool to identify the correct noun type that these words belong to;

1. Train for the common cases: Create a rule that if the word after 'India' is a name, deduce that India is also a name in this context.

2. Implement another library; NLU, a subsidiary of NLP is about structuring the input so that the machine can act upon it. RASA feels like a good option, given that you can perform intent detection. Intent detection would be perfect in this scenario. For example, "I want to buy a Chinese." returns highly likely that the writer is talking about food as opposed to the nationality.
   Unfortunately, RASA server is linked to TensorFlow which would not install for me even after creating a virtual environment for Python which was an older version (3.6). I will need to spend time playing around with Conda to install this package. Given that it's not necessary for the How-to guide, I decided to postpone its installation until next weekend. This was quite frustrating but hopefully I can get it working if needs be.

I watched a YouTube tutorial on training SpaCy for NER but when I was following along, there was a mistake in the code where 'model = None' was showing an 'Invalid Syntax' message despite me doing exactly what he did. I read the comments, thinking "have other people also got stuck on this?" and there are in fact, two other people also couldn't solve it. So I thought that for my tutorial, if I can make a trainer which runs perfectly with no syntax errors, I've made something which benefits at the very least, two other people. Here is a link to the GitHub repository containing that code from the tutorial which doesn't work in Visual Studio:

https://github.com/Jcharis/Natural-Language-Processing-Tutorials/blob/master/Training%20the%20Named%20Entity%20Recognizer%20in%20SpaCy.ipynb

The error appears here:

```
## plac is wrapper for argparser
@plac.annotations(
    model=("Model name. Defaults to blank 'en' model.", "option", "m", str),
    output_dir=("C:\\Users\This PC\Documents\JLabs\JFlow", "option", "o", Path),
    n_iter=("Number of training iterations", "option", "n", int))

# Define our variables
model = None
output_dir=Path("C:\\Users\\This PC\\Documents\\JLabs\\JFlow")
n_iter=100
```

'Unexpected Token: model'

I thought to myself; "That's got nothing to do with model and must be caused by the plac annotations. I moved those below the variables and the if statement had the exact same error message. For my tutorial, I can exclude plac because I'm not using a notebook style of command prompt and that code has no effect on the NLP code. This will fix the problem. Other people in the comments complained that their model was performing poorly. I wouldn't attribute that to the code, but rather poor-quality training data or them incorrectly labelling where the named entity is, which is easily done. I was manually going through each sentence to find the start and end position of the 'Ronan' which took a while at the start.

Follow-up

This is my meta.json for the model after running it 10 times. I noticed that with each consecutive run, it made less mistakes. By the 7th run, it correctly identified 'Ronan' 100% of the time which is very exciting as SpaCy's existing library never picks up my forename correctly.

```
"accuracy":{
  "token_acc":99.8698372794,
  "ents_p":84.9664503965,
  "ents_r":85.6312524451,
  "uas":91.7237657538,
  "tags_acc":97.0403350292,
  "ents_f":85.2975560875,
  "las":89.800872413
},
```


I've searched around for an answer as to what some of the other fields in accuracy mean but I haven't found anything so I'll post in the NLP Facebook Group next week so that an expert in the field can explain it to me.

03/02/2020 Interactive tool for teaching NER – Writing to displaCy and receiving a displaCy diagram in a new page

I will implement a form on the website which accepts text as an input, then outputs all NER tokens beside their NER type. Much like how the official SpaCy tutorials encourage readers to interact with the site, I feel that interactive learning will make the user understand the theory in a more applied sense. This is achieved by importing the requests tool from Flask and using it to manage GET and POST requests that involve a text box containing the target text.

04/02/2020 Implementing the tool conceptualised on 03/02/2020.

I created the input box on the Tutorial2 form for the user to enter their text. I didn't dwell too much on the appearance of the form; I am working towards a rapidly approaching deadline so demonstrating the results of my way of thinking seems more important. My main challenge today was finding out how show a displaCy render through Flask. The first step is generating the NER render which is achieved through the render = 'displacy.render(text, style='ent')', then passing render as an argument in the render_template() for the page that it will be displayed on. The text from the user is sent to this page via a POST request. Once I got familiar with this process, I passed in the get_ents() method from the Tutorial code but instead of returning the entities, I returned the count of entities to add more material to the blank looking results page. If I have time, I'm going to investigate counting the number of one entity which appears in the text. I think this can be achieved by using the collections library. However, that seems quite complicated as there are so many NER types. after reading the examples of test cases that demonstrate SpaCy NER performing well and in reverse, performing poorly.

05/02/2020 Training SpaCy to Recognise a new Named Entity

When creating the test cases, I discovered that SpaCy does not recognise 'Ronan' as a name and instead classifies it as an 'Organisation' which got me thinking that it may not be trained to deal with Irish names. It may be useful to learn how to train SpaCy in recognising names which it currently classifies incorrectly.

I watched a tutorial on training SpaCy and reproduced it with some alterations to the structure of the code, making it more readable for a novice user and improving my understanding of how it works. In order to be able to remove code from a project, you need to know if this will affect it. The tutorial used a very small dataset and I wanted to train something slightly bigger, given that I had a bit of extra time. I wrote my own testing and

training datasets and ran with 100 shuffles and a dropout of 0.3. Those concepts will be explained on my website in the NER tutorial.

