Document Event Extraction for Calendar Integration

FINAL PROJECT PAPER

Alberto Mejia*
Natural Language Processing Spring 2019
Rensselaer Polytechnic Institute
Troy, NY 12180
albertomejia295@gmail.com

April 25, 2019

ABSTRACT

Google Calendar, or calendars in general, play an essential role in the productivity of many users. Students and professionals alike use such calendars to record all their upcoming assignments and deadlines. I found that such a method proved to be effective only when the individual frequently adds these deadlines to their calendar. The issue with this is that not every person will want to spend the time to do that. This is where the idea to automate this process sprouted. The main idea was to create a model that could take in a schedule as input (e.g. syllabus, work schedule, etc.) and extract important information & dates to integrate them into your Google Calendar. The overall goal of this project was to create an API in which various task managing apps can make use of the model. For the purposes of time, an API is not readily available yet.

Keywords Google Calendar · Natural Language Processing · Schedule · Event Extraction

1 Introduction

The biggest challenge this project faced dealt with the myriad of variations/forms a schedule could take. This made parsing every PDF a task in and of itself. Some schedules/syllabuses are just paragraphs describing the date for certain assignments or exams. This would allow standard Natural Language Processing methods to work very well. However, many others are just tables or bullets showing dates and topics due that day; in some cases, a mix of both. An implementation for this model would have to put more weight into the words in the file over the structure of the file. The issue with this is that a common trend found with the bullet format is the use of incomplete sentences. This means that, although we are prioritizing the content over format, we must take into consideration the effects of the different formats. The use of incomplete sentences stops us from using many methods that rely on the structure and syntax of the English language. We could make use of a Part of Speech Tagger, but it would fail to give accurate dependencies on incomplete sentences. This is because returning the correct POS tag for a word depends on the surrounding words (the context).

As of now, I could not find any research done towards this topic. I could only find about three other attempts at a similar project. Only one of those three did not throw an error when I uploaded my syllabus (I used my Biology syllabus from 2016 for all examples). However, the one that did work did not extract all the information and had some other inaccuracies as well (such as listing the wrong year for a date that does not include the year). That project seems to have been a couple years in the making with dedicated engineers tackling the problem, but I plan on creating my own open source version. Refer to the figures below to see the issues I found with their project.

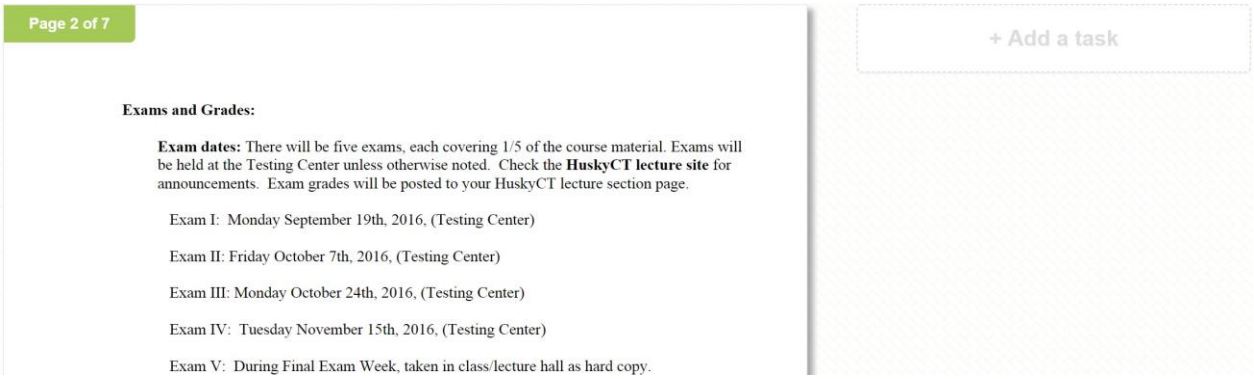

Figure 1a..

