Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Summer NLP Project

Daniel Barron

Date 23/08/23

Overview

The Goals of the Project

- To classify and annotate a set of scientific abstracts manually using the GUI developed by Luke and Matteo.
- To use this set of labelled abstracts to train a classifier model.
- To use our combined sets of labelled abstracts to train a Named Entity Recognition (NER) model.
- To use these models to classify a large set of unseen scientific abstracts and extract a clean dataset of chemical compounds and their corresponding band gaps.
- To compare this dataset to a manually curated one and determine its accuracy.

Abstract Labelling

The initial steps of the project

- After installing a virtual box to run Ubuntu, the program and the GUI was sent to me to begin labelling scientific abstracts.
- 100 abstracts were sent to me initially to label, and an additional 500 were sent after training the initial classifier model.
- Each abstract was classified as either relevant or irrelevant based on whether it was likely to contain a band gap.
- As well as classifying the abstracts, I was also required to label individual words or tokens as named entities.
- Most of these named entities were then related to another named entity using a relationship attribute.

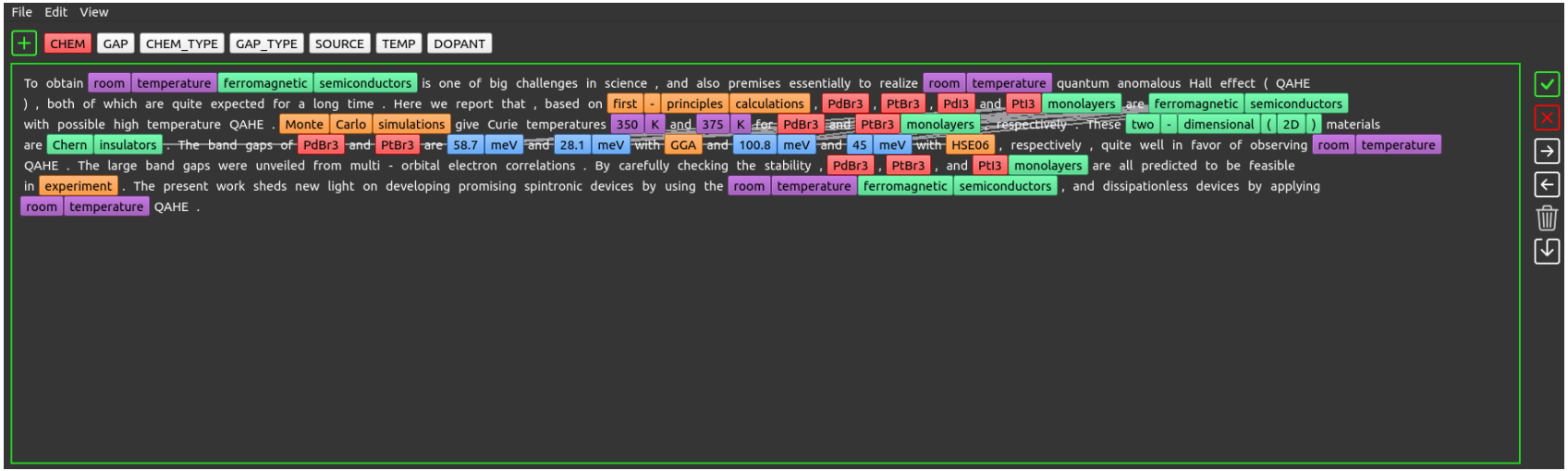
Abstract Labelling

The entities that were used to annotate the abstracts

- CHEM: Any chemical compound's name or formula (no relation)
- GAP: Any band gap value along with the units (related to CHEM)
- CHEM_TYPE: Any additional information about the compound (related to GAP)
- GAP_TYPE: The type of band gap being described (related to GAP)
- SOURCE: The source of the band gap value (related to GAP)
- TEMP: The temperature the band gap was recorded at (related to GAP)
- DOPANT: The dopants used (related to CHEM)