What have I learnt this week?

Identified a problem while writing the code to demonstrate SpaCy NER in my tutorial: There are Irish names which SpaCy doesn't recognise. To manage this, I learned how to train SpaCy to classify a new word into an NER category.

How to host a Flask website which allows the user to enter text, then to be able to view the named entities from pre-trained SpaCy.

How to structure a tutorial to make it more interactive and digestible. This was done by re-watching existing tutorials several times, reviewing the comments left by other users, then acting upon those comments by making my tutorial meet their needs.

How far do I feel I've come this week?

Last week, I had doubts about the feasibility of the project. However, once those doubts were set aside, I felt much more motivated to produce a tutorial covering and making use of lots of the technical skills that I've gained in the process so far. Going from minimal HTML knowledge after coming out of the previous meeting to being able to produce a website that I am proud of feels quite rewarding. I found the error messages produced by JinJa to be pleasantly specific for markup syntax errors. I came across a few challenges when uploading images to the website directly from my laptop. It turns out that some IDEs automatically create the 'templates' and 'static' folders that you are certain to use when developing a website with Flask however, Visual Studio did not do this for me and it took a bit of research because the StackOverflow solution referred to 'static' which I presumed was included in the project folder. I spent around 30-45 minutes looking at fixes for 'HTML images not loading' which all suggested I was using the wrong file path. I only realised afterwards that it was a Flask problem and not a HTML problem. Given that this was a challenge that I had to overcome, I've described my approach in the troubleshooting section of the website.

06/02/2020

I felt the meeting with John was very informative today as he gave me guidance about restructuring my diary to be less formal (make it more like a brain dump) by creating two other documents that run alongside it which will cover the technical skills I am acquiring as well as a 'plan' for the project. From this point forward, this diary will serve as a brain dump for my feelings and from Monday onward, I'll have two new documents where I'll record technical progress. However, he reiterated that the end goal of the module is unrealistic (Everest) and that the creation of something of lasting value, even if that's just making a quality tutorial about an area of NLP or making something small yet useful that others can build upon. John also brought up the point that another student is creating a tool to generate movie scripts so it might be interesting to have him generate a script and me extract information from it. I'm going to improve the appearance of my How-to guide tonight because John said that unlike the diary, it needs to look attractive to retain a user's attention. I didn't make much of an effort in this regard so tonight, it will be worth adding some more visual features. He added that although the website will be assessed, more importance is placed on having a continuously updated diary that describes why you've done something, not just what you've done.

I am becoming more accustomed to working with libraries that I've never used before. When John described the course as 'sympathetic', it made me realise that it is much more

important to throw myself into the areas that I'm looking at currently than to stick to what I've done before starting this module because if I developed something linear, I wouldn't run into the learning challenges that I'm currently facing and then what's the point in the module being called 'Computer Science Challenges' if there are none? Up to now, the process nearly feels like I'm training myself as if I were an NLU AI; the "training data" was all the tutorials I read about these areas of Computing that I previously knew next to nothing about, the "training process" was writing my own projects, applying what I had learnt and now, the "testing data" is being able to explain what I've learnt to a novice user. I genuinely have seen parallels between me learning about NLP and an AI learning about something new; both of us are going to make lots of mistakes at the start and the number of mistakes made gradually decreases as we understand the material that we have taught ourselves. Going forward, I want to read even more research articles about my project brief.

07/02/2020

I'm worried that when the other projects start up for the other modules, that I'll find myself conflicted in terms of the amount of time to spend at each project. This module is the most demanding, however, Software Design Principles is team-based so if I dedicate time that's meant for SDP to CSC, then the mark of the whole team could suffer. Thankfully, I've done lots of work with databases in the past so there are very few new skills I'll be acquiring from that module. I would say that in the past week, my work ratio has been 60:30:10; CSC:SDP:Databases. Next week, I'm going to try to make it 55:35:10.

GOALS FOR THE WEEK

- Celebrate and reflect upon the result attained in Submission 1.
- Read some of 'Thinking Machines by Luke Dormehl'.
- Focus on consolidating work in other modules with close deadlines.

11/02/2020

In my feedback provided by John, he suggested that I focus more on understanding the datastructures in NLP and that I should study the goal of the project more closely prior to productising it. John identified a lack of research into formalising what sentences mean. I've got some new sites to read up on before the next meeting; codepen – it lets you embed editable code in a webpage which could prove very useful since the SpaCy website actually avails of it.

12/02/2020

Until the meeting on 13th February, I'm not going to do any CS Challenges theory and instead, focus on gaining back the time lost in my other modules which have deadlines approaching. In project management, I consider today until my meeting with John a float period. In my leisure time, I'm reading 'Thinking Machines" to gain an insight into the foundations of AI and its pioneers.