Did not extract the Exams (Could be their time identification methods)

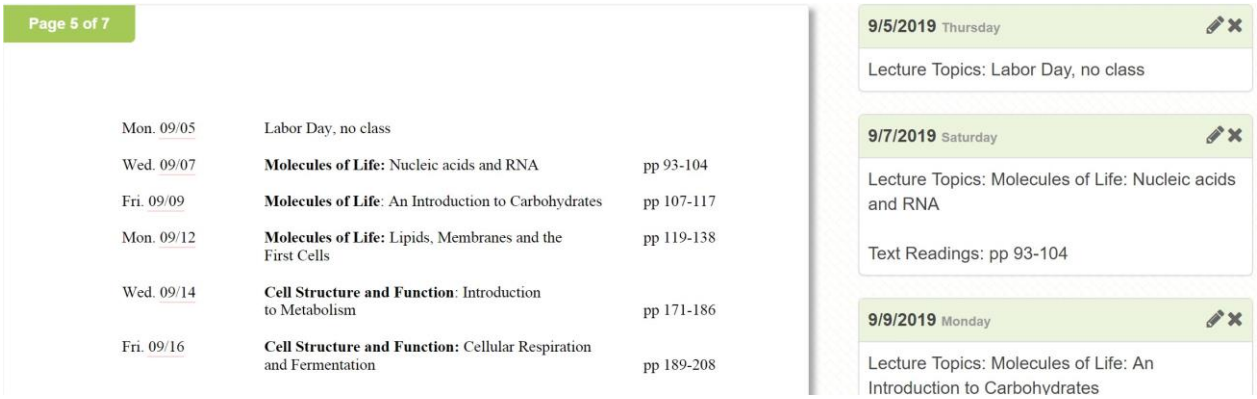

Figure 1b.

Wrong year. As seen from figure 1a, the syllabus is from 2016 not 2019

In this paper, I will cover a novel approach to this problem and potential solutions to improve on the current model.

2 Approach

The model presented in this paper is a Statistical Named Entity Recognition model. To combat the varying formats of the files, we integrate some rules into the model; much of the information we are in search of follows a structured pattern. Some of these structured patterns are emails, phone numbers, and professor names. Thus, we can use rules to leverage our statistical model in handling specific cases while boosting accuracy. In this case, boosting accuracy through the rules is very important since we are using a pre-trained model. The reason for using a pre trained

model is because we do not have the resources in which we could train a model to familiarize itself with. It would require us to have a large dataset of labeled examples, which would take an unreasonable amount time to create.

3 Model Details

In an Information Extraction (IE) system, two of the main components are Event Recognition and Entity Recognition. Entities and events are closely related in the fact that entities are often the participants of events [1]. For this reason, we will implement the Event Extraction of the document using Entity Recognition. For this paper, we will be adopting the ACE definitions for entities and events:

- **Event mention:** An entity is defined as an object or set of objects in the domain. Therefore, an entity mention is a reference to an entity in the form of a noun phrase or a pronoun. [1]
- **Event trigger:** An event trigger is a word or phrase that clearly expresses its occurrence. Event triggers can be verbs, nouns, and occasionally adjectives like "dead". [1]
- **Event argument:** event arguments are entities that fill specific roles in the event. They mainly include participants (i.e., the entities that are involved in the event) and general event attributes such as place and time, and some event-type specific attributes that have certain values (e.g., JOB-TITLE, CRIME). [1]

We will use the event trigger to extract the participants of the event and the attributes (event arguments). Originally, there was thought to revolve the algorithms around the entity mentions (e.g. Exam, Quiz, Project, Homework, etc.) and then gather the event arguments (like time and place) from there but given the structure of a syllabus, an easier method was later discovered. The purpose of a schedule/syllabus document is to inform the individual of all upcoming assignments. This works hand in hand with Calendar Integration as the calendar needs some sort of temporal content to create an event. This allows us to only want to look at places where dates are mentioned, and then extract important information around that trigger. Thus, we can effectively use dates as the event triggers instead of just event arguments. With the dates set as the event triggers, we then look in the same sentence/line to see if any event arguments contain any event mentions.

I used one of the pre-trained statistical models from spaCy (Natural Language Processing Python library), specifically en_core_web_sm and en_web_core_lg. They are both English multi-task CNNs trained on OntoNotes. I began to add rules to the model such as expanding the Person entities component and time identification component (Still in progress). I then used regular expressions to extract structured information like emails and phone numbers.