Abstract Labelling

An example of a labelled abstract viewed in the GUI



Classifier Training

The steps in training the classifier

- Once all the abstracts were labelled, a BERT classifier model was trained to classify unseen abstracts as either relevant or irrelevant.
- Initially, only 100 abstracts were used in the training. It was found to be difficult to train a classifier on such a small dataset, which is why a further 500 abstracts had to be labelled.

```
def combine_json_files(file1, file2, output_file):
  with open(file1, 'r') as f1, open(file2, 'r') as f2:
    data1 = json.load(f1)
    data2 = json.load(f2)

  combined_data = {'corpus': data1['corpus'] + data2['corpus']}

  with open(output_file, 'w') as output:
    json.dump(combined_data, output, indent=4)
```

Classifier Training

Methods used in the training of the classifier

- Most of the training involved changing the hyperparameters such as learning rate and batch size.
- One method I employed was an automatic changing of the learning rate as can be seen here.
- Later when using my trained classifier, I found it to be too selective, i.e., it had a lower than desired recall. I solved this problem by retraining the classifier using Luke's, Oran's and my combined datasets.

```
BERT_VERSION = "m3rg-iitd/matscibert"
MAX_LEN = 256
BATCH_SIZE = 4
EPOCHS = 20
LEARNING_RATE = 5e-05
MAX_GRAD_NORM = 8
MODEL_DIR = "models/classifier"
CORPUS = "./drive/MyDrive/summer_proj/corpus/total_combined.json"
MAIN_DIR = "./drive/MyDrive/summer_proj"
```

```
if val_loss >= val_last:
  LEARNING_RATE = LEARNING_RATE/2
val_last = val_loss
```

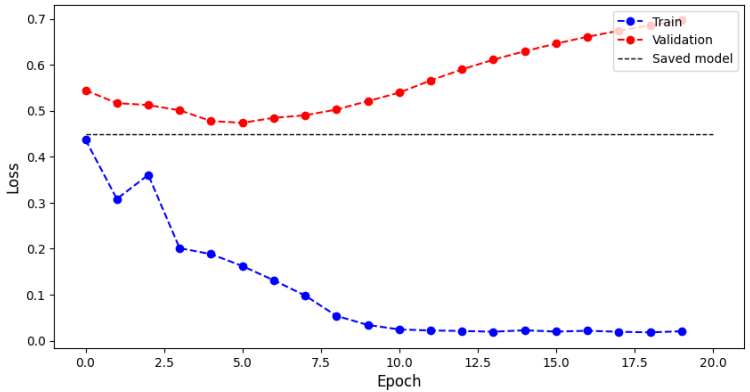
Classifier Training

Results of the classifier training step

Train	
Accuracy	0.98
Precision	0.96
Recall	0.96
F1	0.96
support	256
Size	1190

Val	
Accuracy	0.91
Precision	0.83
Recall	0.77
F1	0.8
support	62
Size	255

Test	
Accuracy	0.91
Precision	0.81
Recall	0.85
F1	0.83
support	66
Size	255



- The result of training the classifier on the combined dataset was a seemingly worse classifier, however I believe this is only due to disparities in our individual labelling.
- My own labelling is probably what led to my earlier model being stricter, as I was stricter with what I deemed relevant or irrelevant.
- The graph shows how the train and validation loss functions progressed with the number of epochs.

NER Training

The steps in training the NER model

- After training a classifier model, an NER model needed to be trained for the nomenclature condition.
- A dataset of 1700 labelled abstracts was used to train this model, which had been labelled between myself, Luke and Oran.
- Initially the model was only trained to label CHEM and GAP entities, before the code was adapted to incorporate the other five labelled entities as well: CHEM_TYPE, GAP_TYPE, SOURCE, TEMP, and DOPANT.