16/02/2020 and 17/02/2020

Today I began using Google Scholars to search for papers which might contain information about Formalising bodies of text to extract information from them. My first search criteria was 'Rule-based matching' and I selected a paper called "RULE-BASED DATABASE OBJECT MATCHING WITH COMPARISON CERTAINTY". Before reading, I knew that this paper may not point me in the direction I would like to go. However, it would serve as a good foundation for learning **how to** read a paper, which would then help me condense

information found in a more relevant one. My first impression as a student is that this paper is not very concise; the publishers use language which makes it difficult to summarise the key points of their diagram. In my previous meeting, John pointed out that publishers may write discursively and with a certain 'arrogance' to disguise any shortcomings in their paper which I feel can be very off-putting for students like myself, who are used to explanations being far less long-winded. It also contains information tables that I find basic as a year 1 student such as an explanation of AND, OR and NOT. In summary, the paper describes how manual certainty tests can be applied, require user verification, then be stored to be used again without user verification (automatically). It also encourages the development of a text-entry/check-box GUI for the user to define rules with ease. This is very similar to what SpaCy achieved on their website with their matcher. It also describes how to prioritise rules; if you have a high, medium and low priority rule, a certain match might be true for any number of these. It also describes the source-field modifiers which are simply String functions like substring. My learning log did not benefit from any new definitions gained from this paper given that from a critical perspective, this paper almost describes the steps to reproduce SpaCy's rule-based matcher. I need to be aware of the time difference between this paper being published and this tool being released because if the paper came first, then it's likely that SpaCy implemented some of the ideas described in it.

In terms of my feelings this week, I am conscious of balancing my time between searching for research papers, an external Spanish qualification I'm completing to improve my CV and helping my team with the Software Design Principles project. If I spend more time on CompSci Challenges than SDP this week, I'm worried that my team's mark will suffer. This term has really placed me in a somewhat 'Project Manager' position where I have to oversee and directly affect the success of two projects. Hence, I have to use my past judgement to prioritise deadlines and this week, I felt it was more valuable to learn how to find research papers as opposed to diving into reading two of them. Who better to speak to about this than someone on my course who has already finished a Physics degree. He agreed with me that a good approach would be to look for highly cited papers as these tend to be more reliable. Once you have found these, it's likely they'll contain terms you're unfamiliar with so consulting a source such as the BCS Glossary of Computing may provide you with an explanation. I searched for 'Formalising text AI'. The first result is called 'Formalisation method for the text expressed knowledge' which aligns appropriately with the goal of this research project. However, you need to pay a fee to download the PDF:

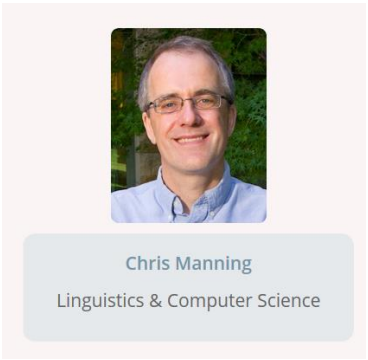
https://www.sciencedirect.com/science/article/pii/S0957417414001353

It only has 18 citations. I found it being hosted on a Croatian website as well, however once again, I would need to pay to access it which is annoying since the abstract would suggest that this paper is more discursive than others; talking not just about syntactical analysis but also current limitations of their method. The limitations are nearly of more interest to me that their current approach because I can then search for other papers and see if other researchers have overcome them; if not, why not? It's frustrating that I can't get access to this paper because none of the others from that search results set feel as relevant to my goal. In the next week, I will start comparing papers which involve formalising text/natural language understanding. However, I'm expecting this to be challenging given that they could be written with two different purposes.

24/02/2020

With the SDP project submitted 1 minute before the deadline and the Databases test out of the way, I can finally focus fully on CSC once again. In accordance with my plan, my first

step today was to visit the Stanford NLP Group website and look for people who specialise in NLP and Linguistics in the hope that they've got some useful information. Luckily, they have a people section which is always a good place to start.



The first person to research is Professor Manning whose specialisms are Linguistics and CS. He is the Stanford Artificial Intelligence Laboratory Director. "His research goal is computers that can intelligently process, understand, and generate human language material."

https://nlp.stanford.edu/pubs/snli_paper.pdf

https://arxiv.org/pdf/1612.08083.pdf

This goal sounds relatively similar to ours so it will be interesting to see what research he has carried out. His bio says he is a leader in applying Deep Learning to NLP. My first potential 'jackpot' may have just come in the form of a list of his papers. From this list of papers, I was able to get the name of a conference on Deep Learning: ICLR

At ICLM 2011, Prof. Manning published a paper called "Parsing Natural Scenes and Natural Language with Recursive Neural Networks" which starts by proposing that NLP ideas could be applied to Computer Vision (see Learning Log Submission 2); using neural networks (see LLS2). I'm not familiar with neural networks so I did some research into them. It turns out that the process I described in my website tutorial is actually very similar to training a neural network; SpaCy gradually improved as it was shown more sample data to identify a named entity from. This paper addresses Computer Vision and Scene Understanding more than NLU so I think it's best to look for another paper co-written by Professor Manning. I visited semanticscholar.org and found a paper titled "Better Word Representations with Recursive Neural Networks for Morphology". From the abstract, it's clear that he has identified a problem with existing Word-vector (See LLS2) approaches; complex/unseen words are represented crudely.

