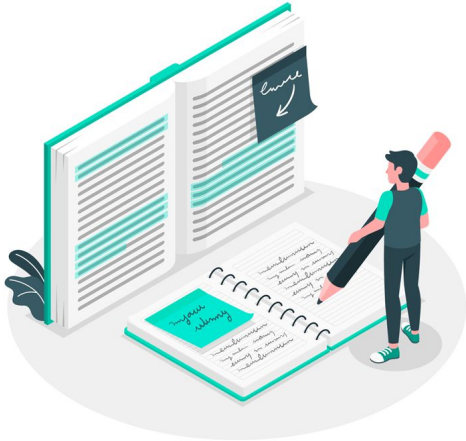
Dartmouth
Course Planning Chatbot

Just a project to make me and my classmates' lives easier!

Table of contents

1. PROJECT SUMMARY



Our Problem

Planning courses at Dartmouth is a **very daunting process** as it involves considering:

- **Prerequisites**
- **Major Classes**
- **All distributives**
- **On and Off terms ****
- **Terms when classes offered**

**** At Dartmouth, students take 12 ON terms and 3 OFF terms in 4 years!**

Solution using NLP

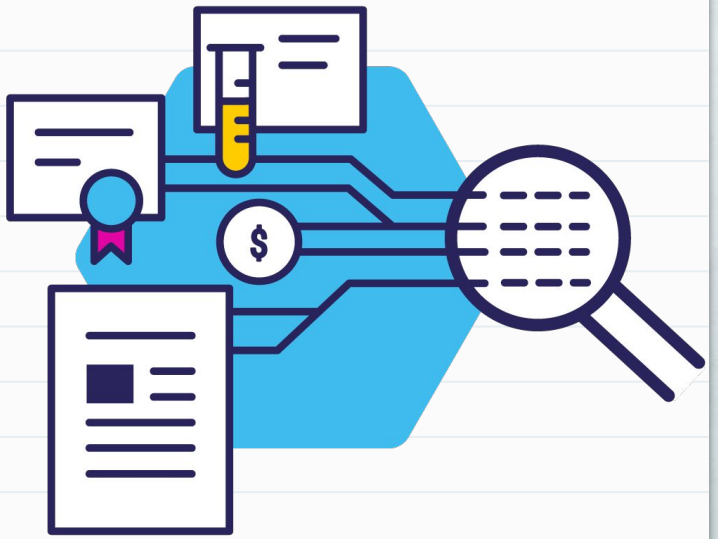
Therefore, I am working on a **Course-Planning Chatbot!**

Input: Natural language input with major and on-off terms

Example: "I am a computer science major off in **25W, 26S and 27W**"

Output: Academic course selection plan with major courses and distributives across available terms!

2. DATA



LING 48/COSC 72 Accelerated Computational Linguistics
Professor Rolando Coto-Solano

Data Source

I collect required data from:

1. ORC
2. different department websites.

The data needs to be processed as it is in:

1. scattered web-pages
2. cannot be categorically accessed by the chatbot (distributives or terms offered)

Data Processing

To process the data, I use:

1. the Spacy package
2. store output into a sqlite database

This database contains categorised information based on these entities -

1. classes
2. prerequisites
3. terms offered
4. distributives covered

3. METHODOLOGY

LING 48/COSC 72 Accelerated Computational Linguistics
Professor Rolando Coto-Solano

Methodology

It uses two NLP components:

1. rule-based Named Entity Recognition (NER) system
    - extracts structured entities from ORC course descriptions

2. regex-based extraction system
    - parses user input based on keyword and pattern matching
    - identifies student's major and off-terms

1. Named Entity Recognition

Package: Python NLP library SpaCy
Input: Data collected

Method: I use SpaCy's EntityRuler which
- identifies types of entities (course codes, terms, prerequisites, distributives)
- matches token patterns with customized entity labels

Output: list of named entities extracted, example:
"COSC 10" → "MAJOR", "Fall" → "TERM", "COSC 1" → "PREREQUISITE", "TLA" → "DISTRIBUTIVE

2. Regex-based extraction system

Module: Python's re library

Input: Natural language input from user

Method: Identifies relevant keywords and match with predefined categories

Output: converts unstructured input into a structured dictionary, example: {"MAJOR": "computer science", "OFF_TERM": ["25W", "26S", "27S"]}.

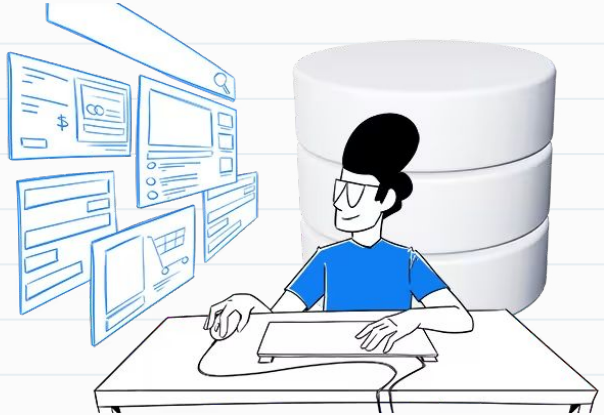
4. PRELIMINARY RESULT

LING 48/COSC 72 Accelerated Computational Linguistics
Professor Rolando Coto-Solano

Currently, the chatbot is able to parse and produce a single year plan for the Cognitive Science major

	Course	Prerequisite	Term	Distributive
6	<PHIL 4>	NaN	25F	TMV
2	<PSYC 6>	NaN	25F	SCI
2	<WRIT 5>	NaN	25F	WREQ
4	<COSC 1>	NaN	25W	TLA
2	<PSYC 6>	NaN	25W	SCI
2	<WRIT 7>	WRIT 5	25W	WREQ
4	<COSC 1>	NaN	25S	TLA
1	<PSYC 1>	NaN	25S	SOC
5	<LING 1>	NaN	25S	QDS

DATA BEHAVIOR



LING 48/COSC 72 Accelerated Computational Linguistics
Professor Rolando Coto-Solano

NER issues

1. Captured **every all-cap combination** (including IP/AI/ML/GPU) as distribs
    - have to manually manipulate the input file

2. Cannot yet distinguish between **prerequisites and usual course codes as their forms are the same**
    - have to manually manipulate the input file

3. **Course codes were not processed correctly**, possibly due to being split into multiple tokens
    - have to pre-process course codes to ensure they are treated as single units

5. ETHICS

LING 48/COSC 72 Accelerated Computational Linguistics
Professor Rolando Coto-Solano

Ethics

We intended to crawl the ORC catalog to a certain depth for **data collection.**

Scraping worked everywhere, but was **blocked** by the course description pages.

We could use **Selenium** to scrape instead, which would involve identifying javascript id and going past that block.

We considered the ethical implications of scraping websites that have blockers, and decided to **manually input pdfs of course descriptions from ORC.**

Thank you!