NER Training

Methods used in the training of the NER model

- Like in the classifier training, most of the training involved changing the hyperparameters such as learning rate and batch size.
- I employed a slightly different strategy to the one I used in the classifier training, as can be seen on the right. In this case, the learning rate depended on the number of epochs that had elapsed, making it automatically decrease over time.

```
MAIN_DIR = "./drive/MyDrive/summer_proj/"
BERT_VERSION = "m3rg-iitd/matscibert"
MAX_LEN = 256
TRAIN_BATCH_SIZE = 32
VALID_BATCH_SIZE = 32
EPOCHS = 6
LEARNING_RATE = 2e-05
MAX_GRAD_NORM = 8
MODEL_DIR = "models/ner_model_4/"
```

```
if f1_best == 0:
  LEARNING_RATE = 3e-05
  MAX_GRAD_NORM = 8
else:
  LEARNING_RATE = (2e-05)/(epoch + 1)
  MAX_GRAD_NORM = 8
```

NER Training

Results of the NER training step

```
Validation loss per 100 evaluation steps: 0.03356190398335457
Validation Loss: 0.05395616047998082
Validation Accuracy: 0.979472945872313
              precision    recall  f1-score   support

        CHEM       0.72      0.80      0.76      1816
         GAP       0.60      0.79      0.68       206

   micro avg       0.71      0.80      0.75      2022
   macro avg       0.66      0.79      0.72      2022
weighted avg       0.71      0.80      0.75      2022
```

```
Validation loss per 100 evaluation steps: 0.23643022775650024
Validation Loss: 0.2941962944449119
Validation Accuracy: 0.9219540856063293
              precision    recall  f1-score   support

        CHEM       0.74      0.81      0.77      1816
   CHEM_TYPE       0.53      0.64      0.58      2194
      DOPANT       0.40      0.81      0.53        21
         GAP       0.62      0.77      0.69       206
    GAP_TYPE       0.48      0.57      0.52       108
      SOURCE       0.50      0.62      0.56      1176
        TEMP       0.53      0.79      0.64        63

   micro avg       0.59      0.70      0.64      5584
   macro avg       0.54      0.72      0.61      5584
weighted avg       0.59      0.70      0.64      5584
```

- The final NER model that was trained on only the CHEM and GAP entities ended up having an F1 score of 0.76 for CHEM and 0.68 for GAP.
- Interestingly, the addition of more entities didn't seem to hinder the F1 scores of the CHEM and GAP entities, it instead seemed to improve them.

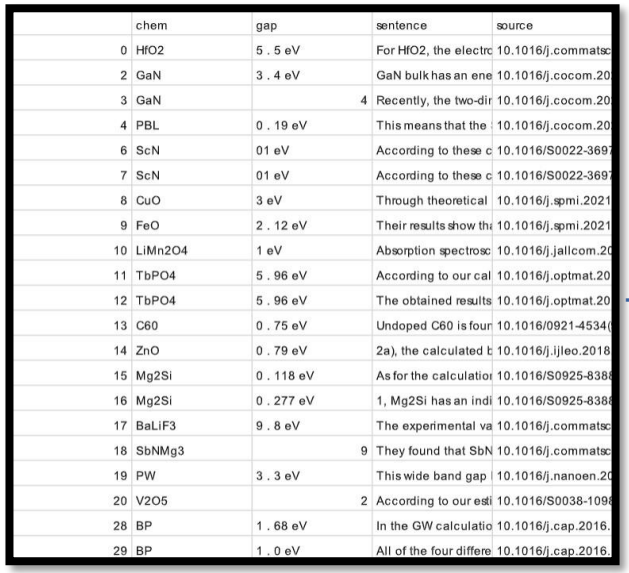
Extracting a Clean Dataset

Deployment of the classifier and NER models

- The dataset of unseen abstracts was initially run through the trained classifier. Seeing that this classifier was too selective, I retrained the classifier on my, Luke and Oran's combined dataset. Out of 1,560,999 abstracts, the classifier deemed 86,251 of them to be relevant.
- The NER model that was trained on two entities was then used to predict CHEM and GAP entities for those 86,251 abstracts, going through them sentence by sentence. In doing this, it extracted a file containing multiple mentions of CHEM and GAP, as well as a readable .csv file containing a single mention of both entities.