I'm also watching Professor Manning's NLP with Deep Learning course in which he described word-vectors as a 'mined-out' area of NLP; the topic area has been more-or-less solved. Despite this, it's useful for me to know how word-vectors work for my how-to guide. I'll be able to extend my current tutorial by making a webpage that explains visually how word-vectors work, then letting the user

This week, I've been looking into the career progression of Santiago Gonzales, someone who at age 14 soared to internet fame for his passion for programming. Now, some 10 years later, he is the poster boy for a somewhat under the radar technology outsourcing / data migration/ content moderation company called Cognizant who have been around for around 25 years. This company hasn't got the best reputation because of an offshore H1-

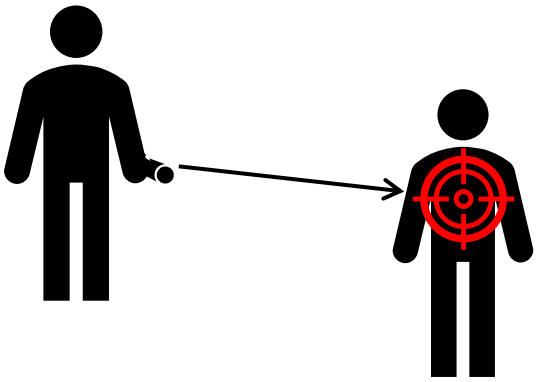
B visa scandal, as well as the treatment of their moderators. Nevertheless, I found that what Santiago is doing is a potential application of the tool that's my Everest. He basically wants to say to a computer, "Show me a horse" and for the computer to GENERATE a new picture of a horse. Hence, there will have been pre-defined rules for characteristics of a horse which is where my Everest comes in – making rules to formalise text eg; for the machine to understand that the phrases "the horse galloped over the fence" refers to a horse and a fence, the horse is in motion, the horse is positioned above the fence and once it knows what a horse is and what a fence is, it can then produce its own image of this sentence which has no human bias in the sense that we may assume that the fence is a certain colour or make other assumptions about the image which is to be drawn such as there being grass which wasn't described in the sentence. This elimination of a subconscious bias is a massive reason why formalising text to then generate images/other media could prove very useful in the context of legal scripts where making and drawing conclusions upon generalisations about a situation.

https://quantumstat.com/word-representation-in-natural-language-processing-prima-parte/

I propose something like this

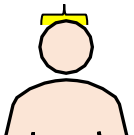
Take a sentence like "The **criminal fired** the **gun at** the **civilian**."

Initially, the tool could produce something like this.



That's not very specific; which is how it should be. From that description, there are 5 words we need to capture "criminal fired gun at civilian". We can then formalise this with rules, for example;

- "Criminal" and "Civilian" are both humans and we have no further description of characteristics such as gender, height, clothing, skin colour. The first rule I would teach the model is if you recognise a human, ask for clarification in the next phase of the image generation.
- Phase 1 essentially consists of tagging all the nouns, then drawing a template of each.
- In phase 2, the tool could go through each named entity and ask for characteristics which are set out in a rule; for example, the properties of an object that have been pre-defined.
- Phase 3 would consist of looking at interactions between the objects in the sentence and verbs / adjectives that are connected to them (dependency parsing)



"Criminal" is a human. Go through human characteristics and ask one-by-one for features

- Skin colour (**Light** ----- Middle ---- Dark ---- Unsure)
- Hair colour (Dark ----- Ginger ----- **Blonde** ---- Unsure)
- Gender (**Male** ----- Unsure ---- Female)

28/02/2020

With the deadline a week away, I'm now concentrating all my effort on this project. This morning, I inquired with the Head of PwC Operate, Andrew Jordan who the best contact in PwC Belfast is with regards to NLP and he told me that Dr. Niloofer Shanavas comes from a research background and is now working for their department to work on projects such as developing a tool which takes a document and checks that it has been hand-signed at the bottom. This saves clients a considerable amount of time when it is used repeatedly. I sent her an email (shown below) explaining the difficulty that I'm facing with coming up with a lack of papers which specifically address the research goal and this has clearly interested her, since she has asked to have a video conference with me after 3rd March. I would be keen to have this call before 6th March so that I can include what was discussed in my diary, as well as being able to read the papers she recommends.

Hi Ronan,

I'm glad to know that you are interested in NLP. I will schedule some time in my calendar to discuss about this. Please let me know your convenient time for a call (after 3rd March) .

Regards,

Niloofer Shanavas
PwC | Operate
Mobile: (+44) 7483301342
Email: niloofer.shanavas@pwc.com
PricewaterhouseCoopers LLP
Waterfront Plaza, 8 Laganbank Road, Belfast, BT1 3LR
www.pwc.co.uk

I came across a paper by chance on Google Scholars by typing 'English language into data structures' that specifically addresses my challenge in the introduction!

"Will a computer program ever be able to convert a piece of English text into a programmer friendly data structure that describes the meaning of the natural language text? Unfortunately, no consensus has emerged about the form or the existence of such a data structure. Until such fundamental Articial Intelligence problems are resolved, computer scientists must settle for the reduced objective of extracting simpler representations that describe limited aspects of the textual information."

This paper has been cited in over 5500 others so there are bound to be some of them which investigate what we're looking for! To condense this, I've done a keyword search for 'English

language into data-structure' as my first step to see if any papers have used that term specifically. I came across one paper.

