Disadvantages

- HMM method is only dependent on every state and its corresponding observed object: The sequence labeling, in addition to having a relationship with individual words, also relates to such aspects as the observed sequence length, word context and others.
- The target function and the predicted target function do not match:
  HMM acquires the joint distribution P(Y, X) of the state and the observed sequence, while in the estimation issue, we need a conditional probability P(Y|X).

Evaluating of the HMM Model

I chose to adopt the same way to evaluating CRF Model. So, the indicator of evaluation is F-score. As mentioned in the coursework description, if we use a machine learning method in this step, we should use a different way in the second. I used a different way in tokenization in the HMM model. For tokenization in this model, I used nltk.RegexpParser and for the CRF model I used the test_sents = [[token for token,tag in sent] for sent in test_set].

Results

I tested the two models in the test data, the results are not very good, for the CRF method the results are :

```
F1 score for class Material = 0.20930232558139536
F1 score for class Task = 0.043887147335423204
F1 score for class Process = 0.20916905444126074
Macro-average f1 score = 0.1541195091193598
```

As we see, this method cannot predict the three tags very well, it predicts the type of word with very high percentage but in our case, to predict task or materiel or process it's very hard for a machine learning problem to do it well without more details. So, this method is efficient compared to the HMM model. HMM model must define the probability of each word and its place in the sentence. With automating this process, the probability is very low, so the result is not good. It seems that the tags in this data are difficult to be predicted. Here are the results of the HMM model:

```
F1 score for class Material = 0.028639618138424822
F1 score for class Task = 0.02040816326530612
F1 score for class Process = 0.05417118093174431
Macro-average f1 score = 0.034406320778491754
```

Task 2: extracting semantic relations

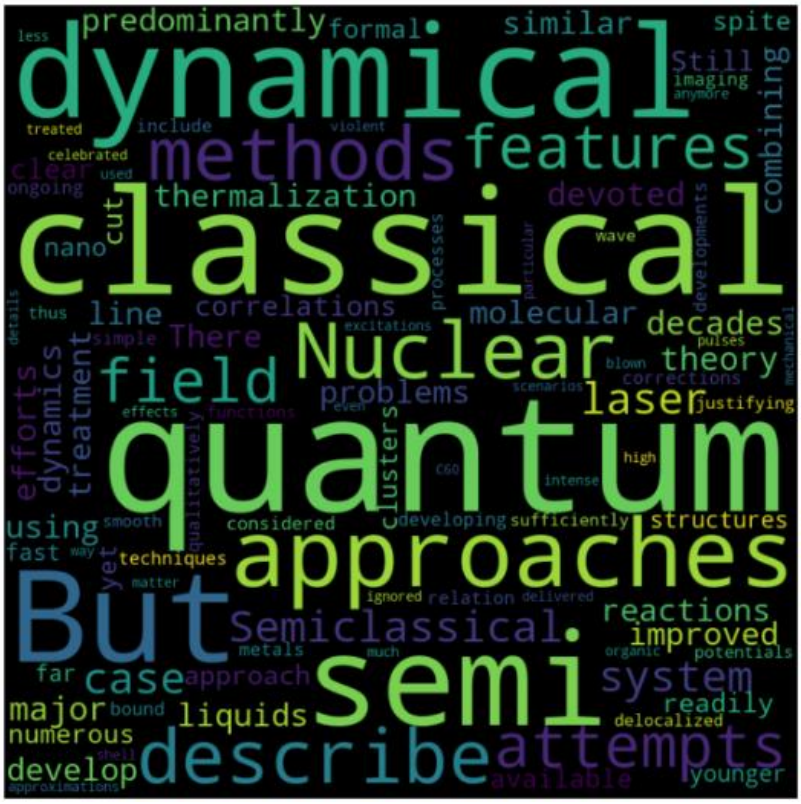
In this task, I decided to use two methods to extract the relation between words which is synonyms and hyponyms. The relation synonym means that the two entity phrases refer to the same thing (<entity A> is the same as <entity B>) whereas the hyponym represents the first entity is a more specific word than the second. The second entity is a category or more general term to which the first belongs.

<u>First method: NLTK method</u>

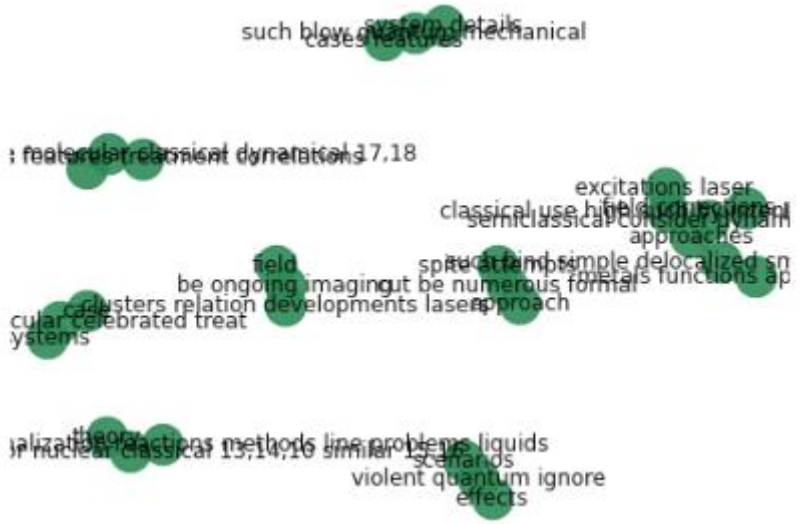
This method is based on taking a word from the input text file and printing the synonyms, definitions and example sentences for the word using WordNet. It separates the synonyms from the synset based on the part-of-speech. For example, synonyms that are verbs and the synonyms that are adjectives are printed separately.

Example for the word flabbergasted the synonyms are 1) flabbergast , boggle , bowl over which are verbs and 2)dumbfounded , dumfounded , flabbergasted , stupefied , thunderstruck , dumbstruck , dumbstricken which are adjectives.

Finally, the method returns the synonym of each word in the same text. This is a basic method that can show us the relation between the words to pass to another complicated method.



Now we will present a graph that shows the relation between the sentences in the text. It shows the relations between some word or a group of words with others in the same text. We do this analysis before we do the second task, to have an idea about the relation between the words.



Now, we draw a graph of word occurrence for the first test in training data (I chose an example). The figure below demonstrates the same results in word cloud plot.