DESCRIPTION OF ISSUE PRICE

We made the training CSV (data) by using regular expression and finding the keywords from those bond documents and labeling the sentences having keyword in the sentence but not the correct sentence as 0 and having keyword in the sentence with correct semantics as 1. Then we made a Count vectorizer or TFIDF to train RNN model with these feature sets. Finally after Training the model we used regular expression to again get the sentences containing the keywords in it and testing these sentence on the model which has been trained by RNN to give the final Output as 0 or 1.

Those sentences having labels as 1 is selected and If in case testing bond document contains multiple 1 as a label by using partition method in python and with the help of regular expression we will get the issue price and we will return maximum issue rice from those.

```
li=[]

y_pred = rfc.predict(tv_test_reviews)

tt=0;

for i in range(0, len(y_pred)):

    if(y_pred[i]==1):

        li+=li3[i]

        s=li3[i]

        s = s.lower()

        sub= "issue price"

        t = s.partition(sub)[2]

        print(s.partition(sub)[1])

        prilist = t.split(" ")

        for i in range(len(prilist)):

            if(not prilist[i].isalnum()):

                print(prilist[i:i+4])
                break
    if(y_pred[i]==0):
        tt++
if(tt==len(y_pred)):
    print("Issue Price:100%")
```

If in case all the labels comes out to be 0 that is bond document does not contain any issue price then we will by default print issue price as 100%.

DESCRIPTION OF NOI AND SENIORITY:

We trained the LSTM on a set of correct sentences so that it can classify the sentences as correct or incorrect. After we have the correct sentence on hand we know that the value of NOI and seniority is within this sentence but still we have to find the correct value. A sentence has a lot of words which are of no interest to us therefore to find the correct value we have to start removing unnecessary words and identify the remaining based on their grammatical structure. We used a variety of natural language processing methods to achieve this, firstly all the stopwords like and ,is ,a etc were removed because they were of no interest to us. Secondly we used named entity recognition to remove all nouns and pronouns because we are searching for a verb in this case therefore names are not required. At the end we used part of speech (POS) tagging to identify the grammatical structures of the sentences and finally using stemming we were able to get the correct value for our fields.

The accuracy of the model was mainly dependent of the number of correct sentences used for training the LSTM, there can be a variety of sentences that can be the correct one therefore a good training dataset is a must in this case. Since the POS and NER algorithms work on grammatical rules they are 100% accurate and can easily find the correct value of fields from the sentence. The only bottleneck is our algorithms are the LSTM and it’s training dataset.