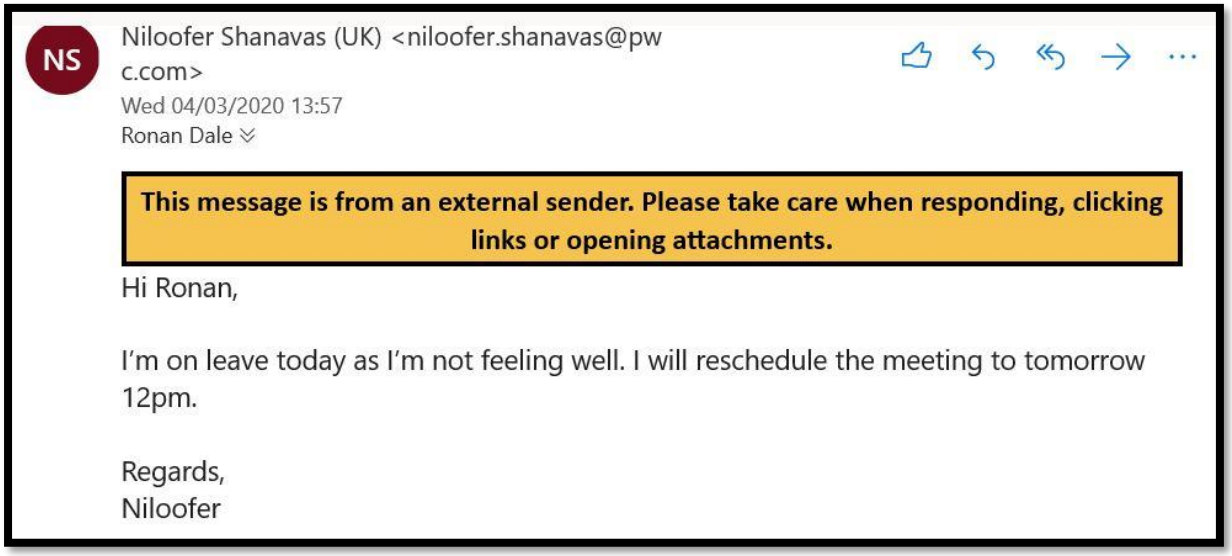
I head a research group (mostly) on computational semantics. That is, we are interested in developing software that enables computers to reason about the meaning of words and sentences. – Sebastián Pado

Approaching the blog post

John stated that the blog is different from the how-to guide because instead of helping the user do something, it encourages the user to take interest in my research. Following on from my approach to the how-to guide, I want to make this blog unique and include perspective from Dr. Niloofer Shanavas

https://pure.ulster.ac.uk/en/publications/supervised-graph-based-term-weighting-scheme-for-effective-text-c-3

If Computer vision is considered the inverse of Computer graphics and Formalising text into a data structure has no existing inverse, I feel that it could be valuable to create a blog which explains what makes formalising shapes in Computer vision



Dr. Krishna Gummadi from Training the future Unbiased AI

https://people.mpi-sws.org/~gummadi/papers/speicher2018a.pdf

Tomas Mikalov from Google

https://cs.stanford.edu/~quocle/paragraph_vector.pdf

Evidence that some journals classify semantic representation as a psychology topic???

https://www.sciencedirect.com/topics/psychology/semantic-representation

RANLP Keynote Speakers and their bio

http://lml.bas.bg/ranlp2019/invited.php

06/03/2020: A retrospective

How do I feel about this deadline? In one word: iffy. I wish it was one week later because the latter half of this week ended up being a logistical nightmare with Dr. Shanavas being sick and

the QCS Formal being on Thursday and it severely affected my vision for the format of the blog. I'd say the entirety of February felt like it lasted 7 days. This year's Turing Talk ended up being about digital twins in a somewhat geographical context which meant that neither speaker could have added any value to my blog. Having said that, I feel like my awareness of the end goal is so much less blurry than it was this time last month and my general knowledge about Computational Linguistics continues to expand.

I feel like I stuck well to the format of the planning and learning log documents but that my brain dump leaves a lot to be desired simply because I prioritised work for other modules over compressing it. I'll be honest and say that there is material in here about some of my figures of interest like Santiago Gonzales that I simply didn't have the time to transfer over to the other documents. I feel like my blog isn't presented to the standard I would usually submit it in. However, that's not to say I don't think the material it contains won't be useful to someone else starting out in Computational Linguistics because lots of the papers, people and videos took a long time to find. If I had more time, I would have included a perspective on RoughWorld in John's github and why D&D is the perfect game for a truly intelligent machine to master in the form of a Dungeon Master given that they need to not only have the cunning/forward thinking skills of a chess player but also the creative mindset that comes with a game with unlimited possibilities for storylines. After seeing how quickly February went in, I'd say there's no better AI than one that can put into perspective how quickly hours turn into days. I can see myself working on RoughWorld during the Development week.

29/03/2020 Analysis of Criticism of Commonsense Knowledge AI

On YouTube on the Doug Lenat TEDx Talk, a user commented a critique of this field rooted in opinion rather than hard research but I think it's useful to look at this to get a better understanding of a potential audience of people who are apprehensive about the prospect of a machine being able to reason.

