Wavelength Capstone Project 2

Employee Contract Information Extraction

The second part of the capstone project involved extracting discrete structured data from free text that describes various aspects of employment contracts, such as holiday entitlement and hours of work. The manipulation of textual data for information extraction is something that the SpaCy package does extremely well, it provides numerous tools such as the tokenizer, dependency parser part of speech (POS) tagger, named entity recognition (NER) models for NLP as well as rule based matching capabilities.

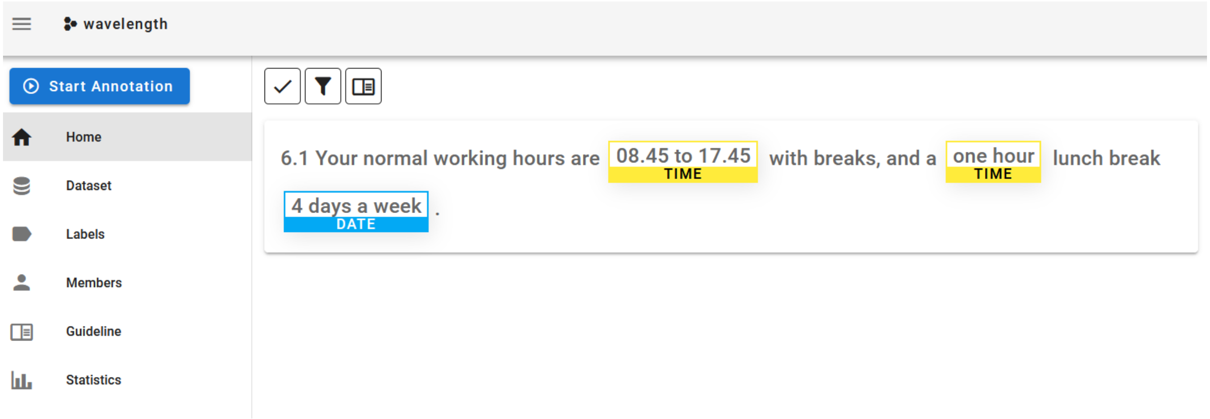
Initial experiments mainly focused on leveraging the benefits of the tokenizer and POS tagger to write enriched conditional statements to extract information. The clauses in the Times [Extract] seemed to have reasonably distinct patterns and initial extraction by iterating over the document examining tokens seemed promising. With examination it became clear that there was in fact a wide variety of branches in the texts that would favour a more sophisticated approach as well as the use of dependency tags to get a better understanding of the links between words and how their meaning changes in different orders and locations.

The standard Spacy NER model was investigated for statistical based recognition which seemed promising. Once added to the pipeline, the NER model creates collections of easily accessible entities for each document which are all conveniently labelled with categories such as TIME and DATE. However, this approach had shortcomings as the NER model seemed to misclassify ranges of days of the week and ranges of times and so there was only a limited increase in accuracy from more basic approaches.

The Matcher and Entity Ruler are rule based engines for Spacy and that allows sophisticated rules to be created to identify text of interest. With the NER model missing many potential entities, the entity Ruler seemed like an appropriate next step. The Entity Ruler engine combines a regex like flexibility to match tokens and attributes of tokens with the ease of access to entities of the NER and together with some helper functions was able to get to 97.5% accuracy on the Days per week specified field.

The downside of the entity ruler is that it does not give the robustness to small changes to the text such as those from OCR or human entry, as demonstrated by several edge cases in the training data, these picked up by tweaking the rules each time a missed entity was found but it showed that this solution could be brittle when encountering noise and previously unseen errors.

One other available option that had not been considered up to this point was to fine tune the standard Spacy NER model with a subset of the training data so that it was better at recognizing the entities likely to be present in the data. This approach required the use of NLP annotation software to create suitable training data, Doccano was selected to do this as it is easy to use and quick to install via a docker image (see figure 1). Data was exported from Pandas into a JSON lines file, imported into Doccano, annotated, and then returned into a pandas dataframe. It was then correctly formatted and used to fine tune the spacy NER model. The results looked promising with all labelled entities correctly classified in the 10% of examples held out from training.



6.1 Your normal working hours are 08.45 to 17.45 TIME with breaks, and a one hour TIME lunch break 4 days a week DATE .

Figure 1 : Doccano annotation software

With the fine-tuned NER model working well, additional code was written to take account of conditions such as half days and multiple days with different hours, this helped increase train accuracy to 98.9% on the first six columns relating to information from the Times [Extract] field. One misclassified example from the training set was on inspection correct and it appeared that the lunch breaks had not been removed from the training data total, another misclassified example was an edge case involving staff meetings one day per week that was not successfully resolved through appropriate conditional statements.

The second extraction was from the Work on Public Holidays [Extract] field. Looking at the structured data labels and how they related to the free text, several patterns were apparent and the Matcher engine in Spacy was used to achieve 100% accuracy on these fields on the training set.

The third text extract was Holiday Entitlement [Extract], this final extract had 10 derived fields from the unstructured text. Looking at the various categories it was clear that many of these described the different type of holiday allowances that an employee would have in their contract. 6 of the fields were mutually exclusive with only 1 of them being the correct classification. This seemed like a good opportunity to try some supervised learning using a bag of words model together with a classifier to identify the correct class of holiday entitlement applicable.

A Tf-Idf vectorizer was used in combination with a custom list of stop words as a pre-processing step to provide numerical features for the classifier as part of an sklearn pipeline. The results of a 5-fold cross validation using GridSearchCV and hyper parameter search are below:

Classifier	CV Mean Test Score
Decision Tree	96.0%
Random Forest	94.6%
Naïve Bayes	87.1%

Table 1: Classifier performance on cross validation

The most effective classifier was a simple decision tree most likely due to the small amounts of information available, hyper parameter tuning did not improve the accuracy substantially. Classes that had only a few instances were merged with similar more common classes and were separated later using code conditions based on the number of days entitled and occurrence of specific words in the text which were picked up with the Matcher engine. This approach yielded 99.9% train accuracy on the holiday entitlement fields.

Upon release of the full data set, the test accuracy was calculated for both complete full dataset and the unseen test data, the accuracy scores are also recorded in the table below:

Extract Name	**Training Accuracy**	**Full Data Set Accuracy**	**Test Set Accuracy**
Times – Entity Matcher	97.5%	-	-
Times – Fine-tuned NER model	98.9%	98.8%	98.6%
Work on Public Holidays	100%	100%	100%
Holiday Entitlement	99.9%	99.1%	97.7%
Nr examples	79	114	35

Table 2: Accuracy on training and full data sets

The first run over the full data set gave a lower accuracy score of 98.0% on the Times extract data and, upon inspection of the test data labels in excel, 3 examples were incorrectly labelled. Once these had been corrected it brought the accuracy up to a similar level to the training set at 98.8%.

The final test involved updating both the fine-tuned Spacy NER model and DTC classifier with all training data and then using these updated models to provide predictions on the unseen portion of the full data set to provide a truer reflection of the performance of the algorithm. The resulting hold out test set accuracies of 97.7% on the holiday entitlement seems to show that the extra training data for the DTC did not increase accuracy overall with 3 examples still classified to the incorrect holiday category. The NER model seemed to perform well on the test set of the times extract with only a small change in accuracy down 0.3% from training to 98.6%.

It would have been useful, time permitting, to have compared the bag of words/DTC approach to a rule-based one given there were so few training examples. Other future work might be to explore the Spacy Text Cat model, which is an inbuilt model for text classification, this also may have resulted in higher accuracy than the bag of words and decision tree approach used in this project.