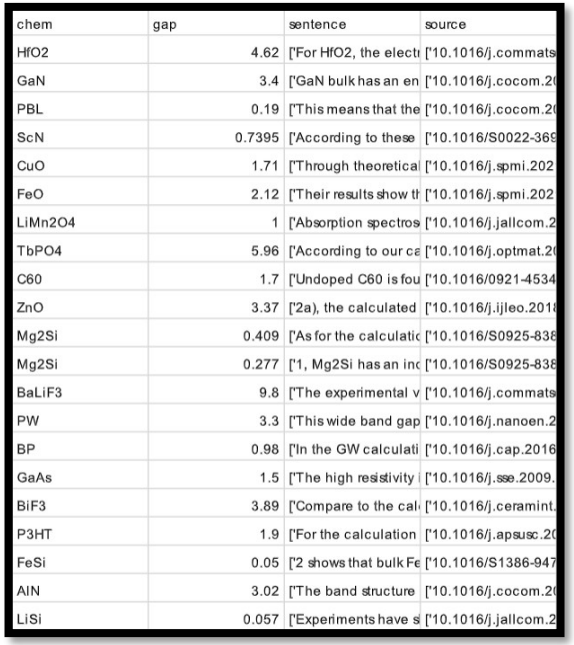
Extracting a Clean Dataset

Cleaning the single mentions file



	chem	gap	sentence	source
0	HfO2	5 . 5 eV	For HfO2, the electro	10.1016/j.commatsc
2	GaN	3 . 4 eV	GaN bulk has an ene	10.1016/j.cocom.20
3	GaN	4	Recently, the two-dir	10.1016/j.cocom.20
4	PBL	0 . 19 eV	This means that the	10.1016/j.cocom.20
6	ScN	01 eV	According to these c	10.1016/S0022-3697
7	ScN	01 eV	According to these c	10.1016/S0022-3697
8	CuO	3 eV	Through theoretical	10.1016/j.spmi.2021
9	FeO	2 . 12 eV	Their results show tha	10.1016/j.spmi.2021
10	LiMn2O4	1 eV	Absorption spectrosc	10.1016/j.jallcom.20
11	TbPO4	5 . 96 eV	According to our cal	10.1016/j.optmat.20
12	TbPO4	5 . 96 eV	The obtained results	10.1016/j.optmat.20
13	C60	0 . 75 eV	Undoped C60 is four	10.1016/0921-4534(
14	ZnO	0 . 79 eV	2a), the calculated b	10.1016/j.ijleo.2018
15	Mg2Si	0 . 118 eV	As for the calculation	10.1016/S0925-838
16	Mg2Si	0 . 277 eV	1, Mg2Si has an indi	10.1016/S0925-838
17	BaLiF3	9 . 8 eV	The experimental va	10.1016/j.commatsc
18	SbNMg3	9	They found that SbN	10.1016/j.commatsc
19	PW	3 . 3 eV	This wide band gap	10.1016/j.nanoen.20
20	V2O5	2	According to our esti	10.1016/S0038-109
28	BP	1 . 68 eV	In the GW calculatio	10.1016/j.cap.2016.
29	BP	1 . 0 eV	All of the four differe	10.1016/j.cap.2016.

- A snippet of the initial .csv file can be seen on the left here. While the data extracted was of a relatively good quality, it needed to be post processed for proper analysis to be done on it.
- The snippet on the right is that of the post processed .csv file. The band gap values were made to be purely numerical, and if any CHEM entities were repeated, the median value of all the corresponding GAP values was taken as the true band gap.