> "Everybody desperately wants to fly but doesn't enjoy the small baby steps, the millions of mistakes that are needed to learn how not to fly are just as important as the successes that do. Thomas Edison stated that every millionth mistake brought him closer to creating a working light bulb. Cyc is another building block to help us achieve general intelligence. It might not be the end all be all nor does it have to. Secondly, I just want the answer to the question I asked Siri, my digital AI assistant. As long as she answered my question successfully, I really don't care if Siri is truly intelligent or alive or has a digital soul BS. Just like today's calculators, it helps solves complex problems in a quarter of the time it would have taken me if I did it all by hand etc. today's planes don't need to simulate a bird's brain not flap its wings in complex mechanical duplications. Just be successful at flying & achieve the intended goals asked. understanding the brain is nice but for now won't solve everyday questions. Looking at my Beta Fish swimming around his aquarium now, his brain is probably more complex than today's supercomputer but I can't ask my Beta Fish who was the US President when Obama was a child either. I can't even ask my betta Fish advanced brain what today's date is nor what 2 + 2 equals, right? Even if we achieved to reach the equivalent of a human baby's complex brain simulations. What practical use would we use connecting it to Siri? can't ask & get any useful questions asking a baby's brain right? we just sit, watch it cry, laugh, spit, scream & sleep & eat. nothing resembling intelligence or anything remotely practical. Lastly, do we really want or need a truly digit AI simulation? I get into my self-driving car & ask it to take me to work & instead my AI "wants" to visit the car dealership to look at new vehicles. we basically want digital AI assistants /slaves. Where & how would a self-conscious AI that has its own ideas, goals & agendas help us today?"

One certain benefit of bestowing human-like intelligence upon a machine is that it can then be placed in situations that require said intelligence but might cause harm to the human if they were in the same situation. However, he is also open to there being a second generation of approaches to common sense knowledge AI because word2vec and Cyc are the closest first generation attempts but neither of them are linked to visual simulations of common sense at the minute. Effectively, they're just Big Datasets which need to be harnessed to formalise common events. What about even using social media messages as a dataset? After all, that's

how humans communicate most often online nowadays. It's almost like using another dialect of English which wouldn't be formally described in Cyc and the machine would likely develop use of emojis and furthermore applying emojis to different pieces of text, sort of like how if you type an object into SwiftKey, it brings up the most relevant emojis for that object. In a way, one could argue that this is a high level way of mapping a piece of text to an image.

07/04/2020 Review of Diary Feedback

I'm going to start this week's entries by reviewing the written diary feedback about my previous submission because it highlights key deliverables for the final submission. That way, I can structure my final two weeks of work properly in my planning section. I feel the mark that I received reflected my effort for submission 2 relative to submission 1 because this time, I had 2 other projects worth lots of marks ongoing and their deadlines hadn't been postponed at that stage. The submission definitely didn't go according to plan when my interview got cancelled because the lady was sick, and I had the formal the following day. But that's life and I feel that my general knowledge of the Commonsense Knowledge Engineering is constantly growing.

"Identifying your diary is a little difficult from the documents in your github repository in future please just have a folder for the diary with 3 parts or one document with 3 parts and leave the others in a temporary directory or equivalent. Your diary contains lots of interesting material and shows that you are starting to get a good insight into the nature of the problem. As you mention quite **a bit of the work in the diary isn't in the blog**, hopefully you will get a chance to polish this over Easter. Please be careful to read the canvas announcements and ensure that your diaries and other deliverables cover the complete material. **Your planning diary doesn't talk about the target audience** and I think overall your work would benefit from a **deep consideration of what success is like for a formal 'rough world' like datastructure of everyday life that could be linked to text and to assess what work would help most to contribute to that and how it could help others**. Initially in the project you were clearly easily distracted by alternative pieces of work you could do, as the project has developed your focus has improved, it would be a very valuable output of the project to create a **clear plan** for how others (or yourself if you wish to keep working on the project) could **work towards this objective and a possible list of what the current outstanding problems are**."

I've since realised that my target audience is considerably broader than simply A-Level and Undergraduate students with an interest in AI and/or Linguistics because online content creators such as Two Minute Papers and Computerphile attract an audience which spans all demographics. Now, I'd go as far as to say my target audience includes anyone who currently owns an AI system and treats it like a black box, taking its output for granted and anyone who is interested in ChatBots because they will benefit immensely from being able to reason using common sense as opposed to just retrieving information from a website that it's linked to based on keywords in the user's sentence. I've actually found another project like Rough World but from MIT Mind called IsisWorld. http://mindmachineproject.org/isisworld/

In my learning log, I described IsisWorld and why it's a valuable contribution towards the end goal of this project specification. I often refer to this resource from MIT: http://alumni.media.mit.edu/~dustin/isisworld.pdf

08/04/2020 Review of Blog Feedback

"Easy to read and reasonably well focused, does a good job of highlighting the challenge of the research subject and makes a reasonable case that the problem is real. It's clear that you haven't found much in terms of actual formalisations of semantics. Hopefully some of the references we have discussed will help you get closer to making progress with the objective of being able to formalise text. I don't quite understand your suggestion for future work. **The key challenge is being able to formalise text using a simulation datastructure**, so ultimately some form of web based tool that **enables text to be associated with a Sims-like datastructure** would seem like it was moving closer to the goal of

the project. There are a number of attempts to formalise common sense and these may be valuable in guiding what a more realistic sims-like datastructure could look like. It is also valuable to have your blog cover all the work of value of your project so that someone can read it and get up to speed with everything you have done."