I also simultaneously started building the application for the service. It currently includes the ability to have these events added to your Google calendar automatically. I do this by working with Google's calendar APIs to receive and use your credentials. That will allow us obtain Authorization for Requests. After that, we can now create and add events to one's calendar. I then used a Python library called Flask to spin up a small server in which we could house the project. It is a simple website for now, but it gives us the ability to upload a syllabus online instead of hardcoding the file in. Here is what the current interface looks:

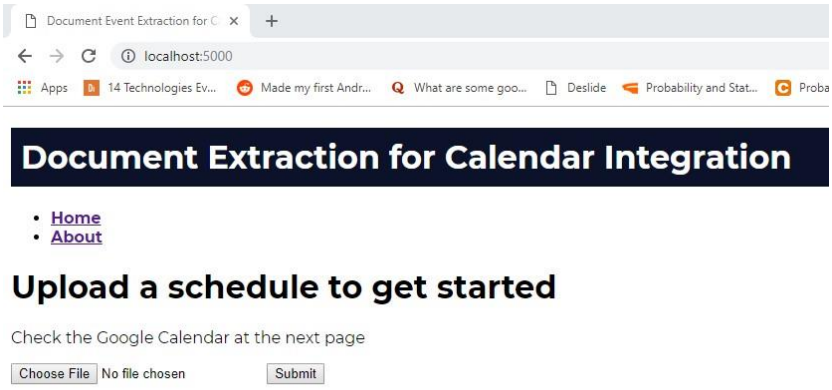

Figure 2a.
Homepage (Contains upload section)

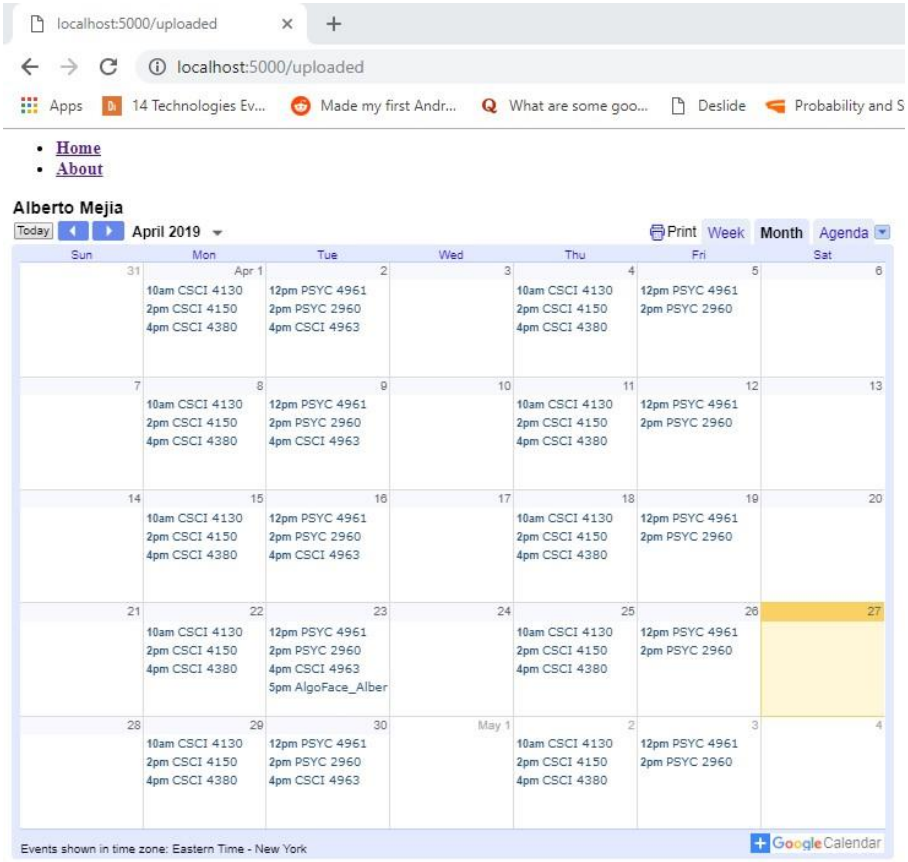

Figure 2b.
Calendar View Page (Currently shows my calendar)

4 Results

As of now, the project managed to extract the exams from the Biology syllabus I provided and add them to my Google Calendar (Something the other website missed). However, the algorithms are still overlooking a lot of information, but I have made a lot of progress to allow myself an easier time for the next steps. So far, my model can now identify some date event triggers. Once

it finds a valid time trigger, it then analyzes the sentence to see if our event arguments contain any of our event mentions. As of now, only exams are supported event mentions but I will extend the set of event mentions to include homework's, readings, projects, etc.
Look at the following figures to visualize the results:

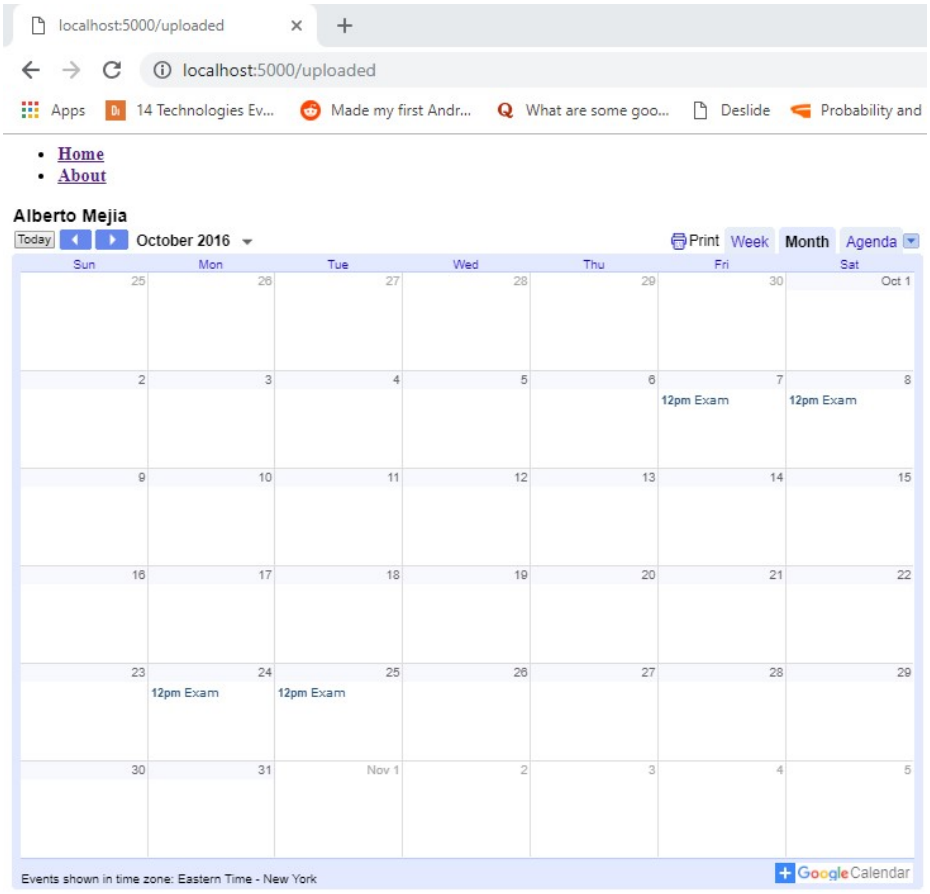

Figure 3a.

Figure 3b.

In figure 3a, we can see that the exams have been integrated in succesfully. It shows the October 2016 exams which match up with the dates on the syllabus (however, there is a small bug that duplicates them into the next day).

In figure 3b, we can even see that the event arguments have also been added into the event by clicking on one of the exams:

5 Error & Analysis

There are many factors of the project that held back it's success. As mentioned in the Introduction section, the schedules tend to come in different layouts. They can also come in structured paragraphs or just a list of due dates. Some even use a combination of the two. Figuring out a model that can deal with all scenarios was the goal of this project. However, this problem proved to be more significant than I originally thought and as a result I was unable to produce an MVP (Minimum Viable Product) for the project. Refer to the following figures to understand how the inconsistencies within these file formats affected popular techniques.

Sentence: "Exam one is on Monday September 19th, 2016 (Testing Center)."