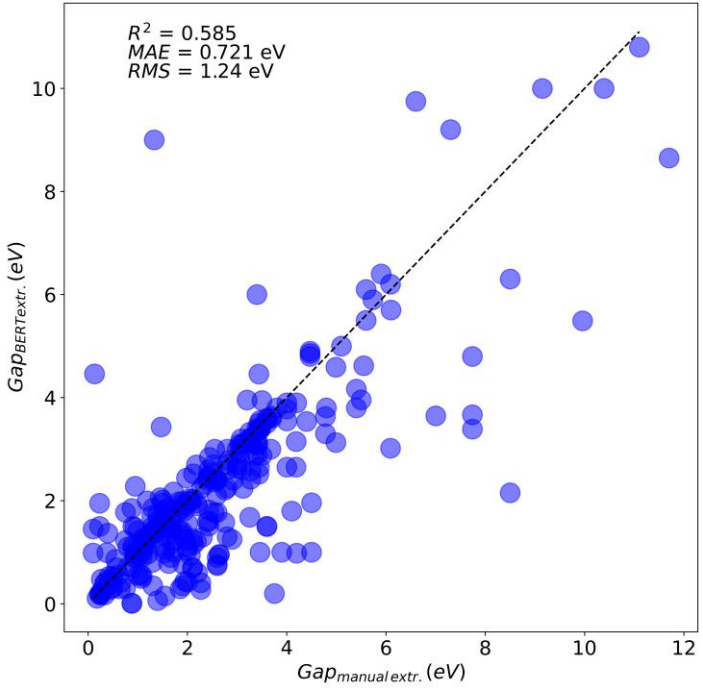


chem	gap	sentence	source
HfO2	4.62	['For HfO2, the electr	['10.1016/j.commats
GaN	3.4	['GaN bulk has an en	['10.1016/j.cocom.2
PBL	0.19	['This means that the	['10.1016/j.cocom.2
ScN	0.7395	['According to these	['10.1016/S0022-369
CuO	1.71	['Through theoretical	['10.1016/j.spmi.202
FeO	2.12	['Their results show th	['10.1016/j.spmi.202
LiMn2O4	1	['Absorption spectros	['10.1016/j.jallcom.2
TbPO4	5.96	['According to our ca	['10.1016/j.optmat.2
C60	1.7	['Undoped C60 is fou	['10.1016/0921-4534
ZnO	3.37	['2a), the calculated	['10.1016/j.ijleo.201
Mg2Si	0.409	['As for the calculatio	['10.1016/S0925-838
Mg2Si	0.277	['1, Mg2Si has an inc	['10.1016/S0925-838
BaLiF3	9.8	['The experimental v	['10.1016/j.commats
PW	3.3	['This wide band gap	['10.1016/j.nanoen.2
BP	0.98	['In the GW calculati	['10.1016/j.cap.2016
GaAs	1.5	['The high resistivity	['10.1016/j.sse.2009.
BiF3	3.89	['Compare to the cal	['10.1016/j.ceramint.
P3HT	1.9	['For the calculation	['10.1016/j.apsusc.2
FeSi	0.05	['2 shows that bulk Fe	['10.1016/S1386-947
AlN	3.02	['The band structure	['10.1016/j.cocom.2
LiSi	0.057	['Experiments have s	['10.1016/j.jallcom.2

Extracting a Clean Dataset

Comparing the extracted dataset to a manually curated one

- The last step was to now determine the accuracy of the clean dataset by comparing it with a manually curated one.
- The graph on the right shows a plot of the BERT extracted data against the manually extracted data.
- The R^2 value was 0.585, the mean absolute error was 0.721 eV and the root mean squared error was 1.24 eV.



Conclusions

What to take away from the project

- Labelling the abstracts less harshly may have led to a more inclusive classifier, but the training of the classifier on the total combined dataset helped to overcome this.
- In the NER model training, the decreasing of the learning rate with the number of epochs greatly improved the F1 scores of the model.
- The final clean dataset that was extracted had a total of 1327 entries. Although this dataset was not perfect, the R^2 value of 0.585, mean absolute error of 0.721 eV and root mean squared value of 1.24 eV when compared with the manually curated one are relatively good metrics.

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

Thank You