In my learning log and also my plan for the future of the project, I'm going to outline my suggested approach to formalising common sense for an AI system. Vector-based word embeddings being combined with a big data repository of concepts like Cyc or ConceptNet, I don't feel there have been any other ways to formalise semantics for a machine and it has been a rather fruitless area of investigation given that the emphasi Doug Lenat says to give him another 10 years at Cyc and it'll . I don't think that understanding text alone is enough for a machine to have common sense.

09/04/2020 Code Feedback and researching simulations

Your code is not assessed for this milestone however it will be for the final deliverable. Remember for the code you submit to be useful then someone like yourself (with the same knowledge as the start of the module) should be able to understand your work and build on it. I would make a call to decide if you are going to explain the code sufficiently for this to be the case or to accept that it is unlikely to be useful to others (and therefore not a source of marks for this module).

I don't think that my code from the first submission is the most useful deliverable for others because my focus has shifted from being distracted by learning NLP Python libraries which is beneficial for my personal development but doesn't help others at the moment because I'm not actually using the practical skills I've acquired to make anything. Instead, I believe that my key deliverable is my research and expanding my blog and learning log to include all that I've found. In terms of running a simulation, I've found Ai2Thor which provides virtual rooms which contain agents and objects that can be interacted with so I feel that it would be a useful test environment. DARPA intends to use Ai2Thor during phase 2 of their plan to teach AI common sense. I noticed that the simulations Ai2Thor offers are mostly domestic so a possible area that could be talked about in my plan is training AI using simulations of other parts of the real world or watching the resources of a city be transacted: https://aeplay.org/citybound and
https://www.researchgate.net/publication/228943276_IsisWorld_An_open_source_commonsense_simulator_for_AI_researchers

IsisWorld is 2010's Ai2Thor because it

Citybound simulates a micro-economy in which a city grows over time and each house has their money and supplies recorded at all times. There are a lot of common-sense concepts that could be deduced from viewing the structured data behind the houses; every house has this data structure (almost like the 'key information' part of a Wikipedia page). Furthermore, it shows the movement of traffic which is another really important common sense concept that is best learned by watching the simulation as opposed to JUST learning about traffic through being trained on the Highway Code (however, training using the simulation AND the Highway Code may actually be a good approach to teaching an AI about customs on the road because it can observe the simulation from something like Citybound or Sims City and has concise, nationally accepted statements about it with which it can reason why the traffic is moving like that. I haven't much knowledge about self-driving cars apart from that there were fears that a mere sticker on a road sign could be a successful adversarial attack on them and make them miss the sign altogether. Overall, I'd say today's research and thinking has made me come to the conclusion that if there was a Highway Code for other parts of life in the real world which lists common sense concepts about say for example, a social media code written by young users instead of postdoc researchers, a chatbot could be trained referring to that and produce texts that are much more like what someone my age would write to a friend. I've played about with

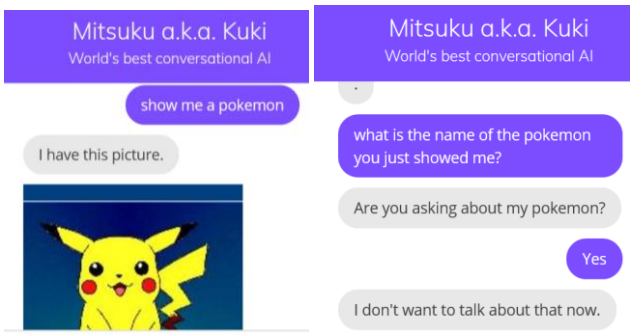
Mitsuku which has an SSA score of 56% (that's Google's latest metric which determines how contextually specific the chatbot's replies are so instead of "can it hold a conversation with a human", it's "can it hold a conversation with a human about a topic and reply to the user in a way that is both human-like and also specific to the context behind that topic ie.

House:	
Money	
Groceries	

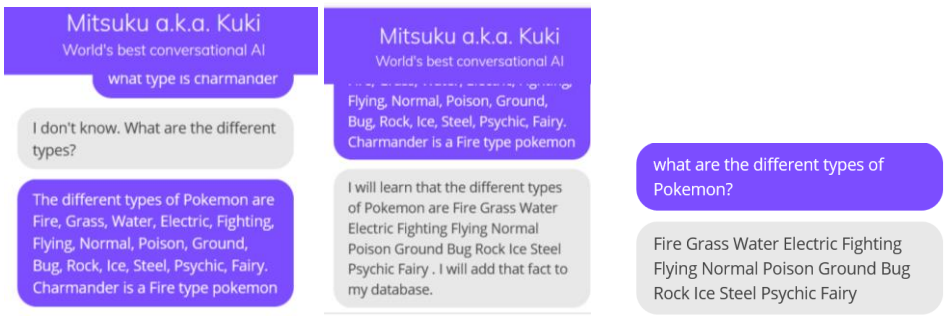
11/04/2020 - 13/04/2020 Working with Mitsuku