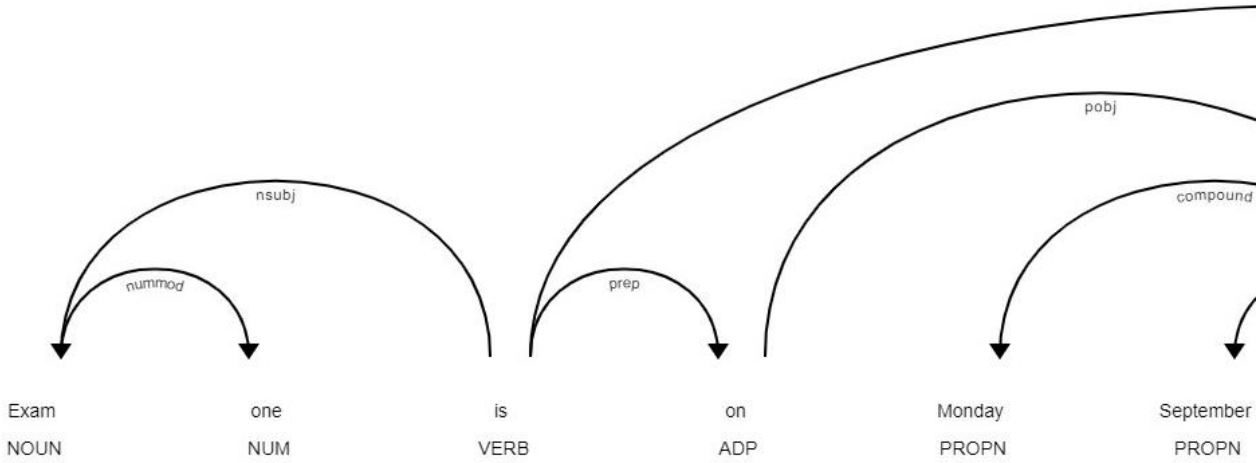

Figure 2a

.

Sentence: "Exam one: Monday September 19th, 2016, (Testing Center)."

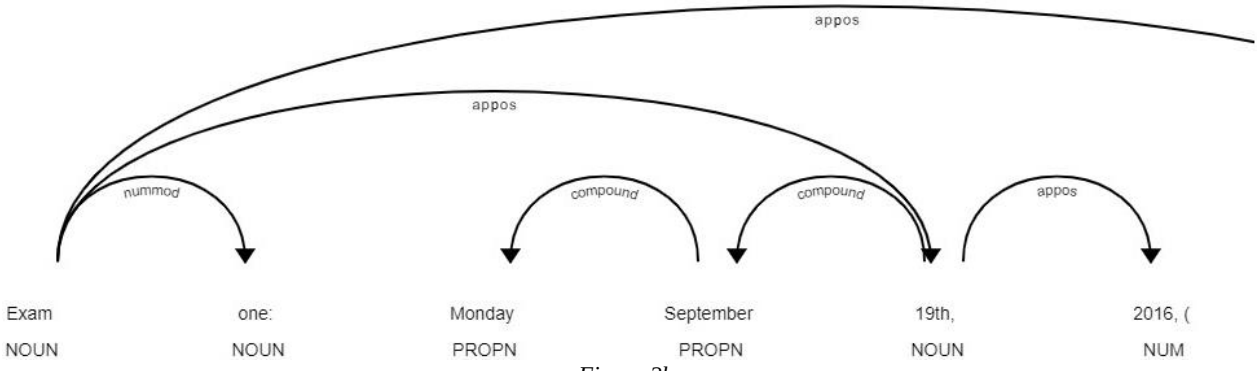

Figure 2b.

I had originally planned on using Part of Speech tagging to get the main subject in the same sentence that the event trigger was met. The issue I had mention earlier is seen above. If the syllabus uses standard English syntax and structure, Figure 1a, we can effectively discover the nominal subject (nsubj) of the sentence. We could then use the nominal subject as a title for that event we add to our Google Calendar. The problem can be seen in the second figure 1b. Here, the document instead listed out the deadlines without using proper sentences. Now, we can no longer use Part of Speech tagging like I had originally intended.

5 Conclusion & Future Work

Overall, the project proved to require more time than I had initially expected. To research and work on the algorithms as well as develop the application was a bit much for one person. Once a more well-rounded model is implemented, I plan on extending support to other file types such as docx and HTML. I had already started working on the HTML extraction but fell short on time. Also, the time identification component of the model must be updated with some rules. It currently overlooks simple formats like 09/16, that is why we could not extract information from those dates like the other project could. I do still plan on furthering this project to create an open source alternative to that one working example I found (www.syllabuddy.com). Many people have commended my project idea and expressed their want for such a product. A lot of progress is still having to be made but with the groundwork and research for the project already done, its progress will accelerate.

[*] Find further information about me @ (www.albertomejia.com)
Follow the progress of the project @ (https://github.com/AlbMej/Natural-Language-Processing)

References

[1] Bishan Yang and Tom Mitchell. Joint Extraction of Events and Entities within a Document Context. NAACL-HLT 2016, pages 289-299. arXiv:1609.03632v1

[2] spaCy documentation

[2] No footnotes