Today I updated my blog to include more about my approach to gathering information (what worked and what didn't) because this in itself is valuable to others who may want to research common sense AI because then they're saving valuable time and have access to resources that have been used in producing the content for the blog and the learning log. I just can't help but feel that there is no better data structure to represent knowledge than word vectors in 2020. Then there's the question do we actually need a data structure to formalise common sense when social media exists? Meena has outperformed all other chatbots by simply being trained on 314GB of social media conversations and using TensorFlow's seq2seq https://google.github.io/seq2seq/

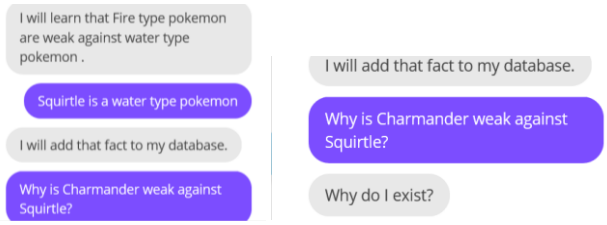
which contains more than 2.6 billion parameters, considerably more than GPT-2. The Tensor Processing Unit used was TPUv3 which was released in 2018 and 2048 of them were used in training Meena in just over 1 month. According to Meena's paper, "The ability to converse freely in natural language is one of the hallmarks of human intelligence and is likely a requirement for true artificial intelligence." By having an SSA of 79%, this demonstrates some level of common sense knowledge because I played around with Mitsuku again today and some of its conversations are much better than others. It hasn't grasped intuitive physics because it asked me to explain how friction is involved in gluing two objects together so I explained the concept in natural language and it said it would remember what I said. By having Mitsuku open to be used by anyone, this is an impressive crowdsourcing tactic because they're demonstrating their AI while also learning common sense rules for free. I tried one more experiment to see just how clever it was.



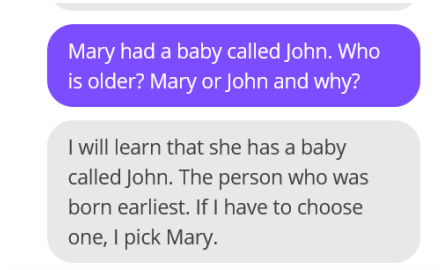
In this case, it lost the context of the conversation very quickly and clearly doesn't have any rules to desrcibe the picture it showed. However once again, the conversation picked up when it asked me to teach it something about Pokemon.



This doesn't demonstrate much about it applying common sense rules about the subject of Pokemon, just that it can learn new rules. So now I'm going to try something else to see if it can apply that knowledge. I taught it that Charmander is a Fire type Pokemon and that Squirtle is a Water type Pokemon and that Fire type Pokemon are weak against Water type Pokemon. Common sense infers that Charmander is weak against Squirtle because Water beats Fire. The response from the chatbot after learning these logical statements **in order** is worrying.



Mitsuku appears to handle more general common sense concepts slightly like temporal reasoning more proficiently:



This example shows that Mitsuku reasoned that the person who was born first is the oldest and since Mary had the baby, she was born first.

17/04/2020 Organising before final submission

I thought it would be beneficial for my viewers if my GitHub organisation improves because according to Y-Combinator, "The enemy is the back button". I'm going to have the following folders: DOCS, BLOG, WEBSITE, SOCIAL MEDIA. By having my project submitted by Sunday night, this gives my post around 20 hours to be viewed by the public. I can then come back on Monday evening and screenshot the analytics of the tweet and compare those with previous general tweets I've made to see if this one has reached a broader audience. A lot of my followers are PwC colleagues so I would expect the interactions to be relatively high.

18/04/2020 Finalising plan //realising that Windows isn't great for simulations.

In reflection, something useful would have been to make a series of video tutorials on IsisWorld because there are absolutely NONE on YouTube. There is nothing to learn from apart from the docs and many programmers prefer watching shadowing another coder to see exactly how the program works. This just struck me today so I'm going to play about with it this evening to see if I can produce some sample simulations and describe them.

EVENING UPDATE

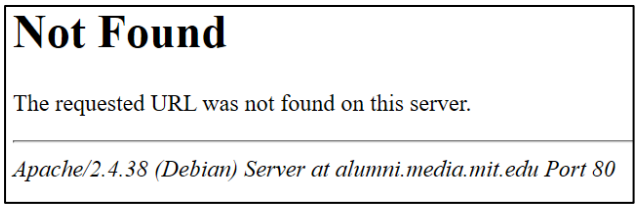


Figure 8: My favourite thing to see during the final week of the project!

http://alumni.media.mit.edu/~dustin/6.868/builds/isis_world_0.3.1_win32.tar.gz

They appear to have removed all Windows implementations of IsisWorld and I have no access to a Mac/Linux OS so I'm going to have to just update my plan to recommend the next person to try again with one of those, but guiding them through what I would have shown off because even though I can't compile/run the simulations, I can still recommend what to cover in tutorials. It's frustrating that Windows systems just don't have any luck with running these simulations so a side-recommendation for something to produce would be one which is accessible to Windows users.

19/04/2020 Final Diary Entry

I'm posting on Twitter tonight and will include a screenshot of the analytics in my project folder tomorrow afternoon. Today I'm just bringing all the diary entries, planning documents and learning log entries together